LOW-RANK DYNAMIC MODE DECOMPOSITION: OPTIMAL SOLUTION IN POLYNOMIAL TIME

P. HÉAS* AND C. HERZET*

Abstract. This work studies the linear approximation of high-dimensional dynamical systems using low-rank dynamic mode decomposition (DMD). Searching this approximation in a data-driven approach can be formalised as attempting to solve a low-rank constrained optimisation problem. This problem is non-convex and state-of-the-art algorithms are all sub-optimal. This paper shows that there exists a closed-form solution, which can be computed in polynomial time, and characterises the ℓ_2-norm of the optimal approximation error. The theoretical results serve to design low-complexity algorithms building reduced models from the optimal solution, based on singular value decomposition or low-rank DMD. The algorithms are evaluated by numerical simulations using synthetic and physical data benchmarks.

Key words. Reduced models, linear low-rank approximation, constrained optimisation, dynamical mode decomposition.

AMS subject classifications. 37M, 49K, 41A29, 68W25

1. Introduction.

1.1. Context. The numerical discretisation of a partial differential equation parametrised by its initial condition often leads to a very high dimensional system of the form:

$$\begin{cases} x_t = f_t(x_{t-1}), \\ x_1 = \theta, \end{cases} \tag{1.1}$$

where $x_t \in \mathbb{R}^n$ is the state variable, $f_t : \mathbb{R}^n \to \mathbb{R}^n$, and $\theta \in \mathbb{R}^n$ denotes an initial condition. In some context, *e.g.*, for uncertainty quantification purposes, one is interested by computing a set of trajectories corresponding to different initial conditions $\theta \in \Theta \subset \mathbb{R}^n$. This may constitute an intractable task due to the high dimensionality of the space embedding the trajectories. For instance, in the case f_t is linear, the overall complexity necessary to compute a trajectory with model (1.1) can scale at worst in $\mathcal{O}(Tn^2)$, which is prohibitive for large values of the dimension n and of the trajectory length T.

To overcome this issue, reduced models aim to approximate the trajectories of the system for a range of regimes determined by a set of initial conditions [6]. A common approach is to assume that the trajectories of interest are well approximated in a low-dimensional subspace of $\mathbb{R}^n$. In this spirit, many tractable approximations of model (1.1) have been proposed in the literature, the most common ones being *Petrov-Galerkin projections* [26]. These methods are however intrusive in the sense they require the knowledge of the equations ruling the high-dimensional system.

Alternatively, there exist non-intrusive data-driven approaches. In particular, linear inverse modeling [25], principal oscillating patterns [12], or more recently, dynamic mode decomposition (DMD) [5, 15, 17, 28, 30] propose approximations of the unknown function f_t by a linear and low-rank operator. This linear framework has been extended in [7] to the quadratic approximation of f_t. Although a linear framework may be in appearance limited, it has recently obtained a new surge of interest

*INRIA Centre Rennes - Bretagne Atlantique, campus universitaire de Beaulieu, 35042 Rennes, France (patrick.heas@inria.fr, cedric.herzet@inria.fr)

because of its prominent role in decompositions known as extended DMD or kernel-based DMD [3, 20, 31, 32, 33]. The latter decompositions can characterise non-linear behaviors under certain conditions [16].

A reduced model based on a low-rank linear approximation uses a matrix $\hat{A}_k \in \mathbb{R}^{n\times n}$ of rank at most $k \leq n$ which substitutes for function f_t as

$$\begin{cases} \tilde{x}_t = \hat{A}_k \tilde{x}_{t-1} \\ \tilde{x}_1 = \theta \end{cases}, \tag{1.2}$$

where elements of $\{\tilde{x}_t\}_{t=1}^T$ denote approximations of the trajectories $\{x_t\}_{t=1}^T$ satisfying (1.1). Obviously, a brute-force evaluation of a trajectory approximation with (1.2) will not induce a low computational cost: the complexity scales as $\mathcal{O}(Tn^2)$. However, exploiting the fact that matrix $\hat{A}_k$ has a rank at most equal to k, this complexity, which we will qualify of *on-line*, can be significantly lowered. An on-line complexity scaling in $\mathcal{O}(Tr^2 + rn)$ is reach if we can determine *off-line* the matrices R, $L \in \mathbb{R}^{n\times r}$ and $S \in \mathbb{R}^{r\times r}$ with $k \leq r \leq n$ such that trajectories of (1.2) correspond to those obtained with the r-dimensional recursion

$$\begin{cases} z_1 = L^\intercal \theta \\ z_t = S z_{t-1} \\ \tilde{x}_t = R z_t \end{cases}. \tag{1.3}$$

The equivalence of systems (1.2) and (1.3) implies that $\prod_{i=1}^t \hat{A}_k = R(\prod_{i=1}^t S)L^\intercal$. We remark that if $S \in \mathbb{R}^{r\times r}$ is a block diagonal matrix and in particular a Jordan matrix, the on-line complexity to run reduced-model (1.3) scales at worst in $\mathcal{O}(s^2T+rn)$ where s denotes the maximum size of the blocks. In the case S is a diagonal matrix, then trajectories of (1.3) can be computed with the advantageous on-line complexity of $\mathcal{O}(rn)$, *i.e.*, linear in the ambient dimension n, linear in the reduced-model intrinsic dimension r and independent of the trajectory length T.

To enjoy this low on-line computational effort, it is first necessary to compute off-line a proper matrix $\hat{A}_k$ in (1.2) and deduce parameters R, L and S at the core of reduced model (1.3). We will refer to the term off-line complexity for the latter computational cost. A standard choice for $\hat{A}_k$ is to select the best trajectory approximations in the ℓ_2-norm, for initial conditions in the set $\Theta \subset \mathbb{R}^n$: matrix $\hat{A}_k$ in (1.2) targets the solution of the following minimisation problem

$$\underset{A:\mathrm{rank}(A)\leq k}{\arg\min} \int_{\theta\in\Theta} \sum_{t=2}^{T} \|x_t(\theta) - A^{t-1}\theta\|_2^2, \tag{1.4}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm. A reduced model of the form of (1.3) based on a low-rank minimiser of (1.4) can then be deduced from its eigen-decomposition: R and L are set as the right and left eigen-vectors and S as the matrix of eigen-values.

Since we focus on non-intrusive *data-driven* approaches, we will assume that we do not know the exact form of f_t in (1.1) and we only have access to a set of representative trajectories $\{x_t(\theta_i)\}_{t=1}^T$, $i = 1, ..., N$ so-called *snapshots*, obtained by running the high-dimensional system for N different initial conditions $\{\theta_i\}_{i=1}^N$ in the set Θ. Using these snapshots, we consider a discretised version of (1.4), which corresponds

to the constrained optimisation problem studied in [5, 15]: matrix $\hat{A}_k$ now targets the solution

$$A_k^\star \in \underset{A:\mathrm{rank}(A)\leq k}{\arg\min} \sum_{t=2,i=1}^{T,N} \|x_t(\theta_i) - Ax_{t-1}(\theta_i)\|_2^2, \tag{1.5}$$

where we have substituted $A^{t-1}\theta_i$ in (1.4) by $Ax_{t-1}(\theta_i)$. By bartering a polynomial objective function for a quadratic one, we consider in (1.5) a simpler constrained least-square problem. This optimisation problem is however still non-convex due to the low-rank constraint.

In the light of (1.3) and (1.5), we introduce the terminology used in the literature [5, 15, 17, 28, 30]. Let columns of R and L be the dominant right and left eigen-vectors of $\hat{A}_k$ and let S be the diagonal matrix gathering the first k eigenvalues. Given this choice for matrices R, L and S, the term "low-rank DMD" of system (1.1) denotes the reduced model (1.3) in the case $\hat{A}_k = A_k^\star$, while the term "DMD" denotes this reduced model in the case $\hat{A}_k$ is the solution of problem (1.4) without the low-rank constraint.

1.2. Problematic. This work deals with the off-line construction of reduced models of the form of (1.3). It focuses on the following questions. Can we compute in polynomial time a solution of (1.5)? How to compute efficiently the parameters R, L and S of (1.3) and in particular the low-rank DMD of (1.1)? We discuss in what follows these two problematics.

Solver for problem (1.5)**.** There has been in the last decade a resurgence of interest for low-rank solutions of linear matrix equations in noise-free [9, 27] or noisy [18, 19, 14] settings. This class of problems includes (1.5) as an important particular case. Problems in this class are generally non-convex and accessing to theirs solutions in polynomial time is often out of reach.

Nevertheless, certain instances with a very special structure admit closed-form solutions [24, 22]. This occurs typically when the solution can be deduced from the Eckart-Young theorem [8]. Surprisingly, previous works [5, 15] presuppose that problem (1.5) is difficult and does not admit a closed-form solution. Therefore, several sub-optimal approaches have been proposed in the literature. The most straightforward approach is to approximate the soluton of problem (1.5), proceeding in two independent steps [15, 30]: in a first step, an unconstrained version of (1.5) (or of a similar problem) is solved; in a second step, a k-th order approximation of the latter solution is obtained either by truncating its SVD or its eigen-decomposition, or by solving a sparse approximation problem. As an alternative, some works propose to approximate problem (1.5) by a regularised version of the unconstrained problem [20, 33]. A regularisation of interest is obtained by particularising convex relaxation techniques to problem (1.5). The relaxed problem may recover exactly the solution $A_k^\star$ under certain condition [9, 18, 19, 23, 27]. Finally, some works tackle directly the constrained minimisation problem. Authors in [5, 15] suggest to approach the targeted solution $A_k^\star$ relying on the assumption that snapshots belong to a pre-determined subspace. Authors in [14] propose an iterative hard thresholding approach with guarantees of exact recovery of $A_k^\star$ under certain conditions.

In summary, none of the state-of-the-art approaches guarantee optimality in general.

Computation of R, L and S in (1.3). The second problem concerns the computation of matrices R, L and S in (1.3), and in particular the computation of low-rank DMD from the solution $A_k^\star \in \mathbb{R}^{n\times n}$. It is not clear that this will not imply a prohibitive computational burden for large n. Indeed the brute-force computation of the SVD or the eigen-decomposition of a matrix in $\mathbb{R}^{n\times n}$ requires a complexity scaling at worst as $\mathcal{O}(n^3)$. Hopefully, since the range of the rows and the range of the columns of $A_k^\star$ belong to a low-dimensional subspace of $\mathbb{R}^n$, the left and right eigen-vectors or singular vectors associated to the non-zero eigen-values or singular values can be computed with a reduced complexity [10]. Exploiting this idea, the authors in [30, 32] propose a method for computing matrices R and S in DMD based on the eigen-decomposition of a square matrix of size $T(N-1)$, involving a linear complexity in n. For low-rank DMD, an analogous low-complexity algorithm approximates R and S (*i.e.*, the right eigen-vectors and associated eigen-values of $A_k^\star$) [15]. In the case $k \ll T(N-1)$ and for a large value of the number of snapshots N or the trajectory length T, the authors in [32] suggest to rely on Krylov methods to approximate R and S, involving a quadratic instead of a cubic complexity in N.

In summary, no state-of-the-art methods enable the exact computation of all low-rank DMD parameters (*i.e.*, of R, L and S) with a linear complexity in n, independently of N or T.

1.3. Contributions. The contribution of this paper is twofold. First, we show that the special structure of problem (1.5) enables the closed-form characterisation of an optimal solution $A_k^\star$, from which we can deduce an efficient polynomial-time solver. Besides, we also characterise the optimal approximation error in (1.5). Second, using this closed-form solution $A_k^\star$, we provide low-complexity algorithms computing reduced models and in particular the low-rank DMD of (1.1).

The paper is organised as follows. In Section 3, we provide a review of state-of-the-art techniques to solve the constrained optimisation problem (1.5) and compute low-rank DMD. Section 4 details an analytical solution for problem (1.5), characterises the optimal approximation error and provides an efficient algorithm to compute this solution. Given this optimal solution, we provide algorithms to compute exactly and with a low off-line complexity SVD-based reduced model and low-rank DMD. Finally, using a synthetic and a physical data benchmark, we compare in Section 5 the proposed algorithms with state-of-the-art. We draw conclusions in a last section.

