Reduced-order Model for Fluid Flows via Neural Ordinary Differential Equations

Carlos J.G. Rojas [1], **Andreas Dengel** [1], **Mateus Dias Ribeiro** [1]

[1] German Research Center for Artificial Intelligence - DFKI

carlos.gonzalez_rojas@dfki.de, andreas.dengel@dfki.de, mateus.dias_ribeiro@dfki.de

Abstract

Reduced order models play an important role in the design, optimization and control of dynamical systems. In recent years, there has been an increasing interest in the application of data-driven techniques for model reduction that can decrease the computational burden of numerical solutions, while preserving the most important features of complex physical problems. In this paper, we use the proper orthogonal decomposition to reduce the dimensionality of the model and introduce a novel generative neural ODE (NODE) architecture to forecast the behavior of the temporal coefficients. With this methodology, we replace the classical Galerkin projection with an architecture characterized by the use of a continuous latent space. We exemplify the methodology on the dynamics of the Von Karman vortex street of the flow past a cylinder generated by a Large-eddy Simulation (LES)-based code. We compare the NODE methodology with an LSTM baseline to assess the extrapolation capabilities of the generative model and present some qualitative evaluations of the flow reconstructions.

Introduction

Modeling and simulation of dynamical systems are essential tools in the study of complex phenomena with applications in chemistry, biology, physics and engineering, among other relevant fields. These tools are particularly useful in the control and design of parametrized systems in which the dependence on properties, initial conditions and other configurations requires multiple evaluations of the system response. However, there are some limitations when performing numerical simulations of systems where nonlinearities, and a wide range of spatial and time scales leads to unmanageable demands on computational resources. The latter is the case of engineering fluid flow problems where the range of scales involved increase with the value of the Reynolds number and the cost of simulating a full-order model (FOM) using techniques such as DNS or LES is very high. One of the possible solutions to reduce the expensive computational cost is to introduce an alternative, cheaper and faster representation that retains the characteristics provided by the FOM without sacrificing the accuracy of the general physical behaviour.

The idea is to construct a methodology able to generalize the physical behaviour for unseen parameters and that can extrapolate forward in time using the minimal amount of full order simulations (Benner, Gugercin, and Willcox 2015).

The projection-based reduced order modeling is one of the most popular approaches to construct surrogate models of dynamical systems. This framework reduces the degrees of freedom of the numerical simulations using a transformation into a suitable low-dimensional subspace. Then, the state variable in the governing equations is rewritten in terms of the reduced subspace and finally the PDE equations are converted into a system of ODEs that can be solved using classical numerical techniques (Benner, Gugercin, and Willcox 2015). In the field of fluid mechanics, the Proper Orthogonal Decomposition (POD) method is widely applied in the dimensionality reduction of the FOM and the Galerkin method is used for the projection onto the governing equations. These methodologies are preferred because an orthogonal normal basis simplifies the complexity of the projected mathematical operators and the truncated basis of the POD is optimal in the least squares sense, retaining the dominant behaviour through the most energetic modes. The projection on the governing equations maintains the physical structure of the model, but the truncation of the modes can affect the accuracy of the results in nonlinear systems and it may also be restricted to stationary and time periodic problems. Furthermore, the projection is intrusive, requiring different settings for each problem, and it is limited to explicit and closed definitions of the mathematical models (San, Maulik, and Ahmed 2019). Some of these problems have been addressed with the search of closure models that compensates the information losses produced by the truncated modes (Mou et al. 2020; Mohebujjaman, Rebholz, and Iliescu 2019; San and Maulik 2018b,a) and with the construction of a data driven reduced "basis" that also provides optimality after the time evolution (Murata, Fukami, and Fukagata 2020; Liu et al. 2019; Wang et al. 2016).

