

RESEARCH ARTICLE

Multifidelity computing for coupling full and reduced order models

Shady E. Ahmed¹, Omer San^{1*}, Kursat Kara¹, Rami Younis², Adil Rasheed^{3,4}

1 School of Mechanical & Aerospace Engineering, Oklahoma State University, Stillwater, OK, United States of America, **2** The McDougall School of Petroleum Engineering, The University of Tulsa, Tulsa, OK, United States of America, **3** Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, **4** Department of Mathematics and Cybernetics, SINTEF Digital, Trondheim, Norway

* osan@okstate.edu

Abstract

Hybrid physics-machine learning models are increasingly being used in simulations of transport processes. Many complex multiphysics systems relevant to scientific and engineering applications include multiple spatiotemporal scales and comprise a multifidelity problem sharing an interface between various formulations or heterogeneous computational entities. To this end, we present a robust hybrid analysis and modeling approach combining a physics-based full order model (FOM) and a data-driven reduced order model (ROM) to form the building blocks of an integrated approach among mixed fidelity descriptions toward predictive digital twin technologies. At the interface, we introduce a long short-term memory network to bridge these high and low-fidelity models in various forms of interfacial error correction or prolongation. The proposed interface learning approaches are tested as a new way to address ROM-FOM coupling problems solving nonlinear advection-diffusion flow situations with a bifidelity setup that captures the essence of a broad class of transport processes.

OPEN ACCESS

Citation: Ahmed SE, San O, Kara K, Younis R, Rasheed A (2021) Multifidelity computing for coupling full and reduced order models. PLoS ONE 16(2): e0246092. <https://doi.org/10.1371/journal.pone.0246092>

Editor: Fang-Bao Tian, University of New South Wales, AUSTRALIA

Received: August 12, 2020

Accepted: January 13, 2021

Published: February 11, 2021

Copyright: © 2021 Ahmed et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019290. O.S. gratefully acknowledges their support. OPWIND: Operational Control for Wind Power Plants (Grant No.: 268044/E20) project funded by the Norwegian Research Council and its industrial partners (Equinor, Vestas, Vattenfall) is also acknowledged. The funders had no role in

Introduction

Numerical simulations are the workhorse for the design, testing, and implementation of scientific infrastructure and engineering applications. While immense advances in computational mathematics and scientific computing have come to fruition, such simulations usually suffer a curse of dimensionality limiting turnaround. The field of multifidelity computing, therefore, aims to address this computational challenge by exploiting the relationship between high-fidelity and low-fidelity models. One such multifidelity approach becomes crucial, especially for multi-query applications, such as optimization, inference, and uncertainty quantification, that require multiple model evaluations in an outer-workflow loop. To this end, sampling-based approaches have been often introduced to leverage information from many evaluations of inexpensive low-fidelity models fused by only a few carefully selected high-fidelity computations. An excellent review of the state-of-the-art multifidelity approaches for outer-loop contexts can be found in [1].

study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Competing interests: A.R. acknowledges funding from Equinor, Vestas, Vattenfall, the industrial partners of the Norwegian Research Council. There are no patents, products in development or marketed products to declare. This does not alter our adherence to PLOS ONE policies on sharing data and materials.

In this paper, we focus on a different type of multifidelity formulation targeting domain decomposition type problems that consist of multiple zones with different characteristics as well as multiphysics systems where different levels of solvers are devoted to coupled physical phenomena. A key aspect of the zonal multifidelity approach is its ability to handle intrinsic heterogeneous physical properties, varying geometries, and underlying governing dynamics. This heterogeneity can be mild as in aerospace applications with spatially varying parameters. However, in media where there is a permittivity such as in electrostatics or porous media, this might be more pronounced. For example, fluid flow in rock often follows Darcy's law, whereas flow in a fracture is modeled as Poiseuille flow. Moreover, a related process in subsurface flows might include a high fidelity approach around wells (that drive the flow) and a low-fidelity model for subdomains in the interior [2, 3]. This discussion can also be extended to an active flow control problem to elucidate the concept of the *zonal multifidelity approach* that we tackle in this work. In general, boundary-layer control poses a grand challenge in many aerospace applications including lift enhancement, noise mitigation, transition delay, and drag reduction. Among many other actuator technologies, blooming jets [4–6] and sweeping jets [7–9] offer new prospective solutions in improving the aerodynamics efficiency and performance of the future air vehicle systems. The size of these actuators is usually orders of magnitude smaller than the length scales of the entire computational domain (e.g., an aircraft wing or tail). Including the full representations of each controller's internal flow dynamics in a comprehensive numerical analysis of the entire system is an extremely daunting approach [10]. Meanwhile, the effective flow physics of these actuators can often be accurately characterized by a latent reduced order space due to the existence of strong coherent structures such as quasi-periodic or time-periodic shedding, pulsation, or jet actuation. Therefore, in practice, those flow actuators can be modeled by considering a reduced order surrogate coupled and tied to the global simulation of the whole wing or tail [11, 12].

The above examples illustrate that different levels of models and descriptions can be devoted to different zones and components of the problem in order to allocate computational resources more effectively and economically. This might be the case for many other coupled multiphysics systems, such as geometric multiscale [13–18] and heterogeneous multiscale [19, 20] problems, fluid-structure interactions [21], and industrial scale applications [22–24]. Since various zones and/or physics in these systems are connected through interfaces, data sharing, and consistent interface treatment among respective models are inevitable. Likewise, multirate and locally adaptive stepping methods can yield a mismatch at the space-time interface, and simple interpolation or extrapolation might lead to solution divergence or instabilities [25]. Moreover, even if we are interested in simulating just one portion of the domain corresponding to some specific dynamics, we still need to specify the physically consistent interface conditions. Running a high fidelity solver everywhere only to provide the flow state at the interface seems to be unreasonable. Therefore, we consider formulating an interface modeling approach to facilitate the development of efficient and reliable multifidelity computing. This should serve and advance the applicability of the emerging digital twin technologies in many sectors [26]. However, just like any technology, it comes with its own needs and challenges [27–32]. In practice, two modeling paradigms are in order.

- *Physics-based modeling:* This approach involves careful observation of a physical phenomenon of interest, development of its partial understanding, expression of the understanding in the form of mathematical equations, and ultimately, solution of these equations. Due to the partial understanding and numerous assumptions along the steps from observation to the solution of the equations, a large portion of the essential governing physics might be, intentionally or unintentionally, ignored. The applicability of high fidelity simulators with

minimal assumptions has so far been limited to the offline design phase only. Despite this significant drawback, what makes these models attractive are sound foundations from first principles, interpretability, generalizability, and existence of robust theories for the analysis of stability and uncertainty. However, most of these models are generally computationally expensive, do not adapt to new scenarios automatically, and can be susceptible to numerical instabilities.

- *Data-driven modeling*: With the abundant supply of big data, open-source cutting edge and easy-to-use machine learning libraries, cheap computational infrastructure, and high quality, readily available training resources, data-driven modeling has become very popular. Compared to the physics-based modeling approach, these models thrive on the assumption that data is a manifestation of both known and unknown physics and hence when trained with an ample amount of data, the data-driven models might learn the full physics on their own. This approach, involving in particular deep learning, has started achieving human-level performance in several tasks that were until recently considered impossible for computers. Notable among these are image classification [33], dimensionality reduction [34], medical treatment [35], smart agriculture [36], physical sciences [37–39] and beyond. Some of the advantages of these models are online learning capability, computational efficiency for inference, accuracy even for very challenging problems as far as the training, validation and test data are prepared properly. However, due to their data-hungry and black-box nature, poor generalizability, inherent bias and lack of robust theory for the analysis of model stability, their acceptability in high stake applications like digital twin and autonomous systems is fairly limited. In fact, the numerous vulnerabilities of deep neural networks have been exposed beyond doubt in several recent works [40–42].

In this work, we put forth a *hybrid analysis and modeling (HAM)* framework as a new paradigm in modeling and simulations by promoting the strengths and mitigating the weaknesses of physics-driven and data-driven modeling approaches. Our HAM approach combines the generalizability, interpretability, robust foundation and understanding of physics-based modeling with the accuracy, computational efficiency, and automatic pattern-identification capabilities of advanced data-driven modeling technologies. In the context of multifidelity computing, we advocate and explore the utilization of statistical inference to bridge low-fidelity and high-fidelity descriptions. In particular, we adopt the long short-term memory (LSTM) neural network to match the reduced order model (ROM) and full order model (FOM) solutions at their intersect. To form the building blocks of our HAM approach for coupling ROM and FOM descriptions, we introduce an array of interface modeling paradigms as depicted in Fig 1 and described next.

The *Direct Prolongation Interface (DPI)* approach utilizes standard projection based ROMs, where the system's state at the interface is obtained by the reconstruction of a Galerkin projection ROM solution. However, traditional Galerkin ROM often yields an inaccurate solution in case of systems with strong nonlinearity. Therefore, we utilize machine learning to correct and augment ROM solution in a hybrid framework, as follows.

In the *Prolongation followed by machine learning Correction Interface (PCI)* methodology, an LSTM is used to rectify the field reconstruction from Galerkin ROM at the interface by learning the correction in the higher dimensional space. Although this seems to be a straightforward implementation, it might amount to learning a high dimensional correction vector, especially for two- and three-dimensional domains.