2. Notations. We will use in the following some matrix notations. The upper script $\cdot^\intercal$ will refer to the transpose operator. I_k will denote the k-dimensional identity matrix. All along the paper, we will make extensive use of the SVD of a matrix $M \in \mathbb{R}^{p\times q}$ with $p \geq q$: $M = U_M \Sigma_M V_M^\intercal$ with $U_M \in \mathbb{R}^{p\times q}$, $V_M \in \mathbb{R}^{q\times q}$ and $\Sigma_M \in \mathbb{R}^{q\times q}$ so that $U_M^\intercal U_M = V_M^\intercal V_M = I_q$ and Σ_M is diagonal. The columns of matrices U_M and V_M will be denoted $U_M = (u_M^1 \cdots u_M^q)$ and $V_M = (v_M^1 \cdots v_M^q)$ while diagonal components of matrix Σ_M will be $\Sigma_M = \mathrm{diag}(\sigma_{M,1}, \cdots, \sigma_{M,q})$ with $\sigma_{M,i} \geq \sigma_{M,i+1}$ for $i = 1, \cdots, q-1$. The Moore-Penrose pseudo-inverse of matrix M is then defined as $M^\dagger = V_M \Sigma_M^\dagger U_M^\intercal$, where $\Sigma_M^\dagger = \mathrm{diag}(\sigma_{M,1}^\dagger, \cdots, \sigma_{M,q}^\dagger)$ with

$$\sigma_{M,i}^\dagger = \begin{cases} \sigma_{M,i}^{-1} & \text{if} \quad \sigma_{M,i} > 0 \\ 0 & \text{otherwise} \end{cases}.$$

The orthogonal projector onto the span of the columns (resp. of the rows) of matrix M will be denoted by $\mathbb{P}_M = MM^\dagger = U_M \Sigma_M \Sigma_M^\dagger U_M^\intercal$ (resp. $\mathbb{P}_{M^\intercal} = M^\dagger M =$

$V_M\Sigma_M^\dagger\Sigma_M V_M^\intercal$) [10].

We also introduce additional notations to derive a matrix formulation of the low-rank estimation problem (1.5). We gather consecutive elements of the i-th snapshot trajectory between time t_1 and t_2 in matrix $X^i_{t_1:t_2} = (x_{t_1}(\theta_i)\cdots x_{t_2}(\theta_i))$ and form large matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$ with $m = N(T-1)$ as

$$\mathbf{X} = (X^1_{1:T-1}\cdots X^N_{1:T-1}) \quad \text{and} \quad \mathbf{Y} = (X^1_{2:T}\cdots X^N_{2:T}).$$

In order to be consistent with the SVD definition and to keep the presentation as simple as possible, this work will assume that $m \leq n$. However, all the result presented in this work can be extended without any difficulty to the case $m > n$ by using an alternative definition of the SVD.

3. State-Of-The-Art for Solving for (1.5) **and Building** (1.3)**.** We begin by presenting state-of-the-art methods solving approximatively the low-rank minimisation problem (1.5). In a second part, we will make an overview of algorithms for the construction of reduced models of the form of (1.3) including low-rank DMD.

3.1. Standard Candidate Solutions for (1.5)**.** Using the notations introduced previously, we want to solve problem (1.5) rewritten as

$$A_k^\star \in \underset{A:\mathrm{rank}(A)\leq k}{\arg\min} \|\mathbf{Y} - A\mathbf{X}\|_F^2, \tag{3.1}$$

where $\|\cdot\|_F$ refers to the Frobenius norm.

3.1.1. Truncation of the Unconstrained Solution. By removing the low-rank constraint, (3.1) becomes a least-squares problem admitting the solution $\mathbf{Y}\mathbf{X}^\dagger$, see [30]. We remark that the latter matrix of rank at most m is also solution of problem (3.1) for $k = m$, *i.e.*,

$$A_m^\star = \mathbf{Y}\mathbf{X}^\dagger, \tag{3.2}$$

and that the cost function at this point vanishes if $\mathrm{rank}(\mathbf{X}) = m$. This solution is computable using the SVD of $\mathbf{X}$: $A_m^\star = \mathbf{Y}V_\mathbf{X}\Sigma_\mathbf{X}^\dagger U_\mathbf{X}^\intercal$. An approximation of the solution of (3.1) satisfying the low-rank constraint $\mathrm{rank}(A) \leq k$ with $k < m$ can then be obtained by a truncation of the SVD or the eigen-decomposition of $A_m^\star$ using k terms.

Since this method relies on the SVD of $\mathbf{X} \in \mathbb{R}^{n\times m}$, it leads to a complexity scaling as $\mathcal{O}(m^2(m+n))$.

3.1.2. Approximation by Projected DMD. The so-called *"projected DMD"* proposed in [28] as a low-dimensional approximation of $A_m^\star$ is further investigated by the authors in [15] in order to approximate $A_k^\star$. It assumes that columns of $A\mathbf{X}$ are in the span of $\mathbf{X}$. This assumption is formalised in [15, 28] as the existence of $A^c \in \mathbb{R}^{m\times m}$, the so-called *companion matrix* of A parametrised by m coefficients,[1]

[1]The particular parametrisation of the companion matrix

$$A^c = \begin{pmatrix} 0 & & & & \alpha_1 \\ 1 & 0 & & & \alpha_2 \\ & \ddots & \ddots & & \vdots \\ & & 1 & 0 & \alpha_{m-1} \\ & & & 1 & \alpha_m \end{pmatrix} \in \mathbb{R}^{m\times m},$$

depending on m coefficients $\{\alpha_i\}_{i=1}^m$ follows from the fact that by definition the first $m-1$ columns of $\mathbf{Y}$ are the last $m-1$ columns of $\mathbf{X}$, see details in [28].

such that

$$A\mathbf{X} = \mathbf{X}A^c. \tag{3.3}$$

Under this assumption, we obtain from (3.3) a low-dimensional representation of A in the span of $U_{\mathbf{X}}$,

$$U_{\mathbf{X}}^{\intercal} A U_{\mathbf{X}} = \tilde{A}^c, \tag{3.4}$$

where $\tilde{A}^c = \Sigma_{\mathbf{X}} V_{\mathbf{X}}^{\intercal} A^c V_{\mathbf{X}} \Sigma_{\mathbf{X}}^{\dagger} \in \mathbb{R}^{m\times m}$. An approximation of $A_k^\star$ is obtained in [15] by plugging (3.3) in the cost function of problem (3.1) and by computing the m coefficients of matrix A^c minimising this cost. Noticing the invariance of the Frobenius norm to unitary transforms, it is straightforward to see that this is equivalent to solve the problem

$$\underset{\tilde{A}^c:\mathrm{rank}(\tilde{A}^c\Sigma_{\mathbf{X}})\leq k}{\arg\min} \|U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}} - \tilde{A}^c\Sigma_{\mathbf{X}}\|_F^2. \tag{3.5}$$

Assuming $\mathbf{X}$ is full-rank, the solution is simply given by the Eckart-Young theorem [8] as the SVD representation of matrix $B = U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}}$ truncated to k terms multiplied by matrix $\Sigma_{\mathbf{X}}^{\dagger}$. Denoting by $\tilde{B}$ this truncated decomposition, we finally obtain the following approximation of (3.1)

$$A_k^\star \approx U_{\mathbf{X}} \tilde{B} \Sigma_{\mathbf{X}}^{\dagger} U_{\mathbf{X}}^{\intercal}. \tag{3.6}$$

This method relies on the SVD of $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times m}$. It thus involves a complexity scaling as $\mathcal{O}(m^2(m+n))$.

3.1.3. Approximation by Sparse DMD. The authors in [15] also propose a two-stage approach they call *"sparse DMD"*. It consists in solving two independent problems. The first stage consists in computing the eigen-decomposition of the approximated solution (3.6) for $k = m$. This first stage yields eigen-vectors ζ_i, for $i = 1, \cdots, m$. In a second stage, the authors assume that a linear combination of k out of the m computed eigen-vectors can approximate sufficiently accurately the data. This assumption serve to design a relaxed convex optimisation problem using a ℓ_1-norm penalisation.[2] Solving this problem, they obtain k eigen-vectors and their associated coefficients. Let us note that the approximation error induced by the sparse DMD method will always be greater than the one induced by the projected approach[3].

[2] We note that the penalisation coefficient must be adjusted to induce $m-k$ coefficients nearly equal to zero.

[3] Exploiting orthogonality and the invariance of the Frobenius norm to unitary transforms, we obtain $\|\mathbf{Y} - \mathbf{X}\tilde{A}^c\|_F^2 = \|U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}} - \tilde{A}^c\Sigma_{\mathbf{X}}\|_F^2 + \|(U_{\mathbf{X}}^{\perp})^{\intercal}\mathbf{Y}\|_F^2$, where the columns of $U_{\mathbf{X}}^{\perp}$ contain the $n-m$ vectors orthogonal to $U_{\mathbf{X}}$. Moreover, any companion matrix A^c obviously satisfies

$$\min_{\tilde{A}^c:\mathrm{rank}(\tilde{A}^c\Sigma_{\mathbf{X}})\leq k} \|U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}} - \tilde{A}^c\Sigma_{\mathbf{X}}\|_F^2 \leq \|U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}} - A^c\Sigma_{\mathbf{X}}\|_F^2,$$

and this inequality is in particular verified by the companion matrix solving the sparse DMD problem. Taking the minimum over the set of low-rank companion matrices, we therefore obtain that

$$\|\mathbf{Y} - \hat{A}_k\mathbf{X}\|_F^2 = \min_{\tilde{A}^c:\mathrm{rank}(\tilde{A}^c\Sigma_{\mathbf{X}})\leq k} \|U_{\mathbf{X}}^{\intercal}\mathbf{Y}V_{\mathbf{X}} - \tilde{A}^c\Sigma_{\mathbf{X}}\|_F^2 + \|(U_{\mathbf{X}}^{\perp})^{\intercal}\mathbf{Y}\|_F^2 \leq \|\mathbf{Y} - \hat{A}_{k,0}\mathbf{X}\|_F^2,$$

where $\hat{A}_k$ denotes the low-rank solution (3.6) given by the projected approach and $\hat{A}_{k,0}$ denotes the solution given by the sparse DMD method.

This method relies on the resolution of a ℓ_1-norm minimisation problem constructed from the computation of the approximation (3.6) and on its eigen-decomposition. The latter can be deduced from the eigen-decomposition of $B\Sigma_{\mathbf{X}}^{\dagger} \in \mathbb{R}^{m\times m}$. The overall compexity scales in $\mathcal{O}(m^2(m+n))$.

3.1.4. Approximation by Solving Regularised Problems. Some works propose to approximate (3.1) by a regularised version of the unconstrained problem. In this spirit, Tikhonov penalisation [20] or penalisation enforcing structured sparsity [33] have been proposed. However, these choices of regulariser are arbitrarily and do not rely on sound theoretical basis. In contrast, the solution of (3.1) may under certain conditions be recovered by the following quadratic program

$$A_k^{\star} \approx \underset{A\in\mathbb{R}^{n\times n}}{\arg\min} \|\mathbf{Y} - A\mathbf{X}\|_F^2 + \alpha\|A\|_*, \tag{3.7}$$

where $\|\cdot\|_*$ refers to the nuclear norm.[4] In the latter optimisation problem, $\alpha \in \mathbb{R}_+$ represents an appropriate regularisation parameter determining the rank of the solution of (3.7). The conditions under which the approximation (3.7) becomes exact are expressed in the theoretical works [19, 14] in terms of a so-called *restricted isometry property* that must be satisfied by the linear operator which maps $A \in \mathbb{R}^{n\times n}$ to $A\mathbf{X} \in \mathbb{R}^{n\times m}$. Program (3.7) is a convex optimisation problem which can be solved by standard convex optimisation techniques [2], or using dedicated algorithms as proposed in [23].

The algorithms solving (3.7) typically involve a complexity per iteration scaling as $\mathcal{O}(mnk)$, *i.e.*, linear with respect to the number m of snapshots on the contrary to the other state-of-the-art approaches

3.1.5. Approximation by Singular Value Projections. In [14], the authors propose an iterative hard thresholding approach to solve a generalisation of (3.1). This algorithm, called singular value projections (SVP), refines the candidate solution iteratively by adapting a classical projected descent gradient. The projection onto the non-convex set $\{A \in \mathbb{R}^{n\times n} : \mathrm{rank}(A) \leq k\}$ is computed efficiently by SVD. A guarantee for the exact recovery of $A_k^{\star}$ is obtained under a condition based on the restricted isometry property.

The algorithm possesses a complexity per iteration scaling as $\mathcal{O}((k+m)^2 n)$.

