

Low-Rank Registration Based Manifolds for Convection-Dominated PDEs

Rambod Mojgani,^{1,2} Maciej Balajewicz^{1,3}

¹ University of Illinois at Urbana-Champaign,

² Rice University,

³ Siemens Technology

{mojgani2,mbalajew}@illinois.edu

Abstract

We develop an auto-encoder-type nonlinear dimensionality reduction algorithm to enable the construction of reduced order models of systems governed by convection-dominated nonlinear partial differential equations (PDEs), i.e. snapshots of solutions with large Kolmogorov n -width. Although several existing nonlinear manifold learning methods, such as LLE, ISOMAP, MDS, etc., appear as compelling candidates to reduce the dimensionality of such data, most are not applicable to reduced order modeling of PDEs, because: (i) they typically lack a straightforward mapping from the latent space to the high-dimensional physical space, and (ii) the identified latent variables are often difficult to interpret. In our proposed method, these limitations are overcome by training a low-rank diffeomorphic spatio-temporal grid that registers the output sequence of the PDEs on a non-uniform parameter/time-varying grid, such that the Kolmogorov n -width of the mapped data on the learned grid is minimized. We demonstrate the efficacy and interpretability of our proposed approach on several challenging manufactured computer vision-inspired tasks and physical systems.

Introduction

Many physical phenomena of engineering interest are described using partial differential equations (PDEs). High accuracy simulations of these equations often require extensive computational resources. Although cheaper and faster processors, as well as highly parallel architectures, have made many large-scale computations viable, reduced order models (ROMs) of these systems remain attractive, especially in many-query simulations and real-time control.

The goal of reduced order modeling is to leverage the vast amount of data generated from high accuracy simulations to learn a low-dimensional model that can accurately and efficiently approximate the underlying dynamical system. For many of these systems a low-rank linear combination of the bases can represent the solution (Li et al. 2020), however, such low-rank reconstructions are especially inaccurate for convection-dominated PDEs, where the Kolmogorov n -width of the snapshots of the solution is relatively large, i.e. the solution cannot be effectively reduced on a linear subspace. Such problems emerge frequently in

a broad range of applications, from Navier-Stokes equations (fluid dynamics) to Schrödinger equation (quantum-mechanical systems) (Mendible et al. 2020). In the machine learning community, the recognition of similar challenges dates back to the 1990s and attempts in classification of handwritten digits (Hinton, Revow, and Dayan 1995), where presence of simple transformations such as translations and rotations in the data-set is well known to dramatically deteriorate the accuracy of linear methods such as principle component analysis (PCA). Fundamentally, other linear manifolds (subspaces) suffer from similar drawbacks, examples include proper orthogonal decomposition (POD), multidimensional scaling (MDS) (Cox and Cox 2008), factor analysis (Friedman, Hastie, and Tibshirani 2001) and independent component analysis (ICA) (Friedman, Hastie, and Tibshirani 2001). Therefore, the high dimensionality of the data on any of these linear manifolds has incentivized a slew of nonlinear manifold learning approaches, such as Iso-map (Tenenbaum 1998), kernel PCA (Mika et al. 1999), locally linear embedding (LLE) (Roweis 2000), Laplacian eigenmaps (LEM) (Belkin and Niyogi 2003), semi-definite embedding (SDE) (Weinberger, Sha, and Saul 2004), auto-encoders (G.E and R.R 2006), t-SNE (Maaten and Hinton 2008), and diffeomorphic dimensionality reduction (Walder and Schölkopf 2009).

Although many of the aforementioned nonlinear methods provide the sought after low-dimensional manifold, only a few provide a mapping from the learned low-dimensional to the high-dimensional manifold, for a survey see (Lee and Carlberg 2020). This is especially important in reduced order modeling of PDEs, since the latent variables evolve on a parameter/temporal space, followed by a mapping of the latent variable from the low-dimensional to the high-dimensional physical manifold. Auto-encoders (AE), specifically convolutional auto-encoders (CAEs) (Masci et al. 2011) and deep convolutional generative adversarial networks (DCGANs) (Radford, Metz, and Chintala 2016), are among the successful methods used in dimensionality reduction of PDEs (Lee and Carlberg 2020; Cheng et al. 2020). However, linear manifolds such as proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD) are still often preferred to the neural network-based AEs, since they provide an interpretable framework for analysis of the system, as well as controlling of the re-

duced system. POD reveals the coherent structures in fluid flows (Noack, Morzynski, and Tadmor 2011; Holmes et al. 2012), and DMD obtains a finite-dimensional, matrix approximations of the Koopman operator, which opens the possibility of utilizing the estimation and control theories developed for the linear systems (Kutz et al. 2016). In a more recent effort, it is shown that deep AE architectures can be trained to transform nonlinear PDEs into linear PDEs (Gin et al. 2020). In this approach, although the transformation is nonlinear, the latent variables lie on a linear subspace. Finally, a similar approach that prioritizes the optimal reducibility via a nonlinear manifold is lacking. Such an approach, by definition, results in a low-rank reduced order model and motivates the present study.

In this paper – inspired by registration based manifold learning approaches, e.g. (Walder and Schölkopf 2009; Taddei 2020) – we develop a new manifold learning algorithm to enable efficient construction of reduced order models of convection-dominated physical systems. To this end, we pose an unsupervised learning problem, that learns a spatio-temporal grid on which the low-rank linear decomposition of the solution of the PDE is optimal. The proposed method: (i) provides a nonlinear map from the latent manifold to the physical high dimensional space, which typically cannot be achieved using standard nonlinear techniques such as LLE, ISOMAP, MDS, etc., (ii) enables accurate predictive extrapolatory simulations (forecasting) for a large and important class of dynamical systems, i.e. hyperbolic/convection-dominated problems, which other methods, including neural network based auto-encoders, cannot achieve, and (iii) provides intuitive and interpretable manifolds for domain specific scientists/engineers.

Preliminaries

In approximation theory, Kolmogorov n -width is used to measure how well the data – the solution of the PDEs in the context of this paper – can be approximated on a linear manifold, i.e. subspace. The connection between the Kolmogorov n -width and POD of the data is rigorously established (Djouadi 2010), leading to a measure on the accuracy/feasibility of a low-rank ROM on a subspace.

