SIAM J. SCI. COMPUT.
Vol. 41, No. 5, pp. A3118–A3151

DATA DRIVEN KOOPMAN SPECTRAL ANALYSIS IN VANDERMONDE–CAUCHY FORM VIA THE DFT: NUMERICAL METHOD AND THEORETICAL INSIGHTS*

ZLATKO DRMAČ†, IGOR MEZIĆ‡, AND RYAN MOHR§

Abstract. The goals and contributions of this paper are twofold. It provides a new numerically robust computational tool for data driven Koopman spectral analysis, based on the natural formulation via the Krylov decomposition with the Frobenius companion matrix, and by using its eigenvectors explicitly—these are the columns of the inverse of the notoriously ill-conditioned Vandermonde matrix. The key step to curb ill-conditioning is the discrete Fourier transform of the snapshots; in the new representation, the Vandermonde matrix is transformed into a generalized Cauchy matrix, which then allows accurate computation by specially tailored algorithms of numerical linear algebra. The second goal is to shed light on the connection between the formulas for optimal reconstruction weights when reconstructing snapshots using subsets of the computed Koopman modes. It is shown how using a certain weaker form of generalized inverses leads to explicit reconstruction formulas that match the abstract results from Koopman spectral theory, in particular the generalized Laplace analysis.

Key words. dynamic mode decomposition, Koopman operator, Krylov subspaces, proper orthogonal decomposition, Rayleigh–Ritz approximation, Vandermonde matrix, discrete Fourier transform, Cauchy matrix, generalized Laplace analysis

AMS subject classifications. 15A12, 15A23, 65F35, 65L05, 65M20, 65M22, 93A15, 93A30, 93B18, 93B40, 93B60, 93C05, 93C10, 93C15, 93C20, 93C57

DOI. 10.1137/18M1227688

1. Introduction.

Dynamic mode decomposition (DMD) is a data driven spectral analysis technique for a time series. For a sequence of snapshot vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{m+1}$ in $\mathbb{C}^n$, assumed driven by a linear operator $\mathbb{A}$, $\mathbf{f}_{i+1} = \mathbb{A}\mathbf{f}_i$, the goal is to represent the snapshots in terms of the computed eigenvectors and eigenvalues of $\mathbb{A}$. Such a representation of the data sequence provides insight into the evolution of the underlying dynamics, in particular on dynamically relevant spatial structures (eigenvectors) and amplitudes and frequencies of their evolution (encoded in the corresponding eigenvalues)—it can be considered a finite dimensional realization of the Koopman spectral analysis, corresponding to the Koopman operator associated with the dynamics under study [1]. This important theoretical connection with the Koopman operator and the ergodic theory, and the availability of numerical algorithm [37] make the DMD a tool of the trade in the computational study of complex phenomena in fluid dynamics; see, e.g., [35], [44]. A peculiarity of the data driven setting is that

*Submitted to the journal's Methods and Algorithms for Scientific Computing section November 19, 2018; accepted for publication (in revised form) June 26, 2019; published electronically October 15, 2019.

https://doi.org/10.1137/18M1227688

Funding: This research was supported by DARPA contract HR0011-16-C-0116 *"On a Data-Driven, Operator-Theoretic Framework for Space-Time Analysis of Process Dynamics,"* by ARL contract W911NF-16-2-0160 *"Numerical Algorithms for Koopman Mode Decomposition and Application in Channel Flow,"* and by DARPA project HR00111890033 *"The Physics of Artificial Intelligence."*

†Faculty of Science, Department of Mathematics, University of Zagreb, 10000 Zagreb, Croatia (drmac@math.hr).

‡Department of Mechanical Engineering and Mathematics, University of California, Santa Barbara, Santa Barbara, CA 93106, and AIMdyn, Inc., Santa Barbara, CA 93101 (mezic@engr.ucsb.edu).

§AIMdyn, Inc., Santa Barbara, CA 93101 (mohrr@aimdyn.com).

direct access to the operator is not available, and thus an approximate representation of $\mathbb{A}$ is achieved using solely the snapshot vectors $\mathbf{f}_i$, whose number $m+1$ is usually much smaller than the dimension n of the domain of $\mathbb{A}$.

In this paper, we revisit the natural formulation of finite dimensional Koopman spectral analysis in terms of Krylov bases and the Frobenius companion matrix. In this formulation, reviewed in section 2 below, a spectral approximation of $\mathbb{A}$ is obtained by the Rayleigh–Ritz extraction using the Krylov basis $\mathbf{X}_m = (\mathbf{f}_1, \ldots, \mathbf{f}_m)$. This means that the Rayleigh quotient of $\mathbb{A}$ is the Frobenius companion matrix, whose eigenvector matrix is the inverse of the Vandermonde matrix $\mathbb{V}_m$, parametrized by its eigenvalues λ_i. The DMD (Koopman) modes, that is, the Ritz vectors $\mathbf{z}_i$, are then computed as $Z_m \equiv (\mathbf{z}_1 \ \ldots \ \mathbf{z}_m) = \mathbf{X}_m \mathbb{V}_m^{-1}$. Then, from $\mathbf{X}_m = Z_m \mathbb{V}_m$ we readily have $\mathbf{f}_i = \sum_{j=1}^{m} \mathbf{z}_j \lambda_j^i$ for $i = 1, \ldots, m$.

Unfortunately, Vandermonde matrices can have extremely high condition numbers, so a straightforward implementation of this algebraically elegant scheme can lead to inaccurate results in finite precision computation. Most other DMD variants bypass this issue by computing an orthonormal basis from the snapshots using a truncated singular value decomposition (SVD) [37], [36], [41]. We note that the original formulation of DMD was based on the SVD ([37], reviewed in Algorithm 1 in this paper), whereas the first connection between DMD and Koopman operator theory was formulated in terms of the companion matrix [35]. In Rowley et al. [35], however, there was no consideration of the deep numerical issues related to working with Vandermonde matrices, which has likely contributed to the prevalence of the SVD-based variants of DMD. There is, however, a certain intrinsic elegance in the decomposition of the snapshots in terms of the spectral structure of the companion matrix, and because this decomposition has a stronger connection than the SVD-based DMD variants to the generalized Laplace analysis (GLA) [31], [29] and to the Koopman operator theory.

Following this natural formulation, we present a DMD algorithm capable of numerically robust computing without resorting to the SVD of $\mathbf{X}_m$. We do this by leveraging high accuracy numerical linear algebra techniques to curb the potential ill-conditioning of the Vandermonde matrix. The key step is to transform the snapshot matrix by the discrete Fourier transform (DFT)—this induces a similarity transformation of the companion matrix and transforms its eigenvector matrix into the inverse of a generalized Cauchy matrix, which then allows accurate computation by specially tailored algorithms. This will be achieved using the techniques introduced in [11], [10]. We present the details of our formulation in section 3. A numerical example presented in section 4, where the spectral condition number of $\mathbb{V}_m$ is above 10^{70}, illustrates the potential of the proposed method.

While the full collection of DMD modes and eigenvalues gives insight into the intrinsic physics, in many applications a reduced-order model is required. Thus reconstruction of the snapshots using a subset of the modes (selected using some criteria that we do not address here) is desired. We take up the subject of optimal reconstruction of the snapshots in section 5. The problem of computing optimal reconstruction weights is formulated in terms of reflexive g-inverses (see [34]), which are somewhat weaker than the more well-known Moore–Penrose pseudoinverse. It is this set of results that is closely linked with the GLA. We compare formulas for the optimal reconstruction weights, derived in section 5, with reconstruction formulas given by GLA theory in section 6. Whereas the GLA reconstruction formulas have the form of a (finite) ergodic average, the optimal reconstruction formulas can be formulated as a nonuniformly weighted (finite) ergodic average which reduces to the GLA version for

unitary spectrum. It is also interesting to note that while the reconstruction formulas from section 5 optimally reconstruct (by definition) any finite set of snapshots, they are not asymptotically consistent in the sense that they do not recover the correct expansion of the observable in terms of eigenvectors in the limit of infinite data. On the other hand, while the GLA reconstruction formulas are suboptimal (for general operators possessing nonunitary spectrum) for any finite set of snapshots, they are asymptotically consistent, recovering the correct expansion in the limit. We also show that the least squares (LS) problem, solved via the g-reflexive inverses defined on a tensor product of coefficient spaces, is equivalent to an eigenvector-adapted Hermitian form in state space.

