Neural Ordinary Differential Equations for Data-Driven Reduced Order Modeling of Environmental Hydrodynamics

Sourav Dutta,[1] **Peter Rivera-Casillas,**[2]**Matthew W. Farthing,**[1]

[1] U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory, Vicksburg, MS,
[2] U.S. Army Engineer Research and Development Center, Information and Technology Laboratory, Vicksburg, MS,
[1, 2] {sourav.dutta, peter.g.rivera-casillas, matthew.w.farthing}@erdc.dren.mil

Abstract

Model reduction for fluid flow simulation continues to be of great interest across a number of scientific and engineering fields. Here, we explore the use of Neural Ordinary Differential Equations, a recently introduced family of continuous-depth, differentiable networks (Chen et al. 2018), as a way to propagate latent-space dynamics in reduced order models. We compare their behavior with two classical non-intrusive methods based on proper orthogonal decomposition and radial basis function interpolation as well as dynamic mode decomposition. The test problems we consider include incompressible flow around a cylinder as well as real-world applications of shallow water hydrodynamics in riverine and estuarine systems. Our findings indicate that Neural ODEs provide an elegant framework for stable and accurate evolution of latent-space dynamics with a promising potential of extrapolatory predictions. However, in order to facilitate their widespread adoption for large-scale systems, significant effort needs to directed at accelerating their training times. This will enable a more comprehensive exploration of the hyperparameter space for building generalizable Neural ODE approximations over a wide range of system dynamics.

Introduction

Despite the trend of hardware improvements and significant gains in the algorithmic efficiency of standard discretization procedures, *high-fidelity* numerical simulation of engineering systems governed by nonlinear partial differential equations still pose a prohibitive computational challenge (Quarteroni, Manzoni, and Negri 2016) for several decision-making applications involving control (Proctor, Brunton, and Kutz 2016), optimal design and multi-fidelity optimization (Peherstorfer, Willcox, and Gunzburger 2016), and/or uncertainty quantification (Sapsis and Majda 2013). *Reduced order models* (ROMs) offer a valuable alternative way to simulate such dynamical systems with considerably reduced computational cost (Benner, Gugercin, and Willcox 2015).

Reduced basis (RB) methods (Quarteroni, Manzoni, and Negri 2016) constitute a family of widely popular ROM techniques that are usually implemented with an offline-online decomposition paradigm. The *offline* stage involves the construction of a solution-dependent, linear basis space spanned by a set of RB "modes", which are extracted from a collection of high-fidelity solutions, also called *snapshots*. The RB "modes" can be thought of as a set of global basis functions spanning a linear subspace that can be used to approximate the dynamics of the high-fidelity model. The most well known method to extract the reduced basis is called *proper orthogonal decomposition* (POD) (Sirovich 1987; Berkooz, Holmes, and Lumley 1993), which is particularly effective when the coherent structures of the flow can be hierarchically ranked in terms of their energy content.

In the *online* stage of traditional RB methods, a linear combination of the reduced order RB modes is used to approximate the high-fidelity solution for a new configuration of flow parameters. The procedure adopted to compute the expansion coefficients leads to the classification of these methods into two broad categories: *intrusive* and *non-intrusive*. In an intrusive RB method, the expansion coefficients are determined by the solution of a reduced order system of equations, which is typically obtained via a Galerkin or Petrov-Galerkin projection of the high-fidelity (full-order) system onto the RB space (Lozovskiy, Farthing, and Kees 2017). Typically this projection and solution involves modification of high-fidelity simulators and hence the label intrusive. For linear systems, Galerkin projection is the most popular choice. However, in the presence of nonlinearities, an affine expansion of the nonlinear (or non-affine) differential operator must be recovered in order to make the evaluation of the projection-based reduced model independent of the number of DOFs of the high-fidelity solution.

Several different techniques, collectively referred to as hyper-reduction methods (Amsallem et al. 2015), have been proposed to address this problem. These include the empirical interpolation method (EIM), its discrete counterpart DEIM (Chaturantabut and Sorensen 2010), "gappy POD" (Willcox 2006), as well as the residual DEIM method (Xiao et al. 2014). Beyond the need for hyper-reduction to recover efficiency, in complex nonlinear problems it is also common that some of the intrinsic structures present in the high-fidelity model may be lost during order reduction using Galerkin projection-based approaches. This is because the Galerkin projection approach inherently assumes that the residual generated by the truncated representation of the high-fidelity model is orthogonal to the reduced basis space which leads to the loss of higher order nonlinear interaction terms in the reduced representation. This can result in qualitatively wrong solutions or instability issues (Amsallem and Farhat 2012). As a remedy, Petrov-Galerkin projection based approaches have been proposed (Carlberg, Bou-Mosleh, and Farhat 2011; Carlberg et al. 2013; Fang et al. 2013).

An alternative family of methods to address the issues of instability and loss of efficiency in the intrusive ROM frameworks is represented by *non-intrusive* reduced order models (NIROMs), and forms the subject of this study. The primary advantage of this class of methods is that complex modifications to the source code describing the physical model can be avoided, thus making it easier to develop reduced models when the legacy or proprietary source codes are not available. In these methods, instead of a Galerkin-type projection, the expansion coefficients for the reduced solution are obtained via interpolation on the space of a reduced basis extracted from snapshot data. However, since the reduced dynamics generally belong to nonlinear, matrix manifolds, a variety of interpolation techniques have been proposed that are capable of enforcing the constraints characterizing those manifolds. Regression-based non-intrusive methods have been proposed that, among others, use artificial neural networks (ANNs), in particular multi-layer perceptrons (Hesthaven and Ubbiali 2018), Gaussian process regression (GPR) (Guo and Hesthaven 2019), and radial basis function (RBF) (Audouze, De Vuyst, and Nair 2013) to perform the interpolation.

Here, we will explore an alternative approach to propagating latent-space dynamics based on Neural ODEs, which are a family of continuous-depth, differentiable networks that can be seen as an extension of ResNets in the limit of a zero discretization step size (Dupont, Doucet, and Teh 2019). Details of our approach follow below. In addition, we consider two NIROM techniques - a) based on linear dimension reduction via POD and latent space evolution via Radial Basis Functions (RBF), and b) Dynamic Mode Decomposition (DMD) that will serve as the benchmarks in our numerical experiments to provide comparisons with the Neural ODE approach. We then proceed with several numerical experiments based on incompressible flow around a cylinder and shallow water hydrodynamics in order to evaluate the methods' performance for fast replay applications in complex fluid-dynamics problems.

Methodology

The standard ROM development framework usually consists of three stages:

1. identification of a low-dimensional latent (or reduced-order) space,
2. determining a latent-space representation of the nonlinear dynamical system in terms of the reduced basis and modeling the evolution of the system of modal coefficients, and
3. reconstruction in the high-fidelity space for validation and analysis.