Constructing the bases from snapshots in the spirit of the POD method can be formulated as a low-rank matrix approximation problem as follows:

For a given snapshot matrix $\mathbf{M} \in \mathbb{R}^{N \times K}$, find a rank- k matrix $\widetilde{\mathbf{M}} \in \mathbb{R}^{N \times K}$ that solves the minimization problem

$$\begin{aligned} & \underset{\widetilde{\mathbf{M}}}{\text{minimize}} \quad \left\| \mathbf{M} - \widetilde{\mathbf{M}} \right\|_F, \\ & \text{subject to} \quad \text{rank}(\widetilde{\mathbf{M}}) = k, \end{aligned} \quad (1)$$

where for an efficient reduction, $k \ll N$ and $k \ll K$. A snapshot matrix, $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K] \in \mathbb{R}^{N \times K}$, is a matrix which its i^{th} column, $\mathbf{m}_i \in \mathbb{R}^N$, contains the states of the system/PDEs of interest at the i^{th} system parameter, time, or the boundary/initial conditions, and $\widetilde{\mathbf{M}} = [\widetilde{\mathbf{m}}_1, \widetilde{\mathbf{m}}_2, \dots, \widetilde{\mathbf{m}}_K] \in \mathbb{R}^{N \times K}$ is the low-rank reconstruction/approximation of the snapshots matrix, and the error is defined as the distance between \mathbf{M} and $\widetilde{\mathbf{M}}$.

In (1), the rank constraint can be taken care of by representing the unknown matrix as $\widetilde{\mathbf{M}} = \mathbf{U}\mathbf{V}$, where $\mathbf{U} \in \mathbb{R}^{N \times k}$ and $\mathbf{V} \in \mathbb{R}^{k \times K}$, so that problem (1) becomes

$$\underset{\mathbf{U}, \mathbf{V}}{\text{minimize}} \quad \left\| \mathbf{M} - \mathbf{U}\mathbf{V} \right\|_F. \quad (2)$$

It is well known that the solution of the low-rank approximation problem of (2) is given by the singular value decomposition (SVD) of \mathbf{M} . Specifically, $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{N \times k}$ and $\mathbf{V} = \boldsymbol{\Sigma}[\mathbf{v}_1, \dots, \mathbf{v}_k] \in \mathbb{R}^{k \times K}$, where $\mathbf{M} = \mathbf{U}^* \boldsymbol{\Sigma}^* \mathbf{V}^{*T}$, $\mathbf{U}^* = [\mathbf{u}_1, \dots, \mathbf{u}_k, \mathbf{u}_{k+1}, \dots, \mathbf{u}_N]$, $\mathbf{V}^{*T} = [\mathbf{v}_1, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_K]$, and $\boldsymbol{\Sigma}^* = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ is a diagonal rank- r matrix of singular values, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$, and $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k)$. This decomposition has a very close connotation to factor analysis (Friedman, Hastie, and Tibshirani 2001) and can be reproduced by artificial neural networks with linear activations (Fyfe 1997). Although the linearity of the learned manifold leads to inefficiencies in convection-dominated PDEs with large Kolmogorov n -width, the existence of a closed form solution, the abundance of computationally efficient approaches such as (Holmes, Gray, and Lee Isbell 2009), as well as its physical interpretability (Holmes et al. 2012), has made POD a predominantly utilized approach in the reduced order modeling of PDEs. Our goal is to extend the norm minimization problem of (2) to an efficient and interpretable nonlinear manifold learning problem.

Low-rank Registration Based Manifold

We generalize the linear manifold learning problem of (1), as a nonlinear manifold learning as follows:

For a given high-dimensional data lying on a manifold learn a map $\mathcal{G}^{-1}(\cdot)$ and its corresponding manifold, on which the mapped data can be accurately expressed by a low-rank linear decomposition. The map from the high-dimensional physical manifold to the learned manifold is denoted by $\mathcal{G}(\cdot)$ and its inverse by $\mathcal{G}^{-1}(\cdot)$. In the case of finite-dimension matrix space, $\mathcal{G}(\cdot) : \mathbb{R}^{N \times K} \rightarrow \mathbb{R}^{N \times K}$ and $\mathcal{G}^{-1}(\cdot) : \mathbb{R}^{N \times K} \rightarrow \mathbb{R}^{N \times K}$, the minimization has the following form:

$$\begin{aligned} & \underset{\mathcal{G}^{-1}(\cdot), \widetilde{\mathbf{M}}}{\text{minimize}} \quad \left\| \mathbf{M} - \mathcal{G}^{-1}(\widetilde{\mathbf{M}}) \right\|_F, \\ & \text{subject to} \quad \text{rank}(\widetilde{\mathbf{M}}) = k_r. \end{aligned} \quad (3)$$

In this case, $\widetilde{\mathbf{M}} = \mathbf{U}\mathbf{V} \in \mathbb{R}^{N \times K}$ is the low-rank linear decomposition of the data on the learned manifold, where $\mathbf{U} \in \mathbb{R}^{N \times k_r}$ and $\mathbf{V} \in \mathbb{R}^{k_r \times K}$. In principle, the compression of the data on the learned manifold is lossless. The proposed minimization of (3) outperforms (1) if and only if for a similar reconstruction/approximation error, $k_r \ll k$.

Diffeomorphism and Interpolation

We impose diffeomorphism as a condition on the mapping to and from the learned manifold. By definition, a map $\mathcal{G}(\cdot)$ is said to be diffeomorphic if $\mathcal{G}(\cdot)$ and $\mathcal{G}^{-1}(\cdot)$ are differentiable (Modersitzki 2009). Bijectivity (i.e. one to oneness) and smoothness guarantee diffeomorphism. Therefore, by

Algorithm 1 The map from the constant grid to the parameter/time-varying grid, $\mathcal{G}(\cdot)$

Input: The constant grid ($\mathbf{X} = \mathbf{x}_0 \odot \mathbf{1} \in \mathbb{R}^{N \times K}$),
The parameter/time-varying grid ($\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$),
The snapshots of the state variables on the constant grid ($\mathbf{M} \in \mathbb{R}^{N \times K}$)
Output: The snapshots of the state variables on the parameter/time-varying grid ($\tilde{\mathbf{M}} \in \mathbb{R}^{N \times K}$)

- 1: **for** $i = 1, 2, 3, \dots, K$ **do**
- 2: $\tilde{\mathbf{m}}_i \leftarrow$ interpolate \mathbf{m}_i stated on \mathbf{x}_0 to $\tilde{\mathbf{x}}_i$
 // Using the interpolation scheme of choice
- 3: **end for**

enforcing the map to be diffeomorphic, we ensure existence and uniqueness of $\tilde{\mathbf{M}}$ given \mathbf{M} , and vice versa. Bijectivity is achieved by ensuring that volume of all the cells remain strictly positive. A negative cell volume leads to the indeterminate derivative of the state parameter, which can be seen as a ‘‘tear’’ in an image. Smoothness of the grid is maintained by penalizing abrupt changes of the grid volume, both in space and parameter/time via regularization terms.