2. Preliminaries. In this section, we set the scene and review results relevant for the later development. We assume that the data vectors $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{m+1}\} \subset \mathbb{C}^n$ are generated by a matrix $\mathbb{A} : \mathbb{C}^n \to \mathbb{C}^n$ via $\mathbf{f}_{i+1} = \mathbb{A}\mathbf{f}_i$ for some given initial $\mathbf{f}_1 \in \mathbb{C}^n$. As a technical simplification, a generic case is assumed, i.e., that $\mathbb{A}$ is diagonalizable and that its Ritz values with respect to the subspaces spanned by the snapshots are simple. However, $\mathbb{A}$ itself is not available. We can think of the snapshots $\mathbf{f}_i$ as being generated by, e.g., black-box numerical software $\mathbb{A}$ for solving a partial differential equation with initial condition $\mathbf{f}_1$ or, e.g., as vectorized images of flames in a combustion chamber, taken by a high-speed camera. The goal is to represent the snapshots $\mathbf{f}_i$ of the underlying dynamics using the (approximate) eigenvectors and eigenvalues of $\mathbb{A}$.

2.1. Spectral analysis using Krylov basis. The data driven framework leaves little room to maneuver; hence we resort to the classical Rayleigh–Ritz approximations from the Krylov subspaces spanned by the snapshots. Let $\mathbf{X}_m$ be the column partitioned matrix $\mathbf{X}_m = (\mathbf{f}_1 \ \ldots \ \mathbf{f}_m)$. The columns of $\mathbf{X}_m$ span the Krylov subpsace $\mathcal{K}_m = \mathrm{span}\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m\}$. Without loss of generality, we assume that $\mathbf{X}_m$ is of full rank. The action of $\mathbb{A}$ on $\mathcal{K}_m$ can be represented as

(2.1)
$$\mathbb{A}\mathbf{X}_m = \mathbf{X}_m C_m + E_{m+1}, \quad C_m = \begin{pmatrix} 0 & 0 & \ldots & 0 & c_1 \\ 1 & 0 & \ldots & 0 & c_2 \\ 0 & 1 & \ldots & 0 & c_3 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & c_m \end{pmatrix}, \quad E_{m+1} = \boldsymbol{r}_{m+1}\boldsymbol{e}_m^T, \quad \boldsymbol{e}_m = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

where the c_i's are the $\mathbf{X}_m$-basis coefficients of the orthogonal projection of $\mathbf{f}_{m+1}$ onto $\mathcal{K}_m$, and, with $\mathbf{c} = (c_i)_{i=1}^m$, $\boldsymbol{r}_{m+1} = \mathbf{f}_{m+1} - \mathbf{X}_m\mathbf{c}$ is orthogonal to $\mathcal{K}_m$; $\mathbf{X}_m^*\boldsymbol{r}_{m+1} = 0$. Hence $C_m = (\mathbf{X}_m^*\mathbf{X}_m)^{-1}(\mathbf{X}_m^*\mathbb{A}\mathbf{X}_m) \equiv \mathbf{X}_m^\dagger\mathbb{A}\mathbf{X}_m$, where $\mathbf{X}_m^\dagger$ is the Moore–Penrose generalized inverse of $\mathbf{X}_m$. Thus, the Frobenius companion matrix C_m is the matrix representation of the Rayleigh quotient $\mathbb{P}_{\mathcal{K}_m}\mathbb{A}\big|_{\mathcal{K}_m}$ in the basis formed by the columns of $\mathbf{X}_m$. In practical computation, the coefficients of $\mathbf{c}$ are obtained from the solution of the LS problem $\|\mathbf{X}_m\mathbf{c} - \mathbf{f}_{m+1}\|_2 \to \min$, where for $\mathbf{v} = (v_i) \in \mathbb{C}^n$, $\|\mathbf{v}\|_2 = \sqrt{\mathbf{v}^*\mathbf{v}} \equiv \sqrt{\sum_{i=1}^n |v_i|^2}$.

The spectral decomposition of C_m has remarkable structure. Assume for simplicity that the eigenvalues λ_i, $i = 1, \ldots, m$, are algebraically simple. It is easily checked that the rows of $\mathbb{V}_m$ are the left eigenvectors of C_m, so the columns of $\mathbb{V}_m^{-1}$ are the (right) eigenvectors of C_m; they are essentially unique. Hence, the spectral decomposition of C_m reads

$$C_m = \mathbb{V}_m^{-1}\Lambda_m\mathbb{V}_m, \quad \text{where } \Lambda_m = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{pmatrix}, \quad \mathbb{V}_m = \begin{pmatrix} 1 & \lambda_1 & \ldots & \lambda_1^{m-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{m-1} \\ \vdots & \vdots & \ldots & \vdots \\ 1 & \lambda_m & \ldots & \lambda_m^{m-1} \end{pmatrix}. \tag{2.2}$$

PROPOSITION 2.1. *The columns of* $\widehat{W}_m \equiv \mathbf{X}_m \mathbb{V}_m^{-1}$ *are the Ritz vectors, with the corresponding Ritz values* $\lambda_1, \ldots, \lambda_m$. *Assume all Ritz values to be algebraically simple. Then, for a Ritz pair* $(\lambda_j, \widehat{W}_m(:,j))$, *the residual is given by*

$$\frac{\|\mathbb{A}\widehat{W}_m(:,j) - \lambda_j \widehat{W}_m(:,j)\|_2}{\|\widehat{W}_m(:,j)\|_2} = \frac{\|\boldsymbol{r}_{m+1}\|_2}{\|\widehat{W}_m(:,j)\|_2} \prod_{\substack{k=1 \\ k\neq j}}^{m} \frac{1}{|\lambda_j - \lambda_k|}. \tag{2.3}$$

Proof. It holds that $\mathbb{A}\widehat{W}_m = \widehat{W}_m \Lambda_m + \boldsymbol{r}_{m+1}\mathbf{e}_m^T \mathbb{V}_m^{-1}$, where for the last row of $\mathbb{V}_m^{-1}$ we can use the formulas from [42] to obtain

$$\mathbf{e}_m^T \mathbb{V}_m^{-1} = \left(\prod_{\substack{k=1 \\ k\neq 1}}^{m} \frac{1}{\lambda_1 - \lambda_k}, \prod_{\substack{k=1 \\ k\neq 2}}^{m} \frac{1}{\lambda_2 - \lambda_k}, \ldots, \prod_{\substack{k=1 \\ k\neq m-1}}^{m} \frac{1}{\lambda_{m-1} - \lambda_k}, \prod_{\substack{k=1 \\ k\neq m}}^{m} \frac{1}{\lambda_m - \lambda_k} \right).$$

Hence, for a particular Ritz pair $(\lambda_j, \widehat{W}_m(:,j))$ we have $\mathbb{A}\widehat{W}_m(:,j) = \lambda_j \widehat{W}_m(:,j) + \boldsymbol{r}_{m+1} \prod_{\substack{k=1 \\ k\neq j}}^{m} \frac{1}{\lambda_j - \lambda_k}$. ∎

Remark 2.2. It is instructive to discuss the computational aspect of the companion matrix approach (2.1) and expression for the residual in Proposition 2.1. Clearly, the matrix $\mathbf{X}_m$ is expected to be ill-conditioned, and hence numerically rank deficient, even if it is exactly of full column rank. A numerical algorithm for LS is designed to detect numerical rank deficiency and, depending on a concrete scheme, will return a particular solution from the solution manifold. If the algorithm uses the SVD, it will truncate it and return, via the pseudoinverse, the uniquely determined $\mathbf{c}$ of smallest Euclidean length; the eigenvalues of C_m will most likely be simple, and (2.3) applies. On the other hand, if we use the rank revealing QR factorization, such as in the MATLAB backslash operator, the computed $\mathbf{c}$ will have at least as many zero entries as is the numerical rank deficiency in $\mathbf{X}_m$. In that case, C_m may have zero as a multiple eigenvalue, and as a consequence, (2.3) does not hold. Details such as these are very important for the design of reliable numerical software. In general, if C_m has multiple eigenvalues, then its generalized eigenvector matrix is the inverse of the confluent Vandermonde matrix $\mathbb{V}_m^{(confl)}$ generated by all distinct eigenvalues, and Λ_m is in Jordan form, with precisely one Jordan block associated with any eigenvalue (geometric multiplicity of an eigenvalue of C_m is always one). See [30], [27]. Extending the technique from this paper, in particular section 3, to the confluent case is the subject of our ongoing and future work.

