The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

EVALUATING FORECAST DIVERGENCE IN FLUID DYNAMICS: A METRIC FOR AUTOREGRESSIVE MODELS IN THE ABSENCE OF TRUE SOLUTION DATA

Rodrigo Abadía-Heredia[1], Manuel Lopez-Martin[1] AND Soledad Le Clainche[1]

[1] ETSI Aeronáutica y del Espacio, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros, 3, Madrid, 28040, Madrid, Spain
e-mail: sr.abadia@upm.es, manuel.lmartin@repsol.com, soledad.leclainche@upm.es

Key words: Turbulence, Reduced order model (ROM), Forecasting Error

Summary. In this work, we introduce a novel metric designed to quantify the error committed by predictions from an autoregressive forecasting model, in the absence of knowledge of the actual solution. This metric provides a means to determine the appropriate point at which to cease predictions, ensuring that the forecast does not diverge significantly from the true dynamics of the physical system under consideration. The proposed metric was tested on the challenging task of forecasting the dynamics of a turbulent flow at a Reynolds number of $Re = 2600$, where it demonstrated a high degree of similarity with the actual prediction errors, effectively capturing the divergence between the predicted and true flow dynamics.

1 INTRODUCTION

Numerical simulation plays a critical role in both industrial and academic settings for the prediction and resolution of complex fluid flow phenomena in fluid mechanics. High-fidelity numerical simulations are often necessary for studying intricate fluid dynamics. This can be done by a direct numerical simulation (DNS), which is computationally expensive and time-consuming. This creates a compelling need for techniques that can effectively reduce computational costs while maintaining, up to some point, the accuracy and reliability offer by a DNS.

In this meaning, reduced order models (ROMs) have been proposed as a viable alternative to DNS, offering a way to decrease computational costs while retaining sufficient accuracy. ROMs are generally categorized into two types: intrusive and non-intrusive ROMs.

Intrusive ROMs involve modifying the governing equations of the physical system under study. A common example of this approach is the Galerkin projection method, where the governing equations are projected onto a reduced basis, thus simplifying the computational effort required to solve them [1].

In contrast, non-intrusive ROMs, also known as surrogate models, do not require alterations to the governing equations. Instead, they rely solely on data to predict the system dynamics. Within this framework, machine learning (ML) models are frequently employed due to their capacity to learn complex patterns and relationships directly from the data.

Galerkin methods have been widely utilized to simulate flow dynamics by projecting the governing equations onto an orthonormal basis formed by proper orthogonal decomposition (POD) modes. In this approach, the primary objective shifts to simulating the mode amplitudes, thereby reducing the complexity of the original problem. While this technique has proven

effective in various applications, it presents some challenges. One significant issue with Galerkin methods is that they can produce unstable iteration schemes [2]. Also, in some problems the dynamics cannot be adequately represented by a few dominant POD modes. This necessitates the inclusion of a larger number of modes to accurately capture the essential structures embedded within the underlying dynamics [3]. However, increasing the number of modes directly raises the computational cost, potentially making the ROM inefficient.

In this context, data-driven forecasting models have emerged as a promising alternative for predicting mode amplitudes without the need to project the governing equations. This approach combines data-driven forecasting models with POD, offering a hybrid methodology that has been extensively explored for predicting the evolution of dynamical systems [4, 5, 6, 7, 8, 9, 10].

Deep learning models are often chosen for this task due to their ability to handle nonlinearities, their relatively low computational cost during the training phase, and their proven effectiveness in time-series forecasting. These models can capture complex temporal dependencies and interactions within the data, making them particularly well-suited for predicting the evolution of fluid dynamics and other complex physical systems. However, a significant challenge with this hybrid methodology lies in determining the appropriate point to stop predictions to prevent them from diverging from the actual dynamics of the system. In a previous work, Kičić et al. [11] introduced a metric specifically designed to measure this divergence within a forecasting framework that relied exclusively on deep learning models, without incorporating POD. Similar works have been carried out for Galerkin methods [12, 13].

Building on this concept, we propose a similar metric tailored to take advantage of the combined use of POD and deep learning forecasting models. This metric is designed to monitor the divergence of the predictions from the actual system dynamics effectively, thereby providing a criterion for halting the predictions when necessary.

This paper is organized as follows. Section 2 describes the hybrid ROM used in this paper, where section 2.1 describes the singular value decomposition method that is used to compute POD modes, and section 2.2 describes the forecasting model. Then, section 3 presents the proposed metric to measure the predictions divergence from the actual dynamics. Section 4 shows the results obtained when the whole framework, which combines the hybrid ROM and the metric, is tasked to predict the dynamics of a turbulent flow. Finally section 5 gives the conclusions obtained from this work.