To this point and to emphasize on the generality of the proposed method, we have intentionally introduced the idea of the mapping between the manifolds from an abstract viewpoint. Hereon, we tie the definition of the map between the manifold to any of the off-the-shelf interpolation schemes, between the constant and parameter/time-varying grid. The snapshot matrix of $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K] \in \mathbb{R}^{N \times K}$ is defined on a constant grid $\mathbf{X} = \mathbf{x}_0 \odot \mathbf{1} = [\mathbf{x}_0, \mathbf{x}_0, \dots, \mathbf{x}_0] \in \mathbb{R}^{N \times K}$ (in physics: Eulerian framework) and we, by construct, associate $\tilde{\mathbf{M}} = [\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \dots, \tilde{\mathbf{m}}_K] \in \mathbb{R}^{N \times K}$ to the snapshots of latent variables on a parameter/time-varying grid $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_K] \in \mathbb{R}^{N \times K}$ (in physics: arbitrary Lagrangian Eulerian or ALE framework). In contrast to most machine vision tasks dealing with images, the data does not necessarily lie on a uniform Cartesian grid. This distinction is important since many of the PDEs are discretized on unstructured computational grids. Any of the off-the-shelf interpolation schemes, between the constant and parameter/time-varying grid, can be utilized. A simple implementation of this procedure is elaborated in Alg. 1 and Alg. 2. Therefore, minimization problem of (3) can be interpreted as a registration task, that minimizes the Kolmogorov n-width of the snapshots of the latent variables on the learned parameter/time-varying grid.

Assumptions of Low-rank Grid for Convection

So far, we have tied the notion of the mapping between manifolds to interpolation of the snapshots between the constant and time/parameter-varying grids. In this section, we impose a structure to the identified grid based on the known physics of the convection-dominated PDEs.

There are two general approaches to formulate the grid deformation in a registration problem. In the first class of approaches, the grid nodes are controlled as the solution

Algorithm 2 The map from the parameter/time-varying grid to the constant grid to, $\mathcal{G}^{-1}(\cdot)$

Input: The constant grid ($\mathbf{X} = \mathbf{x}_0 \odot \mathbf{1} \in \mathbb{R}^{N \times K}$),
The parameter/time-varying grid ($\tilde{\mathbf{X}} \in \mathbb{R}^{N \times K}$),
The snapshots of the state variables on the parameter/time-varying grid ($\tilde{\mathbf{M}} \in \mathbb{R}^{N \times K}$),
Output: The snapshots of the state variables on the constant grid ($\mathbf{M} \in \mathbb{R}^{N \times K}$)

- 1: **for** $i = 1, 2, 3, \dots, K$ **do**
- 2: $\mathbf{m}_i \leftarrow$ interpolate $\tilde{\mathbf{m}}_i$ stated on $\tilde{\mathbf{x}}_i$ to \mathbf{x}_0
 // Using the interpolation scheme of choice
- 3: **end for**

of the a minimization problem and diffeomorphism is enforced by constraining the determinant of the deformation gradient to be strictly positive for all grid cells. This approach leads to a high-dimensional optimization problem which its nonlinearity and ill-posedness makes it computationally challenging (Mang et al. 2018). In the second class of approaches, the mapping is the solution of a transport equation, i.e. flow fields, as in diffeomorphic dimensionality reduction (Walder and Schölkopf 2009). Interestingly, in some special cases, a similar transport equation arises where the frame of references is changed from the Eulerian to the Lagrangian viewpoint, i.e. by solving the hyperbolic PDEs on the corresponding characteristic lines. This change of the reference is proven to be efficient in reduced order modeling of convection-dominated PDEs (Mojgani and Balajewicz 2017; Lu and Tartakovsky 2020). In the present paper, we demonstrate the existence of a low-rank near-optimal grid for many of the convection-dominated PDEs. We extend this change of frame to arbitrary systems in a data-driven setting. In our proposed method the coordinates of the low-rank parameter/time-varying grid is defined as

$$\tilde{\mathbf{X}} := \mathbf{X} + \mathbf{U}_x \mathbf{V}_x \in \mathbb{R}^{N \times K}, \quad (4)$$

where \mathbf{X} is the constant grid, $\mathbf{U}_x \in \mathbb{R}^{N \times r}$ and $\mathbf{V}_x \in \mathbb{R}^{r \times K}$, therefore $\mathbf{U}_x \mathbf{V}_x \in \mathbb{R}^{N \times K}$ is a rank- r matrix. The latent variables can be interpreted as the evolution of the state parameters on the low-rank approximation of the characteristic lines of the hyperbolic PDEs, depicting the direction on which the information travels. This physics-based assumption is one of the key and differentiating elements of the proposed method, (i) reducing the size of the optimization compared to the existing registration-based methods, such as (Taddei 2020), and (ii) resulting in unprecedented extrapolatory predictive models, i.e. accurate models beyond the training range.

Implementation

In this section, we summarize the elements of the proposed algorithm and clarify the implementation of the method. The procedure is designed to learn a low-rank grid, $\tilde{\mathbf{X}}$, on which the mapped snapshots, $\tilde{\mathbf{M}} = \mathcal{G}(\mathbf{M})$, is low-rank (Fig. 1). The final dimensionality reduction problem, with the mini-

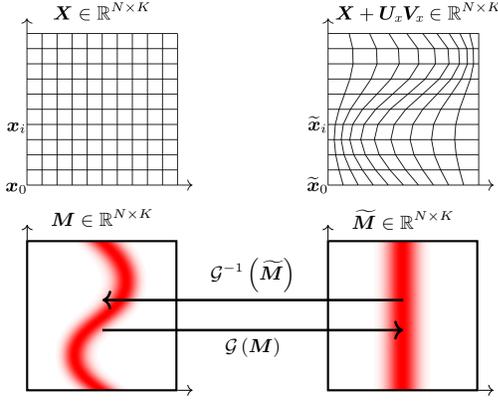


Figure 1: Illustration of the proposed manifold. The solution of a convection-dominated PDE is given on a time-constant grid, $M \in \mathbb{R}^{N \times K}$ expressed on $X \in \mathbb{R}^{N \times K}$. The identified manifold is defined by a time/parameter-varying grid, $\tilde{X} = X + U_x V_x \in \mathbb{R}^{N \times K}$. The manifold is trained such that the snapshot of the mapped states, $\tilde{M} = \mathcal{G}(M)$, is low-rank.