Machine learning techniques can be introduced at any of these stages. For example, many works have explored the use of deep learning-based approaches like autoencoders as a way to introduce a nonlinear alternative for dimension reduction (Lusch, Nathan Kutz, and Brunton 2018; Ghorbanidehno et al. 2021). Combining these methods with data-driven latent-space propagation (for example via fully connected or recurrent neural networks) leads to a fully non-intrusive approach (Gonzalez and Balajewicz 2018). On the other hand, one can also combine nonlinear dimension reduction with intrusive projection to create a hybrid method (Lee and Carlberg 2020; Kim et al. 2020). In this work, we study three different data-driven strategies for accurate learning of system dynamics within the context of linear dimension reduction. In the first two methods we adopt the POD technique for identification of an optimal global basis. For the latent space evolution, we utilize a kernel-based multivariate interpolation method called radial basis function (RBF) interpolation, and a machine learning strategy designed for sequential learning of time-series data called neural ordinary differential equations (NODE). In the third strategy, the three stages of ROM development are combined together by using a classical modal decomposition technique called the dynamic mode decomposition (DMD), that is supported by rigorous mathematical analysis of Koopman mode theory.

Proper orthogonal decomposition

POD is a popular technique for dimension reduction (Antoulas and Sorensen 2001) of the solution manifold of a dynamical system by determining a linear reduced space spanned by an orthogonal basis with an associated energetic hierarchy. (Taira et al. 2020) provides an excellent overview of POD as well as a comparison with other dimension-reduction techniques.

Consider a *snapshot matrix* $\mathbf{S} = [\widehat{\mathbf{v}}^1, \ldots, \widehat{\mathbf{v}}^M] \in \mathbb{R}^{N \times M}$ containing a collection of M high-fidelity snapshots of the solution manifold from time $t = 0$ to $t = T$ such that $\widehat{\mathbf{v}}^k \in \mathbb{R}^N$ is the k^{th} snapshot with the temporal mean value removed, *i.e.*, $\widehat{\mathbf{v}}^k = \mathbf{v}^k - \bar{\mathbf{v}}$ where $\bar{\mathbf{v}} = \sum_{i=1}^{M} \frac{\mathbf{v}^i}{M}$ is the time-averaged solution. The goal of the POD procedure is to identify a linear subspace $\chi = \text{span}\left\{\psi^1, \ldots, \psi^r\right\}, \ (r \ll M)$ which approximates the solution manifold optimally with respect to the L^2-norm.

The POD bases can be efficiently extracted by performing a "thin" singular value decomposition (SVD) of the snapshot matrix $\mathbf{S} = \widetilde{\mathbf{\Theta}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{\Psi}}^T$, where $\widetilde{\mathbf{\Sigma}} = \text{diag}(\sigma_1, \ldots, \sigma_{\mathcal{R}})$ is a $\mathcal{R} \times \mathcal{R}$ diagonal matrix containing the singular values arranged in decreasing order of magnitude, $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_{\mathcal{R}}$ and $\mathcal{R} < \min\{N, M\}$ is the rank of $\mathbf{S}$. $\widetilde{\mathbf{\Theta}}$ and $\widetilde{\mathbf{\Psi}}$ are $N \times \mathcal{R}$ and $M \times \mathcal{R}$ matrices respectively, whose columns are the orthonormal left and right singular vectors of $\mathbf{S}$ such that $\widetilde{\mathbf{\Theta}}^T\widetilde{\mathbf{\Theta}} = \mathbf{I}_{\mathcal{R}} = \widetilde{\mathbf{\Psi}}^T\widetilde{\mathbf{\Psi}}$. The columns $\boldsymbol{\theta}_n$ of the matrix $\widetilde{\mathbf{\Theta}}$ are ordered corresponding to the singular values σ_n and these provide the desired POD basis. Let $\mathbf{\Theta}$ denote the matrix of the first m columns of $\widetilde{\mathbf{\Theta}}$, $\mathbf{\Psi}$ be the matrix containing the first m rows of $\widetilde{\mathbf{\Psi}}$, and $\mathbf{\Sigma}$ be a diagonal matrix containing the first m singular values from $\widetilde{\mathbf{\Sigma}}$, then the high-fidelity solution $\mathbf{v}^n$ at time t^n can be approximated as,

$$\mathbf{v}^n \approx \bar{\mathbf{v}} + \mathbf{\Theta}\mathbf{z}^n = \bar{\mathbf{v}} + \sum_{i=1}^{m} z_i^n \boldsymbol{\theta}_i, \tag{1}$$

where $\mathbf{z}^n \in \mathbb{R}^m$ is a vector of modal coefficients with respect to the reduced basis. The modal coefficient matrix $\mathbf{Z} = \mathbf{\Theta}^T\mathbf{S}$ constitutes our training data for the latent space learning methods. Due to the Eckart-Young-Mirsky theorem, the POD basis provides an optimal rank-m approximation $\widehat{\mathbf{S}} = \mathbf{\Theta}\mathbf{\Sigma}\mathbf{\Psi}^T$ of the snapshot matrix $\mathbf{S}$ with a desired level of accuracy, τ_{POD}.

The POD method has been successfully applied in statistics (Jolliffe 1986), signal analysis and pattern recognition (Deheuvels and Martynov 2008), ocean models (Vermeulen and Heemink 2006), air pollution models (Fang et al. 2014), convective Boussinesq flows (San and Borggaard 2015), and Shallow Water Equation (SWE) models (Stefanescu, Sandu, and Navon 2014; Lozovskiy et al. 2016).

Latent space evolution

In this section, we outline two non-intrusive methods for modeling the evolution of time-series data in the latent space defined by the POD basis. RBF interpolation is a classical, data-driven, kernel-based method for computing an approximate continuous response surface that aligns with the given multivariate data. The second technique called NODE is a neural-network based method to predict the continuous evolution of a vector $\mathbf{c}$ over time, that is designed to preserve memory effects within the architecture.

Radial basis function interpolation For simplicity, let the time evolution of the modal coefficients $\mathbf{z}$ be represented as a semi-discrete dynamical system,

$$\dot{\mathbf{z}} = \mathbf{f}(\mathbf{z}, t), \text{ with } \mathbf{z}^0 = \mathbf{\Theta}^T\left(\mathbf{v}^0 - \bar{\mathbf{v}}\right) \tag{2}$$

where all the information about the temporal dynamics including the effects of any numerical stabilization of the high-fidelity solver and all the nonlinear terms are embedded in $\mathbf{f}(\mathbf{z}, t)$. In the POD-RBF NIROM framework (Dutta et al. 2020), instead of the Galerkin projection, the components of the time derivative function $f_j (j = 1, \dots, m)$ are approximated using RBF interpolation.