2 HYBRID ROM

A physical system is usually represented by a system of nonlinear partial differential equations (PDEs), which in some special cases an analytical solution is easy to find, but in most situations is required a numerical approach to this solution.

Let's assume $\mathcal{P}_x \subset \mathbb{R}^3$ (or $\mathbb{R}^2$ for two-dimensional cases) and $\mathcal{P}_t \subset \mathbb{R}$ are compact sets representing the spatial and temporal dimensions, where to solve the system of PDEs, so $\mathcal{P} = (\mathcal{P}_x, \mathcal{P}_t)$. Consider a Hilbert space $\widetilde{W} = \widetilde{W}(\mathcal{P})$ defined on over the domain $\mathcal{P}$, which represents the space that contains all the possible solutions to the system of PDEs on the domain $\mathcal{P}$, i.e., $\widetilde{W}$ is composed by functions $\tilde{w}$ defined as $\tilde{w} : \mathcal{P}_x \times \mathcal{P}_t \to \mathbb{R}$. Furthermore, the system of nonlinear PDEs can be denoted as a mapping $\Phi : \widetilde{W} \times \mathcal{P} \to \mathbb{R}$, where the problem of interest reads: given $p = (x, t) \in \mathcal{P} = (\mathcal{P}_x, \mathcal{P}_t)$, find a $\tilde{w}(p) \in \widetilde{W}(\mathcal{P})$ such that,

$$\Phi(\tilde{w}(p), p) = 0. \tag{1}$$

In practice, it is of interest to find a function $\tilde{w} \in \widetilde{W}$ that makes (1) as close as possible to zero, and not exactly zero. Furthermore, to approximate the solution a discretization, in either time and space, of the governing equations (1) is required.

This increase the degrees-of-freedom to the order of thousand or millions, scaling the computational cost to the point where the resolution of the equations requires specialize hardware and/or a long time of computation. Being in some scenarios intractable to solve all spatio-temporal scales within the dynamics. In this meaning reduced order models (ROMs) allow to approximate the solution only for the scales of interest.

This paper makes use of a hybrid ROM, already proposed in the literature [4, 5, 6, 7, 8, 9, 10], that combines proper orthogonal decomposition (POD) and deep learning (DL) forecasting models. The methodology followed by this hybrid ROM is explained in the subsequent section.

2.1 Proper orthogonal decomposition

Proper orthogonal decomposition is the first method used by the hybrid ROM, which allows to project the solution of the governing equations onto an orthonormal basis, as follows,

$$\tilde{w}(x,t) = \sum_{i=1}^{K} A_i(t)\mathbf{U}_i(x). \tag{2}$$

The set $\{\mathbf{U}_1, \ldots, \mathbf{U}_K\}$ represents the orthonormal basis or POD modes, while A_i are the mode amplitudes, which individually represents the energy contribution of the POD mode $\mathbf{U}_i$ to the overall dynamics [14]. That is, these amplitudes allows to classify the POD modes based on the information they carry about the dynamics.

This technique has been extensively used to study the structural composition of flow dynamics, in order to find the coherent structures that rely within it [15]. The first one to apply POD to a turbulent flow was Lumley [16] in 1967.

In this context, it is necessary to compute the POD modes $U = [\mathbf{U}_1, \ldots, \mathbf{U}_N]$ and the mode amplitudes $A(t) = A^t = [A_1(t), \ldots, A_N(t)]^\top$, where the latter will be forcasted by the deep learning model. In this work, this is done using singular value decomposition (SVD), also known as the method of snapshots, developed by Sirovich [17] in 1987.

To apply SVD it is required to have a dataset $\mathbf{D}$ composed by a series of snapshots representing the flow dynamics, e.g., the velocity flow field, pressure field, etc. The resolution of these snapshots are directly related to the spatial grid used to discretize the spatial domain.

Let's assume the spatial domain is discretized in $N = \#\mathcal{P}_x$ points $x_j \in \mathbb{R}^3$ (or $\mathbb{R}^2$) and T_1 represents the number of snapshots available, where T_1 is less than the temporal horizon $T = \#\mathcal{P}_t$ where we want solve the problem (1). Then, the dataset $\mathbf{D}$ is a snapshot tensor defined as follows,

$$\mathbf{D} = [\tilde{w}_1, \ldots, \tilde{w}_{T_1}], \tag{3}$$

$$dim(\mathbf{D}) = [C \times N_x \times N_y \times N_z \times T_1]. \tag{4}$$

Where $\tilde{w}_i = \tilde{w}(\cdot, i)$, *dim* stands for dimension of $\mathbf{D}$, C represents the number of components, e.g., the velocity components (streamwise, wall-normal and spanwise), different Reynolds numbers, etc. The variables N_x, N_y and N_z represent the number of points in the x-, y- and z-axis discretization, respectively.