3.2. Standard Methods for Building Reduced-Model (1.3)**.** In this section, we present state-of-the-art methods to compute reduced models of the form of (1.3) from the approximation of $A_k^{\star}$, denoted in what follows by $\hat{A}_k$.

3.2.1. SVD-Based Approach. This approach presupposes the knowledge of a factorisation of the form

$$\hat{A}_k = PQ^{\intercal} \quad \text{with} \quad P, Q \in \mathbb{R}^{n\times r}, \tag{3.8}$$

where $r \leq n$. Assuming this structure, we note that trajectories of (1.2) can be fully determined by the r-dimensional recursion (1.3) with $R = P^{\dagger}$, $L = P$ and $S = Q^{\intercal}P$. According to Section 1.1, the on-line complexity to compute this recursion is $\mathcal{O}(r^2T + rn)$.

Of course, by SVD of $\hat{A}_k$ it is always possible to compute matrices satisfying (3.8) with $r = \mathrm{rank}(\hat{A}_k)$. However, a brute-force computation of the SVD of $\hat{A}_k \in \mathbb{R}^{n\times n}$

[4] The nuclear norm or trace norm of a matrix is the sum of its singular values.

may be prohibitive since it requires a complexity scaling at worst as $\mathcal{O}(n^3)$. Hopefully, there often exist other possible choices satisfying (3.8) and the complexity can often be lowered to $\mathcal{O}(r^2(r+n))$ with $\text{rank}(\hat{A}_k) \leq r \leq m$. For example, in the case $\hat{A}_k$ is obtained by the truncated approach (see Section 3.1.1), we can simply build a reduced model of the form (1.3) by setting $P = U_{\mathbf{Y}}$ and $Q^\intercal = \Sigma_{\mathbf{Y}} V_{\mathbf{Y}}^\intercal \mathbf{X}^\dagger$ in (3.8). In the case $\hat{A}_k$ is obtained by projected DMD (see Section 3.1.2), we can build this reduced model using the factorisation $P = U_{\mathbf{X}}$ and $Q^\intercal = \tilde{B}\Sigma_{\mathbf{X}}^\dagger U_{\mathbf{X}}^\intercal$. In the case $\hat{A}_k$ is obtained by sparse DMD (see Section 3.1.3), we can build this reduced model using the latter factorisation where $\tilde{B}$ is substituted by the approximation of B by a matrix composed of k of its columns, where the column indexes are determined by solving the sparse problem. Approaches based on SVP (see Section 3.1.5) naturally provide a factorisation of $\hat{A}_k$ of the form of (3.8). Indeed, these algorithms iteratively enrich the k-term SVD representation of the solution $\hat{A}_k$, through singular value projections [14]. Unfortunately, regularised approaches (Section 3.1.4) may not naturally exhibit such a factorisation.

3.2.2. Approach by Eigen-Decomposition: Low-Rank DMD. An efficient reduced model may be obtained by rewriting (1.2) in terms of the eigen-decomposition of the linear approximation $\hat{A}_k$

$$\hat{A}_k = RSL^\intercal. \tag{3.9}$$

with L, $R \in \mathbb{C}^{n\times r}$ and $S \in \mathbb{C}^{r\times r}$ is a Jordan-block matrix [10] with $r = \text{rank}(\hat{A}_k)$. Using this decomposition, recursion (1.2) can be rewritten as

$$\tilde{x}_t = RS^{t-1}L^\intercal\theta. \tag{3.10}$$

From the definition of R, L and S, we obtain a direct equivalence of reduced model (3.10) with recursion (1.3). We note $L = (\xi_1 \cdots \xi_r)$ and $R = (\zeta_1 \cdots \zeta_r)$, where $\xi_i \in \mathbb{C}^n$ and $\zeta_i \in \mathbb{C}^n$ are the i-th left and right eigen-vector of $\hat{A}_k$. Assuming that $\hat{A}_k$ is diagonalisable[5], we have $S = \text{diag}(\lambda_1, \cdots, \lambda_r)$ and (3.10) becomes

$$\begin{cases} \tilde{x}_t = \displaystyle\sum_{i=1}^{\text{rank}(\hat{A}_k)} \zeta_i \nu_{i,t}, \\ \nu_{i,t} = \lambda_i^{t-1} \xi_i^\intercal \theta, \quad \text{for} \quad i = 1, \cdots, \text{rank}(\hat{A}_k) \end{cases} \tag{3.11}$$

where $\lambda_i \in \mathbb{C}$ is the i-th (non-zero) eigen-values of $\hat{A}_k$. As mentioned previously, in the literature the latter reduced model is called DMD or low-rank DMD, depending if $\hat{A}_k$ is an approximation of problem (3.1) for $k = m$ or $k < m$ [31]. As detailed in Section 1.1, (3.11) is the most efficient reduced model in terms of on-line complexity, since it scales as $\mathcal{O}(rn)$, with $r = \text{rank}(\hat{A}_k) \leq k$ versus at best $\mathcal{O}(r^2T + rn)$ for the SVD-based reduced model presented in the last section.

However, the eigen-decomposition of a matrix in $\mathbb{R}^{n\times n}$ is in general prohibitive as it implies at worst an off-line complexity of $\mathcal{O}(n^3)$. But for most approximations, this off-line complexity can be significantly reduced as the matrix $\hat{A}_k$ is low-rank and its rows and columns span specific low-dimensional subspaces. In particular, for an approximation obtained by truncation of $A_m^\star$ (see Section 3.1.1), matrices R and S

[5] Diagonalisability is assured if all the non-zero eigenvalues are distinct. This condition is however only sufficient and the class of diagonalisable matrices is larger [13].

can be computed with an off-line complexity scaling as $\mathcal{O}(m^2(m+n))$ by exploiting the SVD of $\mathbf{X}$, see details in [30]. As detailed in [15], an approximation of R is deduced with the same complexity in the case $\hat{A}_k$ is obtained by projected DMD (see Section 3.1.2) exploiting the eigen-vectors of the solution of (3.5). The amplitude $\nu_{i,t}$ (related to S and L) is then obtained in a second stage by solving a convex optimisation problem with an iterative gradient-based method. An approximation using sparse DMD presented in Section 3.1.3 implies the same computation of matrix R followed by a sparse convex minimisation problem to obtain amplitudes $\nu_{i,t}$ (related to S and L). The latter procedure does not increase the off-line complexity of the overall algorithm. No efficient algorithm is provided in the literature for the obtention of matrices R, L and S in the case $\hat{A}_k$ is obtained by SVP as described in Section 3.1.5. However, we can expect from matrix analysis that the eigen-decomposition of $A_k^\star$ can be efficiently computed taking advantage of the k-term SVD of $\hat{A}_k$ given by the SVP algorithm [10]. Finally, unfortunately, the regularised solution presented in Section 3.1.4 does not in general exhibit a clear structure enabling to reduce the complexity of its eigen-decomposition.

4. The Proposed Approach. The existence of a closed-form solution to problem (3.1) remains to our knowledge unnoted in the literature, and no exact algorithm has been proposed yet. The following section intends to fill this gap.

4.1. Closed-Form Solution to (3.1)**.** Let the columns of matrix $\hat{P} \in \mathbb{R}^{n\times k}$ be the left singular vectors $\{u_{\mathbf{Z}}^i\}_{i=1}^k$ associated to the k largest singular values of matrix

$$\mathbf{Z} = \mathbf{Y}\mathbb{P}_{\mathbf{X}^\intercal} \in \mathbb{R}^{n\times m}, \tag{4.1}$$

i.e.,

$$\hat{P} = \begin{pmatrix} u_{\mathbf{Z}}^1 & \cdots & u_{\mathbf{Z}}^k \end{pmatrix}. \tag{4.2}$$

This matrix serve to characterise a closed-form solution to (3.1) given in the following theorem.

THEOREM 4.1. *Problem* (3.1) *admits the solution*

$$A_k^\star = \hat{P}\hat{P}^\intercal \mathbf{Y}\mathbf{X}^\dagger. \tag{4.3}$$

Moreover, the optimal approximation error is given by

$$\|\mathbf{Y} - A_k^\star \mathbf{X}\|_F^2 = \sum_{i=k+1}^{m} \sigma_{\mathbf{Z},i}^2 + \sum_{i=i^*}^{m}\sum_{j=1}^{m} \sigma_{\mathbf{Y},j}^2 \left((v_{\mathbf{X}}^i)^\intercal v_{\mathbf{Y}}^j\right)^2, \tag{4.4}$$

where $i^* = \operatorname{rank}(\mathbf{X}) + 1$.

Therefore, problem (3.1) can be simply solved by computing the orthogonal projection of $\mathbf{Y}\mathbf{X}^\dagger$, which is the solution $A_m^\star$ of problem (3.1) for $k = m$, onto the subspace spanned by the first k left singular vectors of $\mathbf{Z}$. We detail the proof in Appendix A. The ℓ_2-norm of the error is simply expressed in terms of the singular values of $\mathbf{Y}$ and $\mathbf{Z}$ and of scalar products between the right singular vectors of $\mathbf{X}$ and $\mathbf{Y}$. The second term in the right-hand side of (4.4) can be interpreted as the square norm of the projection of the rows of $\mathbf{Y}$ on the orthogonal to the span of the rows of $\mathbf{X}$, *i.e.*,

$$\sum_{i=i^*}^{m}\sum_{j=1}^{m} \sigma_{\mathbf{Y},j}^2 \left((v_{\mathbf{X}}^i)^\intercal v_{\mathbf{Y}}^j\right)^2 = \|\mathbf{Y}(I_m - \mathbb{P}_{\mathbf{X}^\intercal})\|_F^2.$$

Algorithm 1 Computation of a minimiser of (3.1)

inputs: $(\mathbf{X}, \mathbf{Y})$
1) Compute the SVD of $\mathbf{X} = V_{\mathbf{X}} \Sigma_{\mathbf{X}}^{\dagger} U_{\mathbf{X}}^{\intercal}$
2) Compute $\mathbb{P}_{\mathbf{X}^\intercal} = V_{\mathbf{X}} \Sigma_{\mathbf{X}} \Sigma_{\mathbf{X}}^{\dagger} V_{\mathbf{X}}^{\intercal}$.
3) Compute $\mathbf{Z} = \mathbf{Y} \mathbb{P}_{\mathbf{X}^\intercal}$.
4) Compute the first k columns of $V_{\mathbf{Z}}$ and $\Sigma_{\mathbf{Z}}^2$, *i.e.*, the first k eigen-vector/eigen-value of $\mathbf{Z}^\intercal \mathbf{Z}$
5) Compute the columns of $\hat{P}$ defined as the first k columns of matrix $\mathbf{Z} V_{\mathbf{Z}} \Sigma_{\mathbf{Z}}^{\dagger}$
output: $A_k^\star = \hat{P} \hat{P}^\intercal \mathbf{Y} V_{\mathbf{X}} \Sigma_{\mathbf{X}}^{\dagger} U_{\mathbf{X}}^{\intercal}$

Note that if $\mathbf{X}$ is full-rank then we obtain the simplifications $\hat{P} = I_m$ and $\mathbf{Z} = \mathbf{Y}$. In this case $i^* = m + 1$ so that the second term in the right-hand side of (4.4) vanishes and the approximation error reduces to $\|\mathbf{Y} - A_k^\star \mathbf{X}\|_F^2 = \sum_{i=k+1}^{m} \sigma_{\mathbf{Y},i}^2$. We remark that the latter error is independent of matrix $\mathbf{X}$ and is simply the sum of the square of the $m - k$ smallest singular values of $\mathbf{Y}$. This error also corresponds to the optimal error for the approximation $\mathbf{Y}$ by a matrix of rank at most k in the Frobenius norm [8]. Besides we note that the rank of $A_k^\star$ can be smaller than k. Indeed, by the Sylvester's theorem [13] we have that

$$\operatorname{rank}(A_k^\star) \leq \operatorname{rank}(\mathbf{Y}\mathbf{X}^\dagger) \leq \min(\operatorname{rank}(\mathbf{Y}), \operatorname{rank}(\mathbf{X})),$$

which shows that the rank of $A_k^\star$ is smaller than k not only if $\operatorname{rank}(\mathbf{X})$ or $\operatorname{rank}(\mathbf{Y})$ are smaller than k, but also if $\operatorname{rank}(\mathbf{Y}\mathbf{X}^\dagger) = \operatorname{rank}(\Sigma_{\mathbf{Y}} V_{\mathbf{Y}}^\intercal V_{\mathbf{X}} \Sigma_{\mathbf{X}}) < k$. Note that the latter condition does not necessarily imply the former.

