

Piecewise DMD for oscillatory and Turing spatio-temporal dynamics

Alessandro Alla^{a,*}, Angela Monti^b, Ivonne Sgura^c

^a Dipartimento di Scienze Molecolari e Nanosistemi, Università Ca' Foscari Venezia, Via Torino 155, 30170, Mestre, Italy

^b Istituto per le Applicazioni del Calcolo “M. Picone”, CNR, via Amendola 122/D, 70126 Bari, Italy

^c Dipartimento di Matematica e Fisica “E. De Giorgi”, Università del Salento, via per Arnesano, 73100 Lecce, Italy

ARTICLE INFO

Keywords:

Dynamic mode decomposition
Turing patterns
Reaction-diffusion PDE systems
Oscillatory datasets
Spiral waves
Turing-Hopf instability

ABSTRACT

Dynamic Mode Decomposition (DMD) is an equation-free method that aims at reconstructing the best linear fit from temporal datasets. In this paper, we show that DMD does not provide accurate approximation for datasets describing oscillatory dynamics, like spiral waves, relaxation oscillations and spatio-temporal Turing instability. Inspired by the classical “divide and conquer” approach, we propose a piecewise version of DMD (pDMD) to overcome this problem. The main idea is to split the original dataset in N submatrices and then apply the exact (randomized) DMD method in each subset of the obtained partition. We describe the pDMD algorithm in detail and we introduce some error indicators to evaluate its performance when N is increased. Numerical experiments show that very accurate reconstructions are obtained by pDMD for datasets arising from time snapshots of certain reaction-diffusion PDE systems, like the FitzHugh-Nagumo model, a λ - ω system and the DIB morpho-chemical system for battery modeling. Finally, a discussion about the overall computational load and the future prediction features of the new algorithm is also provided.

1. Introduction

The large amount of available temporal datasets has increased in the last decade and with that the study of hidden structures. Specifically, mathematical models can describe rigorously datasets where no information is provided. A data-driven model can help to understand physical phenomena and forecast future development. Recently, the use of machine learning techniques has further improved the capability to discovery mathematical models and has significantly enlarged this area of research. The literature on this topic is rather spread. Here we, first, recall two methods based on different strategies: sparse optimization and neural networks. Finally, we will revoke the Dynamic Mode Decomposition (DMD) which is the building block for the method proposed in this manuscript.

One technique goes back to 2016 where the authors in [1] have used sparse optimization methods, e.g. Lasso algorithm, to recover Ordinary Differential Equations (ODEs). The method relies on a large library including elements that may appear in the model and through a sparse optimization algorithm it is possible to discover the weights of the terms in the library. Thus, if a coefficient is zero, the corresponding element in the library does not appear in the model. This also justifies the use of sparse optimization methods, because only few terms are required

in the searched model. Later, this method has been extended in [2] to PDEs with constant coefficients. The library in this case was built from derivatives of the dataset to include further terms in the model. Finally, extension to PDEs with non constant coefficients was presented in [3].

Another method to construct data-driven modeling is built on Deep Neural Networks (DNNs). Specifically, a new class of DNNs, namely Physics-Informed Neural Networks (PINNs), is trained to solve supervised learning using information from the hidden physical laws one wants to discover. The physical law may be described by an ODE or a PDE model. The great novelty in PINNs is the use of the physical laws together with the mean square error for dataset in the minimization of loss function. Therefore, the output of this approach provides the coefficients needed to discover the model. In this way, the method is forced to converge to the model and to consider the physics behind the dataset. PINNs have been introduced in [4–6].

Another powerful technique for data-driven modeling is the Dynamic Mode Decomposition (DMD). DMD was firstly introduced in [7] and its algorithm finds the best linear fit model without explicit knowledge of the dynamics hidden in the data. Specifically, the DMD algorithm determines eigenvalues and eigenvectors of an approximate linear model. Later, in [8] the authors introduced the *exact* DMD based on a low rank approximation of the original method. Afterwards other algo-

* Corresponding author.

E-mail addresses: alessandro.alla@unive.it (A. Alla), angela.monti@unisalento.it, a.monti@ba.iac.cnr.it (A. Monti), ivonne.sgura@unisalento.it (I. Sgura).

gorithms have been proposed to further improve the above DMD method; we refer to e.g. the optimal DMD [9,10], the extended DMD [11] and the higher order DMD (HODMD, [12,13]) which is based on the fitting of multilinear models. Despite the tremendous effort to improve the method and its computational efficiency, there are (at least) two classes of datasets where DMD still does not work properly, to the best of authors' knowledge.

Specifically, periodic datasets and Turing instability may lead to a wrong (in the first case) or an inaccurate (in the second case) DMD reconstruction, as shown in [14,15] and later in Section 4 of this paper. In [14], the authors show that the DMD algorithm for systems is more accurate when a unique dataset is considered instead of different datasets, one for each variable. They refer to this approach as coupled or uncoupled DMD, respectively. Furthermore, in [14], it is mentioned that DMD may fail in the case of Reaction-Diffusion PDE (RD-PDE) systems. In [15], the uncoupled DMD implementation was applied to Turing instability dynamics leading to stationary pattern solutions. The authors have shown that DMD cannot reconstruct accurately those temporal datasets even if large ranks are used in the algorithm.

The main goal of this paper is to propose an alternative DMD algorithm to face these drawbacks. Stemming from the “divide and conquer” principle, well known in the numerical analysis framework, we introduce a *piecewise* version of the exact DMD, that we will define as pDMD. In the first part of the paper, we will present in details which drawbacks arise when the exact DMD is applied to reconstruct the spatio-temporal dynamics of a selection of RD-PDE systems in two variables. Following the results in [14], in this manuscript, we always use a coupled DMD approach. Furthermore, a randomized DMD algorithm based on the *QB* decomposition (see [16]) is applied to reduce the computational costs. In this context, it is worth mentioning two methods based on the splitting of the data to improve the DMD accuracy: (i) the multiresolution DMD [17] splits the data into slow modes and fast modes and uses a model for each part and (ii) an online DMD method for time varying system in [18]. Both cited methods share the same goal of this paper but they propose different algorithms.

Here, we consider the following RD-PDE systems whose solutions exhibit an oscillatory dynamics: i) the FitzHugh-Nagumo model [19] 1D in space and with relaxation oscillations in time leading to a limit cycle in the phase plane; ii) the λ - ω system from [20,21] with spiral waves solutions. In the first case, also for the full rank approximation, the oscillatory solutions and the corresponding limit cycle cannot be recovered at all. In the second case, for a certain range of ranks, the final spiral wave is approximated with low accuracy, but the amplitude and phase of the oscillating behavior are not preserved. Moreover, for larger ranks, where a better approximation is expected, ill-conditioning of the fitting procedure behind DMD emerges and the time dynamics is lost also at the final time. These results are reported in Section 4.1.

To deepen the discussion started in [15] about DMD defects, we apply here the “coupled” exact randomized DMD to follow the Turing dynamics of the morphochemical RD-PDE model, introduced in [22] and known as DIB model. In Section 4.2, we deal with the Turing instability, where a transient unstable regime is present (said *reactivity zone*) before reaching the spatially inhomogeneous Turing pattern at the steady state (*stabilizing zone*). Finally, in Section 4.3, we also consider an example of Turing-Hopf instability [23] where there exists an interplay between Turing and Hopf instabilities. After the initial unstable behavior, the solutions are patterns that oscillate both in space and time. In the first case (Section 4.2), we show that after a certain rank the DMD error dramatically increases and indeed blows up, because also here the ill-conditioning of the fitting procedure appears. Nevertheless for small ranks it is possible to reconstruct with few accuracy the time dynamics and the final pattern (see e.g. Fig. 8). For the more complicated Turing-Hopf dynamics (Section 4.3), that, at best of author's knowledge, has not been studied so far by DMD, very inaccurate approximations are obtained until the full rank choice. In particular, the best case exhibits large errors both for the final pattern and all the

time history, as documented by the limit cycle approximation (see e.g. Figures 11, 12).

Our proposed pDMD algorithm will work as follows. Given a dataset S , corresponding e.g. to periodic or Turing dynamics in the time interval $[0, T]$, and a tolerance $\text{tol} > 0$, we start with DMD on the whole dataset. If the obtained error, that is the maximum of the worst-approximation in time between the dataset and its DMD reconstruction, is above the threshold tol , then we split $S = [S_1, S_2]$ into $N = 2$ parts and compute DMD in each submatrix S_i . This error indicator is checked at each iteration and, if the i -th dataset is not reconstructed accurately, we directly increase the number of subdivisions N . We iterate this splitting in N parts till we reach the desired accuracy on all submatrices $S_i, i = 1, \dots, N$. This procedure will identify the first acceptable partition size, say N^* . In a second step, we decide to increase the partition size for $N > N^*$ in order to look for a sort of convergence along the whole time dynamics, by controlling another error indicator in the Frobenius norm accounting for the whole time DMD reconstruction. This further piecewise iteration will tend to improve an initial good reconstruction, as we will show in our numerical experiments. More details will be discussed in the formalized algorithms reported in Section 5 and in Section 6. An extensive numerical study of our algorithm is deeply discussed by considering the snapshot matrices obtained by the numerical solutions of the RD-PDE models described above. We will show that pDMD increases the accuracy of the approximation removing the drawbacks discussed. As a consequence, pDMD does not identify a single model, but several different models in each subinterval for a fixed N -partition. This feature will improve accuracy at the cost of increasing the number of degrees of freedom. In Section 6.5, we discuss these questions in details and we introduce some indicators to quantify the computational cost required by pDMD. Hence, for all our simulations, we will show that, beside accuracy, a significant spatial reduction in memory occupation is obtained by pDMD. Also, we devote a brief section to the study of pDMD for future state predictions. It is worth remarking that our pDMD will work with several datasets of small sizes, if many subintervals are required, and this allows to better follow “locally” the dynamics of the problem and then to better capture its behavior with a linear regression method.

The paper is organized as follows. Section 2 recalls the exact DMD method and its variant based on the randomized QB decomposition. In Section 3, we briefly introduce the general reaction-diffusion PDE system in exam and the IMEX Euler method in matrix form for its time approximation, because the obtained numerical solutions are used to build our datasets. As discussed above, in Section 4, we show how the exact (randomized) DMD fails on different datasets for periodic data or Turing dynamics. The pDMD is introduced in Section 5 together with a complete description of our algorithm. Finally, in Section 6, we present our numerical results including the discussion about computational costs and future prediction features. Conclusions are drawn in Section 7. All numerical simulations have been performed in MATLAB (ver. 2019a) on a computer DELL, i7 Intel Core processor 2.8 GHz and 16Gb RAM.

2. Dynamic mode decomposition

The Dynamic Mode Decomposition (DMD) technique aims at reconstructing the best linear dynamical system hidden in a given temporal dataset $S = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{n \times (m+1)}$ where the i -th column of the matrix S corresponds to the data at time t_{i-1} and $t_{i-1} < t_i, i = 1, \dots, m$. DMD fits the following linear model on these data:

$$\begin{aligned} \dot{\mathbf{y}}(t) &= A\mathbf{y}(t) \quad t \in [0, T], \\ \mathbf{y}(0) &= \mathbf{y}_0 \in \mathbb{R}^n, \end{aligned} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is an unknown operator, $\mathbf{y}(t) : [0, T] \rightarrow \mathbb{R}^n$, the initial condition $\mathbf{y}_0$ coincides with the first element of the dataset $\mathbf{x}_0$ and the data are such that $\mathbf{x}_i \approx \mathbf{y}(t_i), i = 1, \dots, m$. To discover the matrix A , DMD starts by splitting the snapshot matrix S into two matrices

$$S_L = \begin{bmatrix} | & | & \dots & | \\ \mathbf{x}_0 & \mathbf{x}_1 & \dots & \mathbf{x}_{m-1} \\ | & | & \dots & | \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad S_R = \begin{bmatrix} | & | & \dots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_m \\ | & | & \dots & | \end{bmatrix} \in \mathbb{R}^{n \times m}$$

and then, assuming that there exists a linear relation between $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$ for $k = 0, \dots, m-1$, tries to find the best fitting matrix $A \in \mathbb{R}^{n \times n}$ such that

$$S_R = AS_L, \iff \mathbf{x}_{k+1} = A\mathbf{x}_k, \quad k = 0, \dots, m-1. \quad (2)$$

Therefore, for this purpose, the following least squares optimization problem can be formulated

$$A := \operatorname{argmin}_{A \in \mathbb{R}^{n \times n}} \|S_R - AS_L\|_F \quad (3)$$

where $\|\cdot\|_F$ is the Frobenius norm. It is well known that (3) can be solved by computing the Moore-Penrose pseudo-inverse of S_L , such that the best fit solution is given by $A = S_R S_L^\dagger$. However, the dimension n of the problem may be large and, from a computational point of view, it is not convenient to calculate A directly by the above product. Indeed, for this reason, the exact DMD algorithm proposed in [8] computes a low rank approximation of the matrix $S_L \approx \Psi_r \Sigma_r V_r^T =: S_{L,r}$ where $\Sigma_r \in \mathbb{R}^{r \times r}$ is a diagonal matrix whose entries $\sigma_i \geq 0$ are the first r singular values of S_L sorted in decreasing order, $\Psi_r \in \mathbb{R}^{n \times r}$ and $V_r \in \mathbb{R}^{m \times r}$ are matrices with orthonormal columns and usually $r \ll n$. Therefore, an approximation A_r of the full matrix A can be obtained by computing the pseudoinverse of $S_{L,r}$ given by $S_{L,r}^\dagger = V_r \Sigma_r^{-1} \Psi_r^T$, such that

$$A_r := S_R S_{L,r}^\dagger = S_R V_r \Sigma_r^{-1} \Psi_r^T \approx A.$$

Then, by projecting it onto the POD modes (the r leading left singular vectors Ψ_r), it is possible to compute the reduced matrix

$$\tilde{A} = \Psi_r^T S_R V_r \Sigma_r^{-1} \in \mathbb{R}^{r \times r}. \quad (4)$$

We observe that $\tilde{A}$ has the same leading r eigenvalues of A (see [8, Theorem 1] for a detailed proof). Thus, we compute the spectral decomposition $\tilde{A}W = W\Lambda$ where the columns of W are the eigenvectors of $\tilde{A}$ and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_r)$ is a diagonal matrix containing the corresponding leading r eigenvalues of the full matrix A . Finally, we reconstruct the high-dimensional DMD modes of A by

$$\Phi = S_R V_r \Sigma_r^{-1} W \in \mathbb{R}^{n \times r},$$

and the state variable as

$$\mathbf{x}_k \approx \tilde{\mathbf{x}}_k := \sum_{i=1}^r \phi_i \lambda_i^k b_i = \Phi \Lambda^k \mathbf{b}. \quad (5)$$

The DMD modes ϕ_i are the columns of Φ (eigenvectors of A), λ_i are the corresponding eigenvalues, while b_i can be obtained by solving in the least squares sense the overdetermined system $\Phi \mathbf{b} = \sum_{i=1}^r \phi_i b_i = \mathbf{x}_0$, where $\mathbf{x}_0$ is the first snapshot. The notation $\tilde{\mathbf{x}}_k$ will indicate the DMD reconstruction of $\mathbf{x}_k$.