mizer \tilde{X} , has the following form:

$$\begin{aligned} & \underset{U, V, U_x, V_x}{\text{minimize}} && \mathcal{J}, \\ & \text{subject to} && v_n \geq v_{\min}, \forall n \in \{1, \dots, K\}, \\ & && \tilde{x}_n|_{\partial\Omega} = x|_{\partial\Omega}, \forall n \in \{1, \dots, K\}, \end{aligned} \quad (5)$$

where

$$\begin{aligned} \mathcal{J} = & \|M - \mathcal{G}^{-1}(UV)\|_F + \|\Gamma_1 U_x\|_F + \dots \\ & + \|\mathbf{V}_x \Gamma_2^T\|_F, \end{aligned} \quad (6)$$

and $\mathcal{G}^{-1}(\cdot)$ interpolates the low-rank mapped snapshots, UV , stated on a low-rank parameter/time-varying grid, \tilde{X} , onto the constant grid, X , i.e. $\mathcal{G}^{-1}(\cdot) : \tilde{X} = X + U_x V_x \rightarrow X$. The Tikhonov matrices, $\Gamma_1 \in \mathbb{R}^{N \times N}$ and $\Gamma_2 \in \mathbb{R}^{K \times K}$, designed to promote grid smoothness, are calibrated using well-known L-curve methods. Also v_n is a vector of cell volumes of the parameter/time-varying grid at the n^{th} parameter/time step, v_{\min} is the minimum admissible cell volume, and $\tilde{x}_n|_{\partial\Omega}$ and $x|_{\partial\Omega}$ are boundary points of the learned grid and the constant grid, respectively. The appropriate scaling of Tikhonov matrices and the minimum cell volumes are hyper-parameters and problem dependent.

Moreover, from a practical standpoint, a weak constraint on the rank reduction is chosen. While (3) implies minimizing over the rank of \tilde{M} , in many cases, the solution of the minimization for a preset size of the decomposition is preferred. Therefore, in (5), we assume $\tilde{M} = UV$, where $U \in \mathbb{R}^{N \times k_r}$, $V \in \mathbb{R}^{k_r \times K}$ and $k_r \ll N$, $k_r \ll K$. Also, to reduce the size of the optimization problem, U_x and V_x are uniformly down-sampled/coarsened, however, the objective (\mathcal{J}) is evaluated on the fine grid.

Remark 1: For $V_x = \mathbf{0} \in \mathbb{R}^{r \times K}$ leading to $\tilde{X} = X$, the minimization problem of (5) reduces to (2), and $\mathcal{G}^{-1}(\cdot)$ acts as an identity map.

Algorithm 3 Training of the proposed low-rank registration based manifold

Input:

Hyper-parameters:

- Γ_1, Γ_2 ,
- Minimum admissible grid volume (v_{\min}),

Reduction parameters:

- Rank of the parameter/time-varying grid (r),
- Rank of the low-dimensional representation (k_r),

The snapshots matrix ($M \in \mathbb{R}^{N \times K}$),

The constant grid ($X = x_0 \odot \mathbf{1} \in \mathbb{R}^{N \times K}$),

Maximum number of iterations (j_{\max}),

Output:

Low-rank parameter/time-varying grid $\tilde{X} = X + U_x V_x \in \mathbb{R}^{N \times K}$,

The maps where $\mathcal{G}(\cdot) : X \rightarrow \tilde{X}$ and $\mathcal{G}^{-1}(\cdot) : \tilde{X} \rightarrow X$

- 1: Initialize the time-varying grid, i.e. $\tilde{X}^{(0)} = X + U_x^{(0)} V_x^{(0)}$, with $U_x^{(0)} \in \mathbb{R}^{N \times r}$ and $V_x^{(0)} \in \mathbb{R}^{r \times K}$ using the SVD decomposition of the constant grid, X , plus a small random perturbation
 - 2: $j \leftarrow 0$
 - 3: **while** $j \leq j_{\max}$ **do**
 - 4: $\tilde{M} \leftarrow \mathcal{G}(M)$
// Interpolate the snapshots, M , onto $\tilde{X}^{(j)}$
 - 5: $UV \approx \tilde{M}$ s.t. $\text{rank}(UV) = k_r$
// Approximate \tilde{M} using its SVD
 - 6: $\mathcal{J} = \|M - \mathcal{G}^{-1}(UV)\|_F + \|\Gamma_1 U_x^{(j)}\|_F + \dots$
 $+ \|\mathbf{V}_x^{(j)} \Gamma_2^T\|_F$
// Evaluate the objective where $\mathcal{G}^{-1}(UV)$ interpolates UV onto the constant grid
 - 7: Update $U_x^{(j)}$ and $V_x^{(j)}$ minimizing \mathcal{J}
// Update the grid bases via the rule of the optimization
 - 8: $\tilde{X}^{(j+1)} = X^{(j)} + U_x^{(j)} V_x^{(j)}$
// Update the grid using the grid bases
 - 9: $j \leftarrow j + 1$
 - 10: **end while**
-

Remark 2: To interpolate the snapshots between the two sets of grid, we simply utilize a p -degree polynomial interpolation scheme in Alg. 1 and Alg. 2. This choice innately incorporates a sparsity pattern into the mapping. The latent space representation using a nearest-neighbor interpolation only requires one data-point, and a p -degree polynomial interpolation requires $p - 1$ entries of the input vector. This choice, in principle, leads to a great reduction in the size the optimization problem compared to the traditional neural networks, where there is no *a priori* assumption on the structure of the connectivity between the nodes.

Finally, the proposed low-rank registration based manifold (Alg. 3) can be utilized as an auto-encoder layer in a broad range of system identification applications, e.g. artificial neural networks (Fig. 2), improving the accuracy and training costs of machine learning architectures.