4.2. Algorithm Solving Problem (3.1)**.** We show hereafter how to compute the solution (4.3) by using a product of easily-computable matrices.

The left singular vectors associated to the k largest singular values of matrix $\mathbf{Z}$ correspond to the first k (real and orthonormal) eigen-vectors of matrix $\mathbf{Z}\mathbf{Z}^\intercal$. Matrix $\mathbf{Z}\mathbf{Z}^\intercal$ is of size $n \times n$. Since n is typically very large, this prohibits the direct computation of an eigen-value decomposition. But it is well-known that the eigen-vectors associated to the non-zero eigen-values of matrix $\mathbf{Z}\mathbf{Z}^\intercal \in \mathbb{R}^{n \times n}$ can be obtained from the eigen-vectors and eigen-values of the smaller matrix $\mathbf{Z}^\intercal \mathbf{Z} \in \mathbb{R}^{m \times m}$. Indeed, using the SVD of $\mathbf{Z}$,

$$\mathbf{Z} = U_{\mathbf{Z}} \Sigma_{\mathbf{Z}} V_{\mathbf{Z}}^\intercal,$$

we see that the columns of matrix $U_{\mathbf{Z}} \in \mathbb{R}^{n \times m}$ are the eigen-vectors of $\mathbf{Z}\mathbf{Z}^\intercal$ while the columns of matrix $V_{\mathbf{Z}} \in \mathbb{R}^{m \times m}$ are the eigen-vectors of $\mathbf{Z}^\intercal \mathbf{Z}$. Since $V_{\mathbf{Z}}$ is unitary, we obtain that the sought vectors are the first k columns of $U_{\mathbf{Z}}$, *i.e.*, of $\mathbf{Z} V_{\mathbf{Z}} \Sigma_{\mathbf{Z}}^{\dagger}$.

In the light of this remark, it is straightforward to design Algorithm 1, which will provide the matrices to compute efficiently the solution $A_k^\star$ of (3.1) as the product of the following matrices $\hat{P} \hat{P}^\intercal \mathbf{Y} V_{\mathbf{X}} \Sigma_{\mathbf{X}}^{\dagger} U_{\mathbf{X}}^{\intercal}$. This algorithm requires the computation of an SVD of matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, an eigen-decomposition of matrix $\mathbf{Z}^\intercal \mathbf{Z} \in \mathbb{R}^{m \times m}$ and matrix multiplications involving m^2 vector products in $\mathbb{R}^n$ or $\mathbb{R}^m$. The complexity of the different step of this algorithm is therefore scaling as $\mathcal{O}(m^2(m + n))$. Obviously, computing each entry of $A_k^\star \in \mathbb{R}^{n \times n}$ in Algorithm 1 would imply a complexity scaling as $\mathcal{O}(n^2 k)$. This would be prohibitive for large n. However, as detailed in the next section, this is not necessary to build reduced models (1.3) or (3.11).

Algorithm 2 SVD-based reduced model

inputs: $(\mathbf{X}, \mathbf{Y})$
1) Compute $\hat{P}$ performing step 1 to 5 of Algorithm 1.
2) Compute $\hat{Q}$ using (4.5).
outputs: $L = \hat{P}$, $R = \hat{P}^\intercal$, $S = \hat{Q}^\intercal \hat{P}$.

Algorithm 3 Low-rank DMD

inputs: $(\mathbf{X}, \mathbf{Y})$
1) Compute step 1 to 5 of Algorithm 1 and use (4.5) to obtain $\hat{Q}$.
2) Let $r = \mathrm{rank}(A_k^\star)$ and solve for $i = 1, \cdots, r$ the eigen-equations

$$(\hat{Q}^\intercal \hat{P}) w_i^r = \lambda_i w_i^r \quad \text{and} \quad (\hat{P}^\intercal \hat{Q}) w_i^\ell = \lambda_i w_i^\ell,$$

where $w_i^r, w_i^\ell \in \mathbb{C}^k$ and $\lambda_i \in \mathbb{C}$ such that $|\lambda_{i+1}| \geq |\lambda_i|$.
3) Compute for $i = 1, \cdots, r$ the right eigen-vectors and left eigen-vector

$$\zeta_i = \lambda_i^{-1} \hat{P} \hat{Q}^\intercal \hat{P} w_i^r \quad \text{and} \quad \xi_i = \lambda_i^{-1} \hat{Q} w_i^\ell. \tag{4.6}$$

4) Rescale the ξ_i's so that ${\xi_i}^T \zeta_i = 1$.
outputs: $L = (\xi_1 \cdots \xi_r)$, $R = (\zeta_1 \cdots \zeta_r)$, $S = \mathrm{diag}(\lambda_1, \cdots, \lambda_r)$.

4.3. Algorithms for Building Reduced-Model (1.3)**.** Given the closed-form expression (4.3) of the optimal low-rank linear approximation $A_k^\star$, we present in what follows two algorithms. The first one builds an SVD-based reduced model while the second one computes low-rank DMD. We introduce to this aim matrix

$$\hat{Q} = (\hat{P}^\intercal \mathbf{Y} \mathbf{X}^\dagger)^\intercal \in \mathbb{R}^{n \times k}, \tag{4.5}$$

with matrix $\hat{P}$ given in (4.2).

4.3.1. SVD-Based Reduced Model. Noticing that $\hat{P}^\dagger = \hat{P}^\intercal$, from the expression of $A_k^\star$ it is straightforward to build an SVD-based reduced model taking the form of recursion (1.3) by setting $r = k$ with $L = \hat{P}$, $R = \hat{P}^\intercal$ and $S = \hat{Q}^\intercal \hat{P}$. This simple reduced-model construction is presented in Algorithm 2. As it relies on the first five steps of Algorithm 1, the off-line complexity to build this SVD-based reduced model goes as $\mathcal{O}(m^2(m+n))$.Therefore, the computational cost is the same as for state-of-the-art algorithms. As mentioned in Section 3.2.1, the on-line complexity to run it then goes as $\mathcal{O}(Tk^2 + kn)$.

4.3.2. Low-Rank DMD. We remark that the solution (4.3) is the product of matrices in $\mathbb{R}^{n \times k}$ and in $\mathbb{R}^{n \times m}$. Therefore we can expect from matrix analysis that eigen-vectors of $A_k^\star$ belong to a k-dimensional subspace [10]. Indeed, as shown in the following proposition, the parameters of low-rank DMD (3.11) can be computed without any approximation by eigen-decomposition of some matrices in $\mathbb{R}^{k \times k}$. The proof of this proposition is given in Appendix C.

PROPOSITION 4.2. *Assume $A_k^\star$ is diagonalisable. The components of the set $\{\zeta_i, \xi_i, \lambda_i\}_{i=1}^{\mathrm{rank}(A_k^\star)}$ generated by Algorithm 3 are the right eigen-vectors, the left eigen-vectors and the eigen-values related to the non-zero eigen-values of $A_k^\star$. Moreover they*

satisfy $\xi_i^\intercal A_k^\star \zeta_i = \lambda_i$.

Assuming $A_k^\star$ is diagonalisable, this proposition justifies Algorithm 3, which deduces the right and left eigen-vectors and eigen-values of $A_k^\star$ from the eigen-decomposition of two matrices in $\mathbb{R}^{k\times k}$.

On *sus* of calling Algorithm 1, Algorithm 3 performs in step 2 the eigen-decomposition of matrix $(\hat{Q}^\intercal \hat{P}) \in \mathbb{R}^{k\times k}$ and its transpose and computes in step 3 matrix multiplications involving $r \times n$ vector products in $\mathbb{R}^m$, with $r = \text{rank}(A_k^\star) \leq k \leq m$. Therefore, the additional off-line complexity due to Algorithm 3 scales at worst as $\mathcal{O}(k^3 + m^2 n))$. We remark that this complexity presents in the case $k \ll m$ an advantage in comparison to state-of-the-art approaches scaling as $\mathcal{O}(m^2(m+n))$. However, the overall off-line complexity necessary to build the low-rank DMD reduced model will be bounded by the complexity associated to the five steps of Algorithm 1, which scales as $\mathcal{O}(m^2(m+n))$. Nevertheless, let us mention that an overall off-line complexity scaling as $\mathcal{O}(k^3+m^2 n))$ can be preserved if we accept to barter the exact eigen-decomposition computation in step 4 of Algorithm 1 by an approximation relying on Krylov methods, as suggested in [32]. The latter approximation may be worth for large values of m. Once this reduced model is built, as mentioned previously, low-rank DMD can be run with an on-line complexity scaling as $\mathcal{O}(rn)$ with $r = \text{rank}(A_k^\star)$, *i.e.*, linear in n and independent of T, which represents a great advantage compared to the previous SVD-based reduced model.

5. Numerical Evaluation. In what follows, we evaluate three different approaches for computing low-rank DMD, namely

- *method a)* denotes the proposed optimal approach, *i.e.*, the SVD-based reduced model given by Algorithm 2 or equivalently low-rank DMD given in Algorithm 3,
- *method b)* denotes a low-rank DMD approximation based on the k-th order truncation of SVD of the unconstrained problem solution (3.2) [30],
- *method c)* denotes a low-rank DMD approximation based on the projected approach, *i.e.*, the k-th order approximation (3.6) [15].

Rather than evaluating the error norm committed by the reduced model, *i.e.*, the error norm of the target problem (1.4), we are interested in the capabilities of the different algorithms to solve the proxy (1.5) (or equivalently (3.1)) for this problem. Therefore, the performance is measured in terms of the cost of the proxy (3.1) for this problem, *i.e.*, the normalised error norm

$$\frac{\|\mathbf{Y} - \hat{A}_k \mathbf{X}\|_F}{\|\mathbf{Y}\|_F}$$

with respect to k, where k denotes the bound on the rank constraint in problem (3.1). Besides, in the analysis perspective adopted most often in the literature on DMD, we are interested in evaluating the capabilities of the algorithms to compute accurately the parameters L, R and S of reduced models of the form of (1.3).

Convex relaxation approaches and iterative hard thresholding algorithms have been omitted in the present evaluation because there does not exist in the literature algorithms coding the recovery of $\hat{A}_k$ yet. We do not provide a comparison with the sparse DMD approach for two reasons. The first one is practical: as formulated in [15], it is not obvious to tune the regularisation coefficient to induce k-term representations

for $k = 1, \cdots, m$. The second reason is theoretical: the error norm induced by the sparse DMD method will always be greater than the one induced by the projected approach, see Section 3.1.3. Concerning method b), we chose to evaluate low-rank DMD based on SVD truncation of the unconstrained solution $A_m^\star$ (rather than on the truncation of its eigen-decomposition) because among two-stage approaches, it is by construction the most performant reduced model in terms of the error norm chosen for evaluation.

We begin by evaluating the low-rank approximations using a toy model and then continue by assessing their performance for the reduction of a Rayleigh-Bénard convective system [4]. We finally evaluate Algorithm 3 and in particular the influence of noise on the capability of the method to extract accuratly the $\zeta_i's$ in the low-rank DMD reduced model (3.11).

5.1. Synthetic Experiments with a Toy Model.

We set $n = 50$ and $m = 40$ and consider a low-dimensional subspace of $r = 30$ dimensions. Matrices $\mathbf{X}$ and $\mathbf{Y}$, are generated using (1.1) and three different definitions for f_t:

- *setting i)*: $f_t(x_{t-1}) = Gx_{t-1}$, where G is chosen so that the exists A^c satisfying $G\mathbf{X} = \mathbf{X}A^c$,
- *setting ii)*: $f_t(x_{t-1}) = Fx_{t-1}$,
- *setting iii)*: $f_t(x_{t-1}) = Fx_{t-1} + F\text{diag}(x_{t-1})\text{diag}(x_{t-1})x_{t-1}$.

Matrices introduced above are random matrices of rank r defined as $F = \sum_{i=1}^{r} \varphi_i \varphi_i^\intercal$ and $G = \mathbf{X}^\dagger F \mathbf{X}$, where the φ_i's are n-dimensional independent samples of the standard normal distribution. The pseudo-inverse $\mathbf{X}^\dagger$ is computed from the SVD of $\mathbf{X}$. We draw the initial condition θ according to the same distribution. The first setting is a linear system satisfying the assumption made in the projected approach [28, 15]. The two next settings do not make this assumption and simulate respectively linear and cubic dynamics.