Note, this dataset is formed by solutions of the system of PDEs given by (1), where typically $T_1 \ll T$. Consequently, to produce the required snapshots for the application of SVD, it is necessary to solve the system (1) over a time horizon T_1 that is as small as possible, ideally ensuring that $T_1 \ll T$. Once $\mathbf{D}$ is obtained, SVD can be performed to yield a matrix decomposition as follows,

$$\mathbf{D} = U\Sigma V^*. \tag{5}$$

Where U is a unitary matrix whose columns represent the POD modes in (2), Σ is a diagonal matrix containing the singular values, and V is a unitary matrix whose columns represents the POD coefficients. The matrix V^* is the conjugate transpose of V. Because Σ is diagonal, it is possible to express (5) as a sum of rank one matrices,

$$\mathbf{D} = \sum_{i=1}^{K} \sigma_i u_i v_i^*. \tag{6}$$

Where σ_i corresponds to the i-th singular value, u_i represents the i-th column in U, i.e., the i-th POD mode, and v_i corresponds to the i-th column of V, i.e., the i-th POD coefficient.

The singular values are hierarchically sorted based on their energy contribution, i.e., $\sigma_i \geq \sigma_j$ for $i > j$. This means that the i-th POD mode carries information about larger-scale structures inside the dynamics than the j-th POD mode. This facilitates the truncation of POD modes in order to keep only the dynamics of interest and not all the spatio-temporal scales.

Note that, from (6) it is clear that the mode amplitudes of (2), for the dataset $\mathbf{D}$, can be expressed as a product of the singular values and the POD coefficients,

$$A_i(t) = \sigma_i v_i^*(t). \tag{7}$$

Once the POD modes has been computed using SVD, the temporal evolution of the corresponding amplitudes, $A(t)$, can be predicted through forecasting models. Note from 2 that, by predicting future states of these amplitudes, the method is predicting the temporal evolution of the POD modes, which represent the structures of interest within the dynamics. The following section gives a description of the forecasting model used in this work.

2.2 Deep learning forecasting model

Since SVD provides historical data on the mode amplitudes, as expressed in equation (7), it is possible to forecast their future evolution by leveraging past information. In contrast to Galerkin methods, where a projection of the governing equations onto the POD modes is required.

As demonstrated in Section 4, this methodology can be effectively applied to forecast datasets derived from experimental data, where the governing equations may be challenging to obtain.

This work employs a deep learning (DL) model to forecast the mode amplitudes. Its architecture must be capable of identifying the temporal correlations within a past sequence

$\{A^{t-m+1}, \ldots, A^t\}$ and extrapolating this information to predict the future state of the amplitudes, $\{A^{t+1}\}$.

In this aim, a Long Short-Term Memory (LSTM) network [18] is employed. This deep learning architecture is characterized by its use of three gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information, of the input sequences, through the network by selectively adding new information (input gate), removing irrelevant information (forget gate), or allowing information to pass through to the next cell (output gate).

In this work, the input gate is represented by i, the forget gate by f and the output gate by o. As well, the cell input is represented by c_i, the cell output is given by c_o and the cell state is denoted by c. The procedure to compute the gates and states are as follows,

$$f_j = \hat{\sigma}(W_f[h_{j-1}, x_j] + b_f) \tag{8}$$
$$i_j = \hat{\sigma}(W_i[h_{j-1}, x_j] + b_i) \tag{9}$$
$$\tilde{c}_t = \tanh(W_c[h_{j-1}, x_j] + b_c) \tag{10}$$
$$c_j = f_j \cdot c_{j-1} + i_j \cdot \tilde{c}_j \tag{11}$$
$$o_j = \hat{\sigma}(W_o[h_{j-1}, x_j] + b_o) \tag{12}$$
$$h_j = o_j \cdot \tanh(c_j) \tag{13}$$

Where W and b are the weights and biases for each gate, respectively, and $\tilde{c}$ is the updated cell state. The activation function is represented by $\hat{\sigma}$, which is typically a Sigmoid.

To enhance the model's flexibility, after applying the LSTM and obtaining the final state, representing the time-ahead prediction, two multi-layer perceptron (MLP) layers are subsequently applied. This process yields the final forecast of the mode amplitudes, $\{A^{t+1}\}$.