Let F_j denote a RBF approximation of the time derivative function f_j, which is defined by a linear combination of N_i instances of a radial basis function ϕ,

$$F_j(\mathbf{z}) = \sum_{k=1}^{N_i} \alpha_{j,k}\, \phi\left(\|\mathbf{z} - \widehat{\mathbf{z}}_k\|\right), \quad j = 1, \dots, m, \tag{3}$$

where $\{\widehat{\mathbf{z}}_k \,|\, k = 1, \dots, N_i\}$ denotes the set of interpolation centers and $\alpha_{j,k}$ $(k = 1, \dots, N_i)$ is the unknown interpolation coefficient corresponding to the k^{th} center for the j^{th} component of the modal coefficient. These interpolation coefficients are computed by enforcing the interpolation function F_j to exactly match the time derivative of the modal coefficients at N_e test points $(N_e \geq N_i)$. Choosing the centers and the test points identically from the set of snapshot modal coefficients as $\{\mathbf{z}^l \,|\, l = 0, \dots, M-1\}$ such that $N_i = N_e = M$, and making some simplifying assumptions leads to a symmetric, linear system of M equations to solve for the unknown interpolation coefficients, $\alpha_{j,k}$

$$\mathbf{A}\boldsymbol{\alpha}^j = \boldsymbol{g}^j, \text{ for } j = 1, \dots, m, \tag{4}$$

where

$$[A_{n,k}] = [\phi\left(\|\mathbf{z}^n - \mathbf{z}^k\|\right)], \quad n, k = 0, \dots M-1,$$
$$\boldsymbol{\alpha}^j = [\alpha_{j,0}, \dots, \alpha_{j,M-1}]^T, \; \boldsymbol{g}^j = [g_{j,0}, \dots, g_{j,M-1}]^T.$$

The coefficients $\boldsymbol{\alpha}^j$ define a unique RBF interpolant which can then be used to approximate eq. (2) and generate a non-intrusive model for the evolution of the modal coefficients In this work, a first-order forward Euler scheme has been employed for the discretization of the time derivative, and a strictly positive-definite Matérn C^0 kernel, given by $\phi(r) = e^{-cr}$ has been adopted, where r is the Euclidean distance and c is the RBF shape factor (Fasshauer 2007).

Adopting RBF interpolation for modeling the latent space evolution of the modal coefficients has been shown to be quite successful for nonlinear, time-dependent partial differential equations (PDEs) (Xiao et al. 2015; Dutta et al. 2020), nonlinear, parametrized PDEs (Audouze, De Vuyst, and Nair 2013; Xiao et al. 2017), and aerodynamic shape optimization (Iuliano and Quagliarella 2013), to name a few.

Neural ordinary differential equations Recurrent neural network (RNN) architectures like LSTM and GRU are often employed to encode time-series data and forecast future states, as their internal memory preserving architecture allows them to incorporate state information over a sequence of input data. Although RNNs have seen great success in natural language processing tasks, they have had relatively limited success in high-fidelity scientific computing applications (Ferrandis et al. 2019; Wang, Ripamonti, and Hesthaven 2020), as it has been observed that a sequence generated by an RNN may fail to preserve temporal regularity of the underlying signal, and thus may not represent true continuous dynamics. (Chen et al. 2018). With deep neural networks (DNN) such as ResNet, the evolution of the features over the network depth is equivalent to solving an ordinary differential equation (ODE) such as $\frac{dz}{dt} = F(z, \theta)$ using the forward Euler method, and this connection between ResNet's architecture and numerical integrators has been explored in details by (Ruthotto and Haber 2019) and others. Several other deep learning methods have been proposed for learning ODEs and PDEs. These include using PDE-based network (Long, Lu, and Dong 2019), training DNNs using physics-informed soft penalty constraints (Raissi, Perdikaris, and Karniadakis 2019), and using sparse regularizers and regression (Brunton et al. 2016; Champion et al. 2019), to name a few.

Chen et al. (2018) proposed a 'continuous-depth' neural network called ODE-Net that effectively replaces the layers in ResNet-like architectures with a trainable ODE solver. The memory efficiency and stability of this neural ordinary differential equation (NODE) approach was further improved in (Gholami, Keutzer, and Biros 2019; Dupont, Doucet, and Teh 2019) and others. (Maulik et al. 2020) applied the NODE framework to obtain latent space closure models for ROMs of a one-dimensional advecting shock problem and a one-dimensional Burgers' turbulence problem that exhibits multiscale behavior in the wavenumber space. Some other notable recent applications of NODE include the identification of ODE or PDE models from time-dependent data (Sun, Zhang, and Schaeffer 2020), modeling of irregularly spaced time series data (Rubanova, Chen, and Duvenaud 2019), modeling of spatio-temporal information in video signals (Kanaa et al. 2019). Finlay et al. (2020) used a combination of optimal transport theory and stability regularizations to propose a neural-ODE generative model that can be efficiently trained on large-scale datasets. Here we further explore the application of the POD-NODE methodology to complex, real-world flows characterized by systems of two-dimensional, nonlinear PDEs.

We assume that the time evolution of the modal coefficients of the high-fidelity dynamical system in the latent space can be modeled using a (first-order) ODE,

$$\frac{d\mathbf{z}}{dt} = \mathcal{F}(t, \mathbf{z}(t)), \text{ with } \mathbf{z}(0) = \mathbf{z}^0,\ \mathbf{z} \in \mathbb{R}^d, d \geq 1. \quad (5)$$

The goal is to obtain a NN approximation $\widehat{\mathcal{F}}$ of the dynamics function $\mathcal{F}$ such that $\frac{d\mathbf{z}}{dt} \approx \text{net}(t, \mathbf{z}) = \widehat{\mathcal{F}}(t, \mathbf{z}, \boldsymbol{\omega})$. The full procedure can be outlined as follows:

1. Compute the time series of modal coefficients $[\mathbf{z}^0, \ldots, \mathbf{z}^{M-1}]$ for $t \in \{0, \ldots, M-1\}$ where $\mathbf{z}^k \in \mathbb{R}^m$.
2. Initialize a NN approximation for the dynamics function $\widehat{\mathcal{F}}(t, \mathbf{z}, \boldsymbol{\omega})$ where $\boldsymbol{\omega}$ represents the initial NN parameters.
3. The NN parameters are optimized iteratively through the following steps.
   (a) Compute the approximate forward time trajectory of the modal coefficients by solving eq. (5) using a standard ODE integrator as,
   $$\hat{\mathbf{z}}^{M-1} = ODESolve(\widehat{\mathcal{F}}, \boldsymbol{\omega}, \mathbf{z}^0, t^0, t^{M-1}) \quad (6)$$
   (b) The free parameters of the NODE model are $\{\boldsymbol{\omega}, t^0, t^{M-1}\}$. Evaluate the differentiable loss function $\mathcal{L}\left(ODESolve(\widehat{\mathcal{F}}, \boldsymbol{\omega}, \mathbf{z}^0, t^0, t^{M-1}) - \mathbf{z}^{M-1}\right)$.
   (c) To optimise the loss, compute gradients with respect to the free parameters. Similar to the usual backpropagation algorithm, this can be achieved by first computing the gradient $\partial\mathcal{L}/\partial\widehat{\mathbf{z}}(t)$, and then a performing a reverse traversal through the intermediate states of the ODE integrator. For a memory-efficient implementation, the adjoint method (Chen et al. 2018) can be used to backpropagate the errors by solving an adjoint system for the augmented state vector $\mathbf{b} = [\frac{\partial\mathcal{L}}{\partial\widehat{\mathbf{z}}}, \frac{\partial\mathcal{L}}{\partial\boldsymbol{\omega}}, \frac{\partial\mathcal{L}}{\partial t}]^T$ backwards in time from t^{M-1} to t^0.
   (d) The gradient $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\omega}}(t=0)$ computed in the previous step is used to update the parameters $\boldsymbol{\omega}$ by using an optimization algorithm like RMSProp or Adam.
4. The trained NODE approximation of the dynamics function can be used to compute predictions for the time trajectory of the modal coefficients.