To mitigate the potential computational challenges dealing with excessively large input/output vectors, a *machine learning Correction followed by Prolongation Interface (CPI)* approach

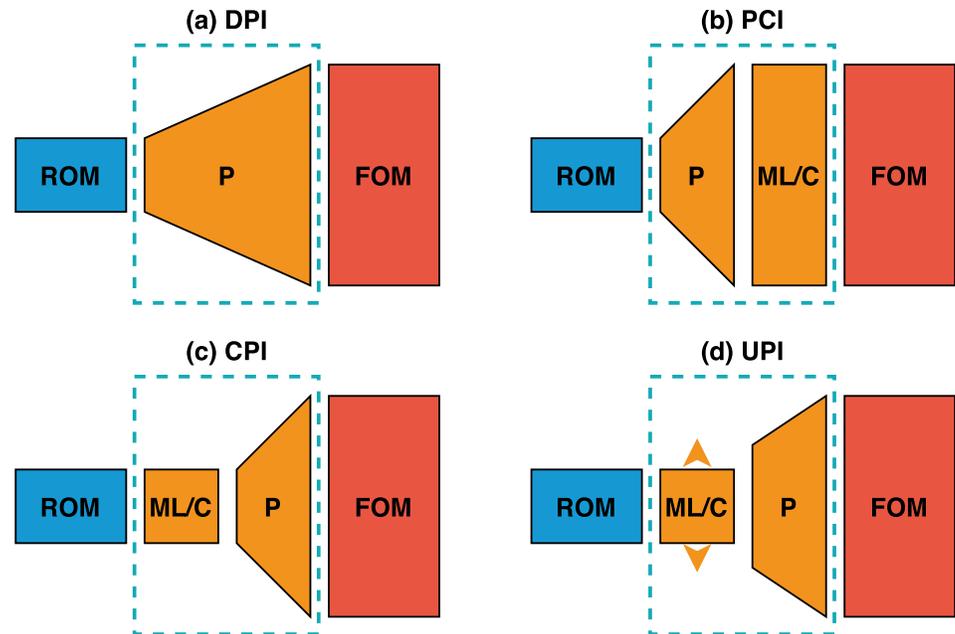


Fig 1. The proposed multifidelity concepts toward hybrid ROM/FOM coupling. Dashed blocks refer to the interface learning approaches introduced in the present work: (a) Direct Prolongation Interface (DPI), (b) Prolongation followed by a machine learning Correction Interface (PCI), (c) machine learning Correction followed by a Prolongation Interface (CPI), and (d) Uplifted Prolongation Interface (UPI) where the latent space is enhanced through machine learning before we apply the prolongation operator.

<https://doi.org/10.1371/journal.pone.0246092.g001>

can be employed to provide a closure effect to remedy the instabilities and inaccuracies of Galerkin ROM due to modal truncation.

For CPI, the LSTM learns the correction terms in ROM space, defined by the number of modes in ROM approximation. As a result, the reconstruction quality will eventually be limited by the Galerkin ROM dimension. Therefore, the *Uplifted Prolongation Interface (UPI)* framework not only corrects the Galerkin ROM solution, but also expands the ROM subspace to enhance the reconstruction quality. Our primary motivation in this paper is to describe and test these four interface modeling approaches to tackle ROM/FOM coupling problems and show how we can elucidate these multifidelity mechanisms within the HAM framework.

ROM-FOM coupling framework

In order to demonstrate the performance of the introduced HAM approaches for ROM-FOM coupling, we consider a coupled system as follows,

$$\frac{\partial u}{\partial t} = f_1(u; \mu_1) + g_1(u, v; \mu_1, \mu_2), \tag{1}$$

$$\frac{\partial v}{\partial t} = f_2(v; \mu_2) + g_2(u, v; \mu_1, \mu_2), \tag{2}$$

where u and v are the coupled variables and g_1 and g_2 define this coupling, while μ_1 and μ_2 denote the set of system’s parameters. We highlight that the coupled variables might represent the state variables at different regions of the domain (e.g., multi-component systems), different physics (e.g., fluid-structure interactions) and/or different scales within the same domain (e.g., multiscale systems). We suppose that the dynamics of u can be approximated by a reduced

order model (ROM) while a full order model resolves v and both solvers need to communicate information to satisfy the coupling. We begin by describing the derivation of a reduced order model of u via Galerkin projection equipped with proper orthogonal decomposition (POD) for basis construction. Then, we formulate the coupling between ROM and FOM solvers.

Reduced order model

Introducing a spatial discretization to Eq (1), it can be rewritten in a semi-discrete continuous-time as follows,

$$\frac{d\mathbf{u}}{dt} = \mathcal{F}(\mathbf{u}, \mathbf{v}; \mu) = \mathcal{L}_1\mathbf{u} + \mathcal{L}_2\mathbf{v} + \mathcal{N}(\mathbf{u}, \mathbf{v}), \tag{3}$$

where the boldfaced symbols represent the arrangement of discretized variables in 1D vector (e.g., $\mathbf{u} \in \mathbb{R}^{n_1}$ and $\mathbf{v} \in \mathbb{R}^{n_2}$, where n_i denotes the spatial resolution), $\mu \in \mathbb{R}^p$ defines the system's parameters, and $\mathcal{F} : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^p \rightarrow \mathbb{R}^{n_1}$ is a deterministic operator with linear and nonlinear components \mathcal{L} , and \mathcal{N} , respectively. These operators depends on the numerical scheme adopted for spatial discretization.

We exploit the advances and developments of ROM techniques to build surrogate models to economically resolve portions of domain and/or physics. The ROM solution can thus be used to infer the flow conditions at the interface so that a FOM solver can be efficiently employed for the sub-domains of interest. The standard Galerkin ansatz is applied for the dynamics of \mathbf{u} as

$$\mathbf{u}(t) \approx \Phi\boldsymbol{\alpha}(t), \tag{4}$$

where the columns of matrix $\Phi = [\phi_1, \phi_2, \dots, \phi_r] \in \mathbb{R}^{n_1 \times r}$ form the orthonormal bases of a reduced subspace with an intrinsic dimension of r , and $\boldsymbol{\alpha}$ defines the projection coordinates associated with Φ . Usually, the basis functions ϕ are constructed to capture the dominant modes or underlying structures of the flow. Proper orthogonal decomposition (POD) is one popular technique to systematically construct Φ such that the solution manifold preserves as much variance as possible when projected onto the subspace spanned by Φ [43–45]. By substituting this approximation into Eq (3) and performing the inner product with Φ , we get the following,

$$\frac{d\boldsymbol{\alpha}}{dt} = \Phi^T \mathcal{L}_1 \Phi \boldsymbol{\alpha} + \Phi^T \mathcal{L}_2 \mathbf{v} + \Phi^T \mathcal{N}(\Phi \boldsymbol{\alpha}, \mathbf{v}). \tag{5}$$

The first coefficient ($\Phi^T \mathcal{L}_1 \Phi$) can be precomputed, so the computational cost for evaluating the linear term depends on r . However, in general, the evaluation of the third term on the right hand side (nonlinear term) depends on the FOM dimension n . Fortunately, most fluid flow systems are characterized by quadratic nonlinear operator, which allows the reduction of Eq (5) into

$$\frac{d\boldsymbol{\alpha}}{dt} = \mathcal{L}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathfrak{N}\boldsymbol{\alpha} + \mathfrak{C}, \tag{6}$$

where \mathcal{L} is an $(r \times r)$ matrix and \mathfrak{N} is an $(r \times r \times r)$ tensor representing the model coefficients while \mathfrak{C} defines the contribution of \mathbf{v} into the ROM of \mathbf{u} . We will see that the computation of \mathfrak{C} may either be computed offline during ROM construction or as part of the online FOM solver of \mathbf{v} with negligible computational overhead. Thus, the floating point operation (flop) count to evaluate the right hand side of the ROM (i.e., Eq (6)) is often $O(r^3)$.