The training of this model is done using the T_1 available amplitudes $\{A^t\}_{t=1}^{T_1}$, which were obtain by applying SVD to the dataset $\mathbf{D}$ as in (5). After training, predictions of the future amplitudes $\{A^t\}_{t=T_1+1}^{T}$ are performed following an iterative procedure where predictions are taken as new inputs for the model, yielding to an autoregressive model. Note, the solutions $\tilde{w}(x,t)$ are then constructed using (5).

The question then arises regarding the optimal duration for predicting these amplitudes before predictions begin to diverge from the actual dynamics. To address this, we propose a novel metric in the following section to quantify the divergence in the predictions. This metric represents the key innovation of our work.

3 METRIC TO MEASURE DIVERGENCE OF PREDICTIONS

As stated above, the problem of interest is to approximate the solution of the system of equations (1) in a time horizon $t \in [0, T]$. In this work, we have chosen a hybrid ROM that forecasts the mode amplitudes A^t and recover the solution using the POD modes, in accordance with equation (2).

To train the forecasting model, a subset of solutions belonging to the horizon $t \in [0, T_1]$, where $T_1 \ll T$, is utilized. After the training phase, the mode amplitudes A^t are then predicted for the extended time range $t \in [T_1 + 1, T]$.

Given the autoregressive nature of the forecasting model, each prediction inherently accumulates errors propagated from preceding predictions. Therefore, it is crucial to establish a means to quantify these incremental prediction errors.

In this study, we propose a metric that measures the divergence between predictions from multiple models, each trained on datasets with different truncation levels in the number of modes.

Singular value decomposition allows to decompose the dataset $\mathbf{D}$ on POD modes u_i, singular values σ_i and POD coefficients v_i^* as shown in (5). Since, singular values are hierarchically sorted based on their energy contribution, it gives a way to truncate the number of modes. In this aiming, four training datasets are defined using different truncation levels in the number of modes,

$$\widetilde{\mathbf{A}}_{k_1} = [A_1, \ldots, A_{K_1}] \in \mathbb{R}^{k_1 \times T_1}, \tag{14}$$

$$\widetilde{\mathbf{A}}_{k_2} = [A_1, \ldots, A_{K_2}] \in \mathbb{R}^{k_2 \times T_1}, \tag{15}$$

$$\widetilde{\mathbf{A}}_{k_3} = [A_1, \ldots, A_{K_3}] \in \mathbb{R}^{k_3 \times T_1}, \tag{16}$$

$$\widetilde{\mathbf{A}}_{k_4} = [A_1, \ldots, A_{K_4}] \in \mathbb{R}^{k_4 \times T_1}. \tag{17}$$

Here k_1, k_2, k_3, k_4 denote the number of modes retained, with $k_1 > k_2 > k_3 > k_4$, where k_1 corresponds to the number of modes required to fully capture all scales of interest. Correspondingly, four forecasting models F_1, F_2, F_3 and F_4 are trained using these datasets.

The rationale behind this strategy is that a lower number of modes simplifies the dynamics by capturing larger spatio-temporal scales. Consequently, training model F_1, which utilizes the largest number of modes, is expected to be more challenging and potentially less stable. However, since k_1 represents the optimal number of modes required to preserve the dynamics of interest, it is crucial to compare the predictions from model F_1 with those from the other models, which are easier to train due to their reduced number of modes.

Although their predictions will vary because of the differing number of modes retained, the overall dynamics should remain consistent. If the predictions begin to diverge, this indicates that the models are forecasting different dynamics, serving as a reliable indicator that the model's predictions can no longer be trusted.

Therefore, defining $T_2 = T - T_1$ and $\mathbf{P_1} \in \mathbb{R}^{k_1 \times T_2}$, $\mathbf{P_2} \in \mathbb{R}^{k_2 \times T_2}$, $\mathbf{P_3} \in \mathbb{R}^{k_3 \times T_2}$ and $\mathbf{P_4} \in \mathbb{R}^{k_4 \times T_2}$ as the sets containing the predicted mode amplitudes $\{A_t\}_{t=T_1+1}^{T}$ from models F_1, F_2, F_3 and F_4, respectively. The procedure starts by computing the standard deviation of the first k_4 mode amplitudes for each $t \in [T_1 + 1, T]$,

$$SD = std(\{\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, \mathbf{P_4}\}) \in \mathbb{R}^{k_4 \times T_2}. \tag{18}$$

Where *std* represents the standard deviation operator. Now compute the euclidean norm of SD for the mode amplitudes, $\widetilde{SD} = \|SD\|_2 \in \mathbb{R}^{T_2 \times 1}$. A smaller $\widetilde{SD}$ indicates that the mode amplitudes from different predictions are closer to each other, implying that the predictions are dynamically similar.