We briefly summarize these steps in Algorithm 1, defined as the exact DMD in [8].

Algorithm 1 (Exact) DMD.

- 1: **INPUT** Snapshots $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_m\}$, rank r
 - 2: **OUTPUT** DMD modes $\{\phi_1, \dots, \phi_r\}$, eigenvalues $\{\lambda_1, \dots, \lambda_r\}$, reduced solution $\{\tilde{\mathbf{x}}_k\}_{k=0}^m$
 - 3: Set $S_L = [\mathbf{x}_0, \dots, \mathbf{x}_{m-1}]$, $S_R = [\mathbf{x}_1, \dots, \mathbf{x}_m]$
 - 4: Compute the truncated SVD of S_L , $S_L \approx S_{L,r} = \Psi_r \Sigma_r V_r^T$
 - 5: Compute $\tilde{A} = \Psi_r^T S_R V_r \Sigma_r^{-1}$
 - 6: Compute the spectral decomposition of $\tilde{A}$, $\tilde{A}W = W\Lambda$
 - 7: Calculate the DMD modes as $\Phi = S_R V_r \Sigma_r^{-1} W$
 - 8: Set $\mathbf{b} = \Phi^\dagger \mathbf{x}_0$
 - 9: Set $\tilde{\mathbf{x}}_k = \sum_{i=1}^r \phi_i \lambda_i^k b_i$, $k = 0, \dots, m$
 - 10: $\tilde{S}(r) = [\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m]$
-

2.1. Randomized DMD

The exact DMD method as presented in Algorithm 1 can be still computationally very expensive if n is very large. Therefore, in this subsection we recall the method introduced in [16] based on the randomized QB decomposition. The aim is to write the dataset $S \approx QB$ where $Q \in \mathbb{R}^{n \times r}$ is a matrix with orthonormal columns and $B \in \mathbb{R}^{r \times (m+1)}$. Once the QB decomposition is obtained for the snapshot matrix S , one will directly apply the DMD algorithm to the matrix $B = [\mathbf{b}_0, \dots, \mathbf{b}_m] \in \mathbb{R}^{r \times (m+1)}$ which is clearly much smaller than the original matrix S . In this way, after splitting the matrix into $B_L = [\mathbf{b}_0, \dots, \mathbf{b}_{m-1}]$ and $B_R = [\mathbf{b}_1, \dots, \mathbf{b}_m]$ we can solve the optimization problem

$$\tilde{A} := \operatorname{argmin}_{\tilde{A} \in \mathbb{R}^{r \times r}} \|B_R - \tilde{A}B_L\|_F. \quad (6)$$

It turns out that the solution of (6) is $\tilde{A} = B_R B_L^\dagger$. We note that $\tilde{A}$ is a low rank approximation of A and differs from (4). This computation is now doable since the dimensions of the matrices B_L and B_R are (eventually¹) very small. Then, we can compute the eigenvalue decomposition $\tilde{A}W = W\Lambda$ and set the DMD modes $\Phi = QB_R V \Sigma^{-1} W$. Therefore, the reduced solution $\mathbf{x}_k, \forall k$ can be obtained as in Algorithm 1.

In the remainder of this section we recall how to obtain the randomized QB decomposition.

First of all, one has to choose the so called *target rank* r and the number of oversampling p usually $5 \leq p \leq 10$. Then, we generate a random test matrix $\Omega \in \mathbb{R}^{(m+1) \times \ell}$ with $\ell = r + p$ drawn from the normal Gaussian distribution. The oversampling needs to guarantee the target rank r (see e.g. [24]), in fact it is common to build slightly larger test matrix to obtain improved basis. The sampling matrix can be computed as $Y = S\Omega \in \mathbb{R}^{n \times \ell}$ or by using the power iteration method as $Y = ((S^T S)^q S)\Omega, q \in \mathbb{N}$. The matrix Q is then obtained from the QR decomposition of the sampling matrix Y . It is shown in [16] that the power iteration improves the quality of the approximated basis matrix Q using just one or two iterations, i.e. $q = \{1, 2\}$. Finally, the low rank matrix B will be such that $B = Q^T S$. The QB algorithm is summarized in Algorithm 2 using Matlab notations.

Algorithm 2 Randomized QB decomposition.

- 1: **INPUT** Snapshots S , target rank r , oversampling p , number of power iterations q
 - 2: **OUTPUT** $Q \in \mathbb{R}^{n \times r}$, $B \in \mathbb{R}^{r \times (m+1)}$
 - 3: $\ell = r + p$
 - 4: $\Omega = \operatorname{rand}(m+1, \ell)$
 - 5: $Y = S\Omega$
 - 6: **for** $j = 1, \dots, q$ **do**
 - 7: $[Q, \cdot] = \operatorname{qr}(Y)$
 - 8: $[Z, \cdot] = \operatorname{qr}(S^T Q)$
 - 9: $Y = SZ$
 - 10: **end for**
 - 11: $[Q, \cdot] = \operatorname{qr}(Y)$
 - 12: $B = Q^T S$
-

3. Full model and its numerical approximation

In this paper, our aim is to apply DMD to reconstruct in time both oscillatory dynamics, leading for example to relaxation oscillations and spiral waves, and Turing pattern formation dynamics, that presents a transient unstable regime, known as reactivity, before reaching a structured spatially inhomogeneous pattern as stationary solution at the steady state. A common feature is that all these time behaviors can characterize the solutions of a RD-PDE system for different choices of the involved parameters. More details can be found e.g. in [25].

¹ Here, a low rank structure for the dataset is assumed, even if this assumption is not guaranteed a priori.

For this reason, here we consider the following general RD–PDE system:

$$\begin{cases} u_t = d_u \Delta u + f(u, v), & \mathbf{z} \in \Omega \subset \mathbb{R}^d, \quad t \in (0, T], \\ v_t = d_v \Delta v + g(u, v), \\ (\mathbf{n} \nabla u)|_{\partial\Omega} = b_u(t), \quad (\mathbf{n} \nabla v)|_{\partial\Omega} = b_v(t), \\ u(x, y, 0) = u_0(x, y), \quad v(x, y, 0) = v_0(x, y), \end{cases} \quad (7)$$

where $d_u, d_v \in \mathbb{R}^+$ are the diffusion coefficients, $T > 0$ the final time of integration, $d \in \{1, 2\}$ is the space dimension of (7). In the case $d = 2$, we have $\mathbf{z} = (x, y)$. The nonlinear reaction terms $f, g : \mathbb{R}^2 \rightarrow \mathbb{R}$ account for biological, chemical and other kind of phenomena. We will consider Neumann boundary conditions, where $\mathbf{n}$ denotes the exterior normal to the boundary $\partial\Omega$ and $b_u(t), b_v(t) : [0, T] \rightarrow \mathbb{R}$ are scalar functions, identically zero in the case of homogeneous Neumann BCs. Our aim is to solve numerically (7) to generate a dataset S of our interest and then apply the DMD directly to this dataset without using any extra information coming from the (known) PDE.

The model (7) depends on various parameters that will be chosen *ad hoc* in order to study different kinds of dynamics, as follows. Indeed, in the next sections we will consider: 1) the FitzHugh-Nagumo model [19] with relaxation oscillations and related limit cycle and 2) a λ - ω system [20,21] with spiral waves. Concerning Turing pattern formation, we consider the DIB morphochemical model [22,23,26,27]. Furthermore, we study this model also in presence of a combination of oscillatory and Turing behaviors, arising from the so-called Turing-Hopf patterns, that are spatial inhomogeneous Turing patterns oscillating both in space and time ([23,28]).

For all the above models, the construction of the snapshot matrix S , to feed the DMD method, follows from the numerical approximation of (7). Hence, for the spatial semi-discretization we apply standard second order finite differences with a total number n of meshpoints inside Ω . For the approximation in time, we apply the IMEX Euler scheme (i.e. we treat implicitly the diffusion part and explicitly the nonlinear reaction terms) on the meshgrid $\tau_{i+1} = \tau_i + h_t$ $i = 0, \dots, n_T$ with timestep $h_t = T/n_T$. To simulate oscillatory solutions and Turing patterns we require both fine spatial meshes and integration for long times ($T \gg 1$) to attain the *standing* asymptotic oscillations (i.e. the limit cycle in the phase space) or the stationary Turing pattern. For this reason the computational load of the usual *vector approach* solving a large sparse linear system at each timestep for the IMEX Euler method, can be very expensive. Then, here we apply the recent *matrix-oriented* approach and in particular the *rEuler* method proposed in [29], solving at each timestep a Sylvester matrix equation in the reduced spectral space.

To further reduce the computational cost, we will store only some of the computed snapshots on a temporal sub-grid $t_{i+1} = t_i + \kappa h_t$, where $\kappa \in \mathbb{N}$ allows to select equidistributed snapshots from the original grid $\{\tau_i\}_{i=0}^{n_T}$. Note that, if $\kappa = 1$ we consider all the snapshots, if $\kappa = 4$ we store snapshots every 4 timesteps from the original grid. Hence, for the simulations presented in the next sections, we build the snapshot matrix $S \in \mathbb{R}^{2n \times (m+1)}$, where $m+1 = n_T/\kappa$ and, for $i = 0, \dots, m$, the $(i+1)$ -th column is the extended vector $\mathbf{x}_{i+1} := [\mathbf{u}_{i+1}; \mathbf{v}_{i+1}] \in \mathbb{R}^{2n}$ given by the concatenation of the numerical solutions for both unknowns u and v , i.e. $\mathbf{u}_{i+1} \approx u(\cdot, t_i)$, $\mathbf{v}_{i+1} \approx v(\cdot, t_i)$.

4. Examples of inaccurate approximations by (exact) DMD for oscillatory dynamics

The examples discussed in this section have in common that the exact randomized DMD method, recalled in Section 2, does not approximate the dataset accurately. In the first two examples for periodic datasets, we will see that DMD completely fails even with a full rank approximation. In the case of Turing instability dynamics, we will show that DMD exhibits poor approximation with also an error behavior dramatically increasing with the rank r in case of stationary patterns.

In all tests presented, to measure the quality of the DMD approximation of rank r , we use the following relative error in the Frobenius norm between the dataset S and its DMD reconstruction $\tilde{S}$:

$$\mathcal{E}(\tilde{S}, r) = \frac{\|S - \tilde{S}\|_F}{\|S\|_F}. \quad (8)$$

Furthermore, for each kind of dynamics considered, we are interested in comparing the behavior in time of the full dataset with that approximated by the DMD for a given rank r . For this reason, we will compare the time behavior of the spatial mean of the full and reduced solutions, defined for u by

$$\langle u(t) \rangle := \frac{1}{|\Omega|} \int_{\Omega} u(x, y, t) dx dy \approx \text{mean}(\mathbf{u}_k), \quad k = 0, \dots, m \quad (9)$$

and $\text{mean}(\tilde{\mathbf{u}}_k)$, $k = 0, \dots, m$, respectively. Similar computations are done for the variable v . In all examples shown in this section, DMD is performed with the QB decomposition from Algorithm 2 and the results presented are referred to the value r which minimizes $\mathcal{E}(\tilde{S}, r)$, as can be extracted by the corresponding reported figures.

4.1. Examples with periodic datasets

The first example will focus on the approximation of the limit cycle generated by the FitzHugh-Nagumo model [19]. The second example concerns the reconstruction of the spiral wave solution and dynamics of the λ - ω system in [20,21]. As already outlined, we are interested in the reconstruction of the whole spatio-temporal history.

4.1.1. FitzHugh-Nagumo model: limit cycle

The one dimensional FitzHugh-Nagumo (FHN) model describes the activation and deactivation dynamics of a spiking neuron and it is a simplified version of the more famous Hodgkin-Huxley model [19]. The nonlinear reaction terms in (7) are given by

$$\begin{cases} f(u, v) = \frac{u(u-0.1)(1-u)}{d_u} - \frac{v}{d_u} + \frac{c}{d_u}, \\ g(u, v) = bu - \gamma v + c. \end{cases} \quad (10)$$

To build the dataset S , here we solve the FHN system on the 1D domain $\Omega = [0, 1]$, for $t \in [0, T]$, $T = 6$, and $d_u = 0.015$, $d_v = 0$, $b = 0.5$, $\gamma = 2$, $c = 0.05$. The initial and boundary conditions are given by

$$\begin{aligned} u_0(x) &= 0, \quad v_0(x) = 0, \quad x \in \Omega, \\ u_x(0, t) &= -(5 \cdot 10^4 t^3 e^{-15t}), \quad u_x(1, t) = 0, \quad t \in [0, T]. \end{aligned} \quad (11)$$

For the spatial meshgrid we consider $n = 1024$ internal points. We integrate in time by applying the IMEX Euler scheme with $h_t = 10^{-3}$, thus we have $m+1 = 6000$. The parameters in (10) and (11) are taken from [19].

In the left panel of Fig. 1, we show the relative error $\mathcal{E}(\tilde{S}, r)$, for $r = 1, \dots, R$, where $R = 51$ is the rank of the snapshot matrix $S \in \mathbb{R}^{2048 \times 6000}$ computed by the Matlab command `rank`. This error is very high and erratic, indicating that the DMD reconstruction for both variables is completely wrong. Even worse, the error increases when we consider higher values for the rank, which is something not expected a priori. The minimum value is reached for $r = 28$, i.e. $\mathcal{E}(\tilde{S}, 28) = 0.9618$. For this r value, in the middle and right panels of Fig. 1 we compare the spatial mean of the data (9) with that of the DMD reconstruction, for u and v respectively. It is clear that the DMD reconstruction does not capture neither the periodic dynamics nor the amplitude of the relaxation oscillations of the FHN model.

In Fig. 2, right panel, we show the DMD reconstructions of u and v in space and time on the domain $\Omega \times [0, T]$. By comparing the dataset S reported in the left panels, it becomes still more evident how the DMD method fails. The effect of this DMD failure on the limit cycle reconstruction is shown in Fig. 3.

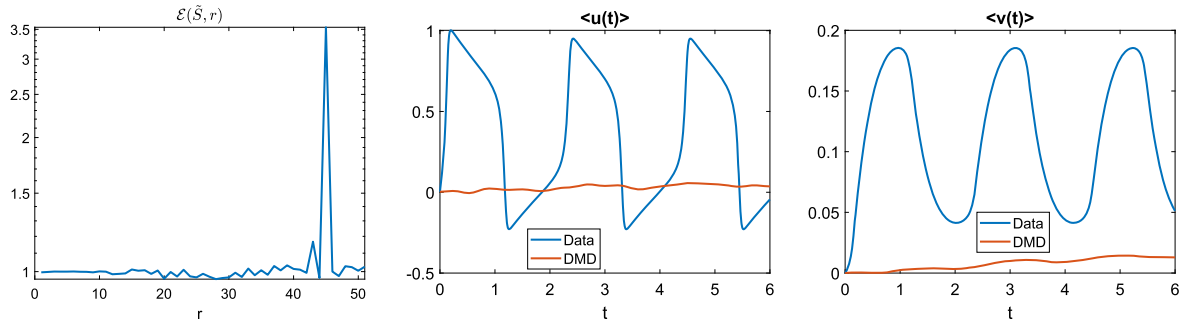


Fig. 1. FHN model, relaxation oscillations. Relative error (8) of the DMD reconstruction $\tilde{S}$ with respect to the dataset S (left). Comparison of the spatial means for the variables u (center) and v (right) for DMD of rank $r = 28$ corresponding to the minimum of the error.