In this work, we utilize the TFDiffEq (https://github.com/titu1994/tfdiffeq) library that runs on the Tensorflow Eager Execution platform to train the NODE models. Although a single layer architecture guarantees upper-bounds according to the universal approximation theorem (Barron 1993), deeper networks with up to four layers as well as several linear and nonlinear activation functions are also explored due to their possibly improved expressibility for more complex nonlinear dynamics (Zhang et al. 2019). RMSProp is adopted for loss minimization with an initial learning rate of 0.001, a staircase decay function with a range of variable decay schedules, and a momentum coefficient of 0.9. NODE predictions of comparable accuracy were obtained for all the numerical experiments by using both the adjoint method as well as by backpropagating gradients directly through the hidden steps of the ode solver. However, for large-scale training data the latter method may lead to memory issues, especially while computing on GPU nodes.

Dynamic mode decomposition

As a final point of comparison, we consider Dynamic mode decomposition (DMD). DMD is a data-driven ROM technique that represents the temporal dynamics of a complex, nonlinear system (Schmid 2010; Kutz et al. 2016) as the combination of a few linearly evolving, spatially coherent modes that oscillate at a fixed frequency, and which are closely related to the eigenvectors of the infinite-dimensional Koopman operator (Koopman 1931; Mezić 2013). Consider the following snapshot matrices containing a few temporally-equispaced snapshots of a high-dimensional dynamical system:

$$\mathbf{X} = \left[\mathbf{v}^0\ \mathbf{v}^1\ \ldots \mathbf{v}^{M-1}\right], \quad \mathbf{X}' = \left[\mathbf{v}^1\ \mathbf{v}^2\ \ldots \mathbf{v}^M\right]$$

where $\mathbf{v}^k \in \mathbb{R}^N$ is the k^{th} solution snapshot, N is the spatial degrees of freedom of the discretized system, and M is the total number of temporal snapshots. DMD involves the identification of the best-fit linear operator $\mathbf{A}_X$ that relates the above matrices as $\mathbf{X}' = \mathbf{A}_X\mathbf{X}$, and computing its eigenvalues and eigenvectors. Computing a least-square approximation of $\mathbf{A}_X$ using the Moore-Penrose pseudoinverse(†) may pose computational challenges due to the size of the discrete dynamical system. For computational efficiency, the exact DMD algorithm (adopted here) avoids computing the Moore-Penrose pseudoinverse(†) by projecting the operator on to a reduced space obtained by POD, as outlined in (Alla and Kutz 2017).

In recent years, Koopman mode theory has provided a rigorous theoretical background for an efficient modal decomposition in problems describing oscillations and other nonlinear dynamics using DMD (Rowley et al. 2009). Several variants of the DMD algorithm have been proposed (Proctor, Brunton, and Kutz 2016; Kutz, Fu, and Brunton 2016; Alekseev et al. 2016; Le Clainche and Vega 2017) and have been successfully applied as efficient ROM techniques for determining the optimal global basis modes for nonlinear, time-dependent problems (Bistrian and Navon 2015, 2017). For non-parametrized PDEs, DMD presents an efficient framework that combines all the three stages of ROM development to learn a linear operator in an optimal least square sense. However, this approach cannot be directly applied to parametrized problems (Alsayyari et al. 2021).

Numerical experiments

In this section, we first assess the performance of different NODE architectures for a benchmark flow problem characterized by the incompressible Navier Stokes equations (NSE), and then further evaluate the relative performance of all three NIROM models for two real-world applications governed by the shallow water equations (SWE). The POD-RBF and DMD NIROM training runs were performed on a Macbook Pro 2018 with a 2.9 GHz 6-Core Intel Core i9 processor and 32 GB 2400 MHz DDR4 RAM. The NODE models were trained in serial on Vulcanite, a high performance computer at the U.S. Army Engineer Research and Development Center DoD Supercomputing Resource Center (ERDC-DSRC). Vulcanite is equipped with NVIDIA Tesla V100 PCIe GPU accelerator nodes and has 32GBytes memory/node.

Flow around a cylinder

This problem simulates a time-periodic fluid flow through a 2D pipe with a circular obstacle. The flow domain is a rectangular pipe with a circular hole of radius $r = 0.05$, denoted by $\Omega = [0, 2.2] \times [-0.2, 0.21] \setminus B_r(0.2, 0)$. The flow is governed by

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nu \Delta \mathbf{u} + \nabla p = 0, \tag{7}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{8}$$

where $\mathbf{u}$ denotes the velocity, p the pressure, $\otimes$ is the outer product (dyadic product) given by $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T$, and $\nu = 0.001$ is the kinematic viscosity. No slip boundary conditions are specified along the lower and upper walls, and on the boundary of the circular obstacle. A parabolic inflow velocity profile is prescribed on the left wall,

$$\mathbf{u}(0, y) = \left(4U \frac{(0.21 - y)(y - 0.2)}{0.41^2}, 0 \right), \tag{9}$$

and zero gradient outflow boundary conditions on the right wall. High-fidelity simulation data is obtained with OpenFOAM using an unstructured mesh with 14605 nodes at $Re = 100$, such that the flow exhibits the periodic shedding of von Karman vertices. 313 training snapshots are collected for $t = [2.5, 5.0]$ seconds with $\Delta t = 0.008$ seconds, and the NIROM predictions are obtained for $t = [2.5, 6.0]$ seconds with $\Delta t = 0.002$ seconds.

A large collection of NODE architectures and hyperparameter configurations were trained for 50000 epochs and details of the best 8 models are presented in Table 1. A fourth-order Runge-Kutta solver was found to be the optimal choice in terms of both accuracy and efficiency among all the available solvers ranging from the fixed-step forward Euler and the midpoint solvers to the adaptive-step Dormand-Prince (dopri5) solver. The "tanh" and "elu" activation functions were found to be the most effective among all the available linear and nonlinear activation functions. Due to the nature of the activation functions, the networks with "tanh" activations were found to train better when every element of the input state vector was individually scaled to be bounded in $[-1, 1]$, while networks with "elu" activations trained better without scaling of input vectors. Augmentation of input states as outlined in (Dupont, Doucet, and Teh 2019) was found to have no significant impact on the training. The RMSProp optimizer paired with either a step decay function or an exponential decay function were found to be equally effective. However, further numerical experiments are necessary to study the efficiency of alternative first-order and second-order optimization methods. The number of decay steps were varied in discrete increments between 5000 to 25000, and decay rates ranging from 0.1 to 0.9 were studied. It was observed that a lower initial learning rate (≈ 0.001) combined with either larger decay steps and smaller decay rates or vice versa led to a desirable training trajectory. Fig. 1 shows the evolution of the 1^{st}, 3^{rd}, and 5^{th} latent-space modal coefficients for the pressure and the x-velocity solutions, obtained using the best 8 NODE models. All the models generate accurate predictions at a finer temporal resolution than the training data, and have excellent agreement with the high-fidelity solution even while extrapolating outside the training data ($5 \leq t \leq 6$ seconds).