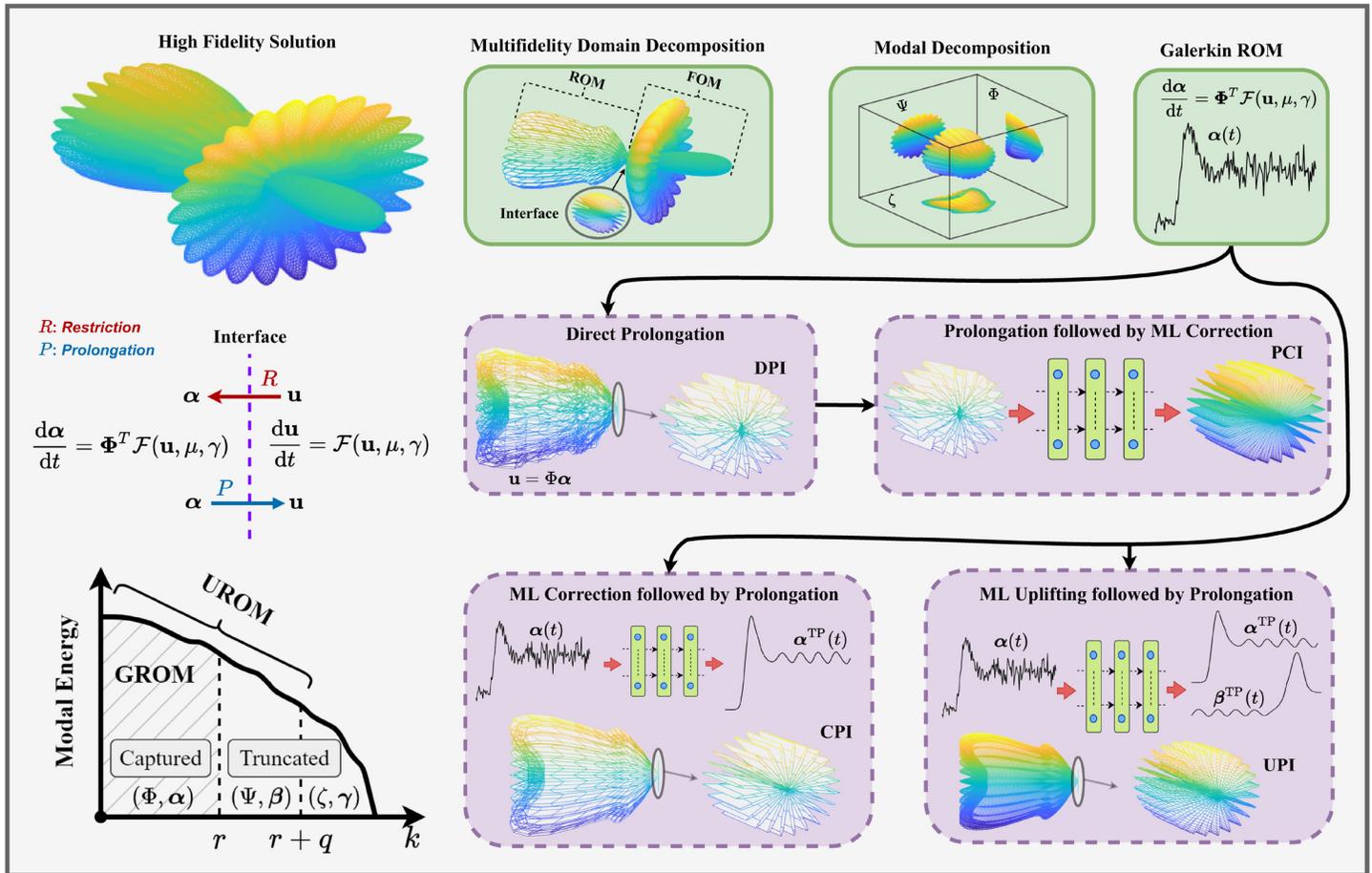


Fig 2. Schematic illustration of the methodologies introduced to utilize ROM to economically provide sound interface conditions in a multifidelity domain decomposition problems. Galerkin ROM yields inaccurate predictions (represented by rough curve), and direct prolongation of these results might be not efficient. PCI adds a correction effect to the prolonged solution in FOM space. Instead, CPI and UPI introduce the correction at ROM level before prolongation. UPI adds an extra superresolution effect to augment solution quality.

<https://doi.org/10.1371/journal.pone.0246092.g002>

In the following, we formulate the four methodologies outlined in Fig 1 to match the ROM solution for α with the FOM solution at the interface. For all cases, a ROM representation is adopted for u , which can be economically solved to compute an estimate of the interface flow condition to feed the FOM solver of v . For example, in multi-component systems like that depicted in Fig 2, the ROM solution at the interface is regarded as a boundary condition for the FOM.

1. *DPI: Direct Prolongation Interface.* The objective of the DPI approach is to recover the flow variables at the interface from the ROM solution (i.e., the time integration of Eq (6)). In other words, we seek to learn a mapping $\mathcal{G}_1 : \mathbb{R}^r \rightarrow \mathbb{R}^d$, that minimizes $\|u^{(i)} - \mathcal{G}_1(\alpha)\|$ where $u^{(i)}$ represent the portion of information at the interface that is shared from the ROM to the FOM solver, with d being the dimension of the interface. For multi-component systems, this interface can be a single point (e.g., for 1D systems), a line (e.g., for 2D systems), or a surface (e.g., for 3D systems). Indeed, this prolongation map naturally results from the Galerkin ansatz, and can be written as

$$\mathcal{G}_1(\alpha) = \Theta\alpha, \tag{7}$$

where Θ represents the portion of the basis Φ that is computed at the interface location. Since the ROM approximation is built upon the assumption of representing the flow within a low order subspace, the approximation given by Eq (4) basically introduces a projection error. This error can be significant for complex systems, where the flow dynamics are characterized by a wide spectrum while only few modes are considered to minimize the computational burden of solving the ROM. Moreover, the nonlinear interactions as well as the modal truncation coupled with the Galerkin projection methodology usually cause Eq (6) to yield erroneous predictions of the coefficients $\alpha(t)$. Therefore, the solution from the DPI approach is potentially inaccurate [46]. Consequently, the reconstruction $\mathcal{G}_1(\alpha)$ is no longer optimal and a correction needs to be introduced.

2. *PCI: Prolongation followed by Correction Interface.* The PCI framework aims to correct the mapping \mathcal{G}_1 to yield more accurate interface condition. To do so, we utilize a long short-term memory (LSTM) neural network to learn a mapping $\mathcal{G}_2 : \mathbb{R}^i \rightarrow \mathbb{R}^i$ such that

$$\mathcal{G}_2(\mathcal{G}_1(\alpha)) = u^{(i)} - \mathcal{G}_1(\alpha). \tag{8}$$

In other words, LSTM is fed with a predictor of $u^{(i)}$ obtained by DPI and approximates the deviation of this value from the true state variables at the interface. Hence, this deviation estimate can be added as a correction term in a predictor-corrector fashion. In PCI, both inputs and outputs of the LSTM lie in the FOM space and thus the LSTM map can be considered as nudging scheme from the ROM prolongation \mathcal{G}_1 to the FOM solution [47]. We highlight that the PCI approach can be feasible for one-dimensional (1D) problems (where the interface can be just a single point). However, for higher dimensional systems, the sizes of input and output vectors grow rapidly (unless a too coarse mesh is adopted). For such cases, PCI becomes prohibitive, and the learning and correction should be performed in a reduced latent space instead.

3. *CPI: Correction followed by Prolongation Interface.* The CPI methodology works by introducing the correction in the latent subspace, rather than the FOM space. This is especially crucial for 2D and 3D configurations. In particular, the CPI aims at curing the deviation in modal coefficients predicted from solving the Galerkin ROM equations, known as closure error. Due to the modal cut-off in ROM approximation, Eq (6) does not necessarily capture the true projected trajectory of $\alpha(t)$. Therefore, we introduce an LSTM mapping $\mathcal{G}_3 : \mathbb{R}^r \rightarrow \mathbb{R}^r$ to provide a closure effect to adjust the Galerkin ROM trajectory. Specifically, the LSTM for CPI takes the values of modal coefficients acquired from integrating Eq (6) in time and predicts the discrepancy between these values and their optimal values. Those are defined by the true projection (TP) of the FOM solution onto the basis functions as follows,

$$\alpha^{TP} = \Phi^T \mathbf{u}. \tag{9}$$

Therefore, the CPI contribution can be written as

$$\mathcal{G}_3(\alpha) = \Phi^T \mathbf{u} - \alpha. \tag{10}$$

We highlight here that the size of the input and output vectors is $O(r)$, independent of the FOM resolution, which offers a potential flexibility dealing with 2D and 3D problems. Once the modal coefficients are corrected, they are prolonged from the ROM space to the FOM space using the reconstruction map \mathcal{G}_1 . For all results, we also show the results obtained from the true projection of FOM solution onto the ROM subspace at the interface

as,

$$\mathbf{u}^{\text{TP}} = \mathcal{G}_1(\boldsymbol{\alpha}^{\text{TP}}). \tag{11}$$

We highlight that u^{TP} represents the best approximation of the true flow field that can be achieved using a linear subspace with an intrinsic dimension of r .

4. *UPI: Uplifted Prolongation Interface.* Although the CPI methodology cures the closure error and provides a stabilized solution, it does not address projection error. Unless a large number of modes are resolved, the projection error can be significant, especially for problems with discontinuities and shocks. To deal with those situations, an uplifting ROM has been proposed [46], where both closure and projection errors are taken care of. For closure, similar to CPI, the Galerkin ROM predictions are tuned to match their true projection values. In addition, following Galerkin ROM solution, the ROM subspace is expanded to capture some of the smaller scales missing in the initial subspace as follows,

$$\mathbf{u} \approx \Phi\boldsymbol{\alpha} + \Psi\boldsymbol{\beta}, \tag{12}$$

where the columns of $\Psi = [\psi_1, \psi_2, \dots, \psi_q]$ form orthonormal basis functions for a q -dimensional subspace complementing that spanned by Φ and $\boldsymbol{\beta}$ defines the corresponding projection coordinates. Similar to Φ , Ψ can be computed through the POD algorithm. Note that Φ and Ψ are orthogonal to each others (i.e., $\Phi^T \Psi = \Psi^T \Phi = \mathbf{0}$). Indeed, Ψ represents the next q basis functions generated by POD after the first r being dedicated to Φ . Those are also constructed a priori during an offline stage using the collected set of snapshot data. We highlight that the Galerkin ROM equations only solve for $\boldsymbol{\alpha}$ to keep the computational cost as low as possible.

Therefore, a complementary model for $\boldsymbol{\beta}$ has to be constructed so that the uplifting approach can be employed. To accomplish this, a mapping from the first r modal coefficients to the next q modes is assumed to exist. Nonlinear Galerkin projection has been pursued to express this mapping as $\boldsymbol{\beta} = \mathcal{H}(\boldsymbol{\alpha})$, but it has been found challenging for most systems [48]. Instead, we exploit the LSTM learning capabilities to infer this map from data. This uplifting approach enhances the quality of prolonged solution by providing a superresolution effect. In particular, the UPI architecture is trained to read the Galerkin ROM prediction for the first r modal coefficients as input, and return the true coefficients of the first $r + q$ modes. Thus, it provides a closure effect for the first r modes and a superresolution effect for the next q modes, simultaneously in a single network as follows,

$$\mathcal{G}_4 : \mathbb{R}^r \rightarrow \mathbb{R}^{r+q} \tag{13}$$

$$\mathcal{G}_4(\boldsymbol{\alpha}) = \begin{bmatrix} \boldsymbol{\alpha}^{\text{TP}} \\ \boldsymbol{\beta}^{\text{TP}} \end{bmatrix}. \tag{14}$$

We note here that the first $r + q$ spatial modes have to be built and stored beforehand, which introduces slightly more storage costs. For the present study, we explore the specific case where $q = r$, but a generalization is straightforward.