Finally normalize $\widetilde{SD}$ over the euclidean norm of the first k_4 mode amplitudes of $\mathbf{P_1}$,

$$MT = \frac{\widetilde{SD}}{\|\mathbf{P_1}\|_2} \in \mathbb{R}^{T_2 \times 1}. \tag{19}$$

Therefore, the prediction should stop when this metric is greater than or equal to some tolerance ϵ, i.e., $MT \geq \epsilon$, where ϵ is a tune-able parameter. As demonstrated in section 4,

this metric, MT, provides a quantitative measure of the model's divergence when it is used to iteratively predict future states of the mode amplitudes.

Regarding the technical details, it is worth to mention that the code for this study was developed using the Python programming language. Singular value decomposition was implemented utilizing the PyTorch library, which was also employed to develop the deep learning forecasting model.

More details can be found in ModelFLOWs-app [19]. The code was run in a computer using a CPU Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz with 16 cores and 64 Gb of RAM. Training each model took around 18.023 seconds for 200 epochs, and the prediction of the subsequent 200 mode amplitudes was around 0.32 seconds.

4 RESULTS

To demonstrate the effectiveness of this method in quantifying the divergence of predictions from the actual dynamics, we applied the hybrid ROM to a dataset characterizing the velocity flow field of a turbulent flow at high Reynolds numbers [20].

This dataset is public and it comprises two-dimensional velocity fields acquired from an experimental study of the three-dimensional wake flow past a circular cylinder with a diameter of $D = 5$ mm. The experiments were conducted in the L10 low-speed wind tunnel at the von Karman Institute, utilizing time-resolved particle image velocimetry (TR-PIV) techniques. The complete dataset spans approximately 4.5 seconds (13200 snapshots) of flow evolution, during which the free-stream velocity transitions between two steady-state conditions. The transitional phase, which captures the change in flow dynamics, lasts for roughly 1 seconds.

In this work, the training and testing set are extracted from the second steady-state condition, characterized by a Reynolds number (calculated with the cylinder diameter) of Re = 2600. The first $T_1 = 200$ snapshots were designated for training, and the subsequent $T_2 = 200$ for testing. In this case the dataset $\mathbf{D}$ has the following dimension,

$$dim(\mathbf{D}) = [C \times N_x \times N_y \times N_z \times T_1] = [2 \times 111 \times 301 \times 1 \times 200]. \tag{20}$$

Note that $N_z = 1$ because the snapshots are two-dimensional, and $C = 2$, reflecting the presence of both streamwise and wall-normal velocity components in the dataset. Singular value decomposition (SVD) is then applied to the dataset $\mathbf{D}$ to extract the POD modes and their corresponding mode amplitudes. The cumulative sum of the singular values, normalize over the maximum value, shows the percentage of energy, $E(k)$, contained for different truncation in the number of modes. This is shown in Fig. 1 for the dataset $\mathbf{D}$ used in this work.

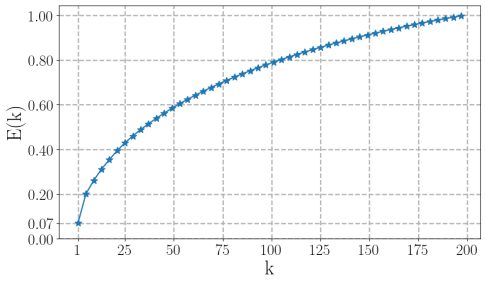


Figure 1: Percentage of energy $E(k)$ contained at the first k POD modes, for different values of k.

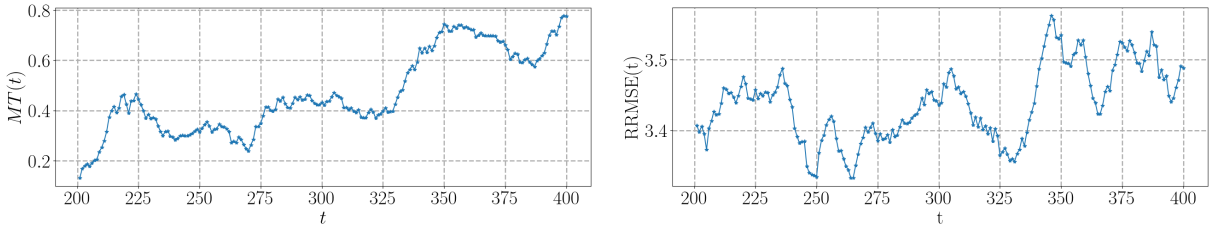
In analyzing complex flows, particularly those within a turbulent regime, it is common practice to truncate the number of modes such that they retain approximately 30% - 40% [21] of the overall energy. Therefore, in this case we selected $k_1 = 30$ ($E(k_1) \sim 40\%$), which represents the truncation with the largest number of modes. This choice was made to ensure that all significant flow scales are adequately captured. The subsequent truncations were chosen so they gradually decrease to $E(k_4) \sim 30\%$, as follows, $k_2 = 25$, $k_3 = 22$, $k_4 = 17$.