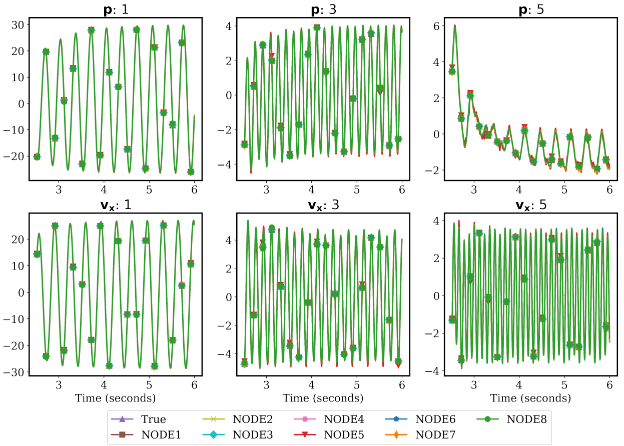


Figure 1: Comparison of NODE models in the latent space for the cylinder example

Fig. 2 compares the time trajectory of the spatial root mean square errors (RMSE) in the high-fidelity space for two of the best NODE models with two DMD NIROM solutions obtained using truncation levels of $r = 20$ and $r = 8$. It is encouraging to note that even though the NODE solutions are computed using a latent-space representation that is roughly comparable to the DMD solution with a smaller truncation level ($r = 8$), they are superior in accuracy to the coarsely truncated DMD solutions. Furthermore, unlike the POD-RBF solution that is trained with a first-order Euler time discretization, the NODE solutions did not exhibit any significant loss in accuracy with time, even while predicting outside the training region. It is, however, important to note that the training time for any new NODE architecture was extremely high (see Table 1) when compared to generating a POD-RBF or a DMD NIROM model, which

Id	Layers	Units	Act.	LR decay steps, rate	Scaling	Augmented	MSE	Training
Range	1-4	32-512	linear, relu, elu, tanh, ...	5000-25000, 0.1-0.9				
NODE1	1	256	elu	10000, 0.3	No	No	8.87e-4	24.45 hrs
NODE2	1	256	tanh	5000, 0.7	Yes	No	9.01e-4	24.56 hrs
NODE3	1	512	elu	5000, 0.5	No	No	8.86e-4	24.39 hrs
NODE4	1	256	tanh	10000, 0.25	Yes	Yes	9.02e-4	22.97 hrs
NODE5	4	64	tanh	5000, 0.5	Yes	No	9.19e-4	27.98 hrs
NODE6	1	256	elu	10000, 0.1	No	No	8.87e-4	24.13 hrs
NODE7	2	128	elu	5000, 0.5	No	No	8.86e-4	25.80 hrs
NODE8	1	512	tanh	5000, 0.5	Yes	Yes	9.00e-4	24.77 hrs

Table 1: Best NODE architectures for the cylinder example. All models were trained for 50000 epochs using the fourth-order Runge-Kutta solver and the RMSProp optimizer with an initial learning rate of 1e-3 and a momentum of 0.9.

usually required less than a minute in most cases. Such long training times may pose a significant challenge for exhaustive explorations of the design space for optimal architectures and hyperparameters, and may hinder the adoption of existing packages for automated architecture search. Thus, a concerted effort needs to be directed towards acceleration of NODE training times and towards constraining the design space by a priori identification of promising architectures.

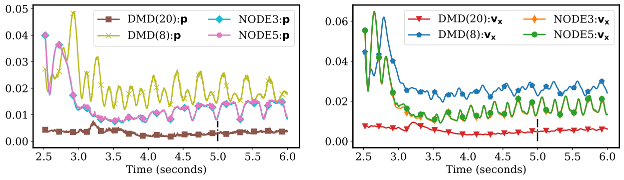


Figure 2: Comparison of RMSE of best NODE models with DMD for the cylinder example

Shallow water equations

The next two numerical examples involve flows governed by the depth-averaged SWE which is written in a conservative residual formulation as

$$\mathfrak{R} \equiv \frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{p}_x}{\partial x} + \frac{\partial \mathbf{p}_y}{\partial y} + \mathbf{r} = 0, \tag{10}$$

where the state variable $\mathbf{q} = [h, u_x h, u_y h]^T$ consists of the flow depth, h, and the discharges in the x and y directions, given by $u_x h$ and $u_y h$, respectively. Further details about the flux vectors $\mathbf{p}_x$, $\mathbf{p}_y$ and the high-fidelity model equations are available in (Dutta et al. 2020). The high-fidelity numerical solutions of the SWE are obtained using the 2D depth-averaged module of the Adaptive Hydraulics (AdH) finite element suite, which is a U.S. Army Corps of Engineers (USACE) high-fidelity, finite element resource for 2D and 3D dynamics (Trahan et al. 2018).

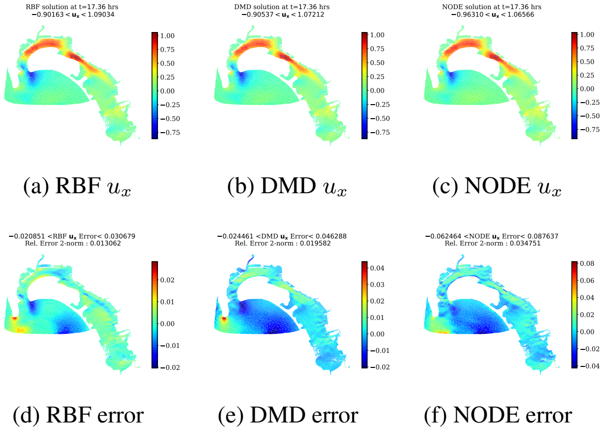
Tidal flow in San Diego bay This numerical example involves the simulation of tide-driven flow in the San Diego Bay in California, USA. The AdH high-fidelity model consists of $N = 6311$ nodes, uses tidal data obtained from NOAA/NOS Co-Ops website at a tailwater elevation inflow boundary and has no flow boundary conditions everywhere else. Further details are available in (Dutta et al. 2020).

The training space is generated using 1801 high-fidelity snapshots obtained between $t = 41$ minutes to $t = 50$ hours at a time interval of $\Delta t = 100$ seconds. The predicted ROM solutions are computed for the same time interval with $\Delta t = 50$ seconds. A latent space of dimension 265 is generated by using a POD truncation tolerance of $\tau_{POD} = 5 \times 10^{-7}$ for each solution component. The RBF NIROM approximation is computed using a shape factor, $c = 0.01$. The simulation time points provided as input to the NODE model are normalized to lie in $t \in [0, 1]$. The 'dopri5' ODE solver is adopted for computing the hidden states both forward and backward in time. Learning from the conclusions of the cylinder example, a network consisting of a single hidden layer with 256 neurons is deployed and the RMSProp optimizer with an initial learning rate of 0.001, a staircase decay rate of 0.5 every 5000 epochs, and a momentum of 0.9 is utilized for training the model over 20000 epochs. For the DMD NIROM, the simulation time points are normalized to an unit time step, and a truncation level of $r = 115$ is used to compute the DMD eigen-spectrum.

Figure 3 shows the NIROM solutions (top row) for u_x at $t = 17.36$ hours and the corresponding error plots.

Figure 4 shows the spatial RMSE over time for the depth (left) and the x-velocity (right) NIROM solutions. The NODE NIROM solution has comparable accuracy to the DMD NIROM solution and unlike the RBF NIROM solution, does not exhibit any appreciable accumulation of error over time.