We present an alternative methodology to evolve the dynamics of the system in the reduced space using a data-driven approach. We use the POD to compute the modes and the temporal coefficients of a fluid flow simulation and then we apply an autoencoder architecture to learn the dynamics of a latent space. The addition of a neural ODE (Chen et al. 2019; Rubanova, Chen, and Duvenaud 2019) block in
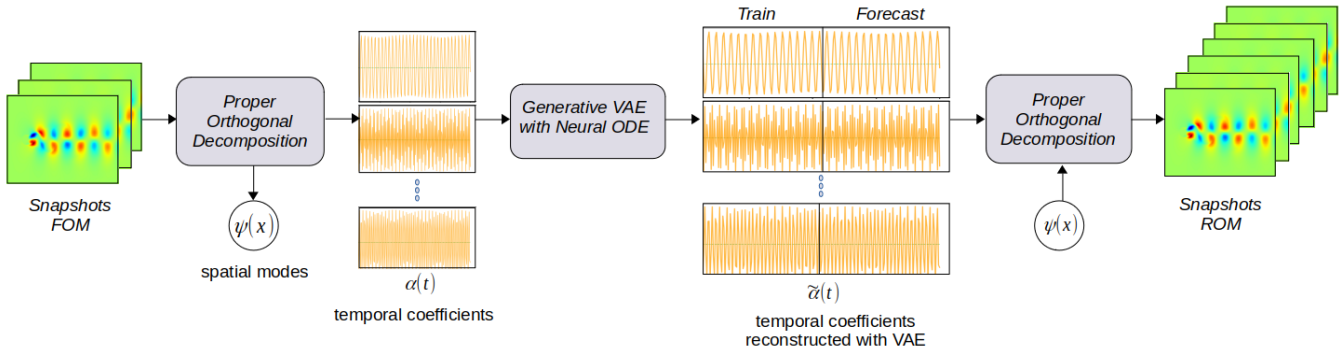
Figure 1: POD-NeuralODE ROM methodology.

the middle of the autoencoder model provides a continuous learning block that is encoded using a feed forward neural network and that can be solved numerically to determine the future states of the input variables. Several works have proposed machine learning models to replace the Galerkin projection step or to improve their capabilities, and different architectures such as feedforward or recurrent networks has been applied with demonstrated good performance in academic and practical fluid flow problems (Pawar et al. 2019; Imtiaz and Akhtar 2020; Eivazi et al. 2020; Lui and Wolf 2019; Portwood et al. 2019; Maulik et al. 2020a,b). The main advantage of the neural ODE generative model is that the learning is posed as a self-supervised task using a continuous representation of the physical behavior. In our view, the neural ODE block can be interpreted as an implicit differential operator that is not restricted to a specific differential equation. This setting provides more flexibility than the projection over the governing equations because it addresses the learning problem with an operator that is informed and corrected by the training data.

Methodology

In this work we use a Large-eddy Simulation (LES) model to approximate the behavior of the fluid flow dynamical system. As it is the case in many fluid flow problems, the discrete solution has a spatial dimension larger than the size of the temporal domain. For this reason, we apply the snapshot POD to construct the reduced order model and have a tractable computation. The POD finds a new basis representation that maximizes the variance in the data, and has the minimum error of the reconstructions in a least squares sense. In addition, the dimensionality reduction is easily performed because the components of the new basis are ordered by their contribution to the recovery of the data.

The main block in the Neural ODE-ROM methodology is concerned with the forecast of the temporal coefficients provided by the snapshot POD. Here we apply the latent ODE (Chen et al. 2019; Rubanova, Chen, and Duvenaud 2019), a generative neural ODE model that takes the temporal coefficients, learns their dynamical evolution and provides an adequate model to extrapolate at the desired time steps. Finally, we can forecast the evolution of the temporal coefficients and reconstruct the behavior of the flow with the

spatio-temporal expansion used in the POD.

The general methodology is represented in the Fig. 1 and more details about each of the building blocks is presented in the following sections.