The ROM-FOM coupling philosophy as well as the introduced interface learning approaches are summarized in the cartoon shown in Fig 2. These methodologies are also applicable to a wide range of computational problems with multifidelity domain decomposition. The depicted system is assumed to be fully characterized by three mutually orthogonal sets of

basis functions, namely Φ , Ψ , and ζ as below

$$\mathbf{u} = \Phi\boldsymbol{\alpha} + \Psi\boldsymbol{\beta} + \zeta\gamma, \quad (15)$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and γ are the corresponding projection coordinates. We also suppose that Galerkin ROM resolves the Φ set of modes (i.e., truncating the contributions of Ψ , and ζ). We reiterate here that the Galerkin ROM yields inaccurate solution (sketched by the noisy (rough) curve of predicted $\boldsymbol{\alpha}$). Consequently, the quality of the direct prolongation mapping is compromised. The PCI aims at correcting the reconstructed solution at the interface. Even though the PCI technique acts only on a small portion of the domain (i.e., the interface), it might amount to excessively large input and output sizes.

On the other hand, CPI treats the Galerkin ROM deficiencies at the ROM level. In particular, it introduces a closure effect to better predict $\boldsymbol{\alpha}$. This closure simply compensates the effects of truncated modes (i.e., Ψ and ζ) onto the dynamics of Φ . This yields a better estimate of $\boldsymbol{\alpha}$, as illustrated by the smooth curve in Fig 2. We note that in CPI, the effects of Ψ and ζ are only considered to improve the prediction of $\boldsymbol{\alpha}$. However, their contributions to the solution manifold reconstruction are not included, resulting in a substantial reconstruction error (projection error). To deal with this caveat, UROM seeks to add a superresolution enhancement by incorporating the Ψ set of basis into the reconstruction step by learning the dynamics of the corresponding $\boldsymbol{\beta}$ coordinates. This is represented by a higher resolution (denser) reconstruction in UPI case, compared to CPI, and DPI. Note that the PCI is still showing the highest resolution (the densest reconstruction) as it nudges the prediction at the interface to its FOM counterpart (i.e., including all Φ , Ψ , and ζ).

Demonstrations

We demonstrate the ROM-FOM coupling methodologies using two examples of varying complexities. In the first one, we describe a fluid flow scenario over a bizonal domain with heterogeneous physical properties using the one-dimensional Burgers problem. For this case, we shall see that the interface between different sub-domains is defined by a single point (i.e., $d = 1$). Second, we consider the Marsigli flow problem represented by the two-dimensional Boussinesq equations to demonstrate the ROM-FOM coupling for multiphysics systems. In particular, a ROM solver is devoted for the mass and momentum transport equations while a FOM is reserved for the energy transport.

The one-dimensional Burgers problem

In order to represent a zonal multifidelity simulation, we consider the following one dimensional (1D) viscous Burgers problem,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{\partial}{\partial x} \left(v\frac{\partial u}{\partial x} \right) - \gamma u, \quad (16)$$

$$(v, \gamma) = \begin{cases} (10^{-2}, 0) & \text{for } 0 \leq x \leq x_b \\ (10^{-4}, 1) & \text{for } x_b < x \leq 1, \end{cases} \quad (17)$$

where x_b is the spatial location of the interface defining the physical heterogeneity. We highlight that Eq (16) includes the $\frac{\partial}{\partial x} \left(v\frac{\partial u}{\partial x} \right)$ term instead of the commonly used $v\frac{\partial^2 u}{\partial x^2}$ term to permit the spatial variation of v . For the given setup, the stiffness and physical properties in the left part dictates higher spatial as well as temporal resolutions than those required for the right

partition. If we opt to a global unified (unique fidelity) solver over the whole domain, the quality of the solution will be dominated by stiffness of the left zone. In other words, a smaller time step would be required to satisfy numerical stability when using an explicit temporal integration scheme. Specifically, assuming that we utilize the forward in time and central in space finite difference scheme (FTCS) to solve Eq (16) with a spatial grid resolution of 4096, a time step of approximately 2.5×10^{-6} will be required for the left part of domain, while a time step of 2.5×10^{-4} would be sufficient if we were able to only resolve the right part. Therefore, domain decomposition approaches might be adopted to segregate partitions with varying numerical requirements. Despite the effectiveness and efficiency of these approaches, idle delays can arise in order to accommodate information transfer from the left zone to the right zone through their common interface. Instead, low-fidelity proxy models can be utilized to avoid such lags by approximating the *effective* dynamics of stiff regions and providing sound interface boundary conditions to the rest of the computational domain.

The discretized domain is divided into a left zone with $\mathbf{u}^L \in \mathbb{R}^{n_1}$ for $x \in [0, x_b]$ and a right zone with $\mathbf{u}^R \in \mathbb{R}^{n_2}$ for $x \in [x_b, 1]$, where $n_1 + n_2 = n + 1$. We build a reduced order model for the left sub-domain as follows,

$$\mathbf{u}^L(t) = \Phi \boldsymbol{\alpha}(t), \tag{18}$$

$$\frac{d\boldsymbol{\alpha}}{dt} = \mathcal{L}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathfrak{N}\boldsymbol{\alpha}, \tag{19}$$

where \mathcal{L} and \mathfrak{N} represent the tensorial ROM coefficients which can be precomputed during the offline stage as,

$$\mathcal{L}_{i,k} = \left\langle \frac{\partial}{\partial x} \left(v \frac{\partial u}{\partial x} \right) - \gamma \phi_i; \phi_k \right\rangle, \tag{20}$$

$$\mathfrak{N}_{i,j,k} = \left\langle -\phi_i \frac{\partial \phi_j}{\partial x}; \phi_k \right\rangle, \tag{21}$$

where the angle parentheses denote the inner product (i.e., $\langle \mathbf{a}; \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b}$).

For data generation, we solve the full order model representing the 1D Burgers equation (Eq (16)) over the entire domain using a spatial mesh resolution of 4096 and time step of 2.5×10^{-6} . For initial condition, we consider a square-like wave defined as,

$$u(x, 0) = 0.5 - 0.5 \tanh \frac{x - x_b}{\epsilon}, \tag{22}$$

where a value of $x_b = 0.75$ is considered as the location of the interface and $\epsilon = 0.005$ is used to define the sharpness of the shock at x_b . Dirichlet boundary conditions are assumed at both boundaries (i.e., $u(0, t) = u(1, t) = 0$). We compute the evolution of the velocity field for $t \in [0, 2]$, and collect snapshots every 100 time steps. That is snapshots are collected every 2.5×10^{-4} time units (working with normalized variables).

For ROM construction, we consider the truncated solution snapshots for the left part of the domain (i.e., $0 \leq x \leq 0.75$) for $t \in [0, 1]$. For POD basis generation, we use only 200 snapshots distributed evenly from $t = 0$ to $t = 1$, to reduce the computational cost of solving the corresponding eigenvalue problem. Once ROM is constructed, it is integrated in time with a time step of 2.5×10^{-4} to match the time step in the right part of the domain (to be handled via a FOM solver). During the deployment phase of the coupled system, the ROM feeds the FOM solver with the boundary condition at x_b . On the other hand, the effects of the interface on

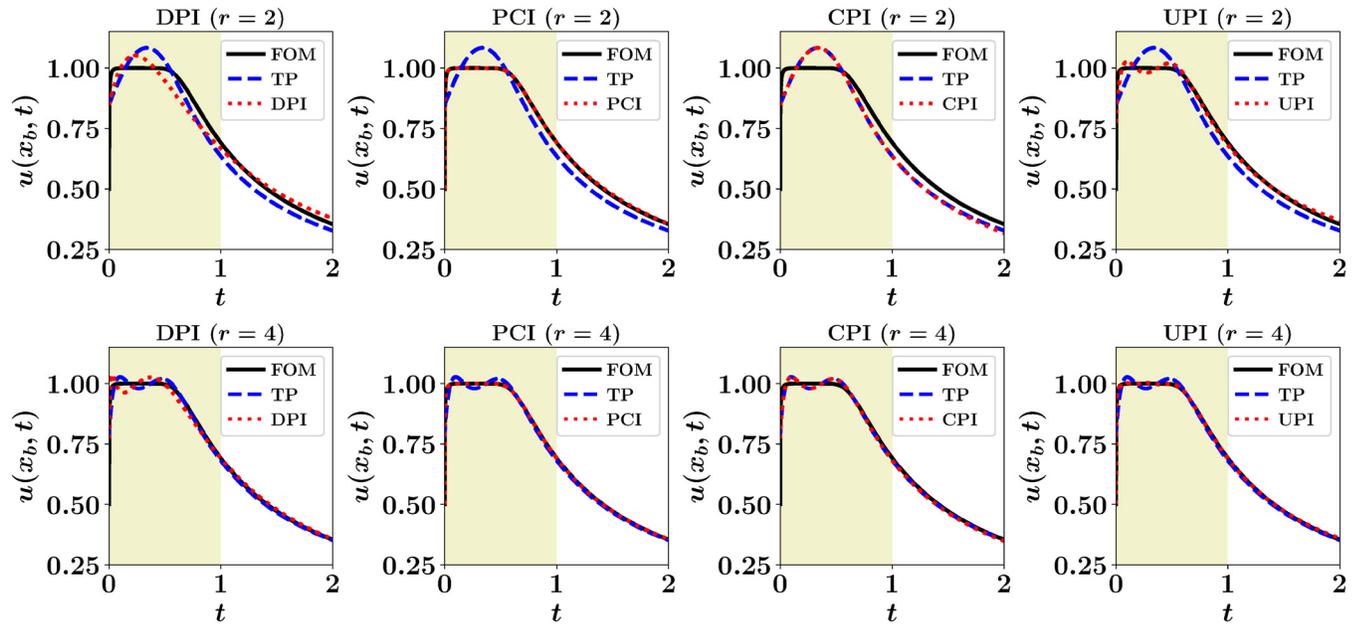


Fig 3. Velocity at the interface obtained by considering ROM in the left part of the domain, with $r = 2$ (top) and $r = 4$ (bottom). FOM solution corresponds to solving the governing equation over the entire domain, while TP denotes the projection of the FOM solution in the left zone onto the corresponding POD subspace.