Riverine flow in Red River The final numerical example involves an application of the 2D SWE to simulate riverine flow in a section of the Red River in Louisiana, USA. The AdH high-fidelity model uses $N = 12291$ nodes, has a natural inflow velocity condition upstream, a tailwater elevation boundary downstream, and no flow boundary along the river bank. For further details see (Dutta et al. 2020).



(a) RBF u_x (b) DMD u_x (c) NODE u_x

(d) RBF error (e) DMD error (f) NODE error

Figure 3: NIROM solutions of u_x and errors at $t = 17.36$ hours for the San Diego example

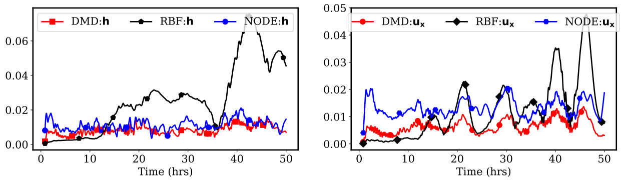
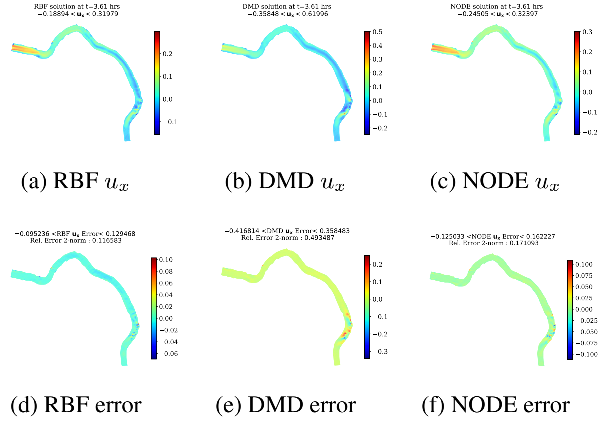


Figure 4: NIROM RMSEs for the San Diego example

The training space is generated by using 1081 high-fidelity snapshots obtained between $t = 16.67$ minutes to $t = 9.3$ hours at a time interval of $\Delta t = 30$ seconds. The predicted ROM solutions are computed for the same time interval with $\Delta t = 10$ seconds. A latent space spanned by 54 modes is generated by using a POD truncation tolerance of $\tau_{POD} = 0.01$ for each solution component. The RBF NIROM approximation is computed using a shape factor, $c = 0.05$. For consistency, the NODE network architecture is kept identical to the San Diego example and the training is also performed for 20000 epochs. Also, similar to the previous example, the simulation time points for DMD input are normalized to an unit time step. However, a smaller truncation level of $r = 30$ is used to compute the DMD eigen-spectrum.

Figure 5 shows the NIROM solutions (top row) for u_x at $t = 3.61$ hours and the corresponding error plots.

Figure 6 shows the spatial RMSE over time of the depth (left) and the x-velocity (right) NIROM solutions for the Red River example. It can be seen that the DMD NIROM solution has a relatively higher RMSE owing to the lower truncation level chosen for this example, while the RBF NIROM is far more accurate. The NODE NIROM solution seems to match the performance of the RBF NIROM solution. This indicates that the NODE NIROM framework is successful with two distinct real-world flow regimes and holds promise for more widespread applicability to model the evolution of latent space dynamics.



(a) RBF u_x (b) DMD u_x (c) NODE u_x

(d) RBF error (e) DMD error (f) NODE error

Figure 5: NIROM solutions of u_x and errors at $t = 3.61$ hours for the Red River example

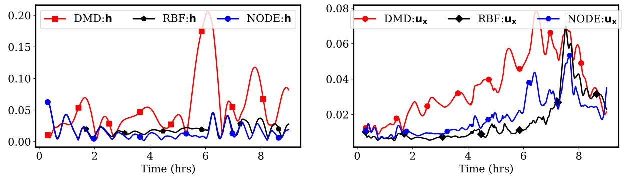


Figure 6: NIROM RMSEs for the Red River example

Conclusion

We have studied Neural ODEs as a non-intrusive machine-learning algorithm to model the evolution of modal coefficients of a system of nonlinear, time-dependent PDEs in the linearly embedded latent space characterized by a truncated POD basis. Numerical experiments were carried out with a benchmark periodic flow problem governed by the incompressible Navier Stokes equations and two real-world applications of estuarine and riverine flow dynamics governed by the two-dimensional shallow water equations.The NODE formulation demonstrated a stable and accurate learning trajectory in modeling reduced basis dynamics, even in comparison to two classical ROM techniques utilizing dynamic mode decomposition and radial basis function interpolation. The DMD NIROM exhibited superior accuracy in most of the examples and was found to be most promising for long-term predictions. However, the POD-RBF NIROM technique is easily applicable to parametrized model reduction scenarios involving parametric training manifolds of very high dimension, whereas the DMD algorithm does not have a natural extension to such a setting. The POD-NODE formulation also produced extremely promising extrapolatory predictions for the flow around a cylinder example. This presents an exciting prospect for future exploration as even for an isolated system, unperturbed by unseen external forcings, truly extrapolative predictions of reduced order dynamics in flow regimes that do not correspond to the training data is a rare feature for most well-established ROM frameworks.

This study leads to several promising avenues of research.

To begin with, an exhaustive search for an optimal NODE network architecture and optimal model hyperparameters needs to be conducted for a wide range of flow dynamics in order to gain insight of the learning trajectory and to design more generalizable NODE NIROM formulations with faster training times. With the goal of long-term predictive formulations in mind, embedding uncertainty estimates in the NODE NIROM framework might facilitate the development of adaptive models capable of re-assessing learning trajectories through in-situ measurements. The construction of a set of response functions for modeling the prediction error using machine learning (Freno and Carlberg 2019) or Gaussian Process Regression (GPR) (Xiao 2019) are some recent works in this direction. Another exciting field of study would be to combine the NODE framework with machine-learning strategies for the generation of nonlinear manifolds (Lee and Carlberg 2020; Kim et al. 2020) that are suitable for an efficient reduced representation of the system dynamics for advection-dominated problems and in the presence of sharp gradients where a truncated linear subspace offers a poor solution representation. All the relevant data and codes for this study will be made available in a public repository at https://github.com/erdc/node_nirom upon publication.

Acknowledgements

This research was supported in part by an appointment of the first author to the Postgraduate Research Participation Program at the U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory (ERDC-CHL) administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and ERDC. The authors would also like to thank Dr. Gaurav Savant for his valuable help in using the Adaptive Hydraulics suite (AdH) (Trahan et al. 2018) for the high-fidelity numerical simulation of the 2D shallow water flow examples. Permission was granted by the Chief of Engineers to publish this information.

References

Alekseev, A.; Bistrian, D.; Bondarev, A.; and Navon, I. 2016. On linear and nonlinear aspects of dynamic mode decomposition. *Int. J. Numer. Methods Fluids* 82(6): 348–371. doi:10.1002/fld.4221.

Alla, A.; and Kutz, J. N. 2017. Nonlinear model order reduction via dynamic mode decomposition. *SIAM J. Sci. Comput.* 39(5): B778—-B796. doi:10.1137/16M1059308.