The performance of the three methods are displayed in Figure 5.1. As predicted by our theoretical results, *method a)*, *i.e.*, the proposed algorithm, yields the smallest error norm. The deterioration of the error norm for *method b)* shows that a two-stage approach is sub-optimal. The error norm increase is moderate in these toy experiments. We mention that, although not displayed in the figure, the gap with the optimal solution becomes important for $k < 30$ if we choose to truncate the eigen-decomposition[6] of $A_m^\star$ instead of its SVD.

Moreover, the experiments show that as expected *method c)*, *i.e.*, the low-rank projected approach, achieves the optimal performance as long as the assumption $G\mathbf{X} = \mathbf{X}A^c$ holds, *i.e.*, for *setting i)*. If the assumption is not satisfied, *i.e.*, in *setting ii)* and *iii)*, the performance of the projected approach deteriorates notably for $k > 10$. Nevertheless, we notice that *method c)* leads to a slight gain in performance compared to *method b)* up to a moderate rank ($k < 5$).

Finally, as expected, the linear operator used to generate the snapshots is accurately recovered by *method a)* and *method b)* for $k \geq r$.

[6] As pointed out previously, this alternative two-stage method is voluntarily not displayed to lighten the presentation.

5.2. Physical Experiments. Rayleigh-Bénard model [4] constitutes a benchmark for convective system in geophysics. It is also famous because of its three-dimensional Galerkin projection, known as the "Lorenz reduced system". The solution of the latter system, when plotted, resembles a butterfly [21]. Convection is driven by two coupled partial differential equations. In order to introduce the model, we need to introduce differential operators. Let $\nabla = (\partial_{s_1}, \partial_{s_2})^\intercal$, $\nabla^\perp = (\partial_{s_2}, -\partial_{s_1})^\intercal$ and $\Delta = \partial^2_{s_1} + \partial^2_{s_2}$ denote the gradient, the curl and the Laplacian with respect to the two spatial dimensions (s_1, s_2). Let operator Δ^{-1} be the inverse of Δ. Boundary conditions are periodic along s_1 and of Dirichlet type[7] along s_2. At any point of the unit cell $\mathbf{s} = (s_1, s_2) \in [0, 1]^2$ and for any time $t \geq 1$, the temperature $\tau(\mathbf{s}, t) \in \mathbb{R}$, the buoyancy $b(\mathbf{s}, t) \in \mathbb{R}$ and the velocity $\mathbf{v}(\mathbf{s}, t) \in \mathbb{R}^2$ in the cell satisfy

$$\begin{cases} \partial_t b(\mathbf{s}, t) + \mathbf{v}(\mathbf{s}, t)^\intercal \nabla b(\mathbf{s}, t) - \sigma \Delta b(\mathbf{s}, t) - \sigma\nu \partial_{s_1} \tau(\mathbf{s}, t) = 0, \\ \partial_t \tau(\mathbf{s}, t) + \mathbf{v}(\mathbf{s}, t)^\intercal \nabla \tau(\mathbf{s}, t) - \Delta \tau(\mathbf{s}, t) - \partial_{s_1} \Delta^{-1} b(\mathbf{s}, t) = 0, \end{cases} \tag{5.1}$$

where velocity is in equilibrium with buoyancy according to

$$\mathbf{v}(\mathbf{s}, t) = \nabla^\perp \Delta^{-1} b(\mathbf{s}, t).$$

The regime of the convective system is parametrised by two quantities: 1) the Rayleigh number $\nu \in \mathbb{R}_+$, which balances thermal diffusion and the tendency for a packet of fluid to rise due to the buoyancy force; 2) the Prandtl number $\sigma \in \mathbb{R}_+$ which controls the importance of viscosity compared to thermal diffusion.

In our experiments, we assume the initial condition takes the form of a solution of a Lorenz reduced model [21]. This initial condition, corresponding to a still fluid with a difference of temperature between the bottom and the top of the cell, is defined as

$$\begin{aligned} b(\mathbf{s}, 1) =& \kappa_b \sin(a_b s_1) \sin(\pi s_2), \\ \tau(\mathbf{s}, 1) =& \kappa_{\tau_1} \cos(a_\tau s_1) \sin(\pi s_2) - \kappa_{\tau_2} \sin(2\pi s_2). \end{aligned} \tag{5.2}$$

It is easy to verify that for this parametrisation and in the particular case where $\nu = 0$ and $\kappa_b = \sigma^{-1}(\pi a)^{-2}$, the non-linear system (5.1) simplifies into a linear evolution of the temperature driven by a buoyancy force evolving in time according to a Taylor vortex [29]

$$\begin{cases} b(\mathbf{s}, t) = \sigma^{-1}(\pi a_b)^{-2} \exp^{-\sigma\pi^2 a_b^2 t} \sin(a_b s_1) \sin(\pi s_2), \\ \partial_t \tau(\mathbf{s}, t) + \mathbf{v}(\mathbf{s}, t) \cdot \nabla \tau(\mathbf{s}, t) - \Delta \tau(\mathbf{s}, t) - \partial_{s_1} \Delta^{-1} b(\mathbf{s}, t) = 0. \end{cases} \tag{5.3}$$

We use a finite difference scheme on (5.1) and obtain a discrete system of the form of (1.1) with $x_t = \begin{pmatrix} b_t \\ \tau_t \end{pmatrix} \in \mathbb{R}^n$, and $n = 1024$, where b_t's and τ_t's are spatial discretisations of buoyancy and temperature fields at time t.

We assume that we have at our disposal three datasets of snapshots of the discretised system trajectories. More precisely, we choose $m = 50$ and consider the three following settings:

[7] In order to simplify the Fourier-based numerical implementation of the model, we will assume periodicity for the discretised system in the two spatial directions.

- *setting iv*): $N = 50$ short trajectories ($T = 2$) of the linear system (5.3) obtained by fixing the initial condition on temperature (5.2) with random parameters $(a_\tau, \kappa_{\tau_1}, \kappa_{\tau_2})$ and setting $a_b = 1$,
- *setting v*): $N = 5$ long trajectories ($T = 11$) of the linear system (5.3) obtained by setting randomly parameters $(a_\tau, \kappa_{\tau_1}, \kappa_{\tau_2})$ of the initial condition on temperature and letting $a_b = 1$,
- *setting vi*): $N = 5$ long trajectories ($T = 11$) of the non-linear system (5.1) obtained by setting parameters $(a_\tau, \kappa_b, \kappa_{\tau_1}, \kappa_{\tau_2})$ of the initial condition randomly and letting $a_b = a_\tau$.

Parameters $(a_\tau, \kappa_b, \kappa_{\tau_1}, \kappa_{\tau_2})$ are randomly sampled so that each set of initial conditions correspond to N realisations of the uniform distribution over an hyper-cube of dimensionality $r = 10$.

The performances of *method a) b)* and *c)* are displayed in Figure 5.2 for these three settings. We first comment on results obtained in *setting iv*). We remark that, as expected for the situation $T = 2$, the error obtained by *method a*) vanishes for $k \geq r$, *i.e.*, a dimensionality greater than the initial condition dimensionality. The sub-optimal solution provided by *method b)* induces an important error which vanishes only for $k = m$, *i.e.*, for a dimensionality equal to the number of snapshots. Concerning *method c)*, it produces a fairly good solution up to $k \leq 8$, but the solution is clearly sub-optimal for greater dimensions and is associated to an error saturating to a non-negligible value.

In *setting v*), we have longer sequences ($T > 2$) so that the dimension embedding the initial condition does not necessarily match the dimension embedding the snapshots. However, we remark that the optimal solution provided by *method a)* induces an error nearly vanishing for $k \geq 10$. This attests of the fact that, for this linear model, trajectories are concentrate near the subspace spanned by the initial condition. This explains the quasi-optimal performances of *method c)* which relies on a strong assumption of linear dependance of snapshots. *Method b)* is again clearly sub-optimal and behaves analogously to *setting iv*).

In the more realistic geophysical *setting vi*), we see that the optimal performances achieved by *method a)* are far from being reached by *method b)* and c). As in the linear settings, we observe that the optimal error nearly vanishes for $k \geq 10$. However, we clearly notice that the assumption on which rely *method c)* is for this non-linear model invalid, which induces an error saturating to a non-negligible value. We observe again in this case the poor performance of *method b)*.

5.3. Robustness to Noise. In the following, we intend to evaluate the capabilities of the different methods to extract the parameters of (3.11) in the presence of noise. To this aim, we build a dataset of $N = 5$ long trajectories with $T = 11$ (so that we get $m = 50$ snapshots) satisfying (3.11) with $k = 3$ and the parameters $\{(\xi_i, \zeta_i, \lambda_i)\}_{i=1}^3$. The latter are extracted using Algorithm 3 from the geophysical dataset described in *setting vi*). In other words, matrices $\mathbf{X}$ and $\mathbf{Y}$ are generated using (1.1) and the model $f_t(x_{t-1}) = Gx_{t-1}$ where

$$G = \begin{pmatrix} \zeta_1 & \zeta_2 & \zeta_3 \end{pmatrix} \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3) \begin{pmatrix} \xi_1 & \xi_2 & \xi_3 \end{pmatrix}^\intercal .$$

We then consider the two following configurations:

- *setting vii*): the original version of this dataset,

- *setting viii*): a noisy version, where we have corrupted the snapshots with a zero-mean Gaussian noise so that the peak-to-signal-ratio[8] is 20 dB.

Results are displayed in Figures 5.3 and 5.4. As expected we recover a vanishing optimal error for *method a)* in the case $k \geq 3$. We observe only a slight increase of the error in the presence of noise. This attests of the robustness of *method a)* to noise. In the noiseless case *method c)* reproduces almost exactly the optimal behaviour, while its performance slightly deteriorates for $k \geq 2$ in the noisy case. The quasi-optimal performance of this method in the noise-less setting can be interpreted as the fact that there exists a matrix A_c such that assumption $G\mathbf{X} = \mathbf{X}A_c$ is nearly valid. This assumption no longer holds when snapshots are corrupted by noise. *Method b)* produces clearly sub-optimal solutions in the noiseless setting. More importantly, the performance of this method becomes dramatic in the presence of noise. The deterioration is clearly visible for eigen-vector ζ_3 re-arranged in the form of a spatial map in Figure 5.4. The spatial structure of the eigen-vector estimated by *method b)* is completely rubbed out in this noisy setting while it is fairly preserved by *method a)* and roughly recovered by *method c)*. This illustrates the usefulness of solving the low-rank minimisation problem instead of truncating the solution of the unconstrained problem.

6. Conclusion. This work characterises an optimal solution of the non-convex problem related to low-rank linear approximation. As shown in Theorem 4.1, the closed-form solution is in fact the orthogonal projection of the unconstrained problem solution $\mathbf{Y}\mathbf{X}^\dagger$ onto a low-dimensional subspace. This subspace is the span of the first k left singular vectors of a matrix $\mathbf{Z}$, which is defined as the multiplication of $\mathbf{Y}$ by the projector onto the span of the rows of $\mathbf{X}$. The theorem also provides a characterisation of the error between the low-rank approximation and the true trajectory. The expression of the ℓ_2-norm of the error is in fact closed-form and depends on the singular values of $\mathbf{Y}$ and $\mathbf{Z}$ and on the scalar product between right singular vectors of $\mathbf{X}$ and $\mathbf{Y}$. Based on this theorem, Proposition 4.2 then shows that the eigen-vectors and eigen-values in low-rank DMD can be deduced from the eigen-decomposition of matrices in $\mathbb{R}^{k\times k}$, independently from the number of observations.

These theoretical results yield a method to compute the optimal solution in polynomial time. The results also serve to design algorithms computing optimal low-rank approximation with a low-computation effort. In particular we propose an algorithm building exactly low-rank DMD with an off-line complexity going as $\mathcal{O}(m^2(m+n))$, which is the same as for state-of-the-art sub-optimal methods. This reduced model can then be run with an attractive on-line complexity scaling as $\mathcal{O}(kn)$, *i.e.*, independently of the trajectory length T.

Finally, we illustrate through numerical simulations in synthetic and physical setups, the significant gain in accuracy brought by the proposed algorithm in comparison to state-of-the-art sub-optimal approaches.

[8] The peak-to-signal-ration is defined as $20\log_{10}\frac{\max_{t,i}\|x_t(\theta_i)\|_\infty}{\sigma}$, where σ denotes the standard deviation of the Gaussian law.

Appendix A. Proof of Theorem 4.1.