LES Model For The Flow Past a Cylinder

The dynamics of the Von Karman vortex street of the flow past a cylinder were solved by the LES filtered governing equations for the balance of mass (1), and momentum (2), which can be written as follows:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial x_i} = 0 \tag{1}$$

$$
\begin{aligned}
\frac{\partial(\bar{\rho}\tilde{u}_i)}{\partial t} + \frac{\partial(\bar{\rho}\tilde{u}_i\tilde{u}_j)}{\partial x_j} &= \frac{\partial}{\partial x_j}\left[\bar{\rho}\bar{\nu}\left(\frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j}\right)\right. \\
&\left. - \frac{2}{3}\bar{\rho}\bar{\nu}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} - \bar{\rho}\tau_{ij}{}^{sgs}\right] - \frac{\partial \bar{p}}{\partial x_i} + \bar{\rho}g_i
\end{aligned}
\tag{2}
$$

In the previous equations, u represents the velocity, ρ is the fluid density, and ν is the dynamic viscosity. These equations are solved numerically using the PIMPLE algorithm (Weller et al. 1998), which is a combination of PISO (Pressure Implicit with Splitting of Operator) by Issa (1986) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) by Patankar (1980). This approach obtains the transient solution of the coupled velocity and pressure fields by applying the SIMPLE (steady-state) procedure for every single time step. Once converged, a numerical time integration scheme (e.g. backward) and the PISO procedure are used to advance in time until the simulation is complete. Furthermore, the unresolved subgrid stresses, $\tau_{ij}{}^{sgs}$, are modeled in terms of the subgrid-scale eddy viscosity ν_T using the dynamic k-equation approach by Kim and Menon (1995).

The setup of the problem is described as follows. The computational domain comprehends a 2D channel with 760 mm in the stream-wise direction and 260 mm in the direction perpendicular to the flow. The cylinder is located between the upper and bottom walls of the channel at 115 mm away from the inlet (left wall). A constant radial velocity of 0.6 m/s with random radial/vertical fluctuations in combination with a zero-gradient outflow condition and non-slip walls on the top/bottom/cylinder walls are imposed as

boundary conditions. Furthermore, a laminar dynamic viscosity of $1 \times 10^{-4} m^2/s$ and a cylinder diameter of 40 mm further characterizes the flow with a Reynolds number of 240 ($Re = 0.6 \times 0.04/1 \times 10^{-4} = 240$). The Central differencing scheme (CDS) was used for the discretization of both convective and diffusive terms of the momentum equation, as well as an implicit backward scheme for time integration. A snapshot of the velocity components in both radial and axial directions at time = 100 is shown in the Fig. 2.
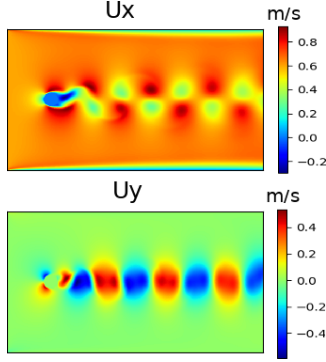

Figure 2: Snapshot of the flow field at t = 100.

Proper Orthogonal Decomposition

The proper orthogonal decomposition (POD) is known under a variety of names such as Karhunen-Loeve expansion, Hotelling transform and principal component analysis (Liang et al. 2002). In addition, one can perform the POD defining a linear autoencoder and setting the loss function metric to the mean squared error. This tool was developed in the field of probability theory to discover interdependencies within vector data and introduced in the fluid mechanics community by Berkooz, Holmes, and Lumley (1993). Once the interdependencies in the data are discovered, it is possible to reduce its dimensionality.

The formulation of the dimensionality reduction starts with some samples of observations provided by experimental results or obtained through the numerical solution of a full order model that characterizes the physical problem. These samples are rearranged in an ensemble matrix of snapshots Y where each row has the state of the dynamical system at a given time step. Then, the correlation matrix of the elements in Y is computed and their eigenvectors are used as an orthogonal optimal new basis for the reduced space.

In the following list we summarize the main steps used for the construction of the snapshot POD:

- Take snapshots : simulate the dynamical system and sample its state u as it evolves.