2.1.1. Modal representation of the snapshots. Once the Ritz pairs have provided useful spectral information on $\mathbb{A}$, we would like to analyze the snapshots $\mathbf{f}_i$ in terms of spectral data. To that end, write

$$\mathbf{X}_m = (\mathbf{X}_m \mathbb{V}_m^{-1})\mathbb{V}_m \equiv \widehat{W}_m \mathbb{V}_m, \qquad \mathbb{A}\mathbf{X}_m = \widehat{W}_m(\Lambda_m \mathbb{V}_m) + \boldsymbol{r}_{m+1}\mathbf{e}_m^T, \tag{2.4}$$

and, with the column partition $\widehat{W}_m = \begin{pmatrix} \widehat{\mathbf{w}}_1 & \ldots & \widehat{\mathbf{w}}_m \end{pmatrix}$ of $\widehat{W}_m$, define

$$\mathfrak{a}_j = \|\widehat{\mathbf{w}}_j\|_2, \quad D_{\mathfrak{a}} = \mathrm{diag}(\mathfrak{a}_j)_{j=1}^m, \quad W_m = \widehat{W}_m D_{\mathfrak{a}}^{-1} = \begin{pmatrix} \mathbf{w}_1 & \ldots & \mathbf{w}_m \end{pmatrix}. \tag{2.5}$$

Then, from the first relation in (2.4), we have

$$\mathbf{f}_i = \sum_{j=1}^{m} \widehat{\mathbf{w}}_j \lambda_j^{i-1} \equiv \sum_{j=1}^{m} \mathbf{w}_j \mathfrak{a}_j \lambda_j^{i-1}, \;\; i = 1, \ldots, m \iff \mathbf{X}_m = \widehat{W}_m \mathbb{V}_m \equiv W_m D_{\mathfrak{a}} \mathbb{V}_m, \tag{2.6}$$

and from the last column in the second relation in (2.4) we have

$$\mathbf{f}_{m+1} = \sum_{j=1}^{m} \widehat{\mathbf{w}}_j \lambda_j^m + \boldsymbol{r}_{m+1} = \sum_{j=1}^{m} \mathbf{w}_j \mathfrak{a}_j \lambda_j^m + \boldsymbol{r}_{m+1}. \tag{2.7}$$

It is important to note that the decompositions (or reconstructions) (2.6), (2.7) are by definition attached to the Ritz pairs of $\mathbb{A}$, formed using the spectral decomposition (2.2) of the matrix representation C_m of $\mathbb{P}_{\mathcal{K}_m}\mathbb{A}\big|_{\mathcal{K}_m}$. If $\mathcal{K}_m$ is close to being an $\mathbb{A}$-invariant subspace, then $\mathbb{A}\mathbf{w}_j \approx \lambda_j \mathbf{w}_j$; i.e., the decompositions (2.6), (2.7) are approximately in terms of the eigenpairs of $\mathbb{A}$.

Remark 2.3. Note that $\|\mathbf{w}_j\|_2 = 1$, $j = 1, \ldots, m$, and that $D_{\mathfrak{a}}\mathbb{V}_m(:,1) = W_m^{\dagger}\mathbf{X}_m(:,1) = (\mathfrak{a}_j)_{j=1}^m \in \mathbb{R}^m$. Of course, we can replace each term $w_j\mathfrak{a}_j$ with $(\mathbf{w}_j \mathbf{e}^{\mathrm{i}\psi_j})(\mathbf{e}^{-\mathrm{i}\psi_j}\mathfrak{a}_j)$ (thus redefining $\mathbf{w}_j$ and $\mathfrak{a}_j$) without affecting the decompositions (2.6), (2.7), but this normalization to real $\mathfrak{a}_j$'s seems reasonable if we interpret those numbers as amplitudes. Further, the amplitudes and the scalings of the Ritz vectors can be done in some other appropriate norm instead of in $\|\cdot\|_2$. However, if we pursued the computation of modes from more than one initial condition, the complex form of the amplitudes would be enforced by the requirement that modes are independent of initial conditions [28].

2.2. On computing the eigenvalues of C_m. Since the spectral decomposition (2.2) of C_m is given explicitly from its eigenvalues, it remains to compute the λ_i's efficiently and in a numerically robust way. Computing the eigenvalues of C_m is equivalent to finding the zeros of its characteristic polynomial $\wp_m(z) = z^m - \sum_{j=1}^m c_j z^{j-1}$, but this is only an elegant theoretical connection that has limited value in practical computation. In fact, the most robust polynomial root finding procedure is based on solving the matrix eigenvalue problem while taking the structure of the companion matrix C_m into account. For an excellent mathematical elucidation we refer the reader to [19], [33].

From the software point of view, if we truly want high performance, both in terms of numerical robustness and run time efficiency, using an eigenvalue method for general matrices (such as, e.g., `eig()` in MATLAB) is not the best choice. Namely, in that case the eigenvalues are extracted from the Schur form with the backward error δC_m that is small in the sense that $\|\delta C_m\|_2/\|C_m\|_2$ is of the order of the machine roundoff $\boldsymbol{\varepsilon}$, but $C_m + \delta C_m$ is not a companion matrix, and there is no proper information on the size of the backward error in the coefficients c_i. Further, the computation requires $O(m^2)$ memory space and $O(m^3)$ *flops*.

A proper method for this case is the one that preserves the structure of the companion matrix and for which the computed eigenvalues correspond exactly to the eigenvalues of a companion matrix $\widetilde{C}_m$ with coefficients $\widetilde{c}_i = c_i + \delta c_i$, where $\max_i |\delta c_i|/|c_i|$ is small. Note that backward stability in terms of the coefficients c_i is a proper framework for assessing the numerical accuracy because the coefficients c_i are the results from previous computation—solving the LS problem $\|\mathbf{X}_m\mathbf{c} - \mathbf{f}_{m+1}\|_2 \to \min$. The desired complexity is $O(m)$ in memory space and $O(m^2)$ in *flop* count.

Two methods that satisfy the above requirements are presented in [5], [2] and should be used in high performance software implementations. Particularly interesting

is the unitary-plus-rank-one formulation

$$C_m = \begin{pmatrix} 0 & 0 & \dots & 0 & c_1 \\ 1 & 0 & \dots & 0 & c_2 \\ 0 & 1 & \dots & 0 & c_3 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & c_m \end{pmatrix} = \begin{pmatrix} 0 & 0 & \dots & 0 & \pm 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} + \begin{pmatrix} \mp 1 + c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \end{pmatrix} \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \end{pmatrix} \equiv U_m + \widehat{\mathbf{c}} \mathbf{e}_m^T, \tag{2.8}$$

which has been exploited in [5] to construct an efficient implicit QR iteration process that requires $O(m)$ memory and $O(m^2)$ *flops*. This splitting, when plugged into (2.1), provides the following interesting form of the Krylov decomposition.

PROPOSITION 2.4. *If in the splitting* (2.8) *we choose* $(U_m)_{1,m} = 1$, *then we can write* (2.1) *as* $\mathbb{A}\mathbf{X}_m = \mathbf{X}_m U_m + (\mathbf{f}_{m+1} - \mathbf{f}_1)\mathbf{e}_m^T$.

Proof. Note that $\mathbb{A}\mathbf{X}_m = \mathbf{X}_m U_m + ((\mathbf{X}_m \mathbf{c} - \mathbf{f}_1) + \boldsymbol{r}_{m+1})\boldsymbol{e}_m^T$, where $\boldsymbol{r}_{m+1} = \mathbf{f}_{m+1} - \mathbf{X}_m \mathbf{c}$. ∎

An important remark is in order.