<https://doi.org/10.1371/journal.pone.0246092.g003>

the ROM dynamics are considered during the offline stage of data generation and basis construction.

We utilize labelled data at the time interval of $t \in [0, 1]$ for the training and validation of the LSTM neural networks. In particular, four fifths of the collected data set are randomly selected for training while the remaining one fifth is reserved for validation purposes (to avoid overfitting). We highlight that numerical experiments reveal that the performance in our case is not significantly sensitive to the neural network hyperparameters. Thus, we manually tune the hyperparameters during the training and validation stages by repeating the previous procedure multiple times while varying the seeds for the random number generator and using averaged loss values for assessment. We adopt an LSTM architecture of 2 layers with 20 cells in each layer. In the meantime, we note that automated hyperparameter search tools can be utilized for optimized architectures. Testing of the proposed schemes is performed for $t \in [0, 2]$, corresponding to a temporal extrapolation behavior with respect to the training interval.

Since the coupling between ROM and FOM is represented by the physical interface at x_b , we notice that the Θ in the DPI map (i.e., $\mathcal{G}_1(\alpha) = \alpha$) is simply the last row of the matrix Φ . In Fig 3, we plot the velocity at the interface obtained from adopting ROM for the left part of the domain, and corrected with machine learning architectures as described before for $r = 2$ and $r = 4$. We note that FOM solution corresponds to the velocity at the interface obtained by applying a FOM solver all over the domain using a time step of 2.5×10^{-6} , while the true projection (TP) represents the projection of the truncated velocity field in the left zone onto the POD subspace. The shaded area in Fig 3 stands for the time interval utilized for POD basis generation, ROM formulation, and LSTMs training.

It can be seen that DPI results, especially for $r = 2$, are not very accurate due to the mutual effects of modal truncation and model nonlinearity in Galerkin ROM. On the other hand, the PCI solution gives almost perfect match with FOM. As the PCI approach nudges the prolonged ROM solution to its FOM counterpart, it gives even higher accuracy than TP. That is

Table 1. ℓ_2 norm for the deviation of the velocity at the interface with respect to its FOM value for $t \in [0, 2]$.

Setup	$r = 2$	$r = 4$	$r = 8$
TP	4.56	1.25	0.21
DPI	3.67	1.46	0.44
PCI	0.23	0.07	0.03
CPI	4.57	1.27	0.23
UPI	1.28	0.24	0.11

<https://doi.org/10.1371/journal.pone.0246092.t001>

TP is limited by the maximum quality that can be obtained using a rank- r approximation. For CPI, since the LSTM introduces a closure effect, it steers the ROM results to match the TP solution. Finally, the UPI recovers some of the smaller scales (truncated modes) so it yields better reconstruction than TP since it spans a larger subspace. For $r = 2$, UPI uplifts the solution to a rank-4 approximation, while for $r = 4$, it is uplifted to a rank-8 approximation.

For quantitative assessment, we document the ℓ_2 norm of the deviation of the predicted velocity at the interface compared to the FOM solution in Table 1. Results are reported for $r \in \{2, 4, 8\}$. We see that the error in the CPI case almost matches that of TP, while PCI gives the highest accuracy since it is trained to learn the correction with respect to the FOM solution. Also, CPI results are significantly close to TP, illustrating the closure effect introduced by CPI to account for the effect of modal truncation on ROM dynamics. Another interesting observation is that UPI quality at a given value of r is equivalent to TP with twice that value. This indicates that UPI is able to give a superresolution effect up to $2r$ (since we select $q = r$). Also, we notice that at $r = 2$, DPI yields lower ℓ_2 norm than TP. This is because TP solution is obtained by the projection of the FOM solution onto the POD subspace generated using data at $t \in [0, 1]$. So, this subspace is optimal only for $t \in [0, 1]$, while testing is performed up to $t = 2$. Therefore, TP solution no longer represents the best rank- r approximation beyond $t = 1$.

Finally, we investigate the coupling efficiency by solving the right part (i.e., $0.75 \leq x \leq 1$) using a high fidelity FOM solver applied only onto this subdomain. This is fed with a boundary condition $u(0.75, t)$ from the low-fidelity interface learning approaches described before. Fig 4 shows the spatiotemporal velocity profile with $r = 2$, compared to the FOM predictions. Again, we observe that the solution with PCI boundary is similar to this FOM solution. Also, CPI matches the TP results, but they both smooth-out the surface peak because of the low rank limitations. Although it seems that the accuracy of UPI cannot exceed that of CPI with $r + q$ assuming optimal performance for both CPI and UPI, there is a computational side in this comparison. The CPI with $r + q$ modes implies the solution of a Galerkin ROM with a dimension of $r + q$, while the UPI requires the solution of a Galerkin ROM with r . We have seen that for fluid flows with quadratic nonlinearity, the computational cost of solving a Galerkin ROM scales cubically with the number of modes. Thus, the implementation of CPI with $r + q$ with $q = r$ is about 8 times more costly than UPI with $r + q$. At this point, we highlight that the selection of r and q is highly dependent on the problem in hand, the corresponding decay of POD eigenvalues, and the level of accuracy sought. A compromise between computational cost of solving a Galerkin ROM with r and the corresponding stability, the amount of information captured by r and $r + q$ modes, and reliability of UPI with $r + q$ is always in place.

The two-dimensional Boussinesq problem

Boussinesq equations represent a simple approach for modeling geophysical waves such as oceanic and atmospheric circulations induced by temperature differences [49] as well as other situations, like isothermal flows with density stratification. In the Boussinesq approximation,

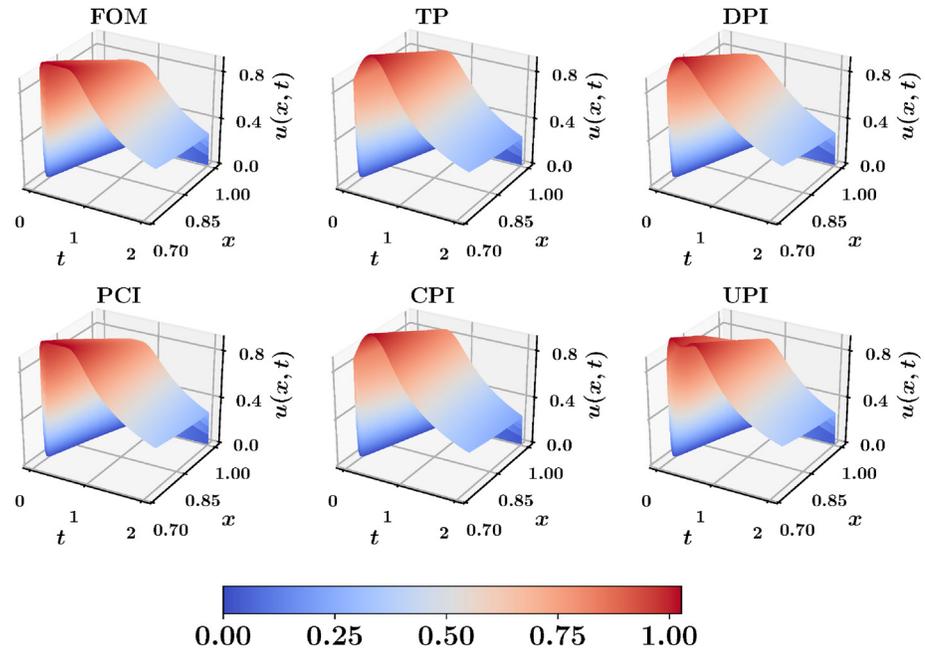


Fig 4. Spatio-temporal velocity profile obtained from applying high fidelity (FOM) solver onto the right subdomain ($0.75 \leq x \leq 1$), fed with interface boundary from a low-fidelity (ROM) solution with $r = 2$.

<https://doi.org/10.1371/journal.pone.0246092.g004>

variations of all fluid properties other than the density are neglected completely. Moreover, the density dependence is ignored in all terms except for gravitational force (giving rise to buoyancy effects). As a result, the continuity equation can be adopted in its constant density form, and the momentum equation can be simplified significantly. The dimensionless form of the 2D incompressible Boussinesq equations in vorticity-streamfunction formulation can be represented by the following two coupled transport equations [50, 51],

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \frac{1}{\text{Re}} \nabla^2 \omega + \text{Ri} \frac{\partial \theta}{\partial x}, \tag{23}$$

$$\frac{\partial \theta}{\partial t} + J(\theta, \psi) = \frac{1}{\text{RePr}} \nabla^2 \theta, \tag{24}$$

where ω , ψ and θ denote the vorticity, streamfunction and temperature fields, respectively. In Boussinesq flow systems, there are three leading physical mechanisms, namely viscosity, conductivity, and buoyancy, giving rise to three dimensionless numbers; Reynolds number Re , Richardson number Ri , and Prandtl number Pr .