We begin by showing the first part of the theorem, namely that $A_k^\star = \hat{P}\hat{P}^\intercal \mathbf{Y}\mathbf{X}^\dagger$ is a solution of (3.1).

We first prove the existence of a minimiser of (3.1). Let us show that we can restrict our attention to a minimisation problem over the set

$$\mathcal{A} = \{\tilde{A} \in \mathbb{R}^{n\times n} : \operatorname{rank}(\tilde{A}) \leq k, \operatorname{Im}(\tilde{A}^\intercal) \subseteq \operatorname{Im}(\mathbf{X})\}.$$

Indeed, we remark that any matrix $A \in \{\tilde{A} \in \mathbb{R}^{n\times n} : \operatorname{rank}(\tilde{A}) \leq k\}$ can be decomposed in two components: $A = A^{\|} + A^{\perp}$ where $A^{\|}$ belongs to the set $\mathcal{A}$, such that columns of $A^{\|}$ are orthogonal to those of $A^{\perp}$, *i.e.*, $A^{\perp}(A^{\|})^\intercal = 0$. From this construction, we have that rows of $A^{\perp}$ are orthogonal to rows of $\mathbf{X}$. Using this decomposition, we thus have that $\|\mathbf{Y} - A\mathbf{X}\|_F^2 = \|\mathbf{Y} - A^{\|}\mathbf{X}\|_F^2$. Moreover, because of this orthogonal property, we have that $\operatorname{rank}(A) = \operatorname{rank}(A^{\|}) + \operatorname{rank}(A^{\perp})$ so that $\operatorname{rank}(A^{\|}) \leq \operatorname{rank}(A)$. In consequence, if A is a minimiser of (3.1), then $A^{\|}$ is also a minimiser since it leads to same value of the cost function and since it is admissible: $\operatorname{rank}(A^{\|}) \leq \operatorname{rank}(A) \leq k$. Therefore, it is sufficient to find a minimiser over the set $\mathcal{A}$.

Now, according to the Weierstrass' theorem [2, Proposition A.8], the existence is guaranteed if the admissible set $\mathcal{A}$ is closed and the objective function $\|\mathbf{Y} - A\mathbf{X}\|_F^2$ is coercive. Let us prove these two properties. We first show that $\mathcal{A}$ is closed. According to [11, Lemma 2.4], the set of low-rank matrices is closed. Moreover, it is well-known that a linear subspace of a normed finite-dimensional vector space is closed [1, Chapter 7.2], so that the set of matrices $\mathcal{A} = \{\tilde{A} \in \mathbb{R}^{n\times n} : \operatorname{Im}(\tilde{A}^\intercal) \subseteq \operatorname{Im}(\mathbf{X})\}$ is closed. Since $\mathcal{A}$ is the intersection of two closed sets, we deduce that $\mathcal{A}$ is closed. Next, we show coercivity. Let us consider the SVD of any $A \in \mathcal{A}$: $A = U_A \Sigma_A V_A^\intercal$, where $\Sigma_A = \operatorname{diag}(\sigma_{A,1} \cdots \sigma_{A,k})$. From the definition of the Frobenius norm, we have for any $A \in \mathcal{A}$, $\|A\|_F = (\sum_{i=1}^k \sigma_{A,i}^2)^{1/2}$. We remark that $\|A\|_F \to \infty$ if a non-empty subset of singular values, say $\{\sigma_{A,j}\}_{j\in\mathcal{J}}$, tend to infinity. Therefore, we have

$$\begin{aligned}
\lim_{\|A\|_F\to\infty : A\in\mathcal{A}} \|\mathbf{Y} - A\mathbf{X}\|_F^2 &= \lim_{\|A\|_F\to\infty : A\in\mathcal{A}} \|\mathbf{Y}\|_F^2 - 2\operatorname{trace}(Y^\intercal A\mathbf{X}) + \|A\mathbf{X}\|_F^2, \\
&= \lim_{\|A\|_F\to\infty : A\in\mathcal{A}} \|A\mathbf{X}\|_F^2, \\
&= \lim_{\|A\|_F\to\infty : A\in\mathcal{A}} \|\Sigma_A V_A^\intercal \mathbf{X}\|_F^2, \\
&= \lim_{\sigma_{A,j}\to\infty : A\in\mathcal{A}, j\in\mathcal{J}} \sum_{j=1}^n \sigma_{A,j}^2 \|\mathbf{X}^\intercal v_A^j\|_2^2 = \infty.
\end{aligned}$$

The second equality is obtained because the dominant term when $\|A\|_F \to \infty$ is the quadratic one $\|A\mathbf{X}\|_F^2$. The third equality follows from the invariance of the Frobenius norm to unitary transforms while the last equality is obtained noticing that $\|\mathbf{X}^\intercal v_A^j\|_2 \neq 0$ because $v_A^j \in \operatorname{Im}(\mathbf{X})$ since $A \in \mathcal{A}$. This shows that the objective function is coercive over the closed set $\mathcal{A}$. Thus, using the Weierstrass' theorem, this shows the existence of a minimiser of (3.1) in $\mathcal{A}$ and thus in $\{\tilde{A} \in \mathbb{R}^{n\times n} : \operatorname{rank}(\tilde{A}) \leq k\}$. We will no longer restrict our attention to the domain $\mathcal{A}$ in the following and come back to the original problem (3.1) implying the set of low-rank matrices.

We next remark that problem (3.1) can be rewritten as the unconstrained min-

imisation

$$A_k^\star \in \underset{A=PQ^\intercal : P,Q\in\mathbb{R}^{n\times k}}{\arg\min} \|\mathbf{Y} - A\mathbf{X}\|_F^2. \tag{A.1}$$

In the following we will use the first-order optimality condition of problem (A.1) to characterise its minimisers. A closed-form expression for a minimiser will then be obtained be introducing an additional orthonormal property. The first-order optimality condition and the additional orthonormal property are presented in the following lemma, which is proven in Appendix B.

LEMMA A.1. *Problem* (A.1) *admits a solution such that*

$$P^\intercal P = I_k \tag{A.2}$$
$$\mathbf{X}\mathbf{Y}^\intercal P = \mathbf{X}\mathbf{X}^\intercal Q. \tag{A.3}$$

To find a closed-form expression of a minimiser of (A.1), we need to rewrite condition (A.3). We can prove that this condition is equivalent to

$$\mathbb{P}_{\mathbf{X}^\intercal}\mathbf{Y}^\intercal P = \mathbf{X}^\intercal Q. \tag{A.4}$$

Indeed, we can show by contradiction that (A.3) implies that, for any solution of the form $PQ^\intercal$, there exists $Z \in \mathbb{R}^{m\times k}$ such that

$$\mathbb{P}_{\mathbf{X}^\intercal}\mathbf{Y}^\intercal P + Z = \mathbf{X}^\intercal Q, \tag{A.5}$$

with columns of Z in $\ker(\mathbf{X})$. Indeed, if $\mathbb{P}_{\mathbf{X}^\intercal}\mathbf{Y}^\intercal P + Z \neq \mathbf{X}^\intercal Q$, then by multiplying both sides on the left by $\mathbf{X}$ we obtain $\mathbb{P}_{\mathbf{X}}\mathbf{X}\mathbf{Y}^\intercal P + \mathbf{X}Z = \mathbb{P}_{\mathbf{X}}\mathbf{X}\mathbf{Y}^\intercal P \neq \mathbf{X}\mathbf{X}^\intercal Q$. Since $\mathbb{P}_{\mathbf{X}}$ is the orthogonal projector onto the subspace spanned by the columns of $\mathbf{X}$, the latter relation implies that $\mathbf{X}\mathbf{Y}^\intercal P \neq \mathbf{X}\mathbf{X}^\intercal Q$ which contradicts (A.3). This proves that (A.3) implies (A.5).

Now, since columns of the two terms in the left-hand side of (A.5) are orthogonal and since columns of the matrix in the right-hand side are in the image of $\mathbf{X}^\intercal$, we deduce that the only admissible choice is Z with columns belonging both to $\ker(\mathbf{X})$ and $\mathrm{Im}(\mathbf{X}^\intercal)$, *i.e.*, Z is a matrix full of zeros. Therefore, we obtain the necessary condition (A.4).

We have shown on the one hand that (A.3) implies (A.4). On the other hand, by multiplying on the left both sides of (A.4) by $\mathbf{X}$, we obtain (A.3) ($\mathbf{X}\mathbb{P}_{\mathbf{X}^\intercal} = \mathbf{X}$ because $\mathbf{X}\mathbf{X}^\dagger$ is the orthogonal projector onto the space spanned by the columns of $\mathbf{X}$). Therefore the necessary conditions (A.3) and (A.4) are equivalent.

We are now ready to characterise a minimiser of (3.1). According to Lemma A.1,

we have

$$\min_{A\in\mathbb{R}^{n\times n}:\mathrm{rank}(A)\leq k}\|\mathbf{Y}-A\mathbf{X}\|_F^2$$

$$= \min_{(\tilde{P},\tilde{Q})\in\mathbb{R}^{n\times k}\times\mathbb{R}^{n\times k}}\|\mathbf{Y}-\tilde{P}\tilde{Q}^\intercal\mathbf{X}\|_F^2 \quad s.t. \quad \begin{cases}\tilde{P}^\intercal\tilde{P}=I_k\\ \mathbf{X}\mathbf{Y}^\intercal P=\mathbf{X}\mathbf{X}^\intercal Q\end{cases}, \tag{A.6}$$

$$= \min_{(\tilde{P},\tilde{Q})\in\mathbb{R}^{n\times k}\times\mathbb{R}^{n\times k}}\|\mathbf{Y}-\tilde{P}\tilde{Q}^\intercal\mathbf{X}\|_F^2 \quad s.t. \quad \begin{cases}\tilde{P}^\intercal\tilde{P}=I_k\\ \mathbb{P}_{\mathbf{X}^\intercal}\mathbf{Y}^\intercal P=\mathbf{X}^\intercal Q\end{cases},$$

$$= \min_{\tilde{P}\in\mathbb{R}^{n\times k}}\|\mathbf{Y}-\tilde{P}\tilde{P}^\intercal\mathbf{Y}\mathbb{P}_{\mathbf{X}^\intercal}\|_F^2 \quad s.t. \quad \tilde{P}^\intercal\tilde{P}=I_k, \tag{A.7}$$

$$= \min_{\tilde{P}\in\mathbb{R}^{n\times k}}\|(\mathbf{Y}-\tilde{P}\tilde{P}^\intercal\mathbf{Y})\mathbb{P}_{\mathbf{X}^\intercal}+\mathbf{Y}(I_m-\mathbb{P}_{\mathbf{X}^\intercal})\|_F^2 \quad s.t. \quad \tilde{P}^\intercal\tilde{P}=I_k,$$

$$= \min_{\tilde{P}\in\mathbb{R}^{n\times k}}\|\mathbf{Z}-\tilde{P}\tilde{P}^\intercal\mathbf{Z}\|_F^2+\|\mathbf{Y}(I_m-\mathbb{P}_{\mathbf{X}^\intercal})\|_F^2 \quad s.t. \quad \tilde{P}^\intercal\tilde{P}=I_k. \tag{A.8}$$

The second equality is obtained from the equivalence between (A.3) and (A.4). The third equality is obtained by introducing the second constraint in the cost function and noticing that projection operators are always symmetric, *i.e.*, $(\mathbb{P}_{\mathbf{X}^\intercal})^\intercal=\mathbb{P}_{\mathbf{X}^\intercal}$, while the last equality follows from the definition of $\mathbf{Z}$ given in (4.1) and the orthogonality of the columns of the two terms. Problem (A.8) is a proper orthogonal decomposition problem with the snapshot matrix $\mathbf{Z}$. The solution of this proper orthogonal decomposition problem is the matrix $\hat{P}$ (with orthonormal columns) given in (4.2), see *e.g.*, [26, Proposition 6.1]. We thus obtain from (A.7) that

$$\min_{A\in\mathbb{R}^{n\times n}:\mathrm{rank}(A)\leq k}\|\mathbf{Y}-A\mathbf{X}\|_F^2=\|\mathbf{Y}-\hat{P}\hat{P}^\intercal\mathbf{Y}\mathbb{P}_{\mathbf{X}^\intercal}\|_F^2. \tag{A.9}$$

Furthermore, we verify that $A_k^\star=\hat{P}\hat{Q}^\intercal$ with

$$\hat{Q}=(\mathbf{X}^\intercal)^\dagger\mathbf{Y}^\intercal\hat{P},$$

is a minimiser of (A.1). Indeed, we check that $(\hat{P},\hat{Q})$ is admissible for problem (A.6) since

$$\mathbf{X}\mathbf{X}^\intercal\hat{Q}=\mathbf{X}\mathbf{X}^\intercal(\mathbf{X}^\intercal)^\dagger\mathbf{Y}^\intercal\hat{P}=\mathbf{X}\mathbf{Y}^\intercal\hat{P}.$$

We also check using (A.4) that

$$\|\mathbf{Y}-\hat{P}\hat{Q}^\intercal\mathbf{X}\|_F^2=\|\mathbf{Y}-\hat{P}\hat{P}^\intercal\mathbf{Y}\mathbb{P}_{\mathbf{X}^\intercal}\|_F^2,$$

i.e., that $(\hat{P},\hat{Q})$ reaches the minimum given in (A.9). In consequence, we have shown that problem (A.1), and equivalently problem (3.1), admit the minimiser $A_k^\star=\hat{P}\hat{Q}^\intercal=\hat{P}\hat{P}^\intercal\mathbf{Y}\mathbf{X}^\dagger$.