Remark 2.5. If the eigenvalues λ_i of C_m are computed as $\widetilde{\lambda}_i$'s with small backward error in the vector of the coefficients $\mathbf{c}$, then $\mathbb{V}_m(\widetilde{\lambda}_i)^{-1}$ is (assuming that all $\widetilde{\lambda}_i$'s are algebraically simple) the exact eigenvector matrix of the companion matrix $\widetilde{C}_m$ defined by $\mathbf{c} + \delta\mathbf{c}$, where $\|\delta\mathbf{c}\|_2/\|\mathbf{c}\|_2$ (or even $\max_i |\delta c_i|/|c_i|$) is small. Since $\mathbf{c}$ is also a computed quantity (by a backward stable LS solver), the fact that $\mathbf{c} + \delta\mathbf{c} \approx \mathbf{c}$ allows us to interpret $\mathbb{V}_m(\widetilde{\lambda}_i)^{-1}$ as an exact eigenvector matrix that corresponds to backward perturbation of the snapshots. To better appreciate this fact, consider the general case. If we compute the eigenvectors of a general square matrix S, then the general form of the backward error δS precludes any interpretation (in the sense of backward stability) that requires a special structure, such as the companion matrix. Further, the accuracy depends on the condition number of the eigenvectors and on the gap between an eigenvalue and it neighbors in the spectrum. More precisely, if λ is a simple eigenvalue of a diagonalizable S, with unit eigenvector $\mathbf{y}$, then with appropriate nonsingular $Y = (\mathbf{y}\ Y_2)$ and $Z = (\mathbf{z}\ Z_2) = Y^{-1}$, the product $Z^* S Y = \left(\begin{smallmatrix} \lambda & \mathbf{0} \\ \mathbf{0} & \Lambda_2 \end{smallmatrix}\right)$ is diagonal. If $(\widetilde{\lambda}, \widetilde{\mathbf{y}})$ is an eigenpair of $S + \delta S$, then, under some additional assumptions, $\|\widetilde{\mathbf{y}} - \mathbf{y}\|_2 \leq \|(\lambda I - \Lambda_2)^{-1}\|_2 \|Y_2\|_2 \|Z_2\|_2 \|\delta S\|_2 \leq \frac{\kappa_2(Y)}{\min_j |\lambda - (\Lambda_2)_{jj}|} \|\delta S\|_2$. So, for instance, if a group of eigenvalues is tightly clustered, then their corresponding eigenvectors are extremely sensitive and difficult to compute numerically. Here we can set $\delta S = -\mathbf{r}\widetilde{\mathbf{y}}^*$, where $\mathbf{r} = S\widetilde{\mathbf{y}} - \widetilde{\lambda}\widetilde{\mathbf{y}}$ is the residual. For more details see, e.g., [8], [38, Chap. V, sect. 2], [7, sect. 3.2.2].

2.3. Computation with Vandermonde matrices. The natural representation of the evolving dynamics $\mathbf{f}_{i+1} = \mathbb{A}\mathbf{f}_i$ by the Krylov decomposition (2.1) and by the simple and elegant snapshot decompositions (2.6), (2.7) has not led to a numerical scheme that can be used in practical computations. This is due to numerical difficulties when computing the matrix $\widehat{W}_m = \mathbf{X}_m \mathbb{V}_m^{-1}$ of the Ritz vectors; these difficulties are due to a potentially extremely high condition number of the Vandermonde matrix $\mathbb{V}_m$.

Indeed, the Vandermonde matrices can be arbitrarily badly ill-conditioned [32]. More precisely, the condition number $\kappa_2(\mathbb{V}_m) \equiv \|\mathbb{V}_m\|_2 \|\mathbb{V}_m^{-1}\|_2$ will depend on the distribution of the eigenvalues λ_i, and it can be as small as one, and it can grow with the dimension m as fast as $O(m^{m+1})$ for harmonically distributed λ_i's; see Gautschi [21], [20]. For example, any real $n \times n$ Vadermonde matrix has condition number greater than $2^{n-2}/\sqrt{n}$. Recall that the classical upper bound on the relative error in

the solution of linear systems is $O(n)\varepsilon\kappa_2(\mathbb{V}_m)$, so that $\kappa_2(\mathbb{V}_m) > 1/(n\varepsilon)$ implies no accuracy whatsoever.

On the other hand, if the λ_i's are the mth roots of unity, then $\mathbb{V}_m$ is the unitary DFT matrix with $\kappa_2(\mathbb{V}_m) = 1$. In fact, if the λ_i's are on the unit circle, then $\kappa_2(\mathbb{V}_m)$ is, under some additional assumptions (e.g., that the nodes are not tightly clustered), usually moderate [3], [4]. If the underlying operator is nearly unitary, then the λ_i's will be close to the unit circle, and $\mathbb{V}_m$ is usually well conditioned. Further, the condition number can be reduced by scaling, and this opens up the possibility for accurate reconstruction, at least in those cases when scaling sufficiently reduces the condition number.

In certain cases, accurate computation is possible independent of the condition number. The Björck–Pereyra algorithm [6] can solve the Vandermonde system to high accuracy if the λ_i's are real, of the same sign, and increasingly ordered; see [24]. It is the distribution of the signs that determines whether a catastrophic cancellation can occur, and in our case, unfortunately, we cannot rely on such a constellation of the Ritz values because the λ_i's are in general complex numbers.

2.4. Reconstruction using Schmid's DMD—an alternative to working with Vandermonde matrices. To alleviate numerical difficulties caused by the inherent ill-conditioning of the matrix $\mathbf{X}_m$ and of the eigenvalues of the companion matrix, Schmid [37] proposed using, instead of the companion matrix C_m, the Rayleigh quotient $S_m = U_m^* \mathbb{A} U_m$, where $\mathbf{X}_m = U_m \Sigma_m V_m^*$ is the SVD of $\mathbf{X}_m$. More precisely, the SVD is used to determine a numerical rank k of $\mathbf{X}_m$, and then the Rayleigh quotient is $S_k = U_k^* \mathbb{A} U_k$, where the columns of U_k are the leading k left singular vectors of $\mathbf{X}_m$, and S_k is computed without explicit use of (unknown) $\mathbb{A}$. The resulting scheme, designated as DMD (dynamic mode decomposition) and outlined in Algorithm 1 below, has become a tool of the trade in computational fluid dynamics.

Algorithm 1. $[Z_k, \Lambda_k] = \mathrm{DMD}(\mathbf{X}_m, \mathbf{Y}_m)$.

Input:

- $\mathbf{X}_m = (\mathbf{x}_1, \ldots, \mathbf{x}_m), \mathbf{Y}_m = (\mathbf{y}_1, \ldots, \mathbf{y}_m) \in \mathbb{C}^{n\times m}$ that define a sequence of snapshot pairs $(\mathbf{x}_i, \mathbf{y}_i \equiv \mathbb{A}\mathbf{x}_i)$. (Tacit assumption is that n is large and that $m \ll n$.)

1: $[U, \Sigma, \Phi] = svd(\mathbf{X}_m)$; {*The thin SVD:* $\mathbf{X}_m = U\Sigma\Phi^*$, $U \in \mathbb{C}^{n\times m}$, $\Sigma = \mathrm{diag}(\sigma_i)_{i=1}^m$}.
2: Determine numerical rank k.
3: Set $U_k = U(:, 1:k)$, $\Phi_k = \Phi(:, 1:k)$, $\Sigma_k = \Sigma(1:k, 1:k)$.
4: $S_k = ((U_k^* \mathbf{Y}_m)\Phi_k)\Sigma_k^{-1}$; {*Schmid's formula for the Rayleigh quotient* $U_k^* \mathbb{A} U_k$}.
5: $[B_k, \Lambda_k] = \mathrm{eig}(S_k)$ {$\Lambda_k = \mathrm{diag}(\lambda_j)_{j=1}^k$; $S_k B_k(:, j) = \lambda_j B_k(:, j)$; $\|B_k(:, j)\|_2 = 1$}. {*We always assume the generic case that* S_k *is diagonalizable, i.e., that in line* 5. *The function* `eig()` *computes the full column rank matrix* B_k *of the eigenvectors of* S_k.}
6: $Z_k = U_k B_k$ {*Ritz vectors*}.

Output: Z_k, Λ_k.

In this way, the matrix $\widehat{W}_m$ of the Ritz vectors and the amplitudes for the representation of the snapshot are computed without having to invert the Vandermonde matrix. The two approaches are algebraically equivalent, and in the following proposition we provide the details.