- Compute the fluctuating components of the velocity u' using the Reynolds decomposition of the flow:

$$u = \overline{u} + u', \qquad (3)$$

where $\overline{u}$ is the temporal mean of the solutions given by the FOM model.

- Assemble the matrix Y with the snapshots in the following form:

$$Y = \begin{bmatrix} u'_x(x_1,y_1,t_1) & ... & u'_y(x_{N_x},y_{N_y},t_1) \\ u'_x(x_1,y_1,t_2) & ... & u'_y(x_{N_x},y_{N_y},t_2) \\ . & . & . \\ . & . & . \\ . & . & . \\ u'_x(x_1,y_1,t_{N_t}) & ... & u'_y(x_{N_x},y_{N_y},t_{N_t}) \end{bmatrix}$$

where each row contains a flattened array with the fluctuating components of the velocity in the x and y directions for a given time step. If the discretization used for the FOM simulation has dimensions N_x, N_y and N_t, then the flattened representation is a vector with length $2 \cdot N_x \cdot N_y$ and the matrix Y has dimensions $N_t \times (2 \cdot N_x \cdot N_y)$.

- Build the correlation matrix K and compute its eigenvectors a_j:

$$K = YY^\top, \qquad (4)$$

$$K_{ij}a_j = \lambda a_i. \qquad (5)$$

Alternatively, one can directly compute the eigenvalues and eigenvectors using the singular value decomposition (SVD) of the snapshot matrix.

- Choose the reduced dimension of the model: As described in the literature, larger eigenvalues are directly related with the dominant characteristics of the dynamical system while small eigenvalues are associated with perturbations of the dynamic behavior. The criterion to select the components for the new basis is to maximize the relative information content $I(N)$ using the minimal amount of components N necessary to achieve a desired percentage of recovery (Schilders et al. 2008).

$$I(N) = \frac{\sum_{i=1}^{N} \lambda_i}{\sum_{i=1}^{N_t} \lambda_i} \qquad (6)$$

- Finally, we compute the spatial modes $\psi_i(x)$ using the temporal coefficients in the reduced dimensional space and the Ansatz decomposition of the POD:

$$u' \approx \sum_{i=1}^{N} \alpha_i(t)\psi_i(x), \qquad (7)$$

$$\psi_i(x) = \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{N} \alpha_i(t_j)u'(t_j). \qquad (8)$$

Neural Ordinary Differential Equations

The neural ordinary differential methodology (Chen et al. 2019) can be interpreted as a continuous counterpart of traditional models such as recurrent or residual neural networks. In order to formulate this model, the authors drew a parallel between the classical composition of a sequence in terms of previous states and the discretization methods used to solve differential equations:
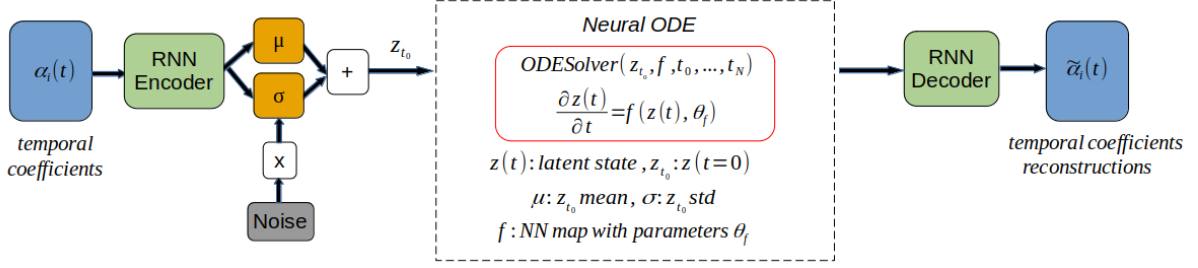
Figure 3: Generative VAE with Neural ODE.

$$h_{t+1} = h_t + f(h_t, \theta). \tag{9}$$

In the limit case of sufficient small steps (equivalent to an increase of the layers) is possible to write a continuous parametrization of the hidden state derivative:

$$\frac{dh(t)}{dt} = f(h_t, \theta), \tag{10}$$

$$h_t = ODESolver(h_0, f(h_t, \theta)). \tag{11}$$

The function f defining the parametrization of the derivative can be approximated using a neural network and the values of hidden states h_t at different time steps are computed using numerical ODE solvers (Chen et al. 2019).