Alsayyari, F.; Perkó, Z.; Tiberga, M.; Kloosterman, J. L.; and Lathouwers, D. 2021. A fully adaptive nonintrusive reduced-order modelling approach for parametrized time-dependent problems. *Comput. Methods Appl. Mech. Eng.* 373: 113483. doi:10.1016/j.cma.2020.113483. URL https://doi.org/10.1016/j.cma.2020.113483.

Amsallem, D.; and Farhat, C. 2012. Stabilization of projection-based reduced order models. *Int. J. Numer. Methods Eng.* 91: 358–377. doi:110.1002/nme.4274.

Amsallem, D.; Zahr, M.; Choi, Y.; and Farhat, C. 2015. Design optimization using hyper-reduced-order models. *Struct. Multi. Opt.* 51(4): 919–940. doi:10.1007/s00158-014-1183-y.

Antoulas, A.; and Sorensen, D. 2001. Approximation of Large-Scale Dynamical Systems: An overview. *Int. J. Appl. Math. Comput. Sci.* 11(5): 1093–1121.

Audouze, C.; De Vuyst, F.; and Nair, P. 2013. Nonintrusive Reduced-Order Modeling of Parametrized Time-Dependent Partial Differential Equations. *Numer. Methods Partial Differ. Equation* 29(5): 1587–1628. doi:10.1002/num.21768.

Barron, A. R. 1993. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* 39(3): 930–945. doi:10.1109/18.256500.

Benner, P.; Gugercin, S.; and Willcox, K. 2015. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* 57(4): 483–531.

Berkooz, G.; Holmes, P.; and Lumley, J. 1993. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annu. Rev. Fluid Mech.* 25(1): 539–575. doi:10.1146/annurev.fl.25.010193.002543.

Bistrian, D.; and Navon, I. 2015. An improved algorithm for the shallow water equations model reduction: Dynamic Mode Decomposition vs POD. *Int. J. Numer. Methods Fluids* .

Bistrian, D.; and Navon, I. 2017. Randomized dynamic mode decomposition for nonintrusive reduced order modelling. *Int. J. Numer. Methods Eng.* 112: 3–25. doi:10.1002/nme.5499.

Brunton, S. L.; Proctor, J. L.; Kutz, J. N.; and Bialek, W. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.* 113(15): 3932–3937. ISSN 10916490. doi:10.1073/pnas.1517384113.

Carlberg, K.; Bou-Mosleh, C.; and Farhat, C. 2011. Efficient nonlinear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *Int. J. Numer. Methods Eng.* 86: 155–181. doi:10.1002/nme.3050.

Carlberg, K.; Farhat, C.; Cortial, J.; and Amsallem, D. 2013. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.* 242: 623–647. doi:10.1016/j.jcp.2013.02.028.

Champion, K.; Lusch, B.; Nathan Kutz, J.; and Brunton, S. L. 2019. Data-driven discovery of coordinates and governing equations. *Proc. Natl. Acad. Sci. U. S. A.* 116(45): 22445–22451. ISSN 10916490. doi:10.1073/pnas.1906995116.

Chaturantabut, S.; and Sorensen, D. 2010. Nonlinear model reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.* 32(5): 2737–2764. doi:10.1137/090766498.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. doi:10.1007/978-3-662-55774-7_3.

Deheuvels, P.; and Martynov, G. 2008. A Karhunen-Loeve decomposition of a Gaussian process generated by independent pairs of exponential random variables. *J. Func. Anal.* 255(9): 2363–2394. doi:10.1016/j.jfa.2008.07.021.

Dupont, E.; Doucet, A.; and Teh, Y. W. 2019. Augmented Neural ODEs. URL http://arxiv.org/abs/1904.01681.

Dutta, S.; Farthing, M. W.; Perracchione, E.; Savant, G.; and Putti, M. 2020. A greedy non-intrusive reduced order model for shallow water equations. URL https://arxiv.org/abs/2002.11329.

Fang, F.; Pain, C.; Navon, I.; ElSheikh, A.; Du, J.; and Xiao, D. 2013. Non-linear Petrov-Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods. *J. Comput. Phys.* 234: 540–559. doi:10.1016/j.jcp.2012.10.011.

Fang, F.; Zhang, T.; Pavlidis, D.; Pain, C.; Buchanan, A.; and Navon, I. 2014. Reduced order modelling of an unstructured mesh air pollution model and application in 2D/3D urban street canyons. *Atmos. Env.* 96: 96–106.

Fasshauer, G. 2007. *Meshfree Approximation Methods with MATLAB* , volume 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing Company.

Ferrandis, J. d. A.; Triantafyllou, M.; Chryssostomidis, C.; and Karniadakis, G. 2019. Learning functionals via LSTM neural networks for predicting vessel dynamics in extreme sea states. URL https://arxiv.org/abs/1912.13382.

Finlay, C.; Jacobsen, J.-H.; Nurbekyan, L.; and Oberman, A. M. 2020. How to train your neural ODE: the world of Jacobian and kinetic regularization. URL http://arxiv.org/abs/2002.02798.

Freno, B. A.; and Carlberg, K. T. 2019. Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations. *Comput. Methods Appl. Mech. Eng.* 348: 250–296. doi:10.1016/j.cma.2019.01.024. URL https://doi.org/10.1016/j.cma.2019.01.024.

Gholami, A.; Keutzer, K.; and Biros, G. 2019. ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs. URL https://arxiv.org/abs/1902.10298.

Ghorbanidehno, H.; Lee, J.; Farthing, M.; Hesser, T.; Darve, E. F.; and Kitanidis, P. K. 2021. Deep learning technique for fast inference of large-scale riverine bathymetry. *Advances in Water Resources* 147: 103715. ISSN 0309-1708. doi:https://doi.org/10.1016/j.advwatres.2020.103715. URL http://www.sciencedirect.com/science/article/pii/S0309170819309418.

Gonzalez, F. J.; and Balajewicz, M. 2018. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems.

Guo, M.; and Hesthaven, J. 2019. Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* 345: 75–99. doi:10.1016/j.cma.2018.10.029.

Hesthaven, J.; and Ubbiali, S. 2018. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.* 363: 55–78. doi:10.1016/j.jcp.2018.02.037.

Iuliano, E.; and Quagliarella, D. 2013. Aerodynamic shape optimization via non-intrusive POD-based surrogate modelling. In *2013 IEEE Congr. Evol. Comput. CEC 2013*, 1467–1474. IEEE. doi:10.1109/CEC.2013.6557736.

Jolliffe, I. 1986. *Principal Component Analysis*. Springer New York, USA.

Kanaa, D.; Voleti, V.; Kahou, S.; and Pal, C. 2019. Simple Video Generation using Neural ODEs. In *Annu. Conf. Neural Inf. Process. Syst. 2019, NeurIPS 2019*. Vancouver, BC, Canada. URL https://voletiv.github.io/docs/publications/2019e_NeurIPSW_EncODEDec.pdf.

Kim, Y.; Choi, Y.; Widemann, D.; and Zohdi, T. 2020. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. URL http://arxiv.org/abs/2009.11990.

Koopman, B. O. 1931. Hamiltonian Systems and Transformation in Hilbert Space. *Proc. Natl. Acad. Sci. U. S. A.* 17(5): 315–8.