PROPOSITION 2.6. *The amplitudes* $\mathfrak{a}_j$ *in the decompositions* (2.6), (2.7) *can be equivalently computed as*

$$\mathfrak{a}_j \equiv \|\mathbf{X}_m \mathbb{V}_m^{-1}\mathbf{e}_j\|_2 = |Z_m^\dagger \mathbf{X}_m(:,1)|, \quad j = 1,\dots,m. \tag{2.9}$$

Proof. Let $S_m\mathbf{b}_j = \lambda_j \mathbf{b}_j$, with $\mathbf{b}_j = B_m(:,j)$, $\|\mathbf{b}_j\|_2 = 1$. Since

$$\begin{aligned} C_m = (\mathbf{X}_m^*\mathbf{X}_m)^{-1}(\mathbf{X}_m^*\mathbb{A}\mathbf{X}_m) &= (\Phi_m\Sigma_m^{-2}\Phi_m^*)(\Phi_m\Sigma_m U_m^*)\mathbb{A}(U_m\Sigma_m\Phi_m^*) \\ &= \Phi_m\Sigma_m^{-1}(U_m^*\mathbb{A}U_m)\Sigma_m\Phi_m^*, \end{aligned}$$

we have $S_m = \Sigma_m\Phi_m^* C_m \Phi_m\Sigma_m^{-1}$ and $C_m(\Phi_m\Sigma_m^{-1}\mathbf{b}_j) = \lambda_j(\Phi_m\Sigma_m^{-1}\mathbf{b}_j)$. Further, since λ_j is assumed simple, $\Phi_m\Sigma_m^{-1}\mathbf{b}_j$ and $(\mathbb{V}_m^{-1})(:,j)$ are linearly dependent, and thus $\mathbf{b}_j$ is collinear with $\Sigma_m\Phi_m^*(\mathbb{V}_m^{-1})(:,j)$. Note that

$$\|\Sigma_m\Phi_m^*(\mathbb{V}_m^{-1})(:,j)\|_2 = \|U_m\Sigma_m\Phi_m^*(\mathbb{V}_m^{-1})(:,j)\|_2 = \|\mathbf{X}_m(\mathbb{V}_m^{-1})(:,j)\|_2.$$

Since $\|\mathbf{b}_j\|_2 = 1$, it holds, with some $\psi_j \in \mathbb{R}$, that

$$\mathbf{b}_j = \Sigma_m\Phi_m^*(\mathbb{V}_m^{-1})(:,j)\frac{\mathbf{e}^{\mathbf{i}\psi_j}}{\mathfrak{a}_j},$$

and the corresponding Ritz vector $\mathbf{z}_j = Z_m(:,j)$ then reads

$$\begin{aligned} \mathbf{z}_j = U_m\mathbf{b}_j &= U_m\Sigma_m\Phi_m^*(\mathbb{V}_m^{-1})(:,j)\frac{\mathbf{e}^{\mathbf{i}\psi_j}}{\mathfrak{a}_j} \\ &= \mathbf{X}_m(\mathbb{V}_m^{-1})(:,j)\frac{\mathbf{e}^{\mathbf{i}\psi_j}}{\mathfrak{a}_j} = \widehat{W}_m(:,j)\frac{\mathbf{e}^{\mathbf{i}\psi_j}}{\mathfrak{a}_j} = W_m(:,j)\mathbf{e}^{\mathbf{i}\psi_j}. \end{aligned}$$

Hence, $Z_m = W_m\Psi$, $\Psi = \mathrm{diag}(\mathbf{e}^{\mathbf{i}\psi_j})_{j=1}^m$, and we seek reconstruction in the form

$$\mathbf{X}_m = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots & \mathbf{z}_m \end{pmatrix} \begin{pmatrix} \widetilde{\mathfrak{a}}_1 & & & \\ & \widetilde{\mathfrak{a}}_2 & & \\ & & \ddots & \\ & & & \widetilde{\mathfrak{a}}_m \end{pmatrix} \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{m-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{m-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \lambda_m & \dots & \lambda_m^{m-1} \end{pmatrix}. \tag{2.10}$$

Note that such a decomposition exists by (2.6), and (2.10) is just a way of expressing it in terms of Algorithm 1. If we equate the first columns on both sides in (2.10), then solving for the $\widetilde{\mathfrak{a}}_j$'s and using (2.6) yields[1]

$$(\widetilde{\mathfrak{a}}_j)_{j=1}^m = Z_m^\dagger \mathbf{X}_m(:,1) = \Psi^* W_m^\dagger \mathbf{X}_m(:,1) = \Psi^* \cdot (\mathfrak{a}_j)_{j=1}^m = (\mathbf{e}^{-\mathbf{i}\psi_j}\mathfrak{a}_j)_{j=1}^m. \tag{2.11}$$

In terms of the quantities computed by Schmid's method, the representations (2.6), (2.7) hold with $\mathbf{z}_j\widetilde{\mathfrak{a}}_j$ instead of $w_j\mathfrak{a}_j$. As discussed in Remark 2.3, we can scale $\mathbf{z}_j$ by $\mathbf{e}^{-\mathbf{i}\psi_j} = \widetilde{\mathfrak{a}}_j^*/|\widetilde{\mathfrak{a}}_j|$ and replace $\widetilde{\mathfrak{a}}_j$ with $\mathfrak{a}_j = |\widetilde{\mathfrak{a}}_j|$. ∎

If Algorithm 1 uses $k < m$, then S_k is a Rayleigh quotient of C_m, and the columns of Z_k are not the same Ritz vectors as in W_m from (2.5). For more details on this connection, we refer the reader to [14].

[1] Note that $Z_m^\dagger Z_m = I_m$.

Remark 2.7. In the framework of Schmid's DMD, the amplitudes are usually determined by the formula (2.11). Since $Z_m = U_m B_m$ (line 6 in Algorithm 1, with $k = m$) and $U_m^* U_m = I_m$, instead of applying the pseudoinverse of the explicitly computed Z_m, we use the more efficient formula

(2.12)
$$Z_m^\dagger \mathbf{X}_m(:,1) = B_m^{-1}(U_m^* \mathbf{X}_m(:,1)) = B_m^{-1} U_m^* U_m \Sigma_m \Phi_m^* \mathbf{e}_1 = B_m^{-1}(\Sigma_m \Phi_m(1,:)^*).$$

Recall that we assume that all λ_j's are mutually distinct, so B_m is of full rank. Since it can be ill-conditioned, we can use the (possibly truncated) SVD of B_m and determine that the $\widetilde{\mathfrak{a}}_j$'s are a LS solution. In the case of numerical rank deficiency in B_m, we can choose/prefer a sparse solution (instead of the least norm).

Remark 2.8. Recently, in [14] we proposed a new computational scheme—refined Rayleigh–Ritz data driven modal decomposition. It follows the DMD scheme, but it further allows data driven refinement of the Ritz vectors and computable data driven residuals.

3. Numerical algorithm for computing $\widehat{W}_m = \mathbf{X}_m \mathbb{V}_m^{-1}$. There is certain intrinsic elegance in the snapshot reconstructions (2.6), (2.7) based on the spectral structure of the companion matrix (2.2). Unfortunately, the potentially high condition number of $\mathbb{V}_m$ precludes exploiting it in numerical computations. Indeed, even in relatively simple examples we can encounter $\kappa_2(\mathbb{V}_m)$ as high as 10^{50}, 10^{70} or higher. If $\widehat{W}_m = \mathbf{X}_m \mathbb{V}_m^{-1}$ is computed in the standard double precision arithmetic with the roundoff $\varepsilon \approx 2.2 \cdot 10^{-16}$, the classical perturbation theory estimates the relative error in the computed result essentially as[2] $\varepsilon \kappa_2(\mathbb{V}_m)$ (e.g., $2.2 \cdot 10^{-16} \cdot 10^{50}$), thus rendering the output as entirely wrong and useless. Schmid's DMD provides an alternative path that avoids $\mathbb{V}_m^{-1}$, as outlined in section 2.4.

We now explore another, SVD-free, way for doing DMD analysis, based on the already available methods of numerical linear algebra [11], [10], [12], [13]. Our starting point, based on the state of the art in numerical linear algebra, is that accurate computation with notoriously ill-conditioned Vandermonde matrices may be possible despite a high classical condition number.

3.1. A review of accurate computation with Vandermonde matrices. For the reader's convenience, we briefly describe the two main components of computational schemes capable of performing linear algebra operations with $\mathbb{V}_m$ accurately in standard floating point arithmetic.

First, the DFT of a Vandermonde matrix $\mathbb{V}_m$ is a generalized Cauchy matrix parametrized by the λ_i's (the original parameters that define $\mathbb{V}_m$) and the mth roots of unity. In our setting, DFT is an allowable and actually meaningfully interpretable transformation, and the result obtained by using the auxiliary Cauchy matrix is easily back substituted into the original formulation. The details are given in section 3.1.1.

Second, in the framework of [11], [10], accurate computation with a generalized Cauchy matrix is possible (e.g., LDU decomposition, SVD) independent of its condition number.

3.1.1. Discrete Fourier transform of $\mathbb{V}_m$. Let $\mathbb{F}$ denote the DFT matrix, $\mathbb{F}_{ij} = \omega^{(i-1)(j-1)}/\sqrt{m}$, where $\omega = \mathbf{e}^{2\pi \mathrm{i}/m}$, $\mathrm{i} = \sqrt{-1}$. Now, recall that DFT transforms

[2] Up to a factor that is polynomial in the dimensions of the problem.