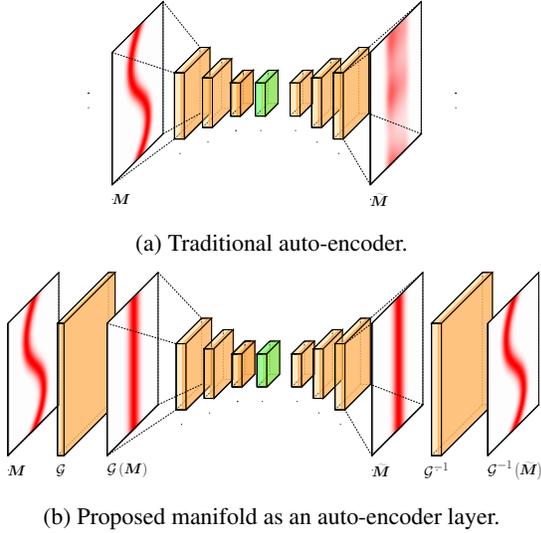


Figure 2: The proposed low-rank registration based map, $\mathcal{G}(\cdot)$ and $\mathcal{G}^{-1}(\cdot)$, as an auto-encoder layer in a deep learning architecture.

Experiments

In this section, we discuss the reducibility/compression of the snapshots and more importantly, leverage the interpretation in extrapolatory predictions of traveling features. We resort to readily available optimization packages capable of solving optimization with nonlinear constraints; such as Sequential Least Squares Programming in `scipy` for the Python implementation, or interior-point in `fmincon` for the MATLAB implementation. Linear and bi-linear interpolation schemes are used for one and two-dimensional problems. The implementations, data and the results are available at <https://github.com/rmojgani/PhysicsAwareAE>.

Manifold for Rotated Character “A”

Consider a computer vision task of learning the nonlinear transformation, rotation, given a data-set comprised of a rotated character “A”. The image of character “A” is stored in a 50×50 matrix and is rotated a total of 90 degrees with 3 degrees increments resulting in a snapshot matrix of dimension 2500×31 . A representative sample of the snapshots is shown in Fig. 3a, and a single POD mode reconstruction is illustrated in Fig. 3b. In this problem, \mathbf{U}_x is down-sampled to size of 7, i.e. the total of 49 control points. Moreover, $\mathbf{v}_{\min} = 0$, $\mathbf{\Gamma}_1 = 100\mathbf{D}_{xx}$ and $\mathbf{\Gamma}_2 = (100/\pi)\mathbf{D}_{\theta\theta}$, where \mathbf{D}_{xx} and $\mathbf{D}_{\theta\theta}$ are the second derivative matrices in the spatial and parameter space, respectively. The boundary point constraints are removed for this particular problem. The optimization problem of (5) approximates the rigid body rotation as in Fig. 3c. In Fig. 3d, the snapshots are approximated using a single basis ($k_r = 1$) on the learned manifold of $r = 1$. The reconstruction on the learned grid, using the proposed approach, is remarkably more accurate compared to the traditional POD. Figure 3e and 3f show the rank-2 grid following the rigid body rotation in the snapshots and the

improvements in the accuracy ($r = 2$).

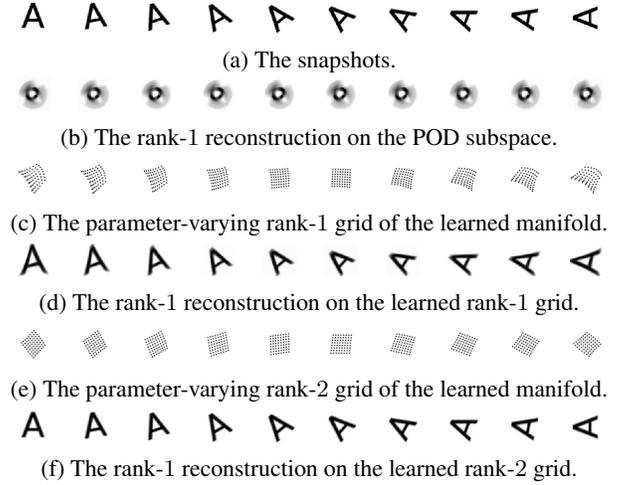


Figure 3: 90 degrees rotation of character “A”.

Manifold of Two-Dimensional Hyperbolic Fluid Flows

Consider the two-dimensional Riemann problem governed by Euler equations of fluid dynamics, $\frac{\partial}{\partial t} \mathbf{q} + \frac{\partial}{\partial x} \mathbf{f}_x + \frac{\partial}{\partial y} \mathbf{f}_y = 0$, where $\mathbf{q} = [\rho, \rho u, \rho v, \rho e]$, $\mathbf{f}_x = [\rho u, \rho u^2 + p, \rho uv, \rho uH]$, $\mathbf{f}_y = [\rho v, \rho uv + p, \rho v^2 + p, \rho vH]$, and $H = e + p/\rho$, $p = \rho(\gamma - 1)(e - 0.5(u^2 + v^2))$ in the domain $(x, y, t) \in [0, 1] \times [0, 1] \times [0, t_{\max}]$, with initial conditions of configuration 3 and 12 as in (Lax and Liu 1998). The snapshots of primitive variables are generated using a high-order artificial viscosity scheme coupled with a 4th-order Runge-Kutta time discretization with $\Delta t = 5 \times 10^{-4}$ on a 150×150 grid. Snapshots are collected at $\delta t = 0.016$ and $\delta t = 0.006$ intervals for simulation range of $t \in [0, 0.80]$ and $t \in [0, 0.25]$ for configuration 3 and 12, respectively. The size of the snapshot matrices in both cases are 10000×50 . A rank-2 time-varying grid ($r = 2$) is learned via (5) setting $k_r = 4$, and $\gamma_1 = \gamma_2 = 0.05$, where $\mathbf{\Gamma}_1 = \gamma_1 \mathbf{D}_{xx} = \gamma_1 \mathbf{D}_{yy}$ is the second derivative matrix in the x and y directions and $\mathbf{\Gamma}_2 = \gamma_2 \mathbf{D}_{tt}$, where \mathbf{D}_{tt} is the second derivative matrix in time. Also, $\mathbf{v}_{\min} = \Delta x_{\min} \Delta y_{\min}$, where $\Delta x_{\min} = \Delta y_{\min} = 6.7 \times 10^{-4}$. The low-dimensional representation of density is compared on the constant grid and the proposed manifold in Fig. 4. The identified grid follows the shock front and therefore the traveling shocks are conserved and the solution is free of non-physical oscillations on the proposed manifold, resulting in a higher accuracy of low-rank reconstruction.