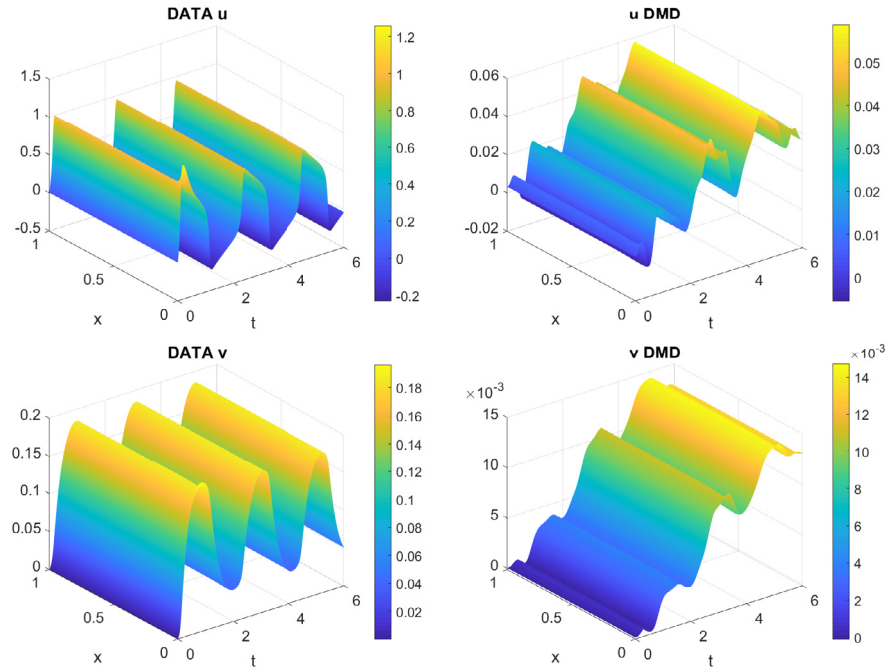


Fig. 2. FHN model, relaxation oscillations. Full model solutions u and v (left panels) and corresponding DMD reconstructions (right panels) for $r = 28$.

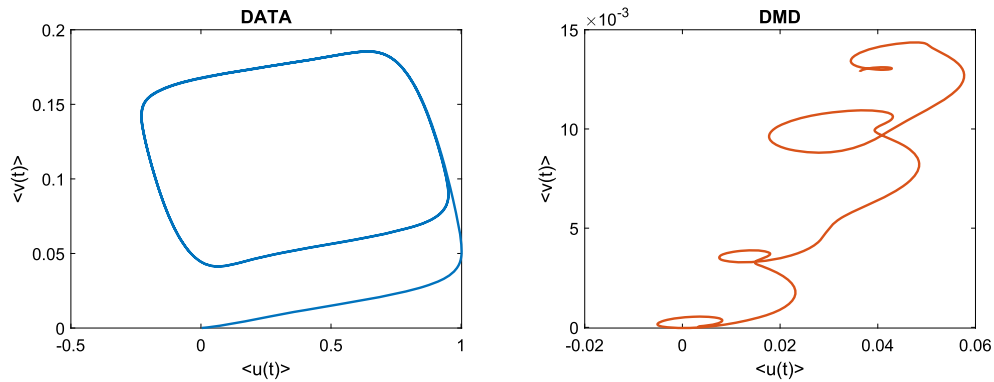


Fig. 3. FHN model, relaxation oscillations. Limit cycle corresponding to the dataset S in the phase plane $(\langle u \rangle, \langle v \rangle)$ (left) and its DMD approximation for $r = 28$ (right).

4.1.2. λ - ω system: spiral waves

Here we consider the λ - ω system from [20,21], with nonlinear kinetics (7) given by

$$\begin{aligned} f(u, v) &= \rho(\lambda(u, v)u - \omega(u, v)v), \\ g(u, v) &= \rho(\omega(u, v)u + \lambda(u, v)v), \end{aligned} \quad (12)$$

where $\lambda = 1 - (u^2 + v^2)$ and $\omega = -\beta(u^2 + v^2)$, $\beta > 0$, $\rho > 0$. On the 2D spatial domain $\Omega = [0, L] \times [0, L]$, with $L = 130$, we choose the parameter

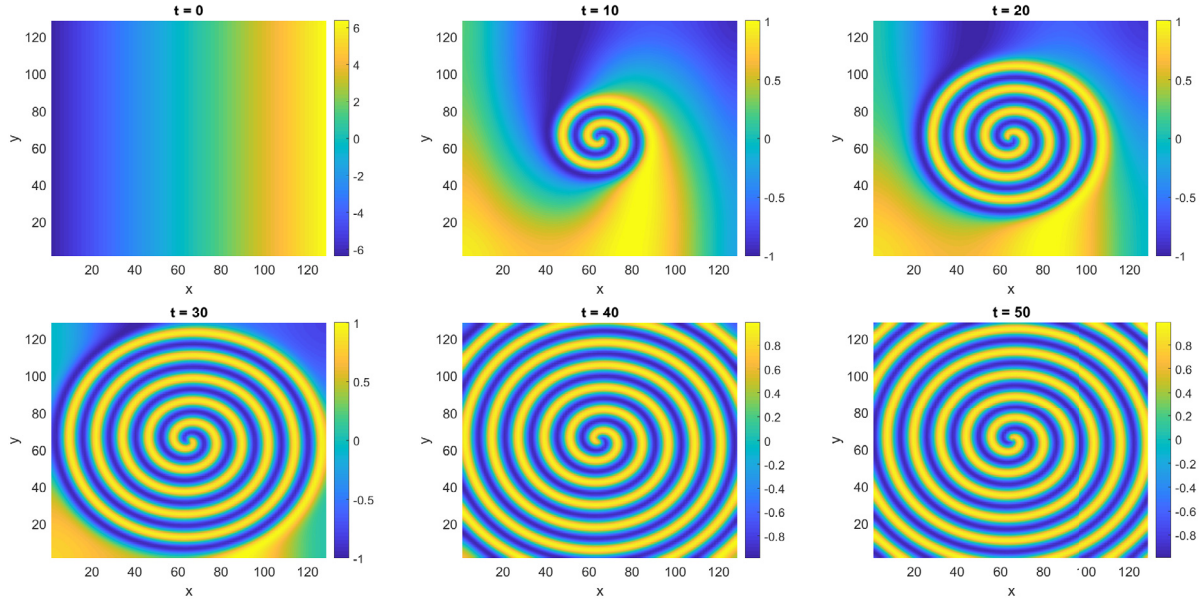


Fig. 4. $\lambda - \omega$ system. Spiral wave generated for the variable u at different times in $[0, T]$, see more details in the main text.

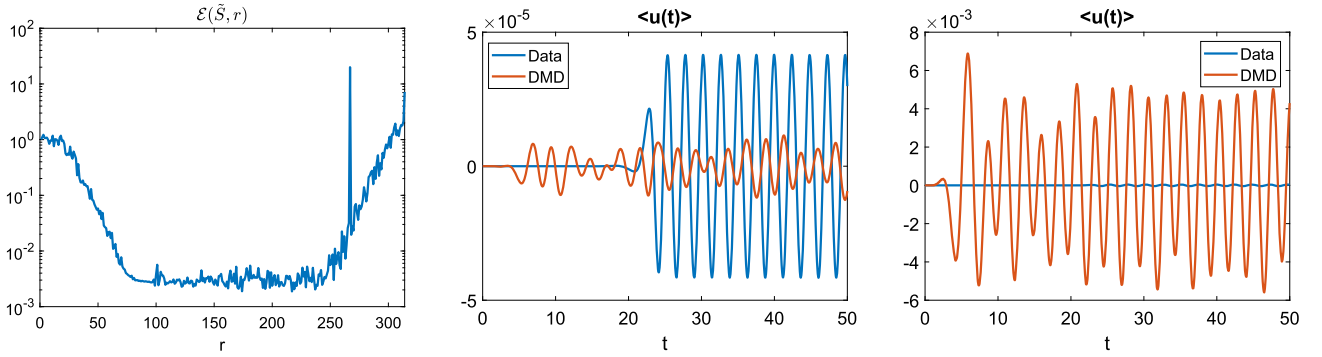


Fig. 5. $\lambda - \omega$ system, spiral wave. DMD relative error in (8) for the dataset S (left plot). Comparison of the spatial means for u : DMD is applied with $r = 217$ (center plot) and $r = 300$ (right plot). The choice of $r = 217$ corresponds to the minimum value of $\mathcal{E}(\tilde{S}, r) = 0.0019$, whereas we choose $r = 300$ which is a value close to the rank of the dataset S .

values and initial conditions from [21] for which spiral waves solutions arise:

$$d_u = 4, d_v = 4, \rho = 10, \beta = 1,$$

$$u_0(x, y) = \frac{1}{10} \left(x - \frac{L}{2} \right), v_0(x, y) = \frac{1}{10} \left(-\frac{y}{2} + \frac{L}{4} \right)$$

and homogeneous Neumann boundary conditions $b_u(t) \equiv 0 \equiv b_v(t)$ in (7). The domain Ω is discretized by $n_x = n_y = 99$ interior points, such that the total number of mesh points is $n = n_x n_y = 9801$. We integrate in time with timestep $h_t = 10^{-3}$ until $T = 50$, by using the IMEX Euler scheme in matrix-oriented form (see [29]). We save the snapshots every four timesteps ($\kappa = 4$), such that the considered dataset S has dimension $2n \times (m + 1)$ and $m + 1 = 12500$.

Departing from the step values of the initial data, for both u and v , in a transient regime the numerical solution starts to form an archimedean spiral wave with “core” (fixed point) in the center of the domain $(x_c, y_c) = (\frac{L}{2}, \frac{L}{2})$ which arms oscillate in space and time until at a certain time, say $\tilde{t}$, when the entire Ω is covered (as shown in Fig. 4). Thereafter, for $t \geq \tilde{t}$ a new time regime arises where the spiral continues indefinitely in its oscillating dynamics such that in the phase plane we can say that a limit cycle is attained by the spatial means $\langle u \rangle, \langle v \rangle$. We show the dynamics of $\langle u(t) \rangle$ in Fig. 5 (center), where the two time regimes in $[0, \tilde{t}]$ and $[\tilde{t}, T]$ with $\tilde{t} \approx 25$ are evident, the u -snapshot at the final time T in Fig. 6 (left) and the corresponding limit cycle in Fig. 7 (left).

In Fig. 5 (left) we show the relative error $\mathcal{E}(\tilde{S}, r)$, obtained by the DMD for $r = 1, \dots, R$, where the rank of the dataset S is $R = 314$. After an initial decreasing trend, for $r \gtrsim 100$ an erratic plateau around $3e-3$ can be observed where the minimum value $\mathcal{E}(\tilde{S}, r) = 0.0019$ is reached for $r = 217$. Moreover, this low accuracy gets worse for $r \geq 250$ when the error dramatically increases due to ill-conditioning of the matrix $\tilde{A}$. We stress this DMD drawback because usually better results are expected by increasing the value of r and this clearly does not happen here. To support this conclusion, in Fig. 5, we also report the spatial mean dynamics for the variable u obtained by DMD for $r = 217$ (middle plot) and $r = 300$ (right plot) when $\mathcal{E}(\tilde{S}, 300) = 0.8471$. In both cases, we note that: i) the approximation of the two distinct time regimes is missed, ii) a large difference in the wave amplitude is present; iii) in the best case $r = 217$ (center plot), the frequency of the oscillations is preserved, but they are in phase opposition as shown in the middle panel of Fig. 5.

In Fig. 6, we show the full model solution (left panel) and the DMD reconstructions with rank $r = 217$ (middle plot) and $r = 300$ (right plot) at the final time $T = 50$. For $r = 300$, DMD fails essentially in the core of the spiral, instead for $r = 217$, where the global Frobenius error is minimum, DMD seems to be in great agreement with the data, even though its time history is really different, as discussed above. To further support the above points (i)–(iii) concerning the defects of DMD in the time dynamics approximation, in Fig. 7 we report the DMD reconstructions

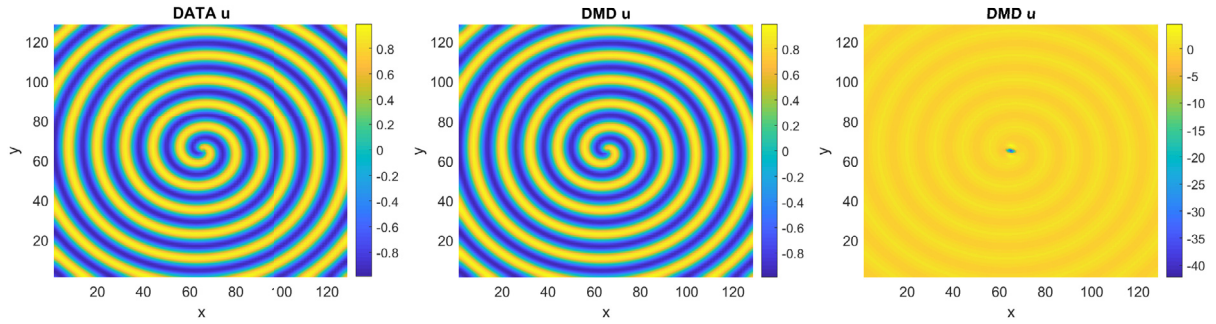


Fig. 6. λ - ω system, spiral wave. Data for the variable u at the final time $T = 50$ (left) and its DMD reconstructions with $r = 217$ (center plot) and $r = 300$ (right plot).

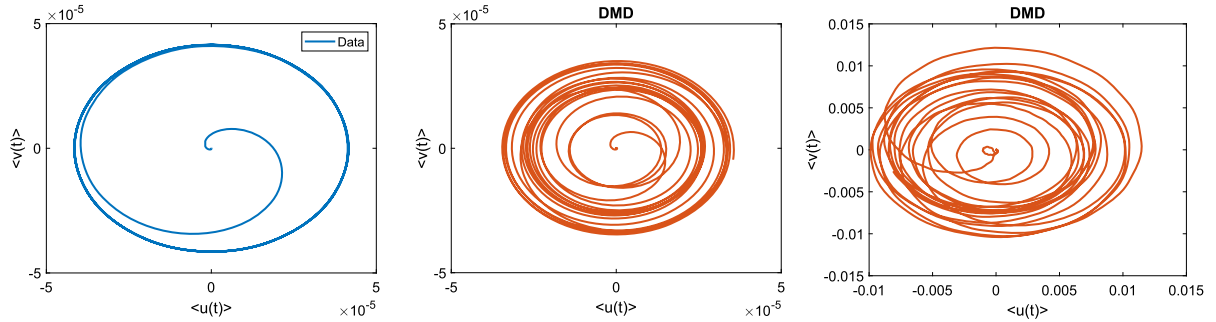


Fig. 7. λ - ω system, spiral wave. Comparison of the limit cycles in the phase plane $(\langle u \rangle, \langle v \rangle)$: data (left) and DMD with $r = 217$ (center) and $r = 300$ (right).

tions for $r = 271$ and $r = 300$ (center and right plot, respectively) in the phase plane $(\langle u \rangle, \langle v \rangle)$. Indeed, by comparing the attained limit cycles with that for the data (left plot) it is still more evident that both DMD reconstruction fails.