We apply the latent ODE generative approach (Chen et al. 2019) presented in the Fig. 3 to model the evolution of the temporal coefficients provided by the proper orthogonal decomposition. This approach can be interpreted as a variational autoencoder architecture with an additional neural ODE block after the sampling of the codings. This block maps the vector of the initial latent state z_{t_0} to a sequence of latent trajectories using the ODE numerical solver while a neural network $f(z_t, \theta)$ learns the latent dynamics necessary to have a good reconstruction of the input data. The variational part of the autoencoder produces the mean μ and standard deviation σ of the initial latent variable z_{t_0} and adds noise to the sampling process improving the quality of the features learned.

After the training process, the latent trajectories are easily extrapolated with the redefinition of the temporal bounds in the ODE solver. Some of the advantages of this strategy are that it does not need an explicit formulation of the physical laws to forecast the temporal coefficients, and in consequence, the method does not resort onto projection methodologies. Furthermore, the parametrization using a neural network gives an accurate nonlinear approximation of the derivative without a predefined mathematical structure.

Results

In this section, we evaluate the performance of the generative neural ODE model in the forecasting of the temporal coefficients. For this assessment, we apply the proper orthogonal decomposition over 300 snapshots of simulated data ob-

tained with the LES code and take the first 8 POD modes achieving a 99 % of recovery according to the relative information content. For the deployment of the neural ODE model (NODE) we take the first 75 time steps for the training set, the following 25 time steps for the validation of the model and the last 200 time steps for the test set. Furthermore, we employ as a baseline model an LSTM sequence to vector architecture as proposed in Maulik et al. with a window size of 10 time steps and a batch size of 15 sequences.

We tuned the hyperparameters necessary for both models adopting a random search and chose the best configuration given the performance on the validation set. The evolution of the loss for the best model is shown in Fig. 4 and the set of hyperparameters employed are presented in Table 1.
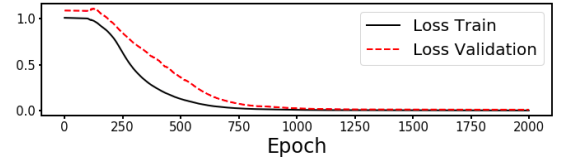

Figure 4: Loss Generative NODE model.

Model	Hyperparameter	Range	Best
	latent dimension	[2,5)	2
	layers encoder	[1,6)	4
	units encoder	[10,50)	10
Neural ODE	layers node	[1,3)	1
	units node	[10,50)	12
	layers decoder	[1,6)	4
	units decoder	[10,50)	41
	learning rate	[0.001, 0.1)	0.0015
	units	[10,60)	49
LSTM	layers	[1,5)	1
	learning rate	[0.001, 0.1)	0.0081

Table 1: Hyperparameters used in the models.