Auto-encoder for Neural Network-Based Models

In this section, the proposed method acts as an auto-encoder layer wrapped around a traditional neural network-based machine learning architecture to decrease the loss in the off-line compression phase and subsequently to improve the predictive capabilities of a recurrent neural network modeling the governing PDEs. We employ a long short-term

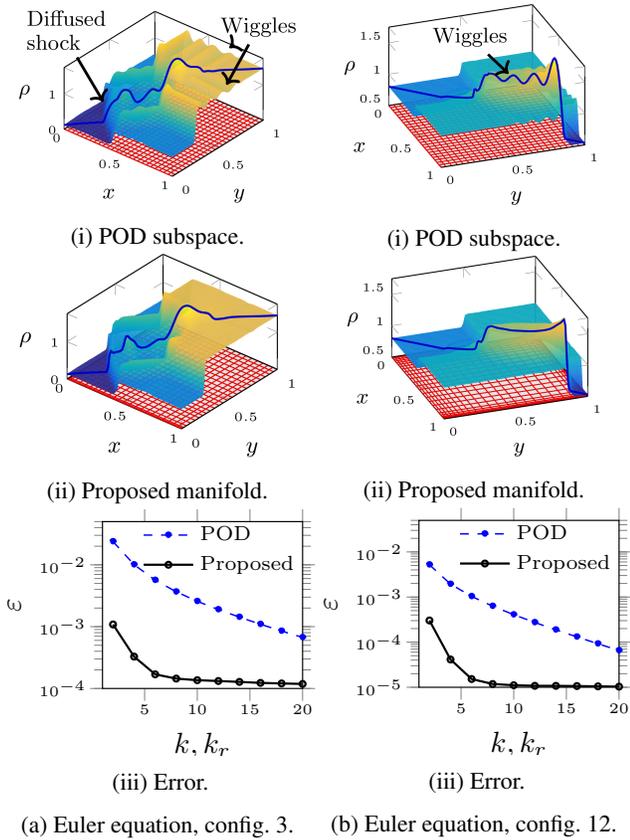


Figure 4: The reconstruction of density of the two-dimensional Riemann problem with (a) configuration 3 and (b) configuration 12 at the last time step of the simulations and the corresponding grid. The snapshots are of (i) $k = 8$ and (ii) $k_r = 8$ on (i) POD subspace and (ii) the proposed rank-2 manifold, respectively. The error convergence on the POD subspace and the proposed manifold for configuration 3 and 12 are compared in a-(iii) and b-(iii), respectively.

memory (LSTM) to approximate the dynamics of the PDE on the learned manifold, for the details see (Parish and Carlberg 2020). To learn a low-dimensional manifold, a 3-layer dense neural network auto-encoder with linear activation functions are used. The densely connected auto-encoder and the LSTM, implemented in Keras (Chollet et al. 2015 Accessed on 03/01/2021), are trained simultaneously on the snapshot matrix of $M \in \mathbb{R}^{N_x \times N_t}$ (Fig. 2a). In the proposed architecture (Fig. 2b), the manifold and the neural network are trained separately.

Consider the scalar, one-dimensional nonlinear convection-diffusion equation, known as viscous Burgers' equation, $\partial_t w(x, t) + w \partial_x w(x, t) = (1/Re) \partial_{xx} w(x, t)$ in the domain $(x, t) \in [x_a, x_b] \times [0, T]$, equipped with initial conditions $w(x, 0) = w_0(x)$, and Dirichlet boundary conditions at x_a , and x_b , where $w(x, 0) = 0.8 + 0.5 e^{-(x-0.5)^2/0.1^2}$, $w(x_a, t) = w(x_b, t) = 0$, for $(x, t) \in [0, 2.5] \times [0, 1]$. An implicit second order time discretization is used with $\Delta t = 8 \times 10^{-3}$ and space

is uniformly discretized where $\Delta x = 1 \times 10^{-2}$. In the proposed architecture, the rank-1 time-varying grid ($r = 1$), representing the low-rank auto-encoder, is learned as in (5) with $k_r = 4$. In this problem, $v_{\min} = 10^{-3}$, $\Gamma_1 = \gamma_1 D_{xx}$ and $\Gamma_2 = \gamma_2 D_{tt}$, where $\gamma_1 = \gamma_2 = 1$, and D_{xx} and D_{tt} are second derivative matrices in space and time. Subsequently, the LSTM is trained to approximate $\mathcal{G}(M)$. The contours and error convergence of the methods are compared in Fig. 5a-(i-iii), showing up to 3 orders of magnitude increase in the accuracy at the most compact networks.

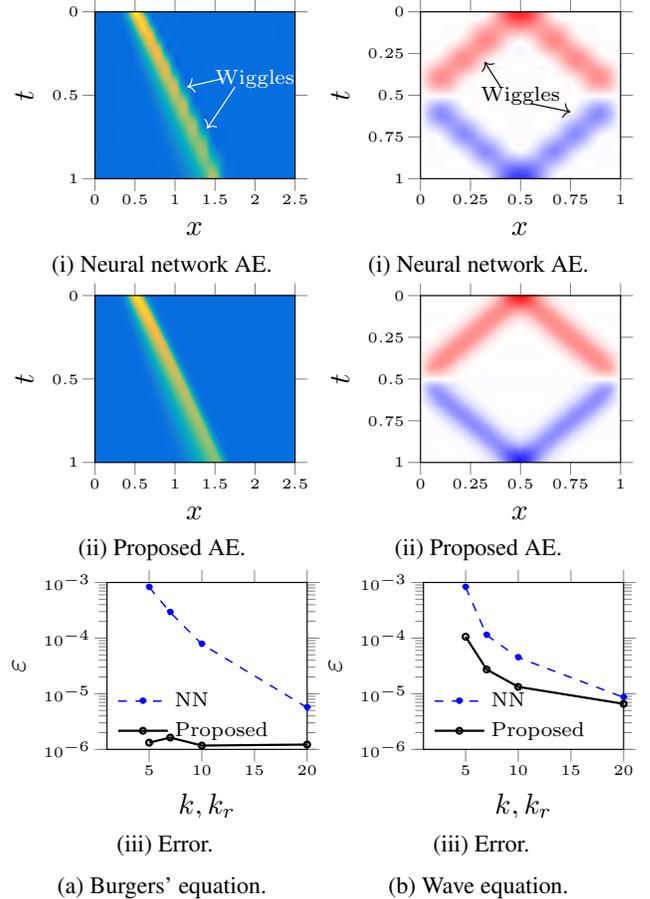


Figure 5: LSTM approximation of the (a) Burgers' and (b) wave equations, on the AE with bottleneck of size 10 and 5. The error convergence of the traditional neural network AE and the proposed AE for Burgers' and wave equations are compared in a-(iii) and b-(iii).