From these truncations, we construct the four distinct training sets: $\widetilde{\mathbf{A}}_{k_1}$, $\widetilde{\mathbf{A}}_{k_2}$, $\widetilde{\mathbf{A}}_{k_3}$ and $\widetilde{\mathbf{A}}_{k_4}$. Correspondingly, the four forecasting models trained. These models are then employed to iteratively predict T_2 time-ahead mode amplitudes. Subsequently, the MT metric is computed following the formulation provided in equation (19).

To demonstrate the effectiveness of this metric as an indicator of the error committed by the model, Figure 2 presents a comparison between the divergence measure from MT and the relative root mean squared error (RRMSE). The RRMSE is calculated by comparing the predictions from F_1, which is the model with the largest number of modes, with the actual solution $w(x,t)$ obtained from the dataset.

To show how this metric is a good indicative of the error committed by the model. Fig. 2 compares the divergence measure from MT and the relative root mean squared error (RRMSE) between predictions from F_1, predictions with the largest number of modes, and the actual solution $w(x,t)$ obtained from the dataset.

The results demonstrate that the divergence measure provided by MT is closely aligned with the actual prediction error, as indicated by the RRMSE. Notably, the increase in error observed around $t = 330$ in the RRMSE, as shown in Fig. 2 (b), is also reflected in the MT, as depicted in Fig. 2 (a). Furthermore, the peaks observed around $t = 225$ and $t = 300$ in the RRMSE are similarly captured by the MT. These observations underscore the reliability of the proposed metric for assessing model accuracy.



(a) Model divergence.

(b) Relative root mean squared error.

Figure 2: Comparison of the metric proposed to measure model divergence, MT (19), (a) and the actual prediction error measured with relative root mean squared error (b).

We determined that a tolerance value of $\epsilon = 0.5$ is effective for deciding when to terminate predictions. This threshold is based on the observation that when $MT(t) \geq \epsilon$ at some time t, the predicted dynamics begin to diverge from the actual system dynamics. In this case, this occurs at $t = 334$, as demonstrated in Fig. 4 bottom row. This figure illustrates the point at which the discrepancy between the forecasted and true dynamics becomes substantial, indicating that predictions beyond this tolerance level are less reliable.

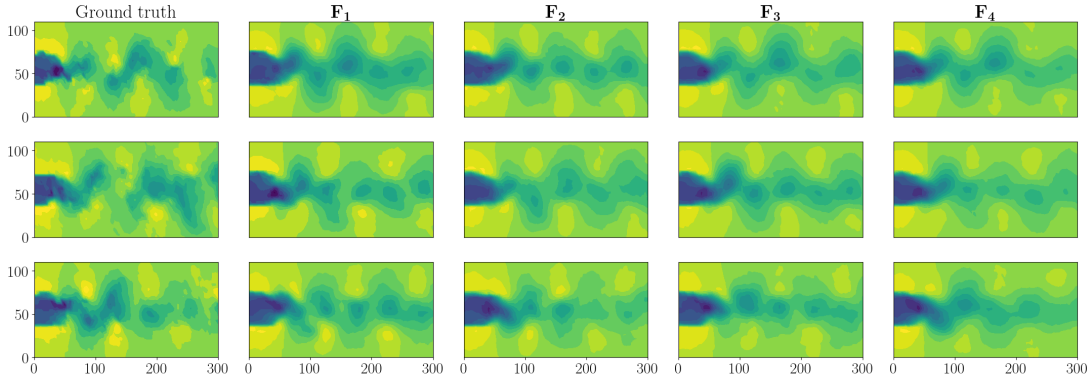


Figure 3: Snapshot comparison for the streamwise velocity at different time instants, from top to bottom, $t = 250$, $t = 300$ and $t = 334$. From left to right, snapshots coming from ground truth dataset, and predictions from models F_1, F_2, F_3 and F_4, respectively.