To conclude, in this section we have shown two examples with oscillatory datasets where DMD fails its reconstruction.

4.2. Example on the Turing instability

In this section, we deal with a RD model that exhibits the so-called Turing instability. In this case, the initial data are spatially random perturbations of the equilibrium of the model in absence of diffusion, say (u_e, v_e) . This peculiar dynamics presents essentially two time regimes: i) the *reactivity zone* where (u_e, v_e) destabilizes from the initial conditions because of diffusion and ii) the *stabilizing regime* where the solution starts to be attracted towards a steady state spatially structured pattern, known as *Turing pattern* of different morphologies, see e.g. [25,30,31]. The challenges for low rank techniques, like Proper Orthogonal Decomposition (POD) and Discrete Empirical Interpolation Method (DEIM), to reconstruct both regimes have been already highlighted in [32]. For the DMD performance an initial study is reported in [15] where the authors have considered an uncoupled approach, that is they have reconstructed independently the unknowns.

This example focuses on the DIB morphochemical system that is an important realistic application for electrochemical phase formation modeling ([22,23,27]). The kinetics in (7) are given by

$$\begin{aligned} f(u, v) &= \rho(A_1(1-v)u - A_2u^3 - B(v-\alpha)), \\ g(u, v) &= \rho(C(1+k_2u)(1-v)[1-\gamma(1-v)] - Dv(1+k_3u)(1+\gamma v)). \end{aligned} \quad (13)$$

If $D = \frac{C(1-\alpha)(1-\gamma+\gamma\alpha)}{\alpha(1+\gamma\alpha)}$, there exists the spatially homogeneous equilibrium $(u_e, v_e) = (0, \alpha)$ that can undergo Turing instability [22]. Here, we consider the parameter values taken from [32]:

$$\begin{aligned} A_1 &= 10, A_2 = 1, \alpha = 0.5, B = 66, C = 3, \gamma = 0.2, \\ d_u &= 1, d_v = 20, k_2 = 2.5, k_3 = 1.5, \rho = \frac{25}{4}. \end{aligned}$$

The initial conditions are spatially random perturbation of the homogeneous equilibrium, given by $u_0(x, y) = u_e + 10^{-5}\text{rand}(x, y)$, $v_0(x, y) = v_e + 10^{-5}\text{rand}(x, y)$. We discretize the spatial domain $\Omega = [0, 20] \times [0, 20]$ with $n_x = n_y = 100$ meshpoints, such that $n = n_x n_y = 10000$ and, for stability reasons, we consider the timestep $h_t = 10^{-3}$ until the final time $T = 40$. We save the snapshots every four timesteps, such that the dataset is $S \in \mathbb{R}^{2n \times 10000}$. In the left panel of Fig. 8, we show the DMD relative error $\mathcal{E}(\tilde{S}, r)$ for $r = 1, \dots, R$, $R = 303$ corresponding to the rank of S . The error dramatically increases for large values of r and indeed blows up for $r \geq 50$. Its minimum is obtained for $\mathcal{E}(\tilde{S}, 22) = 0.1008$ (almost 10%). Nevertheless the time dynamics of the spatial mean reconstruction for $r = 22$ exhibits an oscillating behavior around the mean $\langle u(t) \rangle$ of the dataset, as shown in the right panel of Fig. 8.

In Fig. 9, we show the full model solution, i.e. the labyrinth Turing patterns for the variables u and v attained at the final time $T = 40$ (left plots) and their DMD reconstructions with $r = 22$ (center plots). The DMD approximates quite well the shape of the final patterns, although the amplitude is not correct, as highlighted by computing the spatial absolute errors between them that are reported in the right panels of Fig. 9.

In addition, the relative error gets worse when the rank increases and it never gets lower than 10%. It is worth noting that a similar bad DMD behavior for the DIB model was already discussed in [15]. Even if here we consider a different DMD implementation, that is a coupled approach based on a randomized version of DMD to improve the computational efficiency, we are not able to remove the DMD drawbacks for Turing pattern approximation.

4.3. Example with spatio-temporal oscillatory dynamics in the Turing-Hopf instability

The last class of problems in exam exhibits a Turing-Hopf instability, that is an interplay between Turing and Hopf instabilities [23]. In particular, the solutions of (7) are oscillatory patterns both in space and time. We consider the DIB morpho-chemical model, whose kinetics in (7) are the same as in (13), but the new dynamics arises for different model parameter values given by: $A_2 = 30$, $B = 109$, $C = 2.794$, $\rho = 50$.

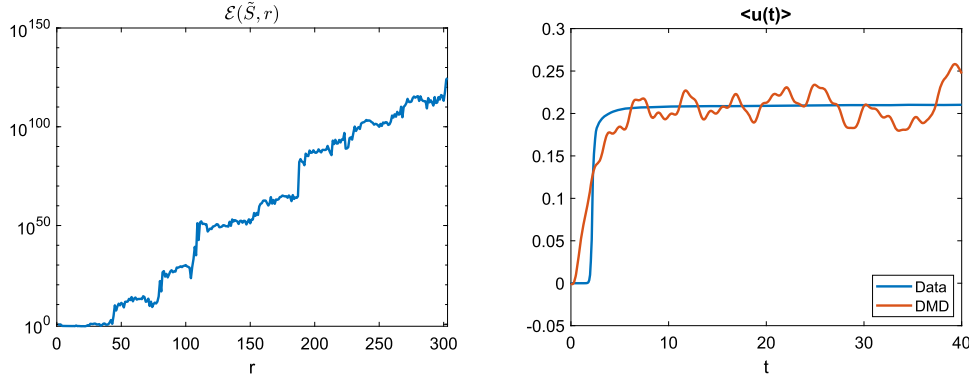


Fig. 8. DIB model, Turing instability. Left plot: relative error (8); right plot: the spatial mean for the variable u in the dataset shows the two time regimes for the reactivity (until $t \approx 4$) and the stabilizing zone, whereas the DMD reconstruction for $r = 22$ destabilizes early and exhibits an oscillating behavior for long times.

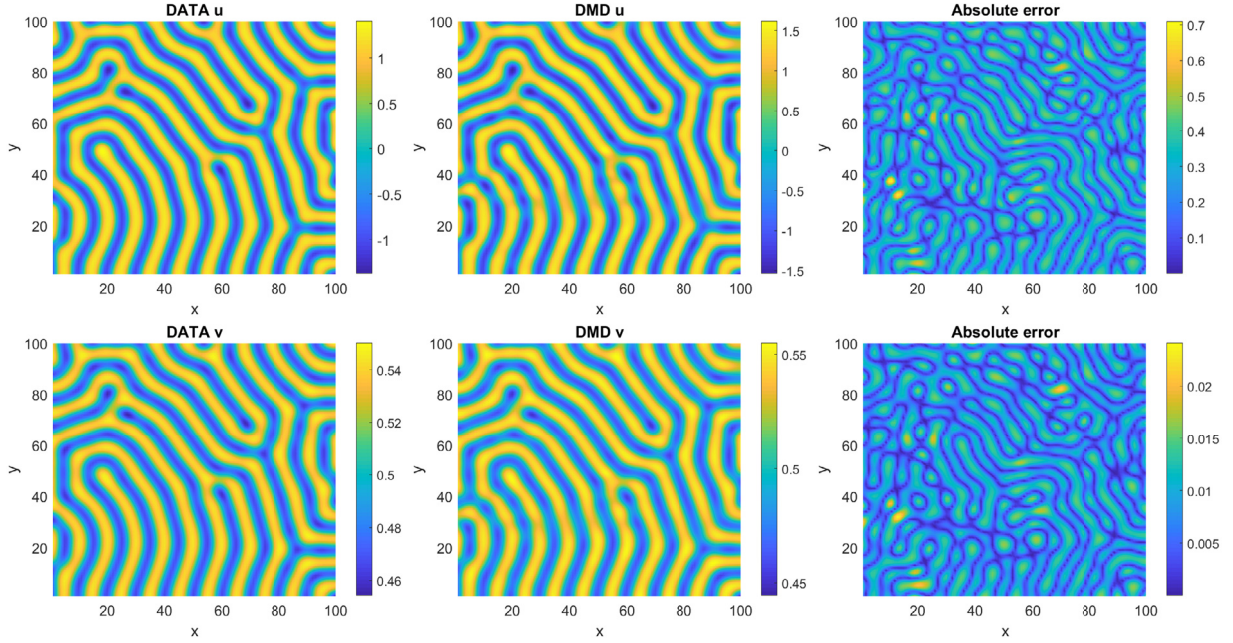


Fig. 9. DIB model: Turing instability. Full model solutions at $T = 40$ (left) and DMD reconstruction (center) with $r = 22$, and absolute errors of the DMD reconstruction with respect to data (right).

The initial conditions are again spatially random perturbation of the homogeneous equilibrium, as in the previous section.

We discretize the rectangular spatial domain $\Omega = [0, 100] \times [0, 70]$ with $n_x = n_y = 100$ meshpoints, thus $n = n_x n_y = 10000$. We integrate in time again by the IMEX-Euler in matrix oriented form with timestep $h_t = 10^{-4}$ and final time $T = 4.5$. We emphasize that, to the best of author's knowledge, the IMEX Euler scheme in the matrix form has never been applied to this kind of problems. This approach reduces significantly the computational execution time with respect to a standard vector form, therefore it allows to speed-up the offline stage, that is the construction of the dataset S . We save the snapshots every four timesteps ($\kappa = 4$, see discussion in Section 3), such that the dataset is $S \in \mathbb{R}^{2n \times (m+1)}$ with $m+1 = 11250$.

In the top left panel of Fig. 10, we show the relative error $\mathcal{E}(\tilde{S}, r)$ for $r = 1, \dots, R$, where $R = 130$ is the rank of the snapshot matrix S . We can observe that the behavior is very erratic and the minimum is $\mathcal{E}(\tilde{S}, 102) = 0.0111$, reached for $r = 102$. Different time dynamics of the spatial means are obtained, as shown in the bottom plots of Fig. 10, for the data u (left) and for DMD reconstruction with $r = 102$ (right). In the top right picture we report a zoom of both spatial means over $0 < t < 0.5$ to show that DMD does not match the mean of the dataset also in the transient regime. Instead for $t \gtrsim 1$, DMD catches the frequency but

not the amplitude of the spatial mean. To further emphasize the above significant difference, in Fig. 11 we also report the corresponding limit cycles obtained in the phase plane ($\langle u \rangle, \langle v \rangle$) for the data (left plot) and DMD reconstruction always for $r = 102$ (right plot).

To complete the discussion on this example, where the more complicated spatio-temporal oscillating dynamics is presented, in Fig. 12 we report the pattern solutions at the final time $T = 4.5$: the left panels concern the full model solutions u and v , the middle ones are for the corresponding DMD reconstructions. As expected from the previous results on the temporal dynamics, DMD does not approximate accurately also the final patterns both for u and v , as confirmed quantitatively from the absolute errors with respect to the data shown in the right panels.

5. The piecewise DMD method

In the previous section, we have shown a selection of examples where DMD fails its reconstruction during the time dynamics and where, in addition, the relative error with respect to the dataset increases when the rank does. It is worth remarking that we have tested the Higher Order DMD (HODMD, [12,13]) on the same examples without any improvement on the quality of the DMD approximation. In this section, we propose a new approach to tackle safely the spatio-

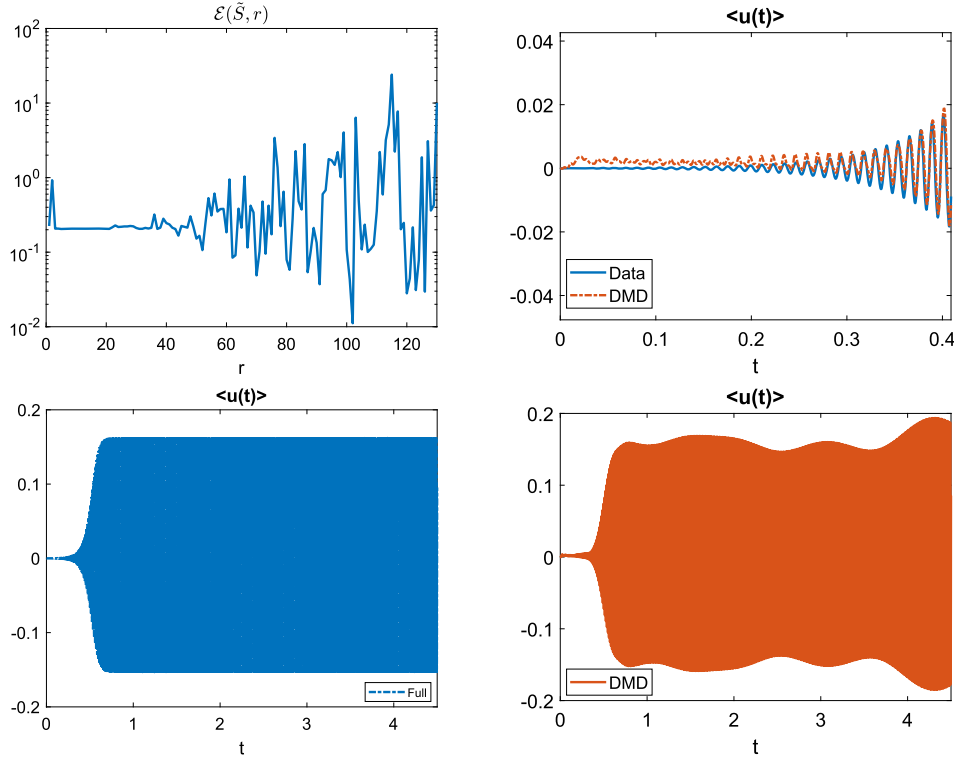


Fig. 10. DIB model: Turing-Hopf instability. Top left plot: DMD relative error in (8) for the dataset S . Top right plot: zoom on the spatial means for short times. Bottom left plot: spatial mean for the full model solution u ; bottom right plot: spatial mean for the DMD reconstruction with $r = 102$ (i.e. minimum error in the top left plot).