We utilize the 2D Boussinesq equation to illustrate the ROM-FOM coupling for multiphysics situations. In particular, we suppose that we are more interested in the temperature field predictions. Thus, we dedicate a FOM solver for Eq (24). However, we see that the solution of this equation requires evaluating the streamfunction field at each time step. The kinematic relationship between vorticity and streamfunction is given by the following Poisson equation,

$$\nabla^2 \psi = -\omega, \tag{25}$$

which implies that the streamfunction is not a prognostic variable, and can be computed from the vorticity field at each timestep. In typical simulations, the solution of Eq (25) consumes significant amount of time and computational resources and is considered the bottleneck for

most incompressible flow solvers. Therefore, we can consider a ROM solver for the vorticity dynamics as follows,

$$\omega(x, y, t) = \bar{\omega}(x, y) + \sum_{k=1}^r \alpha_k(t) \phi_k^\omega(x, y), \tag{26}$$

where $\bar{\omega}$ denotes the time-averaged vorticity field and the POD is performed on the fluctuating part of ω . We note that Eq (25) allows us to assume the same modal coefficients $\alpha_k(t)$ for both ω and ψ as follows,

$$\psi(x, y, t) = \bar{\psi}(x, y) + \sum_{k=1}^r \alpha_k(t) \phi_k^\psi(x, y), \tag{27}$$

where the time-averaged streamfunction field and the corresponding basis functions can be computed from the following relations,

$$\nabla^2 \bar{\psi}(x, y) = -\bar{\omega}(x, y), \tag{28}$$

$$\nabla^2 \phi_k^\psi(x, y) = -\phi_k^\omega(x, y). \tag{29}$$

Thus, the Galerkin ROM of Eq (23) can be written as

$$\frac{d\alpha_k}{dt} = \mathfrak{B}_k + \sum_{i=1}^R \mathfrak{L}_{i,k}^{(\omega,\psi)} \alpha_i + \sum_{i=1}^R \mathfrak{L}_{i,k}^{(\omega,\theta)} \beta_i + \sum_{i=1}^R \sum_{j=1}^R \mathfrak{N}_{i,j,k}^{(\omega,\psi)} \alpha_i \alpha_j, \tag{30}$$

where β represent the projection of the temperature fields onto the reduced subspace defined as

$$\beta_k(t) = \langle \theta(x, y, t_n) - \bar{\theta}(x, y); \phi_k^\theta(x, y) \rangle, \tag{31}$$

where the $\bar{\theta}$ denotes the time-averaged field of θ and Φ^θ are the corresponding orthonormal POD modes. The predetermined coefficients in Eq (30) are calculated as follows,

$$\mathfrak{B}_k = \langle -J(\bar{\omega}, \bar{\psi}) + \frac{1}{\text{Re}} \nabla^2 \bar{\omega} + \text{Ri} \frac{\partial \bar{\theta}}{\partial x}; \phi_k^\omega \rangle, \tag{32}$$

$$\mathfrak{L}_{i,k}^{(\omega,\psi)} = \langle \frac{1}{\text{Re}} \nabla^2 \phi_i^\omega - J(\phi_i^\omega, \bar{\psi}) - J(\bar{\omega}, \phi_i^\psi); \phi_k^\omega \rangle, \tag{33}$$

$$\mathfrak{L}_{i,k}^{(\omega,\theta)} = \langle \text{Ri} \frac{\partial \phi_i^\theta}{\partial x}; \phi_k^\omega \rangle, \tag{34}$$

$$\mathfrak{N}_{i,j,k}^{(\omega,\psi)} = \langle -J(\phi_i^\omega, \phi_j^\psi); \phi_k^\omega \rangle. \tag{35}$$

We notice that the ROM defined by Eq (30) equipped with Eq (27) can be adopted to approximate the instantaneous streamfunction field, which is required to solve Eq (24) for the temperature in FOM space. On the other hand, the solution of the FOM (i.e., Eq (24)) along with Eq (31) feeds the ROM solver with β values. This constitutes a two-way ROM-FOM coupling problem, in contrast to the one-way coupling in the aforementioned 1D Burgers example. We also highlight that the computational cost of the projection step (i.e., Eq (31)) is minimal compared to the solution of Eq (24).

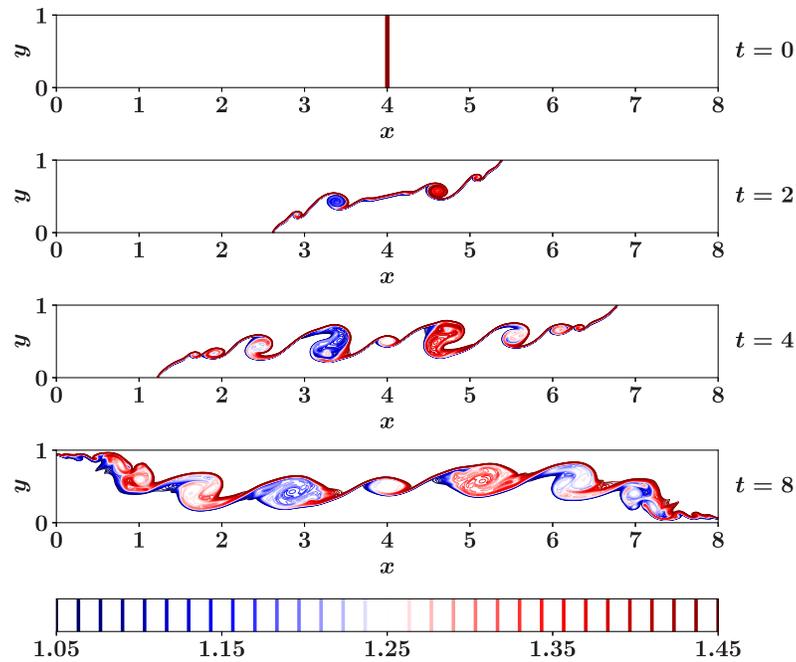


Fig 5. Temperature field at different time instances for 2D Boussinesq problem using 4096×512 grid and $\Delta t = 0.0005$.

<https://doi.org/10.1371/journal.pone.0246092.g005>

For demonstration, we consider a strong-shear flow exhibiting the Kelvin-Helmholtz instability, known as Marsigli flow or lock-exchange problem. The physical process in this flow problem explains how differences in temperature/density can cause currents to form in the ocean, seas and natural straits. For example, Marsigli discovered that the Bosphorus currents are a consequence of the different water densities in the Black and Mediterranean seas [52]. Basically, when fluids of two different densities meet, the higher density fluid slides below the lower density one. This is one of the primary mechanisms by which ocean currents are formed [53].

The problem is defined by two fluids of different temperatures, in a rectangular domain $(x, y) \in [0, 8] \times [0, 1]$ with a vertical barrier dividing the domain at $x = 4$, keeping the temperature, θ , of the left half at 1.5 and temperature of the right half at 1. Initially, the flow is at rest (i.e., $\omega(x, y, 0) = \psi(x, y, 0) = 0$), with uniform temperatures at the right and left regions (i.e., $\theta(x, y, 0) = 1.5 \forall x \in [0, 4]$ and $\theta(x, y, 0) = 1 \forall x \in (4, 8]$). Free slip boundary conditions are assumed for flow field, and adiabatic boundary conditions are prescribed for temperature field. Reynolds number of $Re = 10^4$, Richardson number of $Ri = 4$, and Prandtl number of $Pr = 1$ are set in Eqs (23) and (24). A Cartesian grid of 4096×512 , and a timestep of $\Delta t = 5 \times 10^{-4}$ are used for the FOM simulations. Standard second-order central finite difference schemes are adopted for the discretization of linear terms while the second order Arakawa scheme [54] is used to compute the Jacobian term. The evolution of the temperature field is shown in Fig 5 at $t = 0, 2, 4, 8$. At time zero, the barrier is removed instantaneously triggering the lock-exchange problem, where the higher density fluid (on the right) slides below the lower density fluid (on the left) causing an undercurrent flow moving from right to left. Conversely, an upper current flow moves from left to right, causing a strong shear layer between the counter-current flows. As a result, vortex sheets are produced, exhibiting the Kelvin-Helmholtz instability.

Considering the dimensionality of this problem, we emphasize that the PCI approach becomes highly unfeasible. For example, with the current mesh resolution (i.e., 4096×512), the dimension of the state space of the prolonged interface is $\approx 2 \times 10^6$. Building and training of LSTMs with an input and output vector sizes of two millions become prohibitive. Even the training of convolutional neural network with such high resolutions (which are typical in fluid flow simulations) requires excessive computational resources. Therefore, we present results for ROM-FOM coupling using approaches that operate in ROM space (i.e., DPI, CPI and UPI). We note that 800 time snapshots are stored for POD basis construction. For the Galerkin ROM solver, $r = 8$ modes are retained. We also utilize the dataset of the stored 800 snapshots for LSTM training and validation (80% randomly selected for training and the rest for validation, similar to the previous example). A two-layer LSTM with 20 cell in each layer constitutes our LSTM architecture. During the testing phase, the trained neural networks are deployed at each and every timestep. This corresponds to the application of the presented approach 16000 times.