Also, consider the one-dimensional wave equation, $\partial_{tt} w(x, t) - \partial_{xx} w(x, t) = 0$, in the domain $(x, t) \in [x_a, x_b] \times [0, t_{\max}]$, equipped with initial conditions $w(x, 0) = w_0(x)$, and Dirichlet boundary conditions at x_a , and x_b , where $w(x, 0) = e^{-(x-0.5)^2/0.1^2}$, $w(x_a, t) = w(x_b, t) = 0$, for $(x, t) \in [0, 1] \times [0, 1]$. An implicit second-order time-discretization is used with $\Delta t = 2.5 \times 10^{-3}$ and space is uniformly discretized where $\Delta x = 10^{-2}$. The architecture is set up similar to the Burgers' equation, with the following parameters: the time-varying grid is of rank-

2 ($r = 2$), the reconstruction on the learned manifold is of rank-2 ($k_r = 2$), $v_{\min} = 10^{-3}$, $\gamma_1 = \gamma_2 = 10$, and the size of the grid bases, in both space and time, are down-sampled to 15 control points. The solution and error is plotted in Fig. 5b(i–iii), showing the most increase in the accuracy for the most compact network.

Exploratory Prediction

The proposed low-rank manifold is applied to extrapolatory predictive regimes (forecasting), where the solution of the PDEs does not lie on the manifold of the training dataset. The task of extrapolatory prediction is often considered as a disconcerting task in machine learning (ML) models for PDEs. It is known that ML models are predictive only within the range of the parameters used in the training phase (interpolatory prediction). To address this limitation, some physical properties of the system have to be explicitly incorporated (Brunton, Noack, and Koumoutsakos 2020). Fig. 6 demonstrates application of the proposed low-rank registration in such extrapolatory predictive regimes. In these cases the models/reconstructions are built based on the snapshots collected in $t \in [0, 1]$ and $t \in [0, 0.1]$, and evolved in time for $t \in [0, 1.5]$ and $t \in [0, 0.25]$, respectively for the previously discussed Burgers’ and the two-dimensional Riemann (config. 12). As expected, neither the linear POD subspace nor the nonlinear neural network AE can predict the traveling features outside the training range, Fig. 6a-(i) and Fig. 6b-(i). The low-rank manifold is extended on the linear extrapolation of V_x , following the characteristic lines of the underlying hyperbolic problems, and therefore enabling both POD subspace and LSTM network to track the traveling shock, Fig. 6a-(ii) and Fig. 6b-(ii). The transient error for the two-dimensional Riemann is plotted in Fig. 6a-(iii), depicting a rapid increase of the error past the training range on the POD subspace and an accurate prediction on the proposed manifold, emphasizing that while POD fails to predict the traveling shock, while the proposed manifold captures it. Similar behavior is apparent in the neural network model of the Burgers’ equations. The error, ε , for LSTM of different dimensions are plotted in Fig. 6b-(iii). The neural network AE and nonlinear LSTM alone do not capture the traveling features/convection past the training range, regardless of their dimensions.

Conclusion

Efficient data-driven modeling of convection-dominated partial differential equations (PDEs) is challenging due to the large Kolmogorov n -width typically exhibited by their solutions. For example, it is well known that most existing neural network-based approaches fail to provide reasonable accuracy in the extrapolatory predictive (i.e. forecasting) regimes when trained on data-sets containing solutions of convection-dominated PDEs. In this paper, we propose an auto-encoder-type dimensionality reduction algorithm that significantly improved the predictive qualities of neural network models while, simultaneously, lowering the required training costs. In low-rank registration-based manifold, solution snapshots are mapped onto a low-rank parameter/time-varying grid, such that the Kolmogorov n -width of the latent

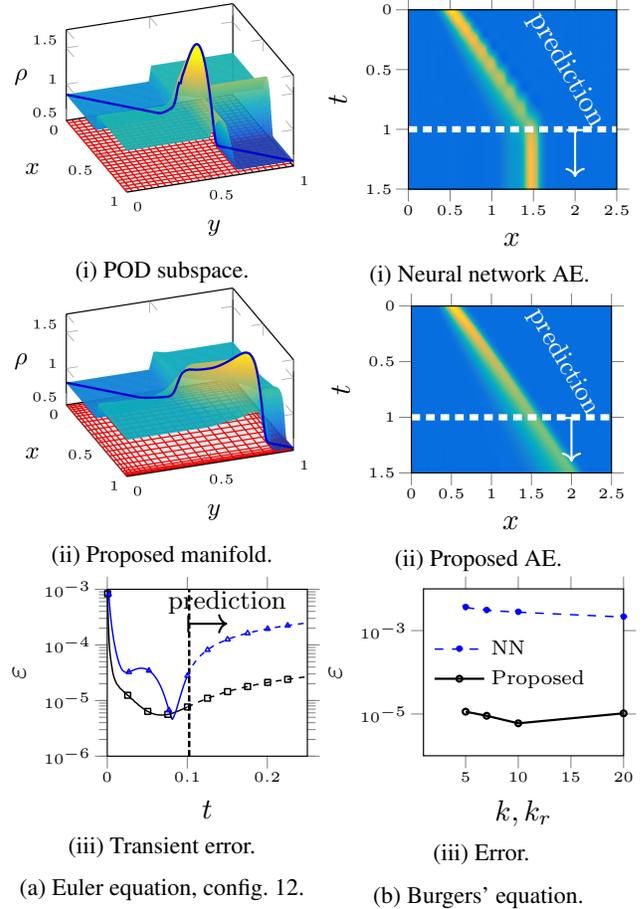


Figure 6: The extrapolatory prediction of the (a) two-dimensional Riemann (Config. 12), and (b) Burgers’ equation. (a) Two-dimensional Riemann (Config. 12): The models are trained in $t \in [0, 0.1]$ and the rank-2 reconstructions are plotted at $t = 0.25$ on the (i) POD subspace, and (ii) the proposed manifold. The transient error is plotted in (iii), where blue line: POD subspace, black line: proposed manifold. (b) Burgers’ equation: The models are trained in $t \in [0, 1.0]$ and LSTM solutions are plotted for $t \in [0, 1.5]$ on the (i) NN auto-encoder, and (ii) the proposed auto-encoder. The transient error and error convergence for config. 12 and Burgers’ equation are plotted in a-(iii) and b-(iii).