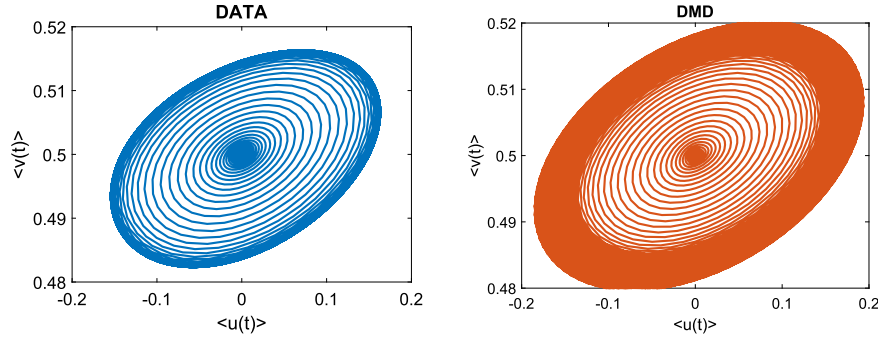


Fig. 11. DIB model: Turing-Hopf instability. Comparison of spatial means in the phase plane for data (left) and DMD with $r = 102$ (right).

temporal features of the peculiar solution dynamics discussed so far, that is datasets with oscillating behaviors and pattern formation by Turing instability.

The main idea here is to propose a *piecewise* version of the DMD algorithm, that we will denote by pDMD. Indeed, we suppose that for oscillatory and Turing spatio-temporal dynamics, the main assumption underlying the original DMD, that is a “global” linear fitting over the full temporal horizon, is not sufficient to recognize different “phenomena” arising along the time pathways. Therefore, instead of performing a DMD approximation on the whole time interval $[0, T]$, we propose to decompose it and the corresponding dataset into $N \geq 1$ parts, as follows.

Consider $v = \lceil \frac{m+1}{N} \rceil \geq v^*$ and the dataset decomposition $S = [S_1, \dots, S_i, \dots, S_N]$, where S_i is the submatrix of v columns of S defined by $S_i = [S_{:, (i-1)v+1}, \dots, S_{:, iv}] \in \mathbb{R}^{2n \times v}$ for $i = 1, \dots, N$. In practice, S_i corresponds to consider those snapshots of S belonging to the time interval $[t_{(i-1)v}, t_{iv-1}]$.

We suppose that $v^* \geq 10$, such that a minimum number of snapshots in each submatrix S_i is guaranteed and a maximum value for N can

be chosen. Then, we apply the DMD technique of rank r_i , using Algorithm 2, on each submatrix $S_i, i = 1, \dots, N$, with the convention that for $N = 1$, we recover the original dataset $S \equiv S_1$. In Algorithm 3 below, we present in details the pDMD, returning in output not only the reconstructed snapshots, but also the vector $\mathbf{r} = [r_1, r_2, \dots, r_N] \in \mathbb{R}^N$ accounting for the ranks considered on each dataset S_i . We can suppose to fix *a priori* the values in $\mathbf{r}$ or to estimate them.

Our idea stems from the “divide and conquer” approach well known in the numerical analysis framework to reduce the “global error” in the approximation under exam, as, for example, the main idea behind piecewise interpolation and composite quadrature rules. Then, due to the basic meaning of the DMD, recalled in Section 2, in each subinterval $[t_{(i-1)v}, t_{iv-1}]$, for $i = 1, \dots, N$ we will get the best linear fit of the form (1) on the dataset portion therein. Hence, this local linearization can help to look at the different solution regimes along time by applying multiple separated linear fittings. Indeed, we will show that local/piecewise linear fit can follow better the switches between these regimes instead of the classical approach, recovered for $N = 1$, that could miss them.

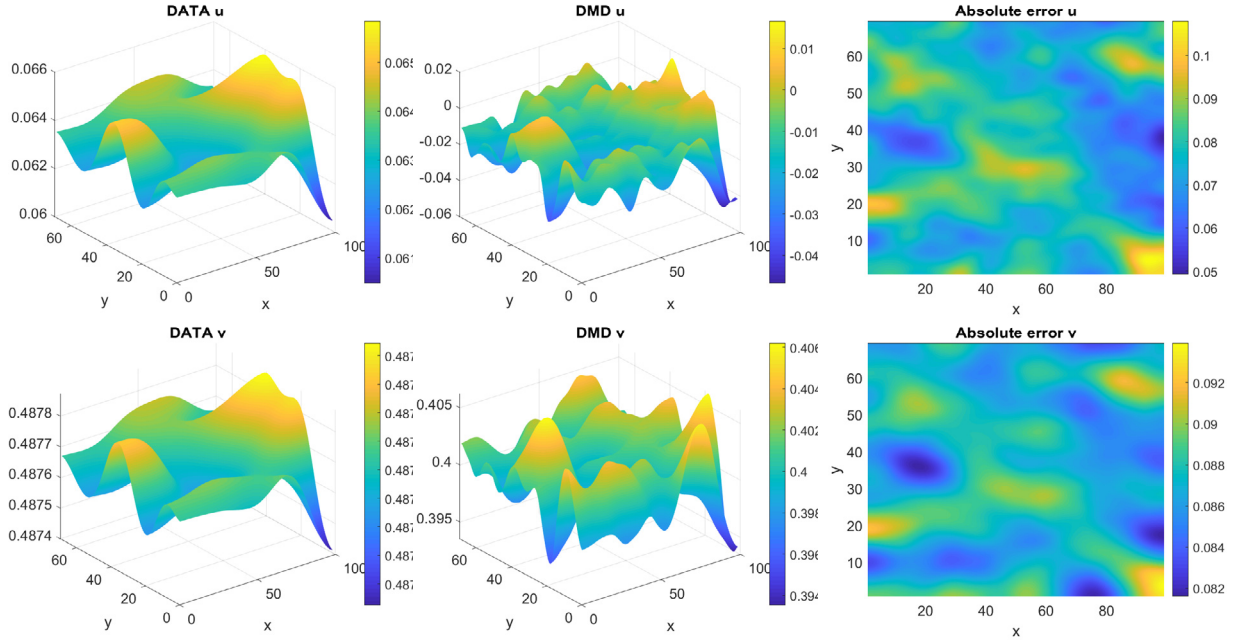


Fig. 12. DIB model: Turing-Hopf instability. Full model solutions (left panels) and DMD reconstruction (center panels) with $r = 102$. Right plots: absolute error of the DMD reconstruction with respect to data.

We summarize the proposed method pDMD in Algorithm 3 and we comment it step by step.

Algorithm 3 Piecewise DMD (pDMD).

- 1: **INPUT** Dataset $S \in \mathbb{R}^{2n \times (m+1)}$ in $[0, T]$, threshold $\overline{\text{tol}} > 0$
 - 2: **OUTPUT** $\tilde{S}^N$ piecewise reconstruction in $[0, T]$, $\mathbf{r}$ vector of ranks for the final partition
 - 3: Choose an initial number N of partitions
 - 4: Split the dataset $S = [S_1, \dots, S_i, \dots, S_N]$
 - 5: **for** $i = 1, \dots, N$ **do**
 - 6: set the target rank $r_i = \text{rank}(S_i)$
 - 7: compute the (randomized) DMD solution $\tilde{S}_i$ of rank r_i using Algorithm 2
 - 8: compute the error $\text{err}(i)$ defined in (14)
 - 9: **if** $\text{err}(i) > \overline{\text{tol}}$ **then**
 - 10: $N = N + 1$
 - 11: go to step 4
 - 12: **end if**
 - 13: **end for**
 - 14: $\tilde{S}^N = [\tilde{S}_1, \dots, \tilde{S}_i, \dots, \tilde{S}_N]$, $\mathbf{r} = [r_1, \dots, r_N]$.
-

Inputs. The inputs of the algorithm are the dataset S and a desired threshold $\overline{\text{tol}} > 0$ for the pDMD.

Initialization. We choose an initial value N to construct the partition of the dataset. One can easily start with $N = 1$, which corresponds to the standard DMD discussed in Section 4. We then split the dataset $S \in \mathbb{R}^{2n \times (m+1)}$ in N parts and build all the matrices $S_i \in \mathbb{R}^{2n \times v}$, $i = 1, \dots, N$, $v = \lceil \frac{m+1}{N} \rceil$.

pDMD. For each matrix S_i we fix the rank r_i using the QB decomposition in Algorithm 2. We then compute the following relative error

$$\text{err}(i) = \max_{(i-1)v+1 \leq k \leq iv} \frac{\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|_\infty}{\|\mathbf{x}_k\|_\infty} \quad (14)$$

that computes the worst approximation in each interval $[t_{(i-1)v+1}, t_{v+1}]$. This error is faster to compute than (8). It is worth noting that the computation of (14) is always possible since we are dealing with DMD reconstruction on the training dataset which is the focus of this work.

If $\text{err}(\tilde{i}) > \overline{\text{tol}}$ for some $\tilde{i} \in \{1, \dots, N\}$, we do not compute DMD for $i > \tilde{i}$, but we directly increase the value of N and restart the method with a new finer partition. That error indicator is, indeed, meant to save computational time whenever possible. In step 10 of Algorithm 3, we propose to increase N by 1, but clearly other choices can be used such as e.g. $N = N + \tilde{N}$ with $\tilde{N} \in \mathbb{N}$. Note that $\tilde{N} = 1$ corresponds to the choice in Algorithm 3. We iterate until the desired convergence is reached.

Output. The Algorithm returns the pDMD reconstruction $\tilde{S}^N$ that is the union of the DMD matrices $\tilde{S}_i$, for $i = 1, \dots, N$ and the ranks $\mathbf{r} = [r_1, \dots, r_N]$ used in each subinterval of the partition.

Remark 5.1 (Choice of the rank). It is important to note that the rank values in this algorithm plays a crucial role. One can always assume to work with a full rank approximation in each S_i , as set in Algorithm 3 but this might be computationally expensive, especially when m is large and N is still small. In some simulations, to avoid the computation of the rank for large matrices, we choose the target rank in step 6 of Algorithm 3 as $r_i = \min\{v, 200\}$, where v is the number of snapshots in S_i . One could also use randomized rank revealing methods as proposed in e.g. [33,34], but this is out of the scope of this paper.

6. Numerical experiments based on the piecewise DMD method

In this section, we apply the proposed pDMD Algorithm 3 to the datasets generated from the RD-PDE models presented in Section 4 where several drawbacks of the classical DMD have been discussed. For each test, we will show different error indicators, as follows. First of all, we consider the relative error $\mathcal{E}_p$ in Frobenius norm between the dataset S and its piecewise DMD reconstruction $\tilde{S}^N$ defined by:

$$\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) = \frac{\|S - \tilde{S}^N\|_F}{\|S\|_F}, \quad (15)$$

depending on the number N used to split the whole dataset S and $\mathbf{r} = [r_1, r_2, \dots, r_N] \in \mathbb{R}^N$ the vector of all ranks considered, such that r_i is used for the subset S_i . We observe that for $N = 1$ we recover the error corresponding to the “global” DMD approach, i.e. $\mathcal{E}_p(\tilde{S}^1, r_1) = \mathcal{E}(\tilde{S}, r_1)$ defined in (8), where r_1 is the number of degrees of freedom.

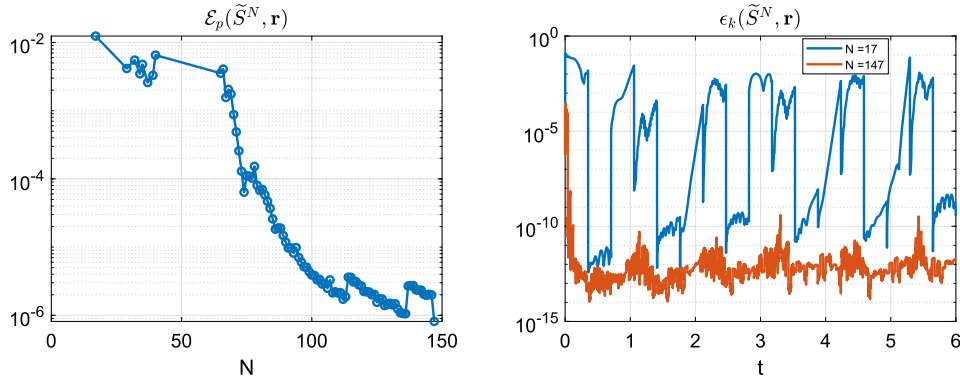


Fig. 13. FHN model. Left plot: relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ in (15) for increasing values of the partition size N . Right plot: relative error $\epsilon_k(\tilde{S}^N, \mathbf{r})$ over time t in (17) for N corresponding to the minimum ($N = 147$) and maximum ($N = 17$) errors in the left plot.

For the pDMD, i.e. $N > 1$, we define the number of degrees of freedom as follows

$$\text{dof}(N) = \|\mathbf{r}\|_1 = \sum_{i=1}^N r_i. \quad (16)$$

In our numerical tests we will provide the behavior of the error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ of the pDMD method for different choices of N , that are obtained by the inner computations of the following Algorithm 4, that stops when $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) \leq \text{tol}$ for a given input threshold $\text{tol} > 0$.

Algorithm 4 Stopping criterion for pDMD.

- 1: **INPUT** Dataset $S \in \mathbb{R}^{2n \times (m+1)}$ in $[0, T]$, threshold $\text{tol} > 0$,
 - 2: **OUTPUT** $\tilde{S}^N$ piecewise reconstruction in $[0, T]$, $\mathbf{r}$ the ranks used along the partition, the error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$
 - 3: Choose an initial number N of subintervals
 - 4: Split the datasets $S = \bigcup_{i=1}^N S_i$
 - 5: **while** $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) > \text{tol}$ **do**
 - 6: compute $\tilde{S}^N$ and $\mathbf{r}$ from Algorithm 3
 - 7: $N = N+1$
 - 8: **end while**
 - 9: $\tilde{S}^N = [\tilde{S}_1, \dots, \tilde{S}_N]$, $\mathbf{r} = [r_1, \dots, r_N]$, $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$
-

As second indicator, for the values of N identified by Algorithm 4, we will also compute the relative error in Euclidean norm over time calculated by:

$$\epsilon_k(\tilde{S}^N, \mathbf{r}) = \frac{\|\mathbf{x}_k - \tilde{\mathbf{x}}_k\|_2}{\|\mathbf{x}_k\|_2}, \quad k = 0, \dots, m \quad (17)$$

where $\{\tilde{\mathbf{x}}_k\}_{k=0}^m$ are the snapshots reconstructed by the pDMD with N subintervals.

We recall that the target rank value r_i for the DMD reconstruction is fixed on each subset $S_i \in \mathbb{R}^{2n \times v}$ and it depends on N , because each $r_i \leq v = \lceil \frac{m+1}{N} \rceil$. Of course, the overall computational load of the pDMD can depend on how large are the r_i values used in the algorithm and also on the total number of degrees of freedom required and calculated by (16). For this reason, in the next simulations we will visualize: i) for a fixed N , the target rank vector $\mathbf{r} \in \mathbb{R}^N$ as a measure of complexity along the subsets $S_i, i = 1, \dots, N$ (that is on the time subintervals of the piecewise technique) and ii) the maximum rank

$$\tilde{r}(N) = \max_{i=1, \dots, N} r_i = \|\mathbf{r}\|_\infty \quad (18)$$

needed by pDMD by varying the partition size N of the original dataset S until the optimal value identified by the Algorithm 4. We will also discuss the memory occupation cost for the pDMD computed as

$$\mathcal{M}(N) = \frac{\text{dof}(N)}{2n} = \frac{\sum_{i=1}^N r_i}{2n}, \quad (19)$$

that is the ratio between the total number of degrees of freedom introduced in (16) and the spatial dimension of the snapshots.