Fig 6 shows the predictions of the temperature field at final time (i.e., $t = 8$) computed from DPI, CPI, and UPI approaches compared to the FOM field. We emphasize that the ROM-FOM coupling results correspond to the solution of the vorticity equation with a ROM solver, which feeds the FOM solver with streamfunction to solve the temperature equation only as opposed to the FOM results which comes from the solution of both the 2D Boussinesq equations using a FOM simulation. Although the CPI results are better than those of DPI, we can observe that the fine details of the flow field are not accurately captured. That is 8 modes are not sufficient to sufficiently represent the flow field. This is a common problem for convection-dominated flows which exhibit slow decay in the POD eigenvalues and the generated global basis functions suffer from modal deformation [55]. On the other hand, the implementation of the UPI approach with $r = 8$ and $q = 8$ recovers an increased amount of the fine flow structures that are not well-represented by the first $r = 8$ modes.

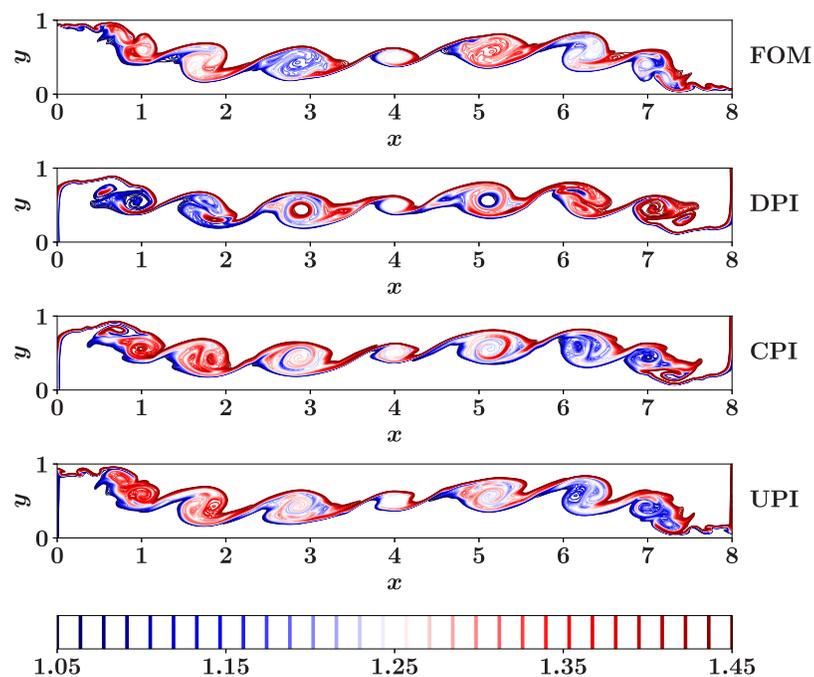


Fig 6. Final temperature fields as obtained from different ROM-FOM coupling approaches, compared to the FOM solution. We note that the PCI becomes infeasible for higher dimensional systems.

<https://doi.org/10.1371/journal.pone.0246092.g006>

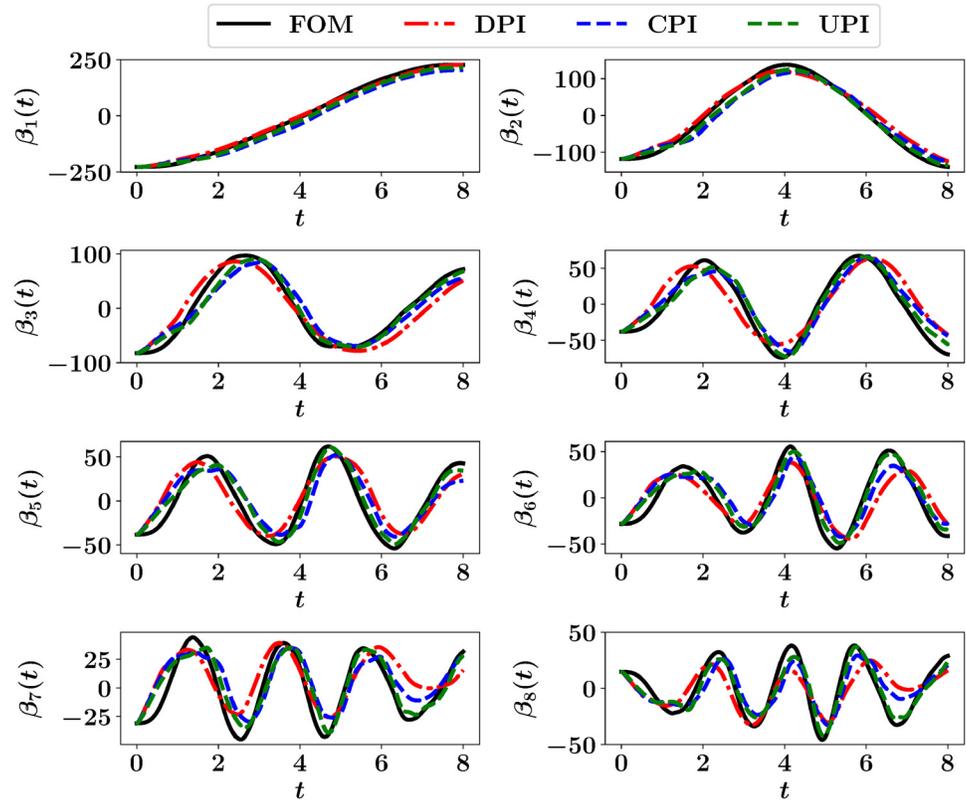


Fig 7. Projection of the predicted temperature fields at different times from FOM, DPI, CPI and UPI onto the POD basis function.

<https://doi.org/10.1371/journal.pone.0246092.g007>

Although the ROM is built for the vorticity equation only, the basis functions of the temperature fields should be generated as well to carry-out the coupling from FOM to ROM. Moreover, in order to illustrate the temporal variation of the coupling quality, we project the resolved temperature fields at different times onto their POD basis. This is depicted in Fig 7, showing the effect of different approaches on the resulting predictions of temperature fields. The FOM trajectory corresponds to the solution of both the 2D Boussinesq equations in FOM space, then projecting the obtained fields on the basis functions of θ (see Eq (31)). For the rest, the streamfunction fields are obtained from ROM predictions and fed into FOM solver to compute the temperature fields.

Conclusions

We provide an interface learning approach via ROM-FOM coupling for multifidelity simulation environments. This learning paradigm is built with a hybrid analysis and modeling (HAM) framework to enhance the ROM approximation of interface conditions. A demonstration with a bizonal 1D Burgers problem is considered to assess the performance of the introduced learning schemes for multi-component systems. For 1D problems, we find that a prolongation followed by a machine learning correction interface (PCI) yields very good predictions. However, this might be unfeasible for 2D and 3D cases, where a correction in ROM subspace is essential. For such, a machine learning correction in ROM space followed by a prolongation interface (CPI) can produce sufficient accuracy. For complex systems where the projection error is significantly large, an uplifted prolongation interface (UPI) methodology can

be adopted to recover some of the truncated scales. This is further illustrated using the lock-exchange problem governed by the 2D Boussinesq problem, where the ROM and FOM solvers address the vorticity and temperature equations, respectively. The coupling from ROM to FOM is represented by the 2D streamfunction fields reconstructed from the ROM solver, saving the run time for Poisson equation, which is the most demanding part of an incompressible flow solver. Owing to the relative simplicity, robustness and ease of these interface learning methods, we expect a growing number of applications in a large variety of interfacial problems in science and engineering. Of particular interest, this ROM-FOM coupling could be a viable method for developing next generation digital twin technologies.

Acknowledgments

O.S. gratefully acknowledges the U.S. DOE Early Career Research Program support. A.R. acknowledges support from Operational Control for Wind Power Plants (OPWIND) project and its industries partners (Equinor, Vestas, Vattenfall).

Author Contributions

Conceptualization: Omer San.

Data curation: Shady E. Ahmed.

Formal analysis: Shady E. Ahmed.

Funding acquisition: Omer San.

Investigation: Shady E. Ahmed.

Methodology: Shady E. Ahmed, Omer San, Kursat Kara, Rami Younis, Adil Rasheed.

Supervision: Omer San.

Validation: Shady E. Ahmed.

Visualization: Shady E. Ahmed.

Writing – original draft: Shady E. Ahmed.

Writing – review & editing: Shady E. Ahmed, Omer San, Kursat Kara, Rami Younis, Adil Rasheed.