Kutz, J. N.; Brunton, S. L.; Brunton, B. W.; and Proctor, J. L. 2016. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, third edition. ISBN 978-1-61197-449-2. doi:10.1137/1.9781611974508.

Kutz, J. N.; Fu, X.; and Brunton, S. L. 2016. Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* 15(2): 713–735. doi:10.1137/15M1023543.

Le Clainche, S.; and Vega, J. M. 2017. Higher order dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* 16(2): 882–925. doi:10.1137/15M1054924.

Lee, K.; and Carlberg, K. T. 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* 404: 108973. doi:10.1016/j.jcp.2019.108973. URL https://doi.org/10.1016/j.jcp.2019.108973.

Long, Z.; Lu, Y.; and Dong, B. 2019. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.* 399: 108925. doi:10.1016/j.jcp.2019.108925. URL https://doi.org/10.1016/j.jcp.2019.108925.

Lozovskiy, A.; Farthing, M.; and Kees, C. 2017. Evaluation of Galerkin and Petrov-Galerkin model reduction for finite element approximations of the shallow water equations. *Comput. Methods Appli. Mech. Eng.* 318: 537–571. doi:10.1016/j.cma.2017.01.027.

Lozovskiy, A.; Farthing, M.; Kees, C.; and Gildin, E. 2016. POD-based Model Reduction for Stabilized Finite Element Approximations of Shallow Water Flows. *J. Comput. Appl. Math.* 302: 50–70.

Lusch, B.; Nathan Kutz, J.; and Brunton, S. L. 2018. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Comm.* 9(1). doi:https://doi.org/10.1038/s41467-018-07210-0.

Maulik, R.; Mohan, A.; Lusch, B.; Madireddy, S.; Balaprakash, P.; and Livescu, D. 2020. Time-series learning of latent-space dynamics for reduced-order model closure. *Phys. D Nonlinear Phenom.* 405: 132368. doi:10.1016/j.physd.2020.132368. URL https://doi.org/10.1016/j.physd.2020.132368.

Mezić, I. 2013. Analysis of Fluid Flows via Spectral Properties of the Koopman Operator. *Ann. Rev. Flu. Mech.* 45(1): 357–378. doi:10.1146/annurev-fluid-011212-140652.

Peherstorfer, B.; Willcox, K.; and Gunzburger, M. A. X. 2016. Optimal model management for multifidelity Monte Carlo estimation. *SIAM J. Sci. Comput.* 38(5): A3163–A3194. doi:10.1137/15M1046472.

Proctor, J. L.; Brunton, S. L.; and Kutz, J. N. 2016. Dynamic mode decomposition with control. *SIAM J. Appl. Dyn. Syst.* 15(1): 142–161. doi:10.1137/15M1013857.

Quarteroni, A.; Manzoni, A.; and Negri, F. 2016. *Reduced Basis Methods for Partial Differential Equations*. Springer, Cham. doi:10.1007/978-3-319-15431-2.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378: 686–707. ISSN 10902716. doi:10.1016/j.jcp.2018.10.045. URL https://doi.org/10.1016/j.jcp.2018.10.045.

Rowley, C.; Mezi, I.; Bagheri, S.; Schlatter, P.; and Henningson, D. 2009. Spectral analysis of nonlinear flows. *J. Fluid Mech.* 641(Rowley 2005): 115–127. doi:10.1017/S0022112009992059.

Rubanova, Y.; Chen, R. T. Q.; and Duvenaud, D. K. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32, 5320–5330. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

Ruthotto, L.; and Haber, E. 2019. Deep Neural Networks Motivated by Partial Differential Equations. *J. Math. Imaging Vis.* 2–10. doi:10.1007/s10851-019-00903-1.

San, O.; and Borggaard, J. 2015. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flow. *Int. J. Numer. Methods Fluids* 78(1): 37–62.

Sapsis, T. P.; and Majda, A. J. 2013. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proc. Natl. Acad. Sci. U. S. A.* 110(34): 13705–13710. doi:10.1073/pnas.1313065110.

Schmid, P. 2010. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656(July 2010): 5–28. doi:10.1109/PIERS-FALL.2017.8293532.

Sirovich, L. 1987. Turbulence and the dynamics of coherent structures. Part I:Coherent structures. *Quart. Appl. Math.* 45: 561–571.

Stefanescu, R.; Sandu, A.; and Navon, I. 2014. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *Int. J. Numer. Methods Fluids* 76(8): 497–521.

Sun, Y.; Zhang, L.; and Schaeffer, H. 2020. NeuPDE: Neural Network Based Ordinary and Partial Differential Equations for Modeling Time-Dependent Data. *Proc. Mach. Learn. Res.* 107(2016): 352–372.

Taira, K.; Hemati, M. S.; Brunton, S. L.; Sun, Y.; Duraisamy, K.; Bagheri, S.; Dawson, S. T.; and Yeh, C. A. 2020. Modal analysis of fluid flows: Applications and outlook. *AIAA J.* 58(3): 998–1022. doi:10.2514/1.J058462.

Trahan, C.; Savant, G.; Berger, R.; Farthing, M.; McAlpin, T.; Pettey, L.; Choudhary, G.; and Dawson, C. 2018. Formulation and application of the adaptive hydraulics three-dimensional shallow water and transport models. *J. Comput. Phys.* 374: 47–90. doi:10.1016/j.jcp.2018.04.055.

Vermeulen, P.; and Heemink, A. 2006. Model-reduced variational data assimilation. *Monthly Weather Review* 134: 2888–2899.

Wang, Q.; Ripamonti, N.; and Hesthaven, J. S. 2020. Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism. *J. Comput. Phys.* 410: 109402. doi:10.1016/j.jcp.2020.109402. URL https://doi.org/10.1016/j.jcp.2020.109402.

Willcox, K. 2006. Unsteady flow sensing and estimation via the gappy Proper Orthogonal Decomposition. *Comput. Fluids* 35(2): 208–226.

Xiao, D. 2019. Error estimation of the parametric non-intrusive reduced order model using machine learning. *Comput. Methods Appl. Mech. Eng.* 355: 513–534. doi:10.1016/j.cma.2019.06.018. URL https://doi.org/10.1016/j.cma.2019.06.018.

Xiao, D.; Fang, F.; Buchan, A.; Pain, C.; Navon, I.; Du, J.; and Hu, G. 2014. Non-linear model reduction for the Navier-Stokes equations using residual DEIM method. *J. Comput. Phys.* 263: 1–18. doi:10.1016/j.jcp.2014.01.011.

Xiao, D.; Fang, F.; Pain, C.; and Hu, G. 2015. Non-intrusive reduced-order modelling of the Navier-Stokes equations based on RBF interpolation. *Int. J. Numer. Methods Fluids* 79: 580–595. doi:10.1002/fld.406.

Xiao, D.; Fang, F.; Pain, C.; and Navon, I. 2017. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Comput. Methods Appl. Mech. Eng.* 317: 868–889. doi:10.1016/j.cma.2016.12.033.

Zhang, H.; Gao, X.; Unterman, J.; and Arodz, T. 2019. Approximation Capabilities of Neural ODEs and Invertible Residual Networks. *CoRR* abs/1907.12998. URL http://arxiv.org/abs/1907.12998.