The time-series prediction for the first four temporal coefficients in the test set is shown in Fig. 5. This plot presents the ground truth values of the POD time coefficients, the baseline produced using an LSTM architecture and the predictions by the proposed generative NODE model for the first 100 time steps in the test window.
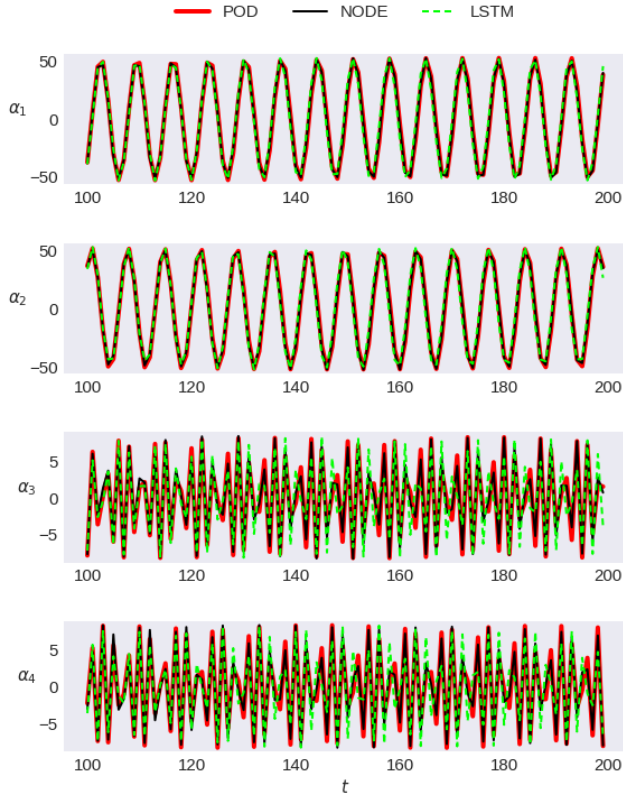
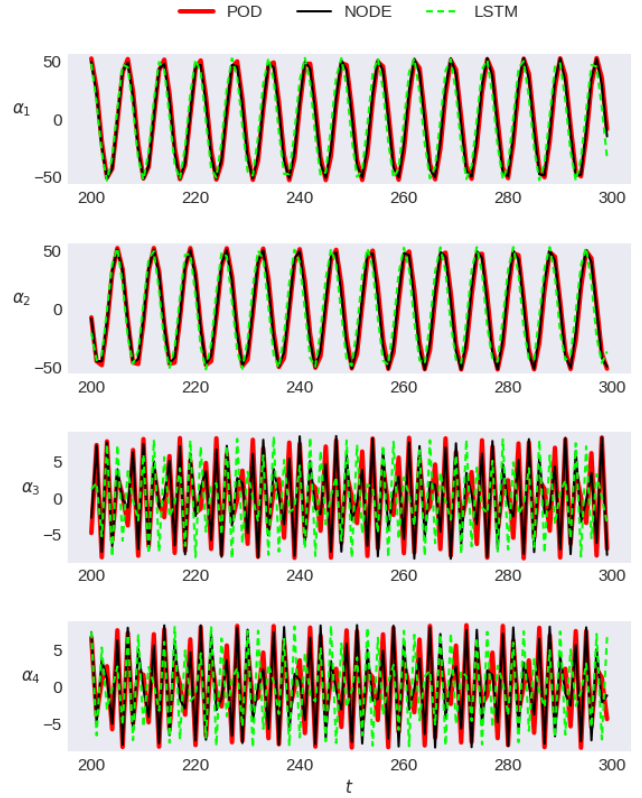Figure 5: Reconstruction of POD temporal coefficients using NODE vs LSTM , $t \in [100, 200]$.

Figure 6: Reconstruction of POD temporal coefficients using NODE vs LSTM, $t \in [200, 300]$.

We notice that the baseline and the NODE model learned adequately the evolution of the two most dominant coefficients, but the performance of the NODE model is significantly better for the third and four time coefficients. Additionally, the quality of the prediction using the LSTM model for the last 100 time steps in the test set deteriorates with the evolution of the time steps even for α_1 and α_2 as seen in Fig. 6 . One of the possible reasons for this is that the autoregressive nature of the predictions in the LSTM model is prone to the accumulation of errors as Maulik et al. pointed out in their study (Maulik et al. 2020b).