It remains to prove the second part of the theorem, namely the characterisation of the approximation error. According to standard proper orthogonal decomposition analysis, see *e.g.*, [26, Proposition 6.1], the first term of the cost function in (A.8) evaluated at $A_k^\star$ is

$$\|\mathbf{Z}-\hat{P}\hat{P}^\intercal\mathbf{Z}\|_F^2=\sum_{i=k+1}^{m}\sigma_{\mathbf{Z},i}^2. \tag{A.10}$$

We can rewrite the second term of the cost function (A.8) as

$$\begin{aligned}
\|\mathbf{Y}(I_m - \mathbb{P}_{\mathbf{X}^\intercal})\|_F^2 &= \|\Sigma_{\mathbf{Y}} V_{\mathbf{Y}}^\intercal V_{\mathbf{X}} (I_m - \Sigma_{\mathbf{X}} \Sigma_{\mathbf{X}}^\dagger) V_{\mathbf{X}}^\intercal\|_F^2, \\
&= \|\Sigma_{\mathbf{Y}} V_{\mathbf{Y}}^\intercal V_{\mathbf{X}} (I_m - \Sigma_{\mathbf{X}} \Sigma_{\mathbf{X}}^\dagger)\|_F^2, \\
&= \left\| \begin{pmatrix} \sigma_{\mathbf{Y},1}(v_{\mathbf{Y}}^1)^\intercal \\ \vdots \\ \sigma_{\mathbf{Y},m}(v_{\mathbf{Y}}^m)^\intercal \end{pmatrix} \begin{pmatrix} v_{\mathbf{X}}^{i^*} & \cdots & v_{\mathbf{X}}^m \end{pmatrix} \right\|_F^2, \\
&= \sum_{i=i^*}^m \left\| \begin{pmatrix} \sigma_{\mathbf{Y},1}(v_{\mathbf{Y}}^1)^\intercal \\ \vdots \\ \sigma_{\mathbf{Y},m}(v_{\mathbf{Y}}^m)^\intercal \end{pmatrix} v_{\mathbf{X}}^i \right\|_2^2, \\
&= \sum_{i=i^*}^m \sum_{j=1}^m \sigma_{\mathbf{Y},j}^2 ((v_{\mathbf{Y}}^j)^\intercal v_{\mathbf{X}}^i)^2. \qquad \text{(A.11)}
\end{aligned}$$

where the first and second equalities follow from the invariance of the Frobenius norm to unitary transforms, and more precisely to the multiplication on the left by $U_{\mathbf{Y}}^\intercal$ and on the right by $V_{\mathbf{X}}$. Gathering error contributions (A.10) and (A.11), we obtain the sought result. □

Appendix B. Proof of Lemma A.1.

We begin by proving that any minimiser of (A.1) can be rewritten as $PQ^\intercal$ where $P^\intercal P = I_k$. Indeed, the existence of the SVD of $\tilde{A}$ for any minimiser $\tilde{A} \in \mathbb{R}^{n \times n}$ guarantees that

$$\|\mathbf{Y} - \tilde{A}\mathbf{X}\|_F^2 = \|\mathbf{Y} - U_{\tilde{A}} \Sigma_{\tilde{A}} V_{\tilde{A}}^\intercal \mathbf{X}\|_F^2,$$

where $U_{\tilde{A}} \in \mathbb{R}^{n \times k}$ possesses orthonormal columns. Making the identification $P = U_{\tilde{A}}$ and $Q = V_{\tilde{A}} \Sigma_{\tilde{A}}$ we verify that $\|\mathbf{Y} - \tilde{A}\mathbf{X}\|_F^2 = \|\mathbf{Y} - PQ^\intercal \mathbf{X}\|_F^2$ and that P possesses orthonormal columns.

Next, any solution of problem $PQ^\intercal$ of (A.1) should satisfy the first-order optimality condition with respect to the j-th column denoted q_j of matrix Q, that is

$$2[-\mathbf{X}\mathbf{Y}^\intercal p_j + \sum_{i=1}^k (p_i^\intercal p_j) \mathbf{X}\mathbf{X}^\intercal q_i] = 0,$$

where the j-th column of matrix P is denoted p_j. In particular, a solution with P possessing orthonormal columns should satisfy

$$\mathbf{X}\mathbf{Y}^\intercal p_j = \mathbf{X}\mathbf{X}^\intercal q_j,$$

or in matrix form

$$\mathbf{X}\mathbf{Y}^\intercal P = \mathbf{X}\mathbf{X}^\intercal Q. \quad \square$$

Appendix C. Proof of Proposition 4.2.

We have $A_k^\star = \hat{P}\hat{P}^\intercal \mathbf{Y}\mathbf{X}^\dagger = \hat{P}\hat{Q}^\intercal$ which implies that

$$\hat{Q}^\intercal \hat{P} = \hat{P}^\intercal \mathbf{Y}\mathbf{X}^\dagger \hat{P} = \hat{P}^\intercal \hat{P} \hat{P}^\intercal \mathbf{Y}\mathbf{X}^\dagger \hat{P} = \hat{P}^\intercal \hat{P} \hat{Q}^\intercal \hat{P}.$$

Using the definition of ζ_i's and ξ_i's in (4.6) and the fact that the w_i^r's and w_i^ℓ's are the right and left eigen-vectors of $\hat{Q}^\intercal \hat{P}$, we verify that

$$A_k^\star \zeta_i = \frac{1}{\lambda_i} \hat{P}\hat{Q}^\intercal \hat{P}\hat{Q}^\intercal \hat{P} w_i^r = \hat{P}\hat{Q}^\intercal \hat{P} w_i^r = \lambda_i \zeta_i,$$

and that

$$(A_k^\star)^\intercal \xi_i = \frac{1}{\lambda_i} \hat{Q}\hat{P}^\intercal \hat{Q} w_i^\ell = \hat{Q} w_i^\ell = \lambda_i \xi_i.$$

Finally, $\xi_i^\intercal \zeta_i = 1$ is a sufficient condition so that $\xi_i^\intercal A_k^\star \zeta_i = \lambda_i$. $\square$

REFERENCES

[1] Auliac, G., Caby, J.: Mathématiques 3e Année: Topologie et analyse. Objectif Licence. Edi-Science (2005)

[2] Bertsekas, D.: Nonlinear Programming. Athena Scientific (1995)

[3] Budišić, M., Mohr, R., Mezić, I.: Applied koopmanism a. Chaos: An Interdisciplinary Journal of Nonlinear Science **22**(4), 047,510 (2012)

[4] Chandrasekhar, S.: Hydrodynamic and hydromagnetic stability. Courier Corporation (2013)

[5] Chen, K.K., Tu, J.H., Rowley, C.W.: Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. Journal of nonlinear science **22**(6), 887–915 (2012)

[6] Cohen, A., Devore, R.: Approximation of high-dimensional parametric PDEs. ArXiv e-prints (2015)

[7] Cui, T., Marzouk, Y.M., Willcox, K.E.: Data-driven model reduction for the Bayesian solution of inverse problems. International Journal for Numerical Methods in Engineering **102**, 966–990 (2015)

[8] Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**(3), 211–218 (1936)

[9] Fazel, M.: Matrix rank minimization with applications, stanford university. Ph.D. thesis (2002)

[10] Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press (2013)

[11] Hackbusch, W.: Tensor spaces and numerical tensor calculus, vol. 42. Springer Science & Business Media (2012)

[12] Hasselmann, K.: PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns. Journal of Geophysical Research: Atmospheres **93**(D9), 11,015–11,021 (1988)

[13] Horn, R.A., Johnson, C.R.: Matrix analysis. Cambridge university press (2012)

[14] Jain, P., Meka, R., Dhillon, I.S.: Guaranteed rank minimization via singular value projection. In: Advances in Neural Information Processing Systems, pp. 937–945 (2010)

[15] Jovanovic, M., Schmid, P., Nichols, J.: Low-rank and sparse dynamic mode decomposition. Center for Turbulence Research Annual Research Briefs pp. 139–152 (2012)

[16] Klus, S., Koltai, P., Schütte, C.: On the numerical approximation of the perron-frobenius and koopman operator. arXiv preprint arXiv:1512.05997 (2015)

[17] Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic mode decomposition: Data-driven modeling of complex systems (2016)

[18] Lee, K., Bresler, Y.: Guaranteed minimum rank approximation from linear observations by nuclear norm minimization with an ellipsoidal constraint. arXiv preprint (2009)

[19] Lee, K., Bresler, Y.: Admira: Atomic decomposition for minimum rank approximation. IEEE Transactions on Information Theory **56**(9), 4402–4416 (2010)

[20] Li, Q., Dietrich, F., Bollt, E.M., Kevrekidis, I.G.: Extended dynamic mode decomposition with dictionary learning: a data-driven adaptive spectral decomposition of the koopman operator. arXiv preprint arXiv:1707.00225 (2017)

[21] Lorenz, E.N.: Deterministic Nonperiodic Flow. Journal of Atmospheric Sciences **20**, 130–148 (1963)

[22] Mesbahi, M., Papavassilopoulos, G.P.: On the rank minimization problem over a positive semidefinite linear matrix inequality. IEEE Transactions on Automatic Control **42**(2), 239–243 (1997)

[23] Mishra, B., Meyer, G., Bach, F., Sepulchre, R.: Low-rank optimization with trace norm penalty. SIAM Journal on Optimization **23**(4), 2124–2149 (2013)
[24] Parrilo, P.A., Khatri, S.: On cone-invariant linear matrix inequalities. IEEE Transactions on Automatic Control **45**(8), 1558–1563 (2000)
[25] Penland, C., Magorian, T.: Prediction of nino 3 sea surface temperatures using linear inverse modeling. Journal of Climate **6**(6), 1067–1076 (1993)
[26] Quarteroni, A., Manzoni, A., Negri, F.: Reduced basis methods for partial differential equations: an introduction, vol. 92. Springer (2015)
[27] Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review **52**(3), 471–501 (2010)
[28] Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics **656**, 5–28 (2010)
[29] Taylor, G., Green, A.: Mechanism of the production of small eddies from large ones. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **158**(895), 499–521 (1937)
[30] Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: Theory and applications. Journal of Computational Dynamics **1**(2), 391–421 (2014)
[31] Williams, M.O., Kevrekidis, I., Rowley, C.: A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. Journal of Nonlinear Science **25**(6), 1307–1346 (2015)
[32] Williams, M.O., Rowley, C.W., Kevrekidis, I.G.: A kernel-based approach to data-driven koopman spectral analysis. arXiv preprint arXiv:1411.2260 (2014)
[33] Yeung, E., Kundu, S., Hodas, N.: Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems. ArXiv e-prints (2017)