snapshots is minimized. We have successfully demonstrated this approach on several challenging engineering problems: (i) a computer vision task (reconstruction of images under nonlinear transformation), (ii) a neural network-based reduced order model of nonlinear convection-dominated flows (Burgers’ and wave equations), and (iii) dimensionality reduction of nonlinear two-dimensional fluid flows (Riemann problem governed by the Euler equations). The proposed approach is general, in that: a) it is, in principle, applicable to PDEs of arbitrary spatial dimensions, and b) is also applicable to a broad range of other reduced-order modeling approaches, including physics-based and gray-box modeling approaches such as projection-based approaches.

Acknowledgments

This material is based upon work supported by the Air Force Office of Scientific Research under Grant No. FA9550-17-1-0203.

References

- Belkin, M.; and Niyogi, P. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation* 15(6): 1373–1396. ISSN 0899-7667.
- Brunton, S. L.; Noack, B. R.; and Koumoutsakos, P. 2020. Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics* 52(1): 477–508.
- Cheng, M.; Fang, F.; Pain, C.; and Navon, I. 2020. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering* 365: 113000. ISSN 00457825.
- Chollet, F.; et al. 2015 Accessed on 03/01/2021. Keras. <https://keras.io>.
- Cox, M. A. A.; and Cox, T. F. 2008. Multidimensional Scaling. In *Handbook of Data Visualization*, 315–347. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Djouadi, S. M. 2010. On the connection between balanced proper orthogonal decomposition, balanced truncation, and metric complexity theory for infinite dimensional systems. *Proceedings of the 2010 American Control Conference, ACC 2010* 4911–4916.
- Friedman, J.; Hastie, T.; and Tibshirani, R. 2001. *The elements of statistical learning*. Number 10 in 1. Springer series in statistics New York.
- Fyfe, C. 1997. A neural network for PCA and beyond. *Neural Processing Letters* 6(1-2): 33–41.
- G.E, H.; and R.R, S. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313(July): 504–507. ISSN 0036-8075.
- Gin, C.; Lusch, B.; Brunton, S. L.; and Kutz, J. N. 2020. Deep learning models for global coordinate transformations that linearise PDEs. *European Journal of Applied Mathematics* 1–25.
- Hinton, G. E.; Revow, M.; and Dayan, P. 1995. Recognizing Handwritten Digits Using Mixtures of Linear Models. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*, 1015–1022. MIT Press.
- Holmes, M. P.; Gray, A. G.; and Lee Isbell, C. 2009. QUIC-SVD: Fast SVD using cosine trees. *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference* 673–680.
- Holmes, P.; Lumley, J. L.; Berkooz, G.; and Rowley, C. W. 2012. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 2nd edition.
- Kutz, J. N.; Brunton, S. L.; Brunton, B. W.; and Proctor, J. L. 2016. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM.
- Lax, P. D.; and Liu, X.-d. 1998. Solution of Two-Dimensional Riemann Problems of Gas Dynamics by Positive Schemes. *SIAM Journal on Scientific Computing* 19(2): 319–340.
- Lee, K.; and Carlberg, K. T. 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics* 404: 108973. ISSN 00219991.
- Li, J.; Sun, G.; Zhao, G.; and Li-wei, H. L. 2020. Robust Low-Rank Discovery of Data-Driven Partial Differential Equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 767–774.
- Lu, H.; and Tartakovsky, D. M. 2020. Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena. *Journal of Computational Physics* 407: 109229.
- Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(November): 2579–2605.
- Mang, A.; Gholami, A.; Davatzikos, C.; and Biros, G. 2018. PDE-constrained optimization in medical image analysis. *Optimization and Engineering* 19(3): 765–812. ISSN 1389-4420.
- Masci, J.; Meier, U.; Cireşan, D.; and Schmidhuber, J. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, 52–59. Springer.
- Mendible, A.; Brunton, S. L.; Aravkin, A. Y.; Lowrie, W.; and Kutz, J. N. 2020. Dimensionality reduction and reduced-order modeling for traveling wave physics. *Theoretical and Computational Fluid Dynamics* 34(4): 385–400. ISSN 0935-4964.
- Mika, S.; Schölkopf, B.; Smola, A. J.; Müller, K.-R.; Scholz, M.; and Rätsch, G. 1999. Kernel PCA and De-Noising in Feature Spaces. In Kearns, M. J.; Solla, S. A.; and Cohn, D. A., eds., *Advances in Neural Information Processing Systems 11*, 536–542. MIT Press.
- Modersitzki, J. 2009. *FAIR: Flexible Algorithms for Image Registration*. Society for Industrial and Applied Mathematics.
- Mojgani, R.; and Balajewicz, M. 2017. Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows. *arXiv:1701.04343*.
- Noack, B. R.; Morzynski, M.; and Tadmor, G. 2011. *Reduced-order modelling for flow control*, volume 528 of *CISM International Centre for Mechanical Sciences*. Springer Science & Business Media.
- Parish, E. J.; and Carlberg, K. T. 2020. Time-series machine-learning error models for approximate solutions to parameterized dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 365: 112990. ISSN 00457825.
- Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434*.

Roweis, S. T. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290(5500): 2323–2326. ISSN 00368075.

Taddei, T. 2020. A Registration Method for Model Order Reduction: Data Compression and Geometry Reduction. *SIAM Journal on Scientific Computing* 42(2): A997–A1027. ISSN 1064-8275.

Tenenbaum, J. B. 1998. Mapping a Manifold of Perceptual Observations. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems 10*, 682–688. MIT Press.

Walder, C.; and Schölkopf, B. 2009. Diffeomorphic Dimensionality Reduction. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems 21*, 1713–1720. Curran Associates, Inc.

Weinberger, K. Q.; Sha, F.; and Saul, L. K. 2004. Learning a kernel matrix for nonlinear dimensionality reduction. In *Twenty-first international conference on Machine learning - ICML '04*, 106. New York, New York, USA: ACM Press. ISBN 1581138285.