After the training and validation process, we reconstruct the velocity fluctuating component u'_x using the Ansatz of the proper orthogonal decomposition with the temporal coefficients forecasted for the test set. Observing the Fig. 7 is possible to notice that the contour generated with the reduced order model provides an adequate recovery of the flow features with only slight differences in some vortexes. In addition, we also present the fluctuation time history for a probe located downstream from the cylinder in Fig. 8. This figure shows with more details how the physical response of the reduced order model gives a satisfactory approximation of the flow behavior.
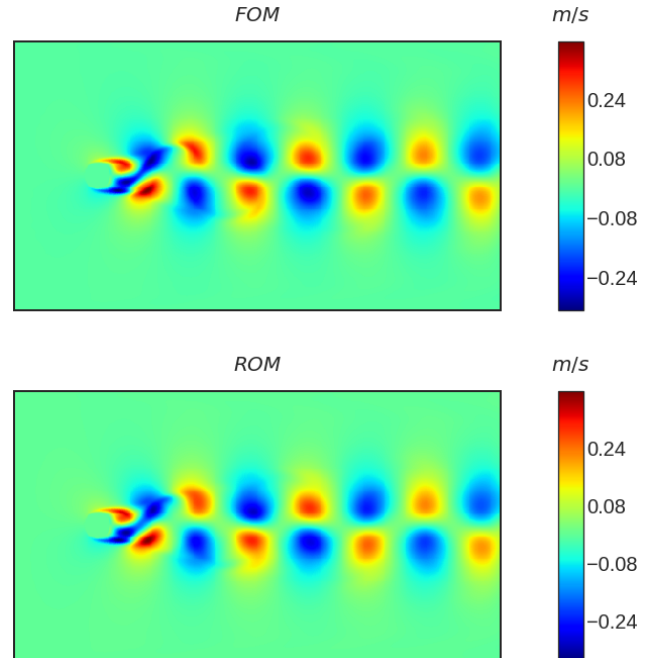

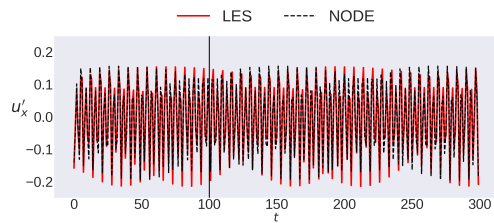Figure 7: Contours of fluctuating component u'_x, $t = 300$.

Figure 8: Probe positioned behind the cylinder.

The data and code that support this study is provided at https://github.com/CarlosJose126/NeuralODE-ROM.

Conclusions

We presented a methodology to produce reduced order models using a neural ODE generative architecture for the evolution of the temporal coefficients. Within this approach, we employ a linear autoencoder (POD) to produce the dimensionality reduction of the model, and after that, we apply a non-linear variational autoencoder to learn the evolution of the temporal coefficients. Although the data is compressed in both autoencoders, the motivations of the dimensional reductions are different. The POD provides an interpretable reduced space used for decomposition and reconstruction of the full order model, while the VAE latent space avoids a trivial copy of the time coefficients.

The neural ODE model was able to learn appropriately the hidden dynamics of the temporal coefficients without having the same propagation of errors common in the autoregressive architectures. Another advantage of this methodology is that learning is posed as a self-supervised task, i.e., the outputs are "self-generated" as equal to the inputs, without splitting the entire sequence into smaller training windows with labels. We also remark that the continuous nature of the neural ODE block is crucial for the good extrapolation capabilities of the methodology. Finally, we expect to test the capabilities of this methodology with other physical problems and also to extend the method for parametric dynamical systems.

References

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

Berkooz, G.; Holmes, P.; and Lumley, J. L. 1993. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annual Review of Fluid Mechanics* 25(1): 539–575.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2019. Neural Ordinary Differential Equations. *arXiv:1806.07366 [cs, stat]* ArXiv: 1806.07366.

Eivazi, H.; Veisi, H.; Naderi, M. H.; and Esfahanian, V. 2020. Deep neural networks for nonlinear model order reduction of unsteady flows. *Physics of Fluids* 32(10): 105104.

Imtiaz, H.; and Akhtar, I. 2020. POD-based Reduced-Order Modeling in Fluid Flows using System Identification Strat-

egy. In *2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 507–512. Islamabad, Pakistan: IEEE.

Issa, R. 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* 62(1): 40 – 65.