Remark 6.1. In Algorithm 4, we suggest to increase N only by one (line 7). This is mainly done to show our study exhaustively. Clearly, one can choose a larger increment to reduce the computational costs.

6.1. FitzHugh-Nagumo model

In this section, we apply the pDMD to the subset S generated by the FitzHugh-Nagumo model introduced in Section 4.1.1. To start the Algorithm 4, we consider the thresholds $\overline{\text{tol}} = 10^{-1}$, $\text{tol} = 10^{-6}$ and $N = 1$.

In the left panel of Fig. 13, we show the behavior of the relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ with respect to N . The plot starts from $N = 17$ because this is the first N value that satisfies the condition $\text{err}(i) \leq \overline{\text{tol}}$ for all $i = 1, \dots, N$, see step 9 of Algorithm 3. We note that the value of $\overline{\text{tol}}$ in this example is a rather mild request. We opt for this choice to show a more complete history of the error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ in Fig. 13, left. Indeed, by choosing for instance $\overline{\text{tol}} = 10^{-3}$, $N = 87$ would have been the first acceptable value. By incrementing the partition size N by one, Algorithm 4 stops for $N = 147$ with stopping criterion $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) < \text{tol}$. We can observe that the relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ is almost decreasing with respect to N , although there are few little jumps, still remaining in the same order of magnitude. We remark that for $N = 147$, the datasets in the partition have dimension $v = 41$, whereas for $N = 17$, $v = 353$.

In the right panel of Fig. 13, we show how the relative error (17) changes in time for $N = 17$ and $N = 147$, corresponding to the maximum and minimum of $\mathcal{E}_p$, that is $\mathcal{E}_p(\tilde{S}^{17}, \mathbf{r}) = 0.0124$ and $\mathcal{E}_p(\tilde{S}^{147}, \mathbf{r}) = 8.1308 \times 10^{-7}$, respectively. In the right plot, we can appreciate that for larger N the error uniformly decreases, especially in the peaks. Then, we compare the spatial mean (9) for the variables u and v obtained by the pDMD reconstruction with $N = 147$, with respect to the data. The results are shown in Fig. 14, left and middle panels, respectively. It is evident that pDMD carefully matches the data, as also confirmed looking at the phase plane in the right panel.

Finally, in the left plot of Fig. 15, we report the target rank vectors $\mathbf{r}$ for $N = 17$ and 147 , to show how the pDMD ranks r_i change in each subset S_i . As post-processing, to study the computational load of pDMD, in the right plot of Fig. 15 we report $\tilde{r}(N)$ calculated by (18). It is easy to see that, this value decreases with N and the maximum rank used $\tilde{r}(17) = 29$ is smaller than $\text{rank}(S) = 51$ for the original dataset. Hence, this indicates that pDMD is also convenient from the computational point of view since we deal with problems of significant small size. Indeed, the memory cost $\mathcal{M}(147) = 821/2048$ corresponds to 40% reduction of the spatial dimension of the snapshot matrix S and the CPU time employed by pDMD is 0.22 s. We remark that the original dataset for this model is already low dimensional and the reduction is

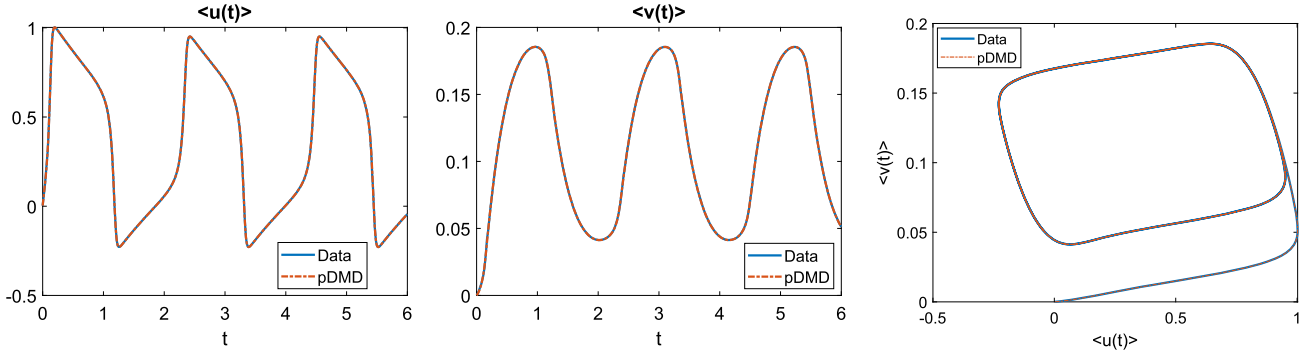


Fig. 14. FHN model. Comparison of the spatial mean for the variable u (left plot) and v (center plot). Right plot: phase plane for the DMD reconstruction with $N = 147$, for which the error has its minimum that is $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) = 8.1308 \times 10^{-7}$.

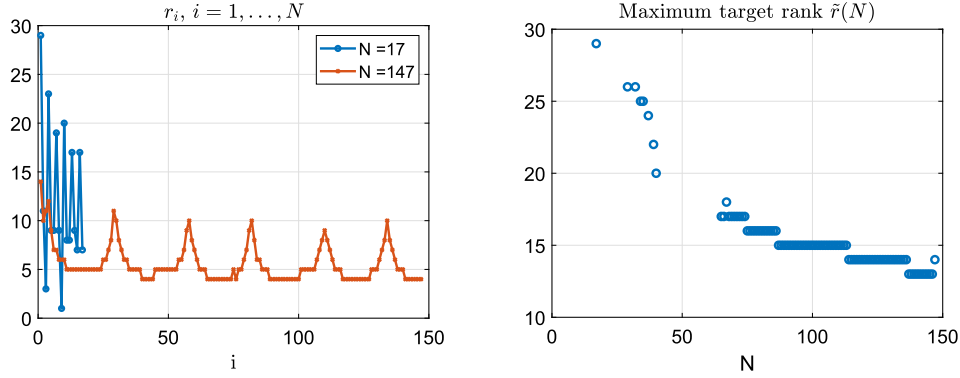


Fig. 15. FHN model. Left plot: vector $\mathbf{r} = [r_1, \dots, r_N]$ of target ranks used on the subsets S_i , $i = 1, \dots, N$, for $N = 17, 147$. Right plot: maximum target rank (18) used by pDMD with respect to N .

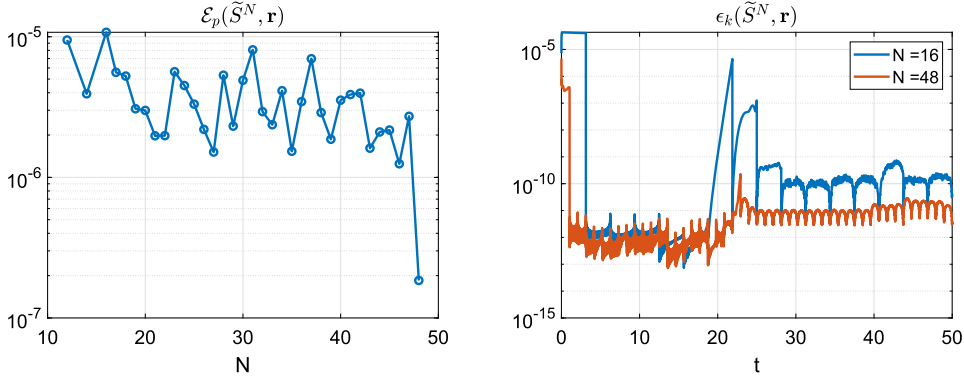


Fig. 16. λ - ω system, spiral wave. Left plot: relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ for $\overline{\text{tol}} = 10^{-2}$ in (15). Right plot: Relative error $\epsilon_k(\tilde{S}^N, \mathbf{r})$ over time in (17) for two significant values of N as discussed in the main text.

not powerful but still active. We will see more impressive reduction in the next examples.

6.2. λ - ω RD-PDE system

We apply pDMD to the same dataset S generated for the λ - ω system discussed in Section 4.1.2 and we consider $\overline{\text{tol}} = 10^{-2}$, $N = 1$ to start the Algorithm 3. In Fig. 16 (left panel), it is shown the error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ defined in (15) when the partition size N , i.e. number of submatrices S_i of S , is increased. For $N = 1$, corresponding to the “global” DMD case, as expected from our results in Section 4.1.2, the condition in step 9 of the algorithm is not satisfied and the first acceptable value is $N = 16$ which corresponds to $\nu = 1042$.

When $\text{tol} = 10^{-6}$, the Algorithm 4 stops for $N = 48$, that is $\nu = 261$. We can see that the error exhibits a slightly oscillating behavior for

$16 \leq N \leq 47$, such that $10^{-5} \leq \mathcal{E}_p(\tilde{S}^N, \mathbf{r}) \leq 10^{-6}$, that is the error remains within the same order of magnitude. As for the previous test, we consider the values of N for which the error has its maximum and minimum, that are $\mathcal{E}_p(\tilde{S}^{16}, \mathbf{r}) = 9.4858 \times 10^{-6}$ and $\mathcal{E}_p(\tilde{S}^{48}, \mathbf{r}) = 1.8478 \times 10^{-7}$. In any case, these error approximation levels are much lower than the best obtained by the “global” DMD in Section 4.1.2. In Fig. 16, right panel, we show the errors in time (17) for $N = 16$ and $N = 48$. We can observe in the error behavior two time regimes, corresponding to those of the spiral wave dynamics, discussed before in Section 4.1.2. Indeed, in both cases, the error rapidly decays immediately after the initial phase until $\tilde{t} \approx 25$, then an almost constant oscillating trend is present in $[\tilde{t}, T]$. We note also that for larger N the error uniformly decreases along all the interval $[0, T]$.

To further show how the pDMD overcomes the drawbacks of the original DMD highlighted in Section 4.1.2, we compare the time dy-

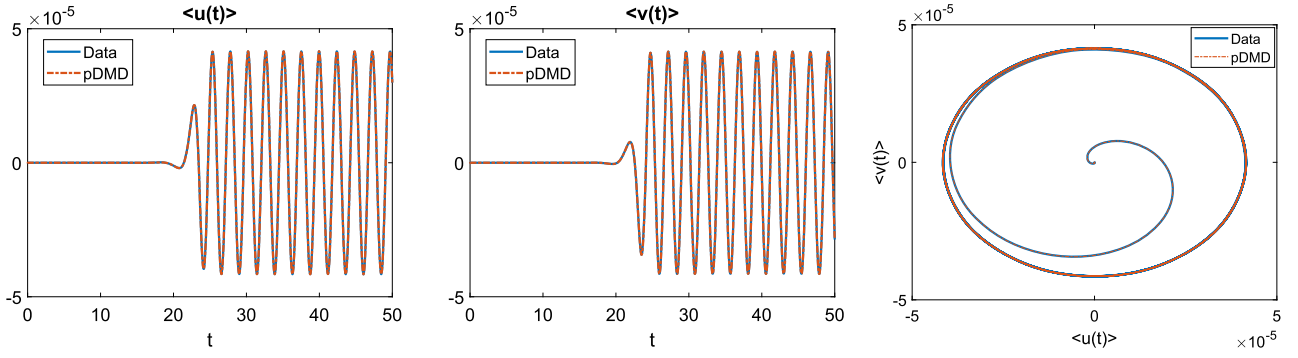


Fig. 17. λ - ω system, spiral waves. pDMD reconstruction of the spatial mean for the variable u (left plot) and v (center plot) with $N = 48$, for which the error has its minimum that is $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) = 1.8478 \times 10^{-7}$. Corresponding limit cycle in the phase plane is shown in the right plot, a very good agreement is obtained with respect to the best approximation by the original DMD, see Section 4.1.2 and Fig. 7.

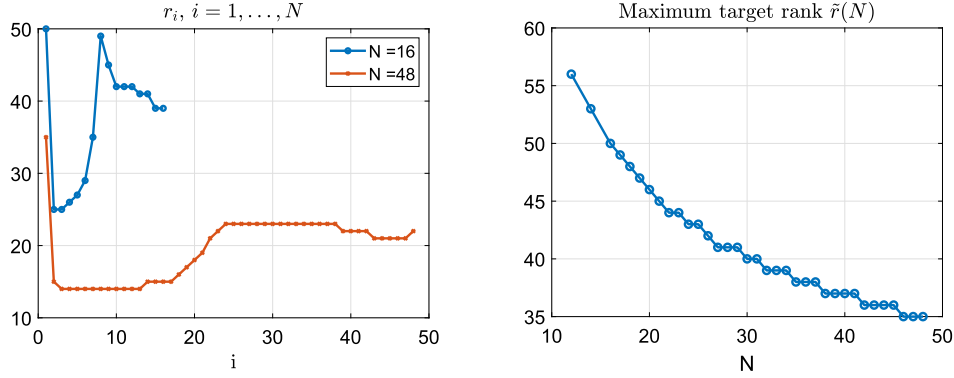


Fig. 18. λ - ω system, spiral waves. Left plot: vector $\mathbf{r} = [r_1, \dots, r_N]$ of target ranks used in the subinterval S_i , by pDMD for $N = 16, 48$. Right plot: maximum target rank $\tilde{r}(N)$ in (18) used by pDMD with respect to N .

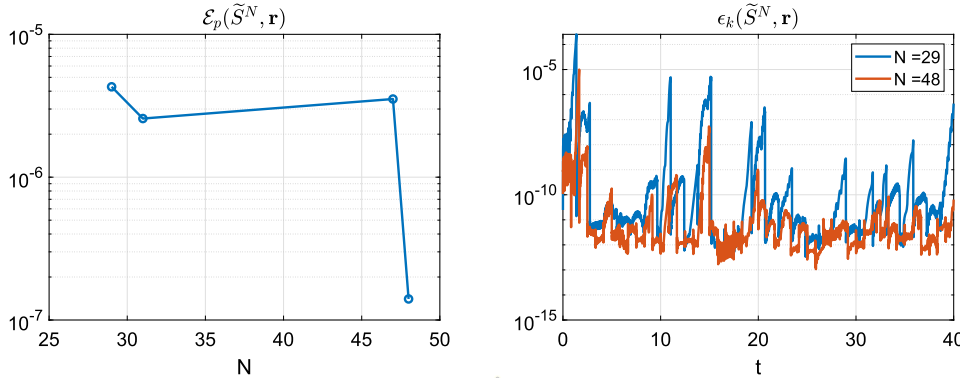


Fig. 19. DIB model: Turing instability. Left plot: relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ in (15). Relative error $\epsilon_k(\tilde{S}^N, \mathbf{r})$ over time in (17) for two meaningful values of N , as discussed in the main text.

namics of the spatial means $\langle u \rangle$ and $\langle v \rangle$ for $N = 48$ with respect to the data in Fig. 17 for u (left plot) and v (middle plot). As for the FHN model, the oscillating time dynamics obtained by pDMD matches perfectly the dataset, as also confirmed looking at the reconstructed limit cycle in the right panel.