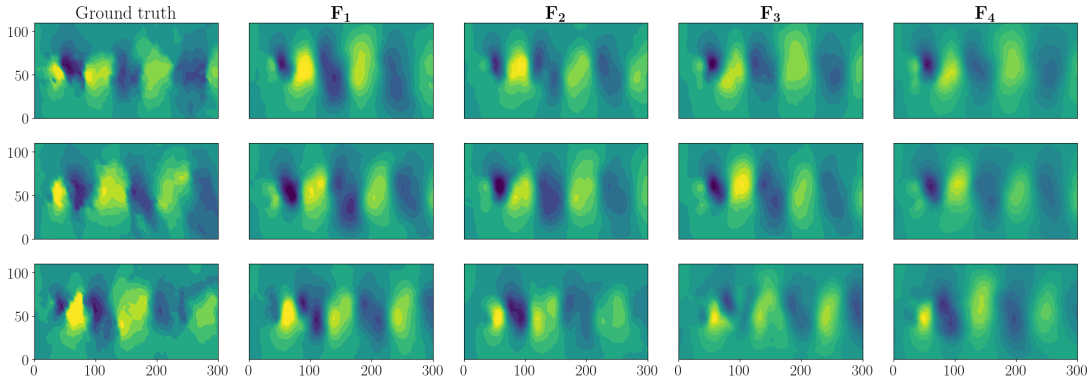


Figure 4: Same as Fig. 3 for the wall-normal velocity.

Figs. 3 and 4 show a comparison of snapshots between the ground truth and the predictions generated by all evaluated models at selected representative time instants.

5 CONCLUSIONS

The proposed work introduces a novel metric designed to measure the divergence of predictions, generated by autoregressive forecasting models, from the actual dynamics of a physical system. This metric facilitates the determination of an appropriate point at which to cease predictions, particularly in scenarios where the actual solution remains unknown. The methodology is rooted in a hybrid reduced order model (ROM) that integrates singular value decomposition with deep learning forecasting models. Analogous to Galerkin methods, this approach targets the prediction of amplitude modes.

The effectiveness of the proposed metric has been validated in a complex case study involving the prediction of flow dynamics in an experimental turbulent flow at a high Reynolds number. In this context, the hybrid ROM not only successfully yielded accurate predictions of the flow dynamics but also demonstrated that the divergence of these predictions, as quantified by the proposed metric, mirrored the actual prediction error trends. These findings suggest that the proposed metric has the potential to serve as a foundation for the development of adaptive (online-offline) models, which could seamlessly integrate classical numerical solvers with data-driven forecasting models.

6 ACKNOWLEDGEMENTS

The authors acknowledge the grants PID2020-114173RB-I00, TED2021- 129774B-C21 and PLEC2022-009235 funded by MCIN/AEI/ 10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR, and S.L.C. acknowledges the support of Comunidad de Madrid through the call Research Grants for Young Investigators from Universidad Politécnica de Madrid. The MODELAIR and ENCODING project has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 101072559 and 101072779, respectively. The results of this publication reflect only the author(s) view and do not necessarily reflect those of the European Union. The European Union can not be held responsible for them.

REFERENCES

[1] D. Rempfer, On low-dimensional galerkin models for fluid flow, Theoretical and Computational Fluid Dynamics 14 (2) (2000) 75–88. doi:10.1007/s001620050131.
URL https://doi.org/10.1007/s001620050131

[2] K. Carlberg, M. Barone, H. Antil, Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction, Journal of Computational Physics 330 (2017) 693–734. doi:https://doi.org/10.1016/j.jcp.2016.10.033.
URL https://www.sciencedirect.com/science/article/pii/S0021999116305319

[3] J. Reiss, P. Schulze, J. Sesterhenn, V. Mehrmann, The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena, SIAM Journal on Scientific Computing 40 (3) (2018) A1322–A1344. arXiv:https://doi.org/10.1137/17M1140571, doi:10.1137/17M1140571.
URL https://doi.org/10.1137/17M1140571

[4] A. T. Mohan, D. V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks (2018). arXiv:1804.09269.
URL https://arxiv.org/abs/1804.09269

[5] S. Pawar, S. M. Rahman, H. Vaddireddy, O. San, A. Rasheed, P. Vedula, A deep learning enabler for nonintrusive reduced order modeling of fluid flows, Physics of Fluids 31 (8) (2019) 085101. arXiv:https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/1.5113494/19764542/085101_1_online.pdf, doi:10.1063/1.5113494.
URL https://doi.org/10.1063/1.5113494