References

1. Peherstorfer B, Willcox K, Gunzburger M. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*. 2018; 60(3):550–591. <https://doi.org/10.1137/16M1082469>
2. Bjørkevoll KS. Use of High Fidelity Models for Real Time Status Detection with Field Examples from Automated MPD Operations in the North Sea. In: *The 2nd IFAC Workshop on Automatic Control in Off-shore Oil and Gas Production*; 2015.
3. Macpherson J. Technology Focus: Drilling Management and Automation. *Journal of Petroleum Technology*. 2015; 67(09):136–136. <https://doi.org/10.2118/0915-0136-JPT>
4. Reynolds WC, Parekh DE, Juvet PJD, Lee MJD. Bifurcating and blooming jets. *Annual Review of Fluid Mechanics*. 2003; 35(1):295–315. <https://doi.org/10.1146/annurev.fluid.35.101101.161128>
5. Tyliczszak A. Multi-armed jets: A subset of the blooming jets. *Physics of Fluids*. 2015; 27(4):041703. <https://doi.org/10.1063/1.4917179>
6. Gohil TB, Saha AK, Muralidhar K. Simulation of the blooming phenomenon in forced circular jets. *Journal of Fluid Mechanics*. 2015; 783:567–604. <https://doi.org/10.1017/jfm.2015.571>
7. Phillips E, Wygnanski I. Use of sweeping jets during transient deployment of a control surface. *AIAA Journal*. 2013; 51(4):819–828. <https://doi.org/10.2514/1.J051683>

8. Koklu M. Effect of a Coanda extension on the performance of a sweeping-jet actuator. *AIAA Journal*. 2016; 54(3):1131–1134. <https://doi.org/10.2514/1.J054448>
9. Koklu M. Effects of sweeping jet actuator parameters on flow separation control. *AIAA Journal*. 2018; 56(1):100–110. <https://doi.org/10.2514/1.J055796> PMID: 31395987
10. Kara K, Kim D, Morris PJ. Flow-separation control using sweeping jet actuator. *AIAA Journal*. 2018; 56(11):4604–4613. <https://doi.org/10.2514/1.J056715>
11. Childs RE, Stremel PM, Kushner LK, Heineck JT, Storms BL. Simulation of sweep-jet flow control, single jet and full vertical tail. In: 54th AIAA Aerospace Sciences Meeting; 2016. p. 0569.
12. Aram S, Shan H. Synchronization effect of an Array of sweeping jets on a separated flow over a wall-mounted hump. In: AIAA Aviation 2019 Forum; 2019. p. 3396.
13. Quarteroni A, Veneziani A. Analysis of a geometrical multiscale model based on the coupling of ODE and PDE for blood flow simulations. *Multiscale Modeling & Simulation*. 2003; 1(2):173–195. <https://doi.org/10.1137/S1540345902408482>
14. D'Angelo C, Quarteroni A. On the coupling of 1d and 3d diffusion-reaction equations: application to tissue perfusion problems. *Mathematical Models and Methods in Applied Sciences*. 2008; 18(08):1481–1504. <https://doi.org/10.1142/S0218202508003108>
15. Passerini T, De Luca M, Formaggia L, Quarteroni A, Veneziani A. A 3D/1D geometrical multiscale model of cerebral vasculature. *Journal of Engineering Mathematics*. 2009; 64(4):319. <https://doi.org/10.1007/s10665-009-9281-3>
16. Quarteroni A, Veneziani A, Vergara C. Geometric multiscale modeling of the cardiovascular system, between theory and practice. *Computer Methods in Applied Mechanics and Engineering*. 2016; 302:193–252. <https://doi.org/10.1016/j.cma.2016.01.007>
17. Boon WM, Nordbotten JM, Yotov I. Robust discretization of flow in fractured porous media. *SIAM Journal on Numerical Analysis*. 2018; 56(4):2203–2233. <https://doi.org/10.1137/17M1139102>
18. Nordbotten JM, Boon WM, Fumagalli A, Keilegavlen E. Unified approach to discretization of flow in fractured porous media. *Computational Geosciences*. 2019; 23(2):225–237. <https://doi.org/10.1007/s10596-018-9778-9>
19. Kevrekidis IG, Samaey G. Equation-free multiscale computation: Algorithms and applications. *Annual Review of Physical Chemistry*. 2009; 60:321–344. <https://doi.org/10.1146/annurev.physchem.59.032607.093610> PMID: 19335220
20. E W, Engquist B, Huang Z. Heterogeneous multiscale method: a general methodology for multiscale modeling. *Physical Review B*. 2003; 67(9):092101. <https://doi.org/10.1103/PhysRevB.67.092101>
21. Van Brummelen E. Partitioned iterative solution methods for fluid–structure interaction. *International Journal for Numerical Methods in Fluids*. 2011; 65(1-3):3–27. <https://doi.org/10.1002/flid.2465>
22. Shankaran S, Alonso J, Liou MF, Liu NS, Davis R. A multi-code-coupling interface for combustor/turbomachinery simulations. In: 39th Aerospace Sciences Meeting and Exhibit; 2001. p. 974.
23. Xu J, Huang C, Duraisamy K. Reduced-Order Modeling Framework for Combustor Instabilities Using Truncated Domain Training. *AIAA Journal*. 2020; 58(2):618–632. <https://doi.org/10.2514/1.J057959>
24. Siddiqui MS, Rasheed A, Tabib M, Kvamsdal T. Numerical investigation of modeling frameworks and geometric approximations on NREL 5 MW wind turbine. *Renewable Energy*. 2019; 132:1058–1075. <https://doi.org/10.1016/j.renene.2018.07.062>
25. Gander MJ, Halpern L. Techniques for locally adaptive time stepping developed over the last two decades. In: *Domain Decomposition Methods in Science and Engineering XX*. Springer; 2013. p. 377–385.
26. Rasheed A, San O, Kvamsdal T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*. 2020; 8:21980–22012. <https://doi.org/10.1109/ACCESS.2020.2970143>
27. Tao F, Zhang H, Liu A, Nee AY. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*. 2018; 15(4):2405–2415. <https://doi.org/10.1109/TII.2018.2873186>
28. Hartmann D, Herz M, Wever U. Model order reduction a key technology for digital twins. In: *Reduced-order modeling (ROM) for simulation and optimization*. Springer; 2018. p. 167–179.
29. Ganguli R, Adhikari S. The digital twin of discrete dynamic systems: Initial approaches and future challenges. *Applied Mathematical Modelling*. 2020; 77:1110–1128. <https://doi.org/10.1016/j.apm.2019.09.036>
30. Chakraborty S, Adhikari S. Machine learning based digital twin for dynamical systems with multiple time-scales. *arXiv preprint arXiv:200505862*. 2020.
31. Kapteyn M, Knezevic D, Huynh D, Tran M, Willcox K. Data-driven physics-based digital twins via a library of component-based reduced-order models. *International Journal for Numerical Methods in Engineering*. 2020.

32. Kapteyn MG, Willcox KE. From Physics-based models to Predictive Digital Twins via Interpretable Machine Learning. arXiv preprint arXiv:200411356. 2020.
33. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: The Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, February 4–9; 2017.
34. Hinton GE, Salakhutdinov RR. Reducing the Dimensionality of Data with Neural Networks. *Science*. 2006; 313(5786):504–507. <https://doi.org/10.1126/science.1127647> PMID: 16873662
35. Liu Z, Yao C, Yu H, Wu T. Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things. *Future Generation Computer Systems*. 2019; 97:1–9. <https://doi.org/10.1016/j.future.2019.02.068>
36. Bu F, Wang X. A smart agriculture IoT system based on deep reinforcement learning. *Future Generation Computer Systems*. 2019; 99:500–507. <https://doi.org/10.1016/j.future.2019.04.041>
37. Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*. 2020; 52:477–508. <https://doi.org/10.1146/annurev-fluid-010719-060214>
38. Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, et al. Deep learning and process understanding for data-driven Earth system science. *Nature*. 2019; 566(7743):195–204. <https://doi.org/10.1038/s41586-019-0912-1> PMID: 30760912
39. Schmidt J, Marques MR, Botti S, Marques MA. Recent advances and applications of machine learning in solid-state materials science. *Computational Materials*. 2019; 5(1):1–36.
40. Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*. 2018; 6:14410–14430. <https://doi.org/10.1109/ACCESS.2018.2807385>
41. Yuan X, He P, Zhu Q, Li X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*. 2019; 30(9):2805–2824. <https://doi.org/10.1109/TNNLS.2018.2886017> PMID: 30640631
42. Hao-Chen HXYM, Deb LD, Anil HLJLT, Jain K. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*. 2020; 17(2):151–178. <https://doi.org/10.1007/s11633-019-1211-x>
43. Sirovich L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics*. 1987; 45(3):561–571. <https://doi.org/10.1090/qam/910463>
44. Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*. 1993; 25(1):539–575. <https://doi.org/10.1146/annurev.fl.25.010193.002543>
45. Holmes P, Lumley JL, Berkooz G, Rowley CW. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, New York; 2012.
46. Ahmed SE, San O, Rasheed A, Iliescu T. A long short-term memory embedding for hybrid uplifted reduced order models. *Physica D: Nonlinear Phenomena*. 2020; 409:132471. <https://doi.org/10.1016/j.physd.2020.132471>
47. Pawar S, Ahmed SE, San O, Rasheed A, Navon IM. Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows. *Physics of Fluids*. 2020; 32(7):076606. <https://doi.org/10.1063/5.0012853>
48. Wan ZY, Vlachas P, Koumoutsakos P, Sapsis T. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLOS One*. 2018; 13(5):e0197704. <https://doi.org/10.1371/journal.pone.0197704> PMID: 29795631
49. Majda A. *Introduction to PDEs and Waves for the Atmosphere and Ocean*. vol. 9. American Mathematical Society, Providence; 2003.
50. Liu JG, Wang C, Johnston H. A fourth order scheme for incompressible Boussinesq equations. *Journal of Scientific Computing*. 2003; 18(2):253–285. <https://doi.org/10.1023/A:1021168924020>
51. Nicolás A, Bermúdez B. 2D thermal/isothermal incompressible viscous flows. *International Journal for Numerical Methods in Fluids*. 2005; 48(4):349–366. <https://doi.org/10.1002/flid.895>
52. Soffientino B, Pilson MEQ. The Bosphorus Strait: A special place in the history of oceanography. *Oceanography*. 2005; 18(2):16–23. <https://doi.org/10.5670/oceanog.2005.38>
53. Gill AE. *Atmosphere-Ocean Dynamics*. vol. 30. Academic Press, San Diego, California, USA; 1982.
54. Arakawa A. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *Journal of Computational Physics*. 1997; 135(2):103–114. <https://doi.org/10.1006/jcph.1997.5697>
55. Ahmed SE, Rahman SM, San O, Rasheed A, Navon IM. Memory embedded non-intrusive reduced order modeling of non-ergodic flows. *Physics of Fluids*. 2019; 31(12):126602. <https://doi.org/10.1063/1.5128374>