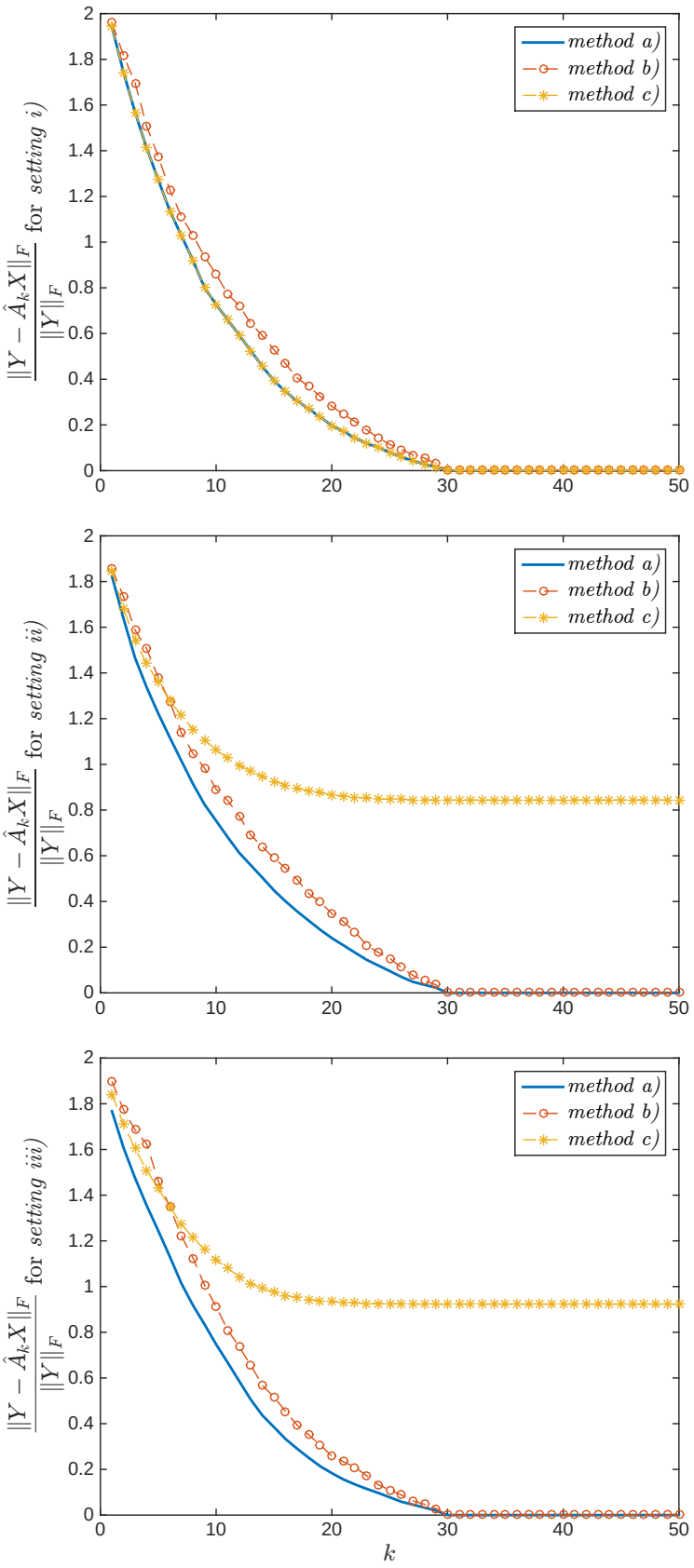

FIG. 5.1. *Error norm as a function of k for setting i), ii) and iii) and for methods a), b) and c). See details in Section 5.1.*

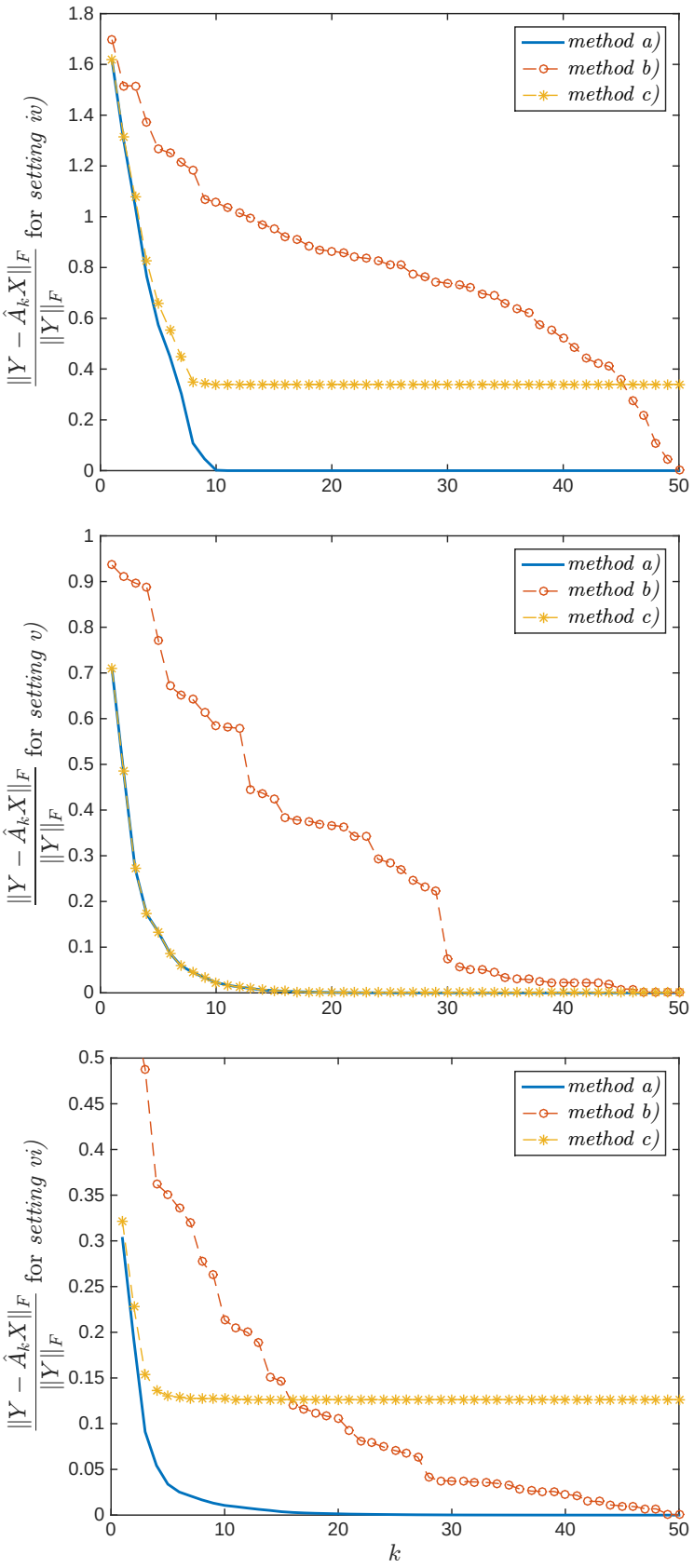

FIG. 5.2. *Error norm as a function of k for setting iv), v) and vi) and for methods a), b) and c). See details in Section 5.2.*

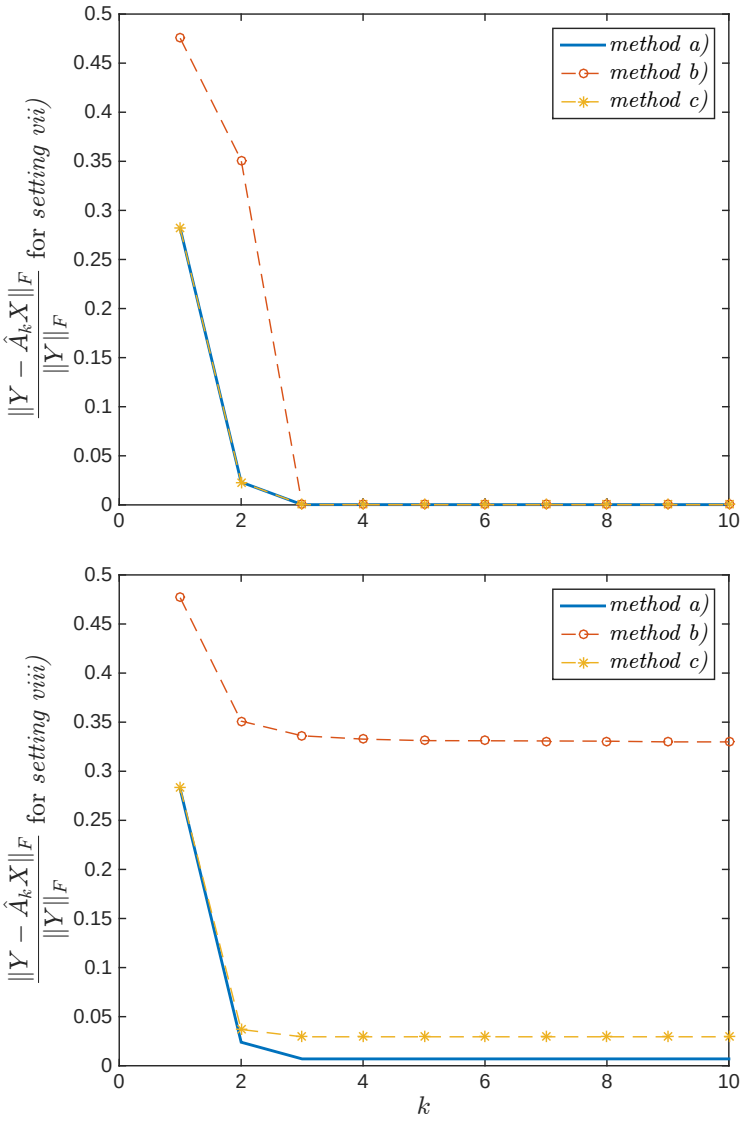

FIG. 5.3. *Error norm as a function of k for setting vii) and viii) and for methods a), b) and c). See details in Section 5.3.*

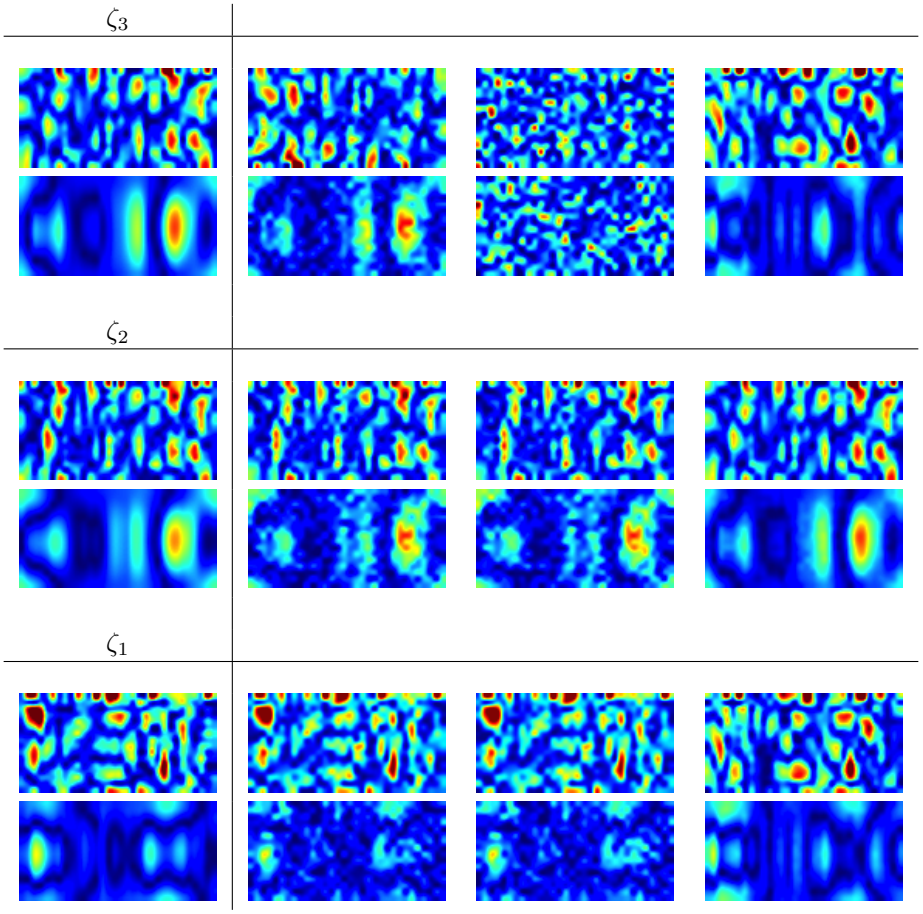

FIG. 5.4. *Amplitude of eigen-vectors ζ_1, ζ_2 and ζ_3 of the low-rank DMD* (3.11) *of a Rayleigh-Bénard system. Left column : reference obtained with method a) in the case of the noiseless setting vii). Eigen-vector reconstruction in the noisy setting viii) with method a) (middle left column), method b) (middle right column) and method c) (right column). Amplitudes are related to temperature (row above) and buoyancy (row below). See details in Section 5.3.*