Finally, in the left panel of Fig. 18, we show the target rank vectors $\mathbf{r}$ for $N = 16$ and 48 , to show how the pDMD ranks r_i changes along the subsets S_i . We note that, for each N , the largest value of the rank is always required in the first subset, but in general lower r_i values are required for larger N . This result is also confirmed by the behavior of the maximum $\tilde{r}(N)$, shown in Fig. 18, right plot, that monotonically decays with respect to N . Moreover, it is worth noting that the maximum rank needed by pDMD is $\tilde{r}(16) = 56$ which is much smaller than the rank of the original dataset S , that is 314. The memory cost

$\mathcal{M}(48) = 937/19602$ corresponds to about 4% of the spatial memory occupation of the original snapshot matrix. The CPU time employed for $N = 48$ is 2.7 s. For this example, we can see how pDMD reduces significantly the original problem obtaining the desired accuracy.

6.3. DIB model: Turing instability

In this section, we apply the pDMD to reconstruct the Turing instability dynamics of the morphochemical DIB model discussed in Section 4.2. We consider $\text{tol} = 10^{-3}$ and $N = 1$ as input of the pDMD Algorithm 3.

In the left panel of Fig. 19 we show the relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ defined in (15) for increasing values of N until it is less than $\text{tol} = 10^{-6}$ in Algorithm 4, reached for $N = 48$. The first acceptable value is $N =$

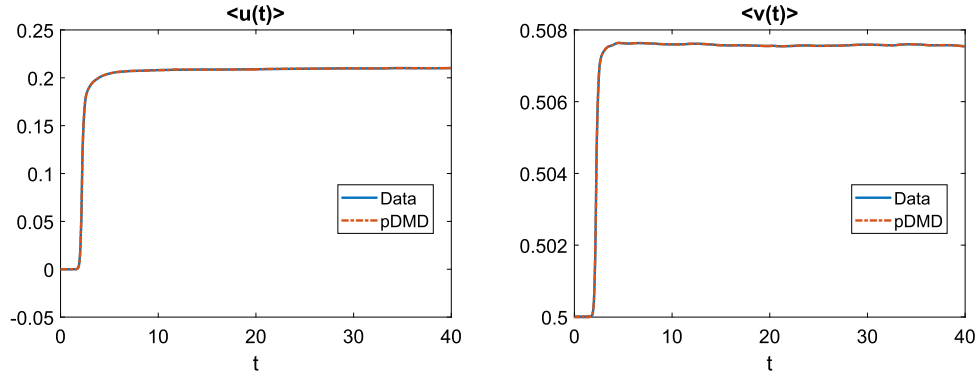


Fig. 20. DIB model: Turing instability. Spatial mean of the variables u (left plot) and v (right plot) for the pDMD reconstruction with $N = 48$ compared with that of the data.

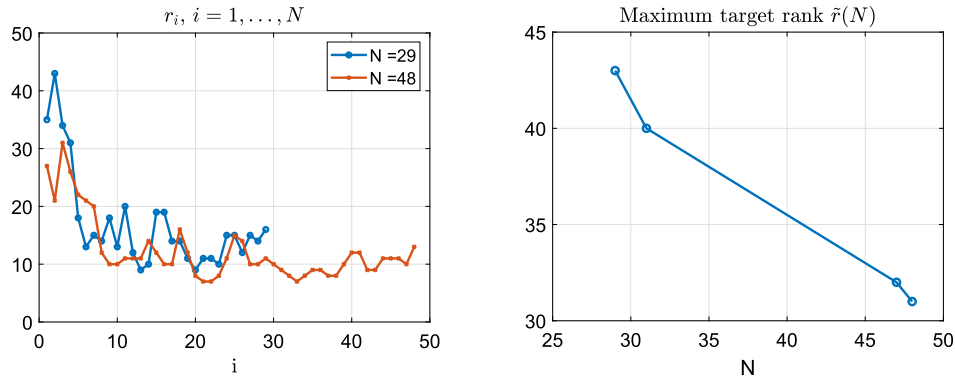


Fig. 21. DIB model: Turing instability. Left plot: vector $\mathbf{r} = [r_1, \dots, r_N]$ of target ranks used in the subsets S_i , $i = 1, \dots, N$, for $N = 29, 48$. Right plot: maximum target rank $\tilde{r}(N)$ in (18) required by pDMD with respect to N .

29, but for $N < 48$ some symbols are missing because for that values of N there exists a subset $S_{\tilde{i}}$ such that $\text{err}(\tilde{i}) > \text{tol}$. We remark that for $N = 29$, each subset has dimension $\nu = 345$ while for $N = 48$, $\nu = 209$ holds.

In the right panel of Fig. 19, we show how the pDMD error (17) evolves in time for $N = 29$ and $N = 48$, corresponding to the maximum and minimum value attained in Fig. 19, left plot, given by $\mathcal{E}_p(\tilde{S}^{29}, \mathbf{r}) = 4.2932 \times 10^{-6}$ and $\mathcal{E}_p(\tilde{S}^{48}, \mathbf{r}) = 1.4039 \times 10^{-7}$, respectively. We note that the maximum error is essentially concentrated in the first zone, that is in the reactivity Turing regime, and that for larger N it decreases almost uniformly with respect to time along $[0, T]$.

To further confirm this trend, in Fig. 20, we show the comparison of the spatial means for the variables u (left panel) and v (right panel) obtained by the pDMD reconstruction with $N = 48$ with respect to the data. The time dynamics of the pDMD solution for all times matches the spatial means of the data (compare with the right plot in Fig. 8).

We emphasize that, for the reconstruction of the Turing instability dynamics the piecewise approach not only is able to remove the ill-conditioning in the “global” DMD (see Fig. 8, left), but also it is able to adapt the choice of the “local” target ranks to the peculiar dynamics along time. For this reason, we report how the rank r_i in each subset S_i changes for $i = 1, \dots, N$ for the previous partitions with $N = 29$ and $N = 48$. The left plot in Fig. 21 highlights that, for both N , in the initial reactivity zone we need to choose higher values of the rank, whereas in the stabilizing zone significantly lower values are required, such that $r_i \leq 20$. In the right plot of Fig. 21, we report the maximum target ranks $\tilde{r}(N)$ for the values of N in Fig. 19(left) (that are then attained in the initial part of the time interval). Also in this case we observe a monotone decay with respect to N and the maximum $\tilde{r}(29) = 43$ is again much smaller than the rank of the original dataset S given by 303. Again the piecewise approach is also able to reduce the overall com-

putational load of the usual DMD implementation. Indeed, the memory cost $\mathcal{M}(48) = 592/10000$ corresponds to about 3% of the original spatial dimension of the snapshot matrix. The CPU time, employed for $N = 48$ is 1.9 s. Also for this example, we can see how pDMD reduces significantly the original problem obtaining the desired accuracy and with reasonable overall computational load in the online phase.

6.4. DIB model: Turing-Hopf instability

The last experiment concerns again the DIB morphochemical RD system but with a different choice of the model parameters that gives rise to the so-called Turing-Hopf instability. We recall from Section 4, that this is the most complicated dynamics considered in this paper, because the PDE solutions exhibit an initial instability and then the formation of a pattern oscillating both in space and time. We apply the pDMD to the same dataset S generated in Section 4 starting the Algorithm 3 with $N = 5$ and $\text{tol} = 10^{-3}$. We obtain the first useful partition for $N = 225$, that is $\nu = 50$. Then, we increment N by 5 and check the error (15) until $\text{tol} = 10^{-5}$ in Algorithm 4. The final value is $N = 535$ with $\nu = 21$ where $\mathcal{E}_p(\tilde{S}^{535}, \mathbf{r}) = 8.5031 \times 10^{-6}$. In Fig. 22 (left plot) we show that the error (15) slowly decreases for increasing N , even though there are small oscillations in a neighborhood of $N = 500$. In this case, it is clear that the dynamics is very complex to catch and many iterations are needed which means that submatrices S_i of low dimensions are required.

As for the previous numerical experiments, we consider the values of N for which the error $\mathcal{E}_p$ has its maximum and minimum, that is $N = 230$ where $\mathcal{E}_p(\tilde{S}^{230}, \mathbf{r}) = 2.6059 \times 10^{-4}$ and $N = 535$ where the pDMD stopped with $\mathcal{E}_p(\tilde{S}^{535}, \mathbf{r}) = 8.503 \times 10^{-6}$. In the right panel of Fig. 22, we show the behavior of the error $\epsilon_k(\tilde{S}^N, \mathbf{r})$ (see (17)) along the integration time interval of the PDE model. For both partition sizes N , this error is larger in the second part of the time interval that corresponds to

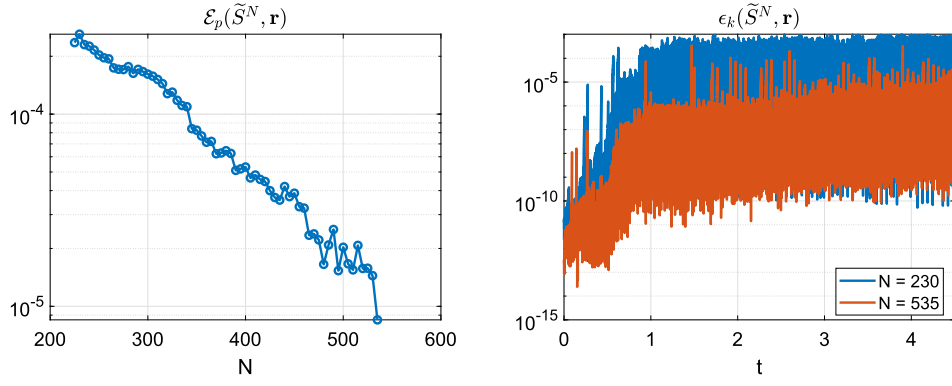


Fig. 22. DIB model: Turing-Hopf instability. Left plot: relative error $\mathcal{E}_p(\tilde{S}^N, \mathbf{r})$ in (15). Relative error $\epsilon_k(\tilde{S}^N, \mathbf{r})$ over time in (17) for two meaningful values of N .

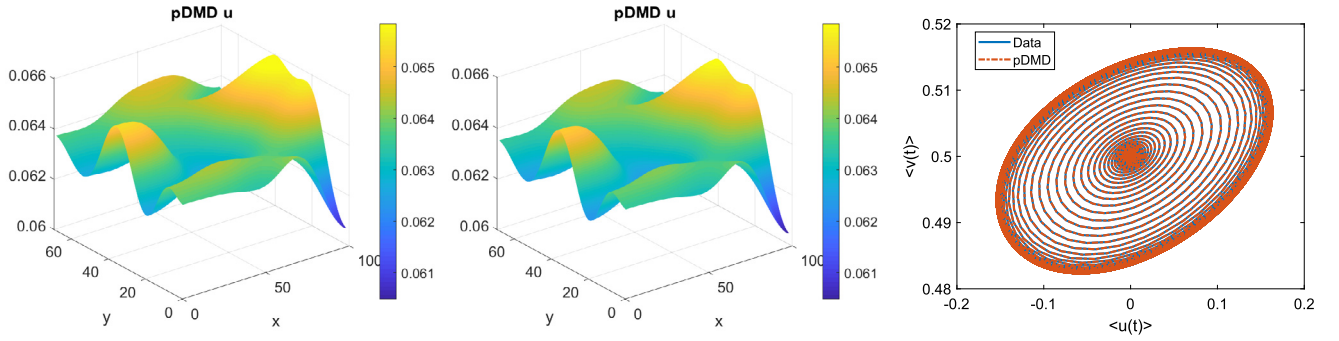


Fig. 23. DIB model: Turing-Hopf instability. pDMD reconstruction of the variable u at the final time $T=4.5$ for $N=230$ (left plot) and $N=535$ (center plot). Right plot: limit cycle for the pDMD solution with $N=535$ in the phase plane of the spatial means, for which the error has its minimum that is $\mathcal{E}_p(\tilde{S}^N, \mathbf{r}) = 8.5031 \times 10^{-6}$.

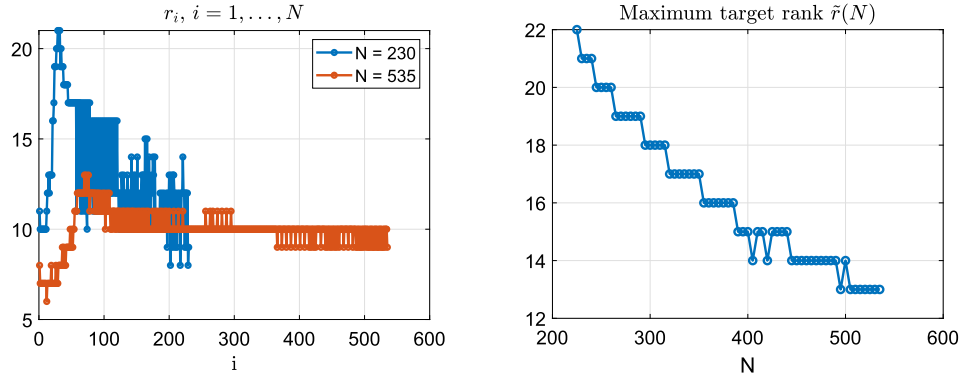


Fig. 24. DIB model: Turing-Hopf instability. Left plot: vector $\mathbf{r}$ of the target ranks r_i used in the subsets S_i , $i=1, \dots, N$, for $N=230, 535$. Right plot: maximum target rank $\tilde{r}(N)$ with respect to the range of partition sizes N identified in Fig. 22.

the oscillatory behavior of the spatial mean (see bottom left plot of Fig. 10), but it decreases on the whole time interval for the larger value $N=535$.

In Fig. 23, we compare the solutions obtained at the final time T for the u variable by applying the pDMD algorithm with $N=230$ (left plot) and $N=535$ (middle plot). The reconstruction with $N=535$ (middle plot) is almost the same as the data (see Fig. 12); for $N=230$ almost the same shape is reconstructed by pDMD (left plot), with very small different amplitudes, as evident by a slight different color distribution. Moreover, in Fig. 23 (right plot) we compare the time dynamics in the phase plane ($\langle u \rangle, \langle v \rangle$) of the spatial mean values obtained by pDMD for $N=535$ with respect to the data. We can observe that there is no evident difference between the limit cycles for pDMD and data.