[6] F. Regazzoni, L. Dedè, A. Quarteroni, Machine learning for fast and reliable solution of time-dependent differential equations, Journal of Computational Physics 397 (2019) 108852. doi:https://doi.org/10.1016/j.jcp.2019.07.050.
URL https://www.sciencedirect.com/science/article/pii/S0021999119305364

[7] E. J. Parish, K. T. Carlberg, Time-series machine-learning error models for approximate solutions to parameterized dynamical systems, Computer Methods in Applied Mechanics and Engineering 365 (2020) 112990. doi:https://doi.org/10.1016/j.cma.2020.112990.
URL https://www.sciencedirect.com/science/article/pii/S0045782520301742

[8] R. Abadía-Heredia, M. López-Martín, B. Carro, J. Arribas, J. Pérez, S. Le Clainche, A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures, Expert Systems with Applications 187 (2022) 115910. doi:https://doi.org/10.1016/j.eswa.2021.115910.
URL https://www.sciencedirect.com/science/article/pii/S0957417421012653

[9] L. Xu, G. Zhou, F. Zhao, Z. Guo, K. Zhang, A data-driven reduced order modeling for fluid flow analysis based on series forecasting intelligent algorithm, IEEE Access 10 (2022) 60163–60176. doi:10.1109/ACCESS.2022.3177223.

[10] A. Corrochano, R. S. M. Freitas, A. Parente, S. L. Clainche, A predictive physics-aware hybrid reduced order model for reacting flows (2023). arXiv:2301.09860.

[11] I. Kičić, P. R. Vlachas, G. Arampatzis, M. Chatzimanolakis, L. Guibas, P. Koumoutsakos, Adaptive learning of effective dynamics for online modeling of complex systems, Computer Methods in Applied Mechanics and Engineering 415 (2023) 116204. doi:https://doi.org/10.1016/j.cma.2023.116204.
URL https://www.sciencedirect.com/science/article/pii/S0045782523003286

[12] F. Terragni, E. Valero, J. M. Vega, Local pod plus galerkin projection in the unsteady lid-driven cavity problem, SIAM Journal on Scientific Computing 33 (6) (2011) 3538–3561. arXiv:https://doi.org/10.1137/100816006, doi:10.1137/100816006.
URL https://doi.org/10.1137/100816006

[13] M.-L. Rapún, F. Terragni, J. M. Vega, Adaptive pod-based low-dimensional modeling supported by residual estimates, International Journal for Numerical Methods in Engineering 104 (9) (2015) 844–868. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4947, doi:https://doi.org/10.1002/nme.4947.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4947

[14] N. Aubry, On the hidden beauty of the proper orthogonal decomposition, Theoret. Comput. Fluid Dynamics 2 (1991) 339–352. doi:10.1007/BF00271473.

[15] P. Holmes, J. L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge Monographs on Mechanics, Cambridge University Press, 1996.

[16] J. L. Lumley, The structure of inhomogeneous turbulent flows, in: A. M. Yaglom, V. I. Tartarsky (Eds.), Atmospheric Turbulence and Radio Wave Propagation, Nauka, Moscow, 1967, pp. 166–178.

[17] L. Sirovich, Turbulence and the dynamics of coherent structures: I, ii, iii, Quart. Appl. Math. 45 (1987) 561–590.

[18] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
URL https://doi.org/10.1162/neco.1997.9.8.1735

[19] A. Hetherington, A. Corrochano, R. Abadía-Heredia, E. Lazpita, E. Muñoz, P. Díaz, E. Maiora, M. López-Martín, S. Le Clainche, Modelflows-app: Data-driven post-processing and reduced order modelling tools, Computer Physics Communications 301 (2024) 109217. doi:https://doi.org/10.1016/j.cpc.2024.109217.
URL https://www.sciencedirect.com/science/article/pii/S0010465524001401

[20] M. A. Mendez, D. Hess, B. B. Watz, J.-M. Buchlin, Multiscale proper orthogonal decomposition (mpod) of tr-piv data—a case study on stationary and transient cylinder wake flows, Measurement Science and Technology 31 (9) (2020) 094014. doi:10.1088/1361-6501/ab82be.
URL https://dx.doi.org/10.1088/1361-6501/ab82be

[21] A. Martínez-Sánchez, E. López, S. Le Clainche, A. Lozano-Durán, A. Srivastava, R. Vinuesa, Causality analysis of large-scale structures in the flow around a wall-mounted square cylinder, Journal of Fluid Mechanics 967 (2023) A1. doi:10.1017/jfm.2023.423.