Vandermonde matrices into Cauchy as follows:

$$(\mathbb{V}_m\mathbb{F})_{ij} = \left[\frac{\lambda_i^m - 1}{\sqrt{m}}\right]\left[\frac{1}{\lambda_i - \omega^{1-j}}\right]\left[\omega^{1-j}\right] \equiv (\mathcal{D}_1)_{ii}\,\mathcal{C}_{ij}\,(\mathcal{D}_2)_{jj},\ 1 \le i,j \le m. \tag{3.1}$$

In other words, $\mathbb{V}_m\mathbb{F} = \mathcal{D}_1\mathcal{C}\mathcal{D}_2$, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are diagonal, and $\mathcal{C}$ is a Cauchy matrix. Note, in addition, that $\mathcal{D}_2 = \mathrm{diag}(\omega^{1-j})_{j=1}^m$ is unitary. For more details see [10].

To avoid singularity (i.e., an expression of the form 0/0) if in (3.1) some λ_i's equal an mth root of unity, we proceed by following [10], as follows. If $\lambda_i = \omega^{1-j}$ for some index j, write $\lambda_i^m - 1 = \prod_{k=1}^m(\lambda_i - \omega^{1-k})$ and replace (3.1) with the equivalent formula for the ith row,

$$(\mathbb{V}_m\mathbb{F})_{ij} = \underbrace{\frac{1}{\sqrt{m}}}_{(\mathcal{D}_1)_{ii}} \prod_{\substack{k=1\\k\neq j}}^{m}(\lambda_i - \omega^{1-k})\,\underbrace{\omega^{1-j}}_{(\mathcal{D}_2)_{jj}}, \quad (\mathbb{V}_m\mathbb{F})_{ik} = 0 \ \text{ for } k \neq j. \tag{3.2}$$

If $\varsigma = (\varsigma_1, \ldots, \varsigma_\ell)$ is a subsequence of $(1, \ldots, n)$, and if $\mathbb{V}_{\varsigma,m}$ is an $\ell \times m$ submatrix of $\mathbb{V}_m$ consisting of the rows with indices $\varsigma_1, \ldots, \varsigma_\ell$, then (3.1), (3.2) trivially hold for the entries of $\mathbb{V}_{\varsigma,m}\mathbb{F}$.

Note that the matrices $\mathcal{D}_1$, $\mathcal{C}$, $\mathcal{D}_2$ are given implicitly by the parameters λ_i (eigenvalues, available on input) and the mth roots of unity $\zeta_j = \omega^{1-j}$, $j = 1, \ldots, m$, (easily computed to any desired precision and, e.g., tabulated in a preprocessing phase of the computation), so that the DFT $\mathbb{V}_m\mathbb{F}$ is not done by actually running an FFT. It suffices to note that the λ_i's and the roots of unity are the parameters that define $\mathbb{V}_m\mathbb{F}$ as in (3.1), (3.2).

Remark 3.1. It is interesting to see how the transformation $\mathbb{V}_m \mapsto \mathbb{V}_m\mathbb{F}$ fits into the framework of the Krylov decomposition (2.1). Postmultiply (2.1) with $\mathbb{F}$ to obtain

$$\mathbb{A}(\mathbf{X}_m\mathbb{F}) = (\mathbf{X}_m\mathbb{F})\mathbb{F}^*(\mathbb{V}_m^{-1}\Lambda_m\mathbb{V}_m)\mathbb{F} + \mathbf{r}_{m+1}\mathbf{e}_m^T\mathbb{F} \tag{3.3}$$

and then, using (3.1), $\mathbb{F}^*C_m\mathbb{F} = \mathbb{F}^*(\mathbb{V}_m^{-1}\Lambda_m\mathbb{V}_m)\mathbb{F} = \mathcal{D}_2^*\mathcal{C}^{-1}\mathcal{D}_1^{-1}\Lambda_m\mathcal{D}_1\mathcal{C}\mathcal{D}_2 = \mathcal{D}_2^*\mathcal{C}^{-1}\Lambda_m\mathcal{C}\mathcal{D}_2$ and

$$\mathbb{A}(\mathbf{X}_m\mathbb{F}) = (\mathbf{X}_m\mathbb{F})((\mathcal{C}\mathcal{D}_2)^{-1}\Lambda_m(\mathcal{C}\mathcal{D}_2)) + \mathbf{r}_{m+1}\mathbf{e}_m^T\mathbb{F} \ \text{ or, equivalently,} \tag{3.4}$$

$$\mathbb{A}(\mathbf{X}_m\mathbb{F}\mathcal{D}_2^*) = (\mathbf{X}_m\mathbb{F}\mathcal{D}_2^*)(\mathcal{C}^{-1}\Lambda_m\mathcal{C}) + \mathbf{r}_{m+1}\mathbf{e}_m^T\mathbb{F}\mathcal{D}_2^*. \tag{3.5}$$

If we think of each row $\mathbf{X}_m(i,:)$ as a time trajectory of the corresponding observable, then $\mathbf{X}_m(i,:)\mathbb{F}$ represents its image in the frequency domain, and (3.4), (3.5) is the corresponding Krylov decomposition. For another insightful connection between (data centered) DMD and temporal DFT see [9].

3.1.2. Rank revealing LDU decomposition of $\mathcal{D}_1\mathcal{C}\mathcal{D}_2$. Applying the DFT to $\mathbb{V}_m$ in order to avoid the ill-conditioning of $\mathbb{V}_m$ may seem a futile effort—since $\mathbb{F}$ is unitary, $\kappa_2(\mathcal{D}_1\mathcal{C}\mathcal{D}_2) = \kappa_2(\mathbb{V}_m\mathbb{F}) = \kappa_2(\mathbb{V}_m)$. Further, Cauchy matrices are also notoriously ill-conditioned, so, in essence, we have traded one badly conditioned structure for another.

Example 3.2. The best known example of an ill-conditioned Cauchy matrix is the Hilbert matrix, $H_{ij} = 1/(i + j - 1)$. For instance, the condition number of the 100×100 Hilbert matrix satisfies $\kappa_2(H) > 10^{150}$. Ill-conditioning is not always obvious in the sizes of its entries—the entries of the 100×100 Hilbert matrix

range from $1/199 \approx 5.025 \cdot 10^{-3}$ to 1. Moreover, in MATLAB, `cond(hilb(100))` returns `ans=4.622567959141155e+19`. We should keep in mind that the matrix condition number is a matrix function with its own condition number. By a result of Higham [23], the condition number of the condition number is the condition number itself, meaning that our computed condition number, if it is above $1/\varepsilon$ (in MATLAB, `1/eps=4.503599627370496e+15`), it might be entirely wrongly computed. This may lead to an underestimate of extra precision needed to handle the ill-conditioning.

Although we have not changed the condition number, we have changed the representation of the data, which will allow more accurate computation. The key numerical advantage of this change of variables is in the fact that for any two diagonal matrices $\mathcal{D}_1$, $\mathcal{D}_2$ and any Cauchy matrix $\mathcal{C}$, the pivoted LDU decomposition

$$\Pi_1(\mathcal{D}_1\mathcal{C}\mathcal{D}_2)\Pi_2 = L\Delta U \tag{3.6}$$

can be computed by a specially tailored algorithm so that all entries of L, Δ, U, even the tiniest ones, are computed to nearly machine precision accuracy, no matter how high the condition number of $\mathcal{D}_1\mathcal{C}\mathcal{D}_2$. More precisely, if $\widetilde{L}$, $\widetilde{\Delta}$, and $\widetilde{U}$ are the computed matrices, then, for all i, j,

$$|\widetilde{L}_{ij} - L_{ij}| \leq \epsilon|L_{ij}|, \;\; |\widetilde{\Delta}_{ii} - \Delta_{ii}| \leq \epsilon|\Delta_{ii}|, \;\; |\widetilde{U}_{ij} - U_{ij}| \leq \epsilon|U_{ij}|, \tag{3.7}$$

where $L\Delta U$ is the pivoted LDU decomposition that is computed exactly from the stored parameters. Essentially, if we take the stored (in the machine memory) eigenvalues λ_i and the roots of unity ω^{1-j} as our initial data, the first errors committed in the floating point LDU decomposition (3.6) are the entrywise small forward errors (3.7). This is achieved by avoiding subtractions of intermediate results and using clever updates of the Schur complements; see [10].

Moreover, as a consequence of pivoting, the matrix L (lower triangular with unit diagonal) and the matrix U (upper triangular with unit diagonal) are well conditioned. All ill-conditioning is conspicuously exposed on the diagonal of Δ. For instance, in the case of the Hilbert matrix from Example 3.2, $\kappa_2(L) = \kappa_2(U) \approx 72.34$ and $\kappa_2(\Delta) \approx 10^{149}$.

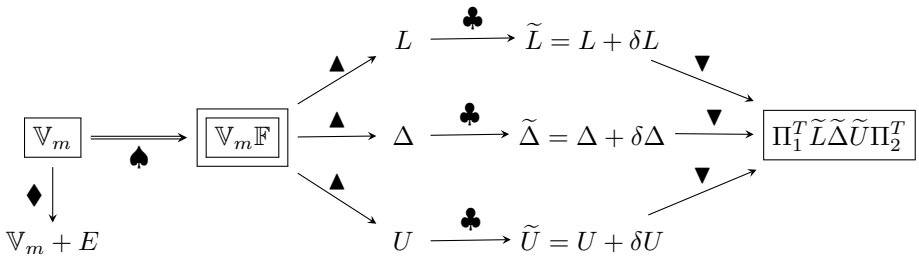


FIG. 3.1. *Reparametrization of $\mathbb{V}_m$ via the DFT and LDU in floating point arithmetic. Legend: ♠ = the DFT of $\mathbb{V}_m$ using the explicit formulas* (3.1)*; ▲ = the pivoted LDU* (3.6) *of $\mathbb{V}_m\mathbb{F}$ by explicit forward stable updates of the Schur complements* [10]*; ♣ = forward errors in the computed factors $\widetilde{L}$, $\widetilde{\Delta}$, $\widetilde{U}$, bounded as in* (3.7)*; ▼ = implicit representation of $\mathbb{V}_m\mathbb{F}$ as the product $\Pi_1^T\widetilde{L}\widetilde{\Delta}\widetilde{U}\Pi_2^T$; ♦ = direct computation with $\mathbb{V}_m$, using standard algorithms, produces backward error E that is small in matrix norm, and the condition number is $\kappa_2(\mathbb{V}_m)$.*

A MATLAB implementation of the decomposition (3.6) is provided in Algorithm 2 of the appendix (section 8), and a scheme of the error analysis is given in Figure 3.1. For detailed analysis we refer to [10], [12], [13].

3.2. Cauchy matrix based reconstruction (in the frequency domain). We now revise relation (2.4) and write $\mathbf{X}_m\mathbb{V}_m^{-1}$ as $(\mathbf{X}_m\mathbb{F})(\mathbb{V}_m\mathbb{F})^{-1}$. This mere insertion of the identity $\mathbb{F}\mathbb{F}^{-1}$ between $\mathbf{X}_m$ and $\mathbb{V}_m^{-1}$ allows a natural interpretation: the time trajectories of the observables (the rows of $\mathbf{X}_m$) have been mapped to the frequency domain, and the inverse Vandermonde matrix of the eigenvectors of C_m has been changed with the inverse of the generalized Cauchy matrix $\mathcal{D}_1\mathcal{C}\mathcal{D}_2$ (the eigenvectors of $\mathbb{F}^*C_m\mathbb{F}$); see Remark 3.1.

Following section 3.1.2, compute the LDU decomposition with complete pivoting $\Pi_1(\mathcal{D}_1\mathcal{C}\mathcal{D}_2)\Pi_2 = L\Delta U$, and then apply $\mathbb{V}_m^{-1}$ through backward and forward substitutions,

$$\widehat{W}_m = (((((\mathbf{X}_m\mathbb{F})\Pi_2)U^{-1})\Delta^{-1})L^{-1})\Pi_1. \tag{3.8}$$

The implementation of this formula depends on a particular software tool. For the reader's convenience, in Algorithm 3 we show a simple MATLAB version,[3] where the function `Vand_DFT_LDU()` computes the decomposition (3.6).

This is obviously more complicated than "backslashing" in MATLAB (i.e., $\widehat{W}_m = \mathbf{X}_m/\mathbb{V}_m$) and not as efficient as the fast Vandermonde inversion techniques, such as the Björck–Pereyra type algorithms that solve the Vandermonde system with $O(m^2)$ complexity. However, Algorithm 3, when combined with Algorithm 2, provides superior accuracy, independent of the distribution of the λ_i's. For an error analysis we refer the reader to [13].

Besides ill-conditioning, one general difficulty with Vandermonde matrices is that the powers λ_i^j may spread, in absolute value, hundreds of orders of magnitude and thus underflows and overflows may cause irreparable exceptions in machine arithmetic. An interesting salient feature of the transformation (3.1) is that the powers of λ_i are changed into $1/(\lambda_i - \omega^{j-1})$ where the only powers are those of the primitive mth root of unity ω; the powers of λ_i are extracted in the form $(\lambda_i^m - 1)/\sqrt{m}$ on the diagonal of the scaling matrix $\mathcal{D}_1$, where they can be kept until the very end of the computation. Namely, since our goal is to compute $\widehat{W}_m = \mathbf{X}_m\mathbb{V}_m^{-1}$ and its column norms, we can rephrase the previous formulas as

$$\widehat{W}_m = \left[(\mathbf{X}_m\mathbb{F})(\mathcal{C}\mathcal{D}_2)^{-1}\right]\mathcal{D}_1^{-1}$$

and compute $\widetilde{W}_m = \left[(\mathbf{X}_m\mathbb{F})(\mathcal{C}\mathcal{D}_2)^{-1}\right]$ by using the LDU of $\mathcal{C}\mathcal{D}_2$, instead of (3.6), and the forward and backward substitutions analogously to (3.8). The diagonal entries of $\mathcal{D}_1^{-1}$ will then be assimilated as multiplicative factors in the computation of $\mathfrak{a}_j$; see (2.5). (Analogously, we can write $\widetilde{W}_m = (\mathbf{X}_m\mathbb{F}\mathcal{D}_2^*)\mathcal{C}^{-1}$ and invert $\mathcal{C}$ via its pivoted LDU, but this does not make a big difference since $\mathcal{D}_2$ is unitary.) This modification can be easily implemented by minor modifications in Algorithms 2 and 3 (section 8); we omit the details for the sake of brevity.

3.2.1. SVD based reconstruction. Regularization. Furthermore, based on the decomposition (3.6), computed to high accuracy (3.7), we can compute accurate SVD of the product $L\Delta U$, $L\Delta U = \Omega\Sigma\Theta^*$, which gives an accurate SVD of $\mathbb{V}_m$, $\mathbb{V}_m = (\Pi_1^T\Omega)\Sigma(\mathbb{F}\Pi_2\Theta)^* \equiv \mathcal{U}\Sigma\mathcal{V}^*$. For details of the algorithm we refer the reader to [16], [11], [10], [17], [18].

Now, instead of computing $\mathbf{X}_m\mathbb{V}_m^{-1}$ as a solution of a linear system of equations, with the inverse $\mathbb{V}_m^{-1} = \mathcal{V}\Sigma^{-1}\mathcal{U}^*$, we change the framework into a regularized LS

[3] Note that we have adapted postmultiplication by $\mathbb{F}$ to the MATLAB definition of the functions `fft()`, `ifft()`.

solution. Let $\varphi_j \geq 0$ be filter factors, and let

$$\Sigma_\varphi^\dagger = \operatorname{diag}\left(\varphi_1 \frac{1}{\sigma_1}, \ldots, \varphi_m \frac{1}{\sigma_m}\right). \tag{3.9}$$

For instance, the commonly used regularization is

$$\Sigma_\varphi^\dagger = \operatorname{diag}\left(\frac{\sigma_1}{\sigma_1^2+\eta^2}, \ldots, \frac{\sigma_m}{\sigma_m^2+\eta^2}\right), \quad \text{where } \varphi_i = \frac{\sigma_i^2}{\sigma_i^2+\eta^2}. \tag{3.10}$$

Then we use the approximation $\mathbf{X}_m \mathbb{V}_m^{-1} \approx \mathbf{X}_m \mathcal{V} \Sigma_\varphi^\dagger \mathcal{U}^* = (\mathbf{X}_m \mathbb{F}\Pi_2)\Theta \Sigma_\varphi^\dagger \mathcal{U}^* \equiv (\mathbf{X}_m \mathbb{F}\Pi_2)\Theta \Sigma_\varphi^\dagger \Omega^* \Pi_1$. The key is that we have an accurate SVD even with extremely large σ_1/σ_m, and that with the tuning parameter $\eta \geq 0$ we can control the influence of the smallest singular values. This may be important in the case of noisy data—an issue that will be addressed in our future work.

4. Numerical example: A case study. To illustrate the preceding discussion, we use the simulation data of a 2D model obtained by depth averaging the Navier–Stokes equations for a shear flow in a thin layer of electrolyte suspended on a thin lubricating layer of a dielectric fluid; see [40], [39] for a more detailed description of the experimental setup and numerical simulations.[4]

The (scalar) vorticity field data consist of n_t snapshots of dimensions $n_x \times n_y$; in this particular example $n_t = 1201 \equiv m+1$, $n_x = n_y = 128$. The $n_x \times n_y \times n_t$ tensor is matricized into $n_x \cdot n_y \times n_t$ matrix $(\mathbf{f}_1, \ldots, \mathbf{f}_{n_t})$, and $\mathbf{X}_m$ is of dimensions 16384×1200.

The computational schemes are tested with respect to reconstruction potential as follows: for a given snapshot $\mathbf{f}_i$, the representation (2.6) is truncated by taking a given number of modes with absolutely largest amplitudes $|\mathfrak{a}_j|$ (abbreviated as *dominant modes*). We note that, in general, selecting the most appropriate modes is a separate nontrivial problem, not considered here.

Our ultimate goal is to develop accurate computation (to the extent deemed possible by the perturbation theory) of the Ritz vectors $\widehat{W}_m = \mathbf{X}_m \mathbb{V}_m^{-1}$ and their use for snapshot reconstruction, independent of the distribution of the λ_i's and independent of the range of their moduli, including the case of $\max_i |\lambda_i| / \min_i |\lambda_i| \gg 1$. This will provide a kernel for a new computational tool for Koopman spectral analysis, based on the natural Krylov decomposition.

4.1. Ill-conditioning of $\mathbb{V}_m$ and diagonal scalings. In the first experiment, we illustrate the problem caused by the high condition number of $\mathbb{V}_m$, and also the effects of simple diagonal scalings. More precisely, we attempt reconstruction of the snapshots as outlined in section 2.1.1, using the modes computed by

1. inversion of the Vandermonde matrix by the backslash operator in MATLAB;
2. inversion of the row scaled Vandermonde matrix by the backslash operator in MATLAB: $\mathbb{V}_m = D_r \mathbb{V}_m^{(r)}$, $\widehat{W}_m = (\mathbf{X}_m (\mathbb{V}_m^{(r)})^{-1}) D_r^{-1}$, where $D_r = \operatorname{diag}(\|\mathbb{V}_m(i,:)\|)_{i=1}^m$;
3. inversion of the column scaled Vandermonde matrix by the backslash operator in MATLAB: $\mathbb{V}_m = \mathbb{V}_m^{(c)} D_c$, $\widehat{W}_m = (\mathbf{X}_m D_c^{-1})(\mathbb{V}_m^{(c)})^{-1}$, where $D_c = \operatorname{diag}(\|\mathbb{V}_m(:,i)\|)_{i=1}^m$.

[4] We thank Michael Schatz, Balachandra Suri, Roman Grigoriev, and Logan Kageorge from the Georgia Institute of Technology for providing us with the data.

We choose the scaling in the ℓ_2-norm, i.e., $\|\cdot\| = \|\cdot\|_2$. The relevant condition numbers, estimated using the MATLAB function `cond()`, are as follows:

$$\texttt{cond}(\mathbb{V}_m) \approx 8.9 \cdot 10^{76}, \quad \texttt{cond}(\mathbb{V}_m^{(r)}) \approx 3.1 \cdot 10^{7}, \quad \texttt{cond}(\mathbb{V}_m^{(c)}) \approx 3.0 \cdot 10^{21}. \tag{4.1}$$

We can also scale in the $\|\cdot\|_\infty$ or $\|\cdot\|_1$-norm, with an effect similar to that in (4.1). (It is known that this scaling, which equilibrates the columns or rows, is nearly optimal in the class of all diagonal scalings; see [43].) This is instructive; we see that the condition number of $\mathbb{V}_m$ can indeed be high (much higher than $1/\varepsilon$) and that certain diagonal scalings may reduce it enough to allow sufficiently accurate computations in machine working precision ε (here assumed to 16 decimal digits).

In Figure 4.1, we display reconstruction results for $\mathbf{f}_{321}$, using 300 dominant modes. The snapshots are visualized using the contour plots over the 2D domain. The results are consistent with (4.1)—with the condition number above $1/\varepsilon$, we do not expect any reasonable accuracy.

The key to the success of row scaling of $\mathbb{V}_m$ is that in this case the distribution of the Ritz values λ_i is such that it allows improvement of the condition number by row scaling. Although this will not be the case in general, it provides a good case study example for numerical analysts. It should be noted that $\mathbb{V}_m^{-1} = (\mathbb{V}_m^{(r)})^{-1} D_r^{-1}$ ($\widehat{W}_m = (\mathbf{X}_m(\mathbb{V}_m^{(r)})^{-1})D_r^{-1}$) is just a different scaling of the eigenvectors of C_m (Ritz vectors of $\mathbb{A}$) that is compensated by the corresponding scaling of the amplitudes; this scaling is, as an allowable transformation, an invariant of the representation.

On the other hand, the column scaling of $\mathbb{V}_m$ did not reduce the condition number enough to ensure accurate inversion, although the reduction was by more than 50 orders of magnitude. Also, this scaling is not interpretable as an invariant of the reconstruction; note that it rescales the snapshots.

It is interesting that in the case of taking all, or almost all, modes, the reconstruction by simply backslashing $\mathbb{V}_m$ provides perfect reconstruction; see Figure 4.2. (This can be explained by the fact that even an inaccurate solution of a linear system of equations may have a small residual.)

For an analysis of optimally selected real nodes and the potential improvement of the condition number of Vandermonde matrices by scaling, we refer the reader to [22] with the caveat that *our nodes are arbitrary complex numbers.*

4.2. Comparing Björck–Pereyera, DMD, and DFT based reconstruction. Next, we perform reconstruction using the following three methods:

1. companion matrix formulation with the Björck–Pereyera method [6] for Vandermonde systems. Although forward stable in the special case of real and ordered λ_i's, this method may be very sensitive in the case of general complex λ_i's and relatively large dimension m.
2. companion matrix formulation with the DFT and inversion of the Cauchy matrix, as described in section 3. Since $\mathbb{F}$ and $\mathcal{D}_2$ in (3.1) are unitary, Algorithm 3 solves a linear system with the matrix $\mathcal{D}_1\mathcal{C} = \mathbb{V}_m\mathbb{F}\mathcal{D}_2^*$ of condition number bigger than 10^{76}. No additional scaling is used; we want to check the claim that such a high condition number cannot spoil the result.
3. Schmid's DMD method as described in section 2.4. Here we expect good reconstruction results, provided it is feasible for given data and the parameters. The SVD is not truncated because the spectral condition number of $\mathbf{X}_m$ is approximately $5.5 \cdot 10^{10}$ ($\sigma_{\max}(\mathbf{X}_m) \approx 4.2 \cdot 10^3$, $\sigma_{\min}(\mathbf{X}_m) \approx 7.5 \cdot 10^{-8}$).

The reconstruction results for $\mathbf{f}_{321}$, shown in Figure 4.3, indicate that row scaled $\mathbb{V}_m$ based computation, as well as DFT based and Schmid's DMD, are capable of reconstructing the snapshots with a relatively small number of dominant modes.

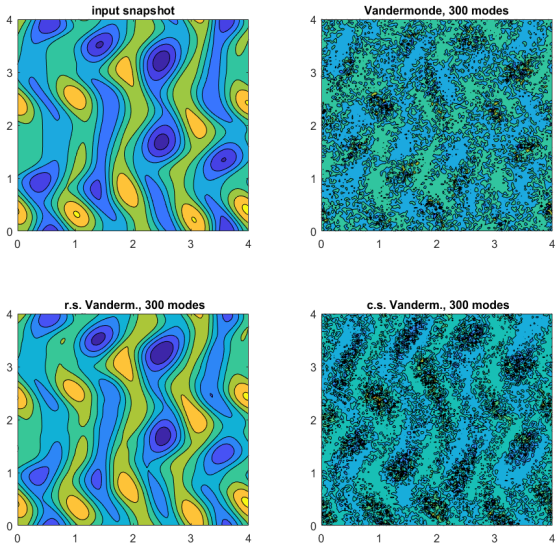


FIG. 4.1. *Reconstruction of* $\mathbf{f}_{321}$ *using* 300 *dominant modes. The linear systems are solved in MATLAB using the backslash operator. Note how using the row scaled* $\mathbb{V}_m