Finally, in the right panel of Fig. 24, always for the partition sizes $N=230$ and $N=535$ we show the ranks r_i chosen in the subsets S_i , for $i=1, \dots, N$. In both cases, in the first part of the time interval, increasing values of the target rank are needed, then after a certain subset (or time, say $\tilde{r} \approx 1$ when the limit cycle is reached) smaller and smaller rank values are sufficient to follow the oscillatory spatio-temporal regime. Moreover, for larger N , that is $N=535$, smaller r_i are needed. To confirm this trend, in the right panel of Fig. 24 we show the behavior of the maximum target rank $\tilde{r}(N)$: it exhibits a decreasing trend and in the worst case, the maximum rank needed is $22 \ll 130 = \text{rank}(S)$, i.e. much smaller than the rank of the original dataset. In this example, the memory cost $\mathcal{M}(535) = 5326/11250$ corresponds to about 27% reduction of the original spatial dimension of the matrix S . The CPU time,

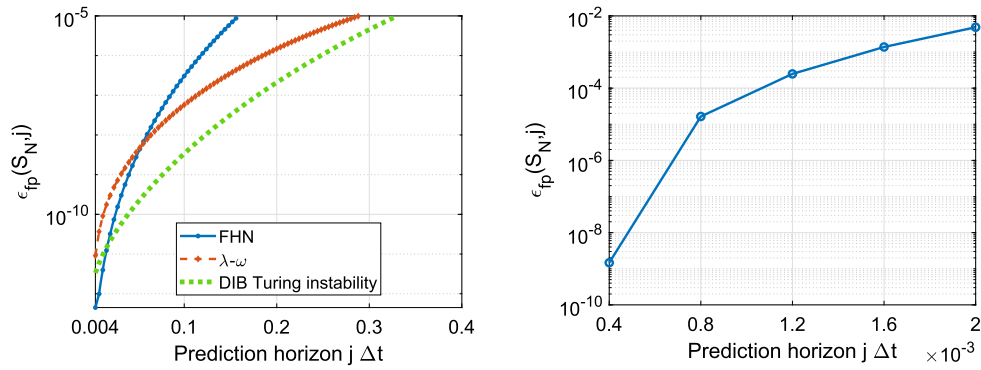


Fig. 25. Error prediction defined in (20). Left plot: FHN, $\lambda - \omega$, DIB model (Turing instability) with $\Delta t = 4 \cdot 10^{-3}$. Right plot: DIB model (Turing-Hopf instability) with $\Delta t = 4 \cdot 10^{-4}$.

related to $N = 535$, is 6 s. This example still provides an important reduction considering the peculiarity and complexity of the Turing-Hopf time evolution combining spatio-temporal oscillatory dynamics.

6.5. Future predictions for pDMD

One of the feature of the DMD algorithm, as common for data-driven techniques, is the capability to predict future state developments using a single model. On the contrary, the pDMD algorithm by using different models in each sub-interval, improves the accuracy with respect to the exact DMD. In this subsection, we briefly study how pDMD performs for future predictions. This, as for the global DMD, strongly depends on the studied data.

For future state predictions, we consider the last model computed by pDMD, that is the pDMD reconstruction corresponding to the snapshot submatrix S_N . Hence, we study the error behavior after the time interval $[0, T]$ as follows. Let us define:

$$\epsilon_{fp}(S_N, j) = \frac{\|\mathbf{x}_{j+m} - \tilde{\mathbf{x}}_{j+m}\|_2}{\|\mathbf{x}_{j+m}\|_2}, \quad \text{for } j = 1, \dots, \bar{j} \quad (20)$$

where $\mathbf{x}_{j+m}$ are the solutions taken from the full model, t_{j+m} for $j = 1, \dots, \bar{j}$ are the timesteps not included in the original dataset and t_m corresponds to the final time in the snapshot matrix S . Then, the final prediction horizon will be defined as $\bar{T} := \bar{j} \Delta t$ and $\bar{j}$ will be chosen such that $\epsilon_{fp}(S_N, j) < \text{tol}_j$ for a fixed desired accuracy $\text{tol}_j > 0$. In the left panel of Fig. 25, we show the error in (20) for $\text{tol}_j = 10^{-5}$ and $\Delta t = 4 \cdot 10^{-3}$. We can observe that pDMD well captures the dynamics beyond the timeframe for the oscillatory models and DIB model with Turing instability. The final prediction horizon is $\bar{T} = 0.156$ for FHN, $\bar{T} = 0.288$ for $\lambda - \omega$ system and $\bar{T} = 0.324$ for the DIB model with Turing instability. On the other hand, the error behavior for the DIB model with Turing-Hopf instability, with $\text{tol}_j = 10^{-2}$ and $\Delta t = 4 \cdot 10^{-4}$ stops early at $\bar{T} = 0.002$, as shown in Fig. 25, right, further confirming the complexity of this spatio-temporal dynamics. This brief study further validates our method, which not only it is able to reconstruct data with accurate results but it is also able to make future predictions for moderate time horizons.

7. Conclusions

Inspired by the classical “divide and conquer” principle, in this paper we have introduced a *piecewise* version of the exact DMD technique, called pDMD. Given a temporal dataset and fixed tolerances, the new approach is implemented in Algorithm 3 and Algorithm 4, when the partition size N of the original dataset is increased towards a desired final accuracy of the reconstruction. The new approach can be applied to a general dataset, even if our study has been motivated by the failure of the original exact DMD on snapshots describing peculiar spatio-temporal dynamics arising in Reaction-Diffusion (RD) PDE systems.

Indeed, in Section 4 we have shown that DMD exhibits very inaccurate reconstructions or ill-conditioning for large target ranks (where a better approximation is expected) for four significant models with: relaxation oscillations (FitzHugh-Nagumo 1D in space), spiral waves ($\lambda - \omega$ system), Turing pattern formation and Turing-Hopf patterns oscillating in space and time (DIB morphochemical system for battery modeling).

In Section 6, for each kind of the above dynamics, we have shown that pDMD is able to remove all drawbacks previously highlighted. In some cases, like the FitzHugh-Nagumo and Turing-Hopf dynamics, we have shown that a suitable partition size N^* can be obtained such that for $N \geq N^*$ an error much lower than the best obtained by the “global” DMD in Section 4. In other cases, like for spiral waves and Turing instability, both the final spiral/pattern and their time histories described by the limit cycle in the phase plane are carefully reconstructed for $N \geq N^*$. In all cases, pDMD is now able to follow the entire spatio-temporal dynamics, including different regimes (e.g. reactivity-stabilization for Turing, unstable-oscillating for spiral waves). In particular, in all simulations, for larger N the error uniformly decreases along all the time interval of the entire dataset and a convergence trend can be observed.

In all our simulations, a discussion about the computational load of the new algorithm, in terms of memory occupation and execution times in the online phase, has also been presented. We show how pDMD reduces the spatial dimension of the original dataset, although increasing the number of subproblems. Indeed, the maximum rank used by the DMDs in each subset is always much smaller than the rank of the original dataset S . Our work confirms that for some time dynamics, the “global” (exact) DMD does not reconstruct the dataset correctly, therefore it is necessary to make extra work, as for the pDMD, to achieve the desired accuracy. The linear fitting behind DMD is also at the origin of the observed ill-conditioning of the original DMD (see e.g. Figs. 5 and 8). Moreover, we have shown that pDMD is also able to produce future predictions for moderate time horizons for almost all the dynamics investigated in this work.

We can conclude that, for oscillatory and Turing spatio-temporal dynamics, the main assumption underlying the original DMD, that is a “global” linear fitting over the full temporal horizon, is not sufficient to recognize different “phenomena” arising along the time pathways. Instead, the “local” linear fitting by pDMD does it when a sufficient dataset partition size N can be identified.

Code availability

The MATLAB source code for the implementations to compute the presented results can be downloaded from <https://github.com/alessandroalla/pDMD> upon request to the corresponding author.

Data availability

Data will be made available on request.

Acknowledgements

AA, AM, IS are members of the INdAM-GNCS activity group. The work of A.A. was carried out within the “Data-driven discovery and control of multi-scale interacting artificial agent systems”, and received funding from the European Union Next-GenerationEU - National Recovery and Resilience Plan (NRRP) – MISSION 4 COMPONENT 2, INVESTIMENT 1.1 Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN) – Project Code P2022JC95T, CUP H53D23008920001-. The work of I.S. is supported by PRIN2022 PNRR “BAT-MEN” (BATtery Modeling, Experiments & Numerics), Project code P20228C2PP_001, CUP F53D23010020001, funded by MIUR (Italian Ministry of University and Research) and European Union – NextGenerationEU. This manuscript reflects only the authors’ views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

- [1] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci.* 113 (15) (2016) 3932–3937, <https://doi.org/10.1073/pnas.1517384113>.
- [2] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.* 3 (4) (2017) e1602614, <https://doi.org/10.1126/sciadv.1602614>.
- [3] S.H. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, *SIAM J. Appl. Dyn. Syst.* 18 (2) (2019) 643–660, <https://doi.org/10.1137/18M1191944>.
- [4] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707, <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [5] G. Karniadakis, I. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (2021) 422–440, <https://doi.org/10.1038/s42254-021-00314-5>.
- [6] J. Yu, L. Lu, X. Meng, G.E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, *Comput. Methods Appl. Mech. Eng.* 393 (2022) 114823, <https://doi.org/10.1016/j.cma.2022.114823>.
- [7] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.* 656 (2010) 5–28, <https://doi.org/10.1017/S0022112010001217>.
- [8] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S.L. Brunton, J.N. Kutz, On dynamic mode decomposition: theory and applications, *J. Comput. Dyn.* 1 (2) (2014) 391–421, <https://doi.org/10.3934/jcd.2014.1.391>.
- [9] P. Héas, C. Herzet, Optimal low-rank dynamic mode decomposition, in: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4456–4460.
- [10] T. Askham, J.N. Kutz, Variable projection methods for an optimized dynamic mode decomposition, *SIAM J. Appl. Dyn. Syst.* 17 (1) (2018) 380–416, <https://doi.org/10.1137/M1124176>.
- [11] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data-driven approximation of the Koopman operator: extending dynamic mode decomposition, *J. Nonlinear Sci.* 25 (18) (2015) 1307–1346, <https://doi.org/10.1007/s00332-015-9258-5>.
- [12] S. Le Clainche, J.M. Vega, Higher order dynamic mode decomposition, *SIAM J. Appl. Dyn. Syst.* 16 (2) (2017) 882–925, <https://doi.org/10.1137/15M1054924>.
- [13] S. Le Clainche, J.M. Vega, Higher order dynamic mode decomposition to identify and extrapolate flow patterns, *Phys. Fluids* 29 (8) (2017) 084102, <https://doi.org/10.1063/1.4997206>.
- [14] A. Viguerie, G.F. Barros, M. Grave, A. Realí, A.L. Coutinho, Coupled and uncoupled dynamic mode decomposition in multi-compartmental systems with applications to epidemiological and additive manufacturing problems, *Comput. Methods Appl. Mech. Eng.* 391 (2022) 114600, <https://doi.org/10.1016/j.cma.2022.114600>.
- [15] B. Bozzini, A. Monti, I. Sgura, Model-reduction techniques for PDE models with Turing type electrochemical phase formation dynamics, *Appl. Eng. Sci.* 8 (2021) 1–10, <https://doi.org/10.1016/j.apples.2021.100074>.
- [16] N.B. Erichson, L. Mathelin, J.N. Kutz, S.L. Brunton, Randomized dynamic mode decomposition, *SIAM J. Appl. Dyn. Syst.* 18 (4) (2019) 1867–1891, <https://doi.org/10.1137/18M1215013>.
- [17] J.N. Kutz, X. Fu, S.L. Brunton, Multiresolution dynamic mode decomposition, *SIAM J. Appl. Dyn. Syst.* 15 (2) (2016) 713–735, <https://doi.org/10.1137/15M1023543>.
- [18] H. Zhang, C.W. Rowley, E.A. Deem, L.N. Cattafesta, Online dynamic mode decomposition for time-varying systems, *SIAM J. Appl. Dyn. Syst.* 18 (3) (2019) 1586–1609, <https://doi.org/10.1137/18M1192329>.
- [19] S. Chaturantabut, D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.* 32 (2010) 2737–2764, <https://doi.org/10.1137/09076649>.
- [20] H. Müller, S. Volkwein, Model reduction by proper orthogonal decomposition for lambda-omega systems, in: P. Wesseling, E. Onate, J. Periaux (Eds.), *Proceedings of European Conference on Computational Fluid Dynamics (ECCOMAS CFD)*, Egmont aan Zee, 2006, 2006.
- [21] A. Borzi, R. Griesse, Distributed optimal control of lambda-omega systems, *J. Numer. Math.* 14 (1) (2006) 17–40, <https://doi.org/10.1515/156939506776382120>.
- [22] B. Bozzini, D. Lacitignola, I. Sgura, Spatio-temporal organization in alloy electrodeposition: a morphochemical mathematical model and its experimental validation, *J. Solid State Electrochem.* 17 (2) (2013) 467–479, <https://doi.org/10.1007/s10008-012-1945-7>.
- [23] D. Lacitignola, B. Bozzini, I. Sgura, Spatio-temporal organization in a morphochemical electrodeposition model: Hopf and Turing instabilities and their interplay, *Eur. J. Appl. Math.* 26 (2) (2015) 143–173, <https://doi.org/10.1017/S0956792514000370>.
- [24] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Rev.* 53 (2) (2011) 217–288, <https://doi.org/10.1137/090771806>.
- [25] J. Murray, *Mathematical Biology II - Spatial Models and Biomedical Applications, Interdisciplinary Applied Mathematics*, vol. 18, Springer-Verlag, Berlin, Heidelberg, 2003.
- [26] D. Lacitignola, B. Bozzini, M. Frittelli, I. Sgura, Turing pattern formation on the sphere for a morphochemical reaction-diffusion model for electrodeposition, *Commun. Nonlinear Sci. Numer. Simul.* 48 (2017) 484–508, <https://doi.org/10.1016/j.cnsns.2017.01.008>.
- [27] I. Sgura, A. Lawless, B. Bozzini, Parameter estimation for a morphochemical reaction-diffusion model of electrochemical pattern formation, *Inverse Probl. Sci. Eng.* 27 (2019) 618–647, <https://doi.org/10.1080/17415977.2018.1490278>.
- [28] G. Settanni, I. Sgura, Devising efficient numerical methods for oscillating patterns in reaction-diffusion systems, *J. Comput. Appl. Math.* 292 (2016) 674–693, <https://doi.org/10.1016/j.cam.2015.04.044>.
- [29] M. D’Autilia, I. Sgura, V. Simoncini, Matrix-oriented discretization methods for reaction-diffusion PDEs: comparisons and applications, *Comput. Math. Appl.* 79 (2020) 2067–2085, <https://doi.org/10.1016/j.camwa.2019.10.020>.
- [30] M. Neubert, H. Caswell, Alternatives to resilience for measuring the responses of ecological systems to perturbations, *Ecology* 78 (1997) 653–665, <https://doi.org/10.1890/0012-9658>.
- [31] M. Neubert, H. Caswell, J. Murray, Transient dynamics and pattern formation: reactivity is necessary for Turing instabilities, *Math. Biosci.* 175 (2002) 1–11, [https://doi.org/10.1016/S0025-5564\(01\)00087-6](https://doi.org/10.1016/S0025-5564(01)00087-6).
- [32] A. Alla, A. Monti, I. Sgura, Adaptive pod-deim correction for Turing pattern approximation in reaction-diffusion pde systems, *J. Numer. Math.* 31 (3) (2023) 205–229, <https://doi.org/10.1515/jnma-2022-0025>.
- [33] N. Heavner, F.D. Igual, G. Quintana-Ortí, P.G. Martinsson, Efficient algorithms for computing a rank-revealing UTV factorization on parallel computing architectures, *ACM Trans. Math. Softw.* 48 (2) (2022), <https://doi.org/10.1145/3507466>.
- [34] P.G. Martinsson, G. Quintana-Ortí, N. Heavner, Randutv: a blocked randomized algorithm for computing a rank-revealing UTV factorization, *ACM Trans. Math. Softw.* 45 (1) (2019), <https://doi.org/10.1145/3242670>.