Kim, W.-W.; and Menon, S. 1995. *A new dynamic one-equation subgrid-scale model for large eddy simulations*.

Liang, Y.; Lee, H.; Lim, S.; Lin, W.; Lee, K.; and Wu, C. 2002. PROPER ORTHOGONAL DECOMPOSITION AND ITS APPLICATIONS—PART I: THEORY. *Journal of Sound and Vibration* 252(3): 527–544. ISSN 0022460X. doi:10.1006/jsvi.2001.4041. URL https://linkinghub.elsevier.com/retrieve/pii/S0022460X01940416.

Liu, Y.; Wang, Y.; Deng, L.; Wang, F.; Liu, F.; Lu, Y.; and Li, S. 2019. A novel in situ compression method for CFD data based on generative adversarial network. *Journal of Visualization* 22(1): 95–108.

Lui, H. F. S.; and Wolf, W. R. 2019. Construction of reduced-order models for fluid flows using deep feedforward neural networks. *Journal of Fluid Mechanics* 872: 963–994.

Maulik, R.; Fukami, K.; Ramachandra, N.; Fukagata, K.; and Taira, K. 2020a. Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Physical Review Fluids* 5(10): 104401.

Maulik, R.; Mohan, A.; Lusch, B.; Madireddy, S.; Balaprakash, P.; and Livescu, D. 2020b. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena* 405: 132368.

Mohebujjaman, M.; Rebholz, L.; and Iliescu, T. 2019. Physically constrained data-driven correction for reduced-order modeling of fluid flows. *International Journal for Numerical Methods in Fluids* 89(3): 103–122.

Mou, C.; Liu, H.; Wells, D. R.; and Iliescu, T. 2020. Data-driven correction reduced order models for the quasi-geostrophic equations: a numerical investigation. *International Journal of Computational Fluid Dynamics* 34(2): 147–159.

Murata, T.; Fukami, K.; and Fukagata, K. 2020. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics* 882: A13.

Patankar, S. V. 1980. *Numerical heat transfer and fluid flow*. Washington: Hemisphere Publishing Corporation.

Pawar, S.; Rahman, S. M.; Vaddireddy, H.; San, O.; Rasheed, A.; and Vedula, P. 2019. A deep learning enabler for nonintrusive reduced order modeling of fluid flows. *Physics of Fluids* 31(8): 085101.

Portwood, G. D.; Mitra, P. P.; Ribeiro, M. D.; Nguyen, T. M.; Nadiga, B. T.; Saenz, J. A.; Chertkov, M.; Garg, A.; Anandkumar, A.; Dengel, A.; et al. 2019. Turbulence forecasting via Neural ODE. *arXiv preprint arXiv:1911.05180* .

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. 2019. Latent ODEs for Irregularly-Sampled Time Series. *arXiv preprint arXiv:1907.03907* .

San, O.; and Maulik, R. 2018a. Machine learning closures for model order reduction of thermal fluids. *Applied Mathematical Modelling* 60: 681–710.

San, O.; and Maulik, R. 2018b. Neural network closures for nonlinear model order reduction. *Advances in Computational Mathematics* 44(6): 1717–1750.

San, O.; Maulik, R.; and Ahmed, M. 2019. An artificial neural network framework for reduced order modeling of transient flows. *Communications in Nonlinear Science and Numerical Simulation* 77: 271–287.

Schilders, W. H. A.; van der Vorst, H. A.; Rommes, J.; Bock, H.-G.; de Hoog, F.; Friedman, A.; Gupta, A.; Neunzert, H.; Pulleyblank, W. R.; Rusten, T.; Santosa, F.; Tornberg, A.-K.; Bonilla, L. L.; Mattheij, R.; and Scherzer, O., eds. 2008. *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Wang, M.; Li, H.-X.; Chen, X.; and Chen, Y. 2016. Deep Learning-Based Model Reduction for Distributed Parameter Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46(12): 1664–1674.

Weller, H. G.; Tabor, G.; Jasak, H.; and Fureby, C. 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics* 12(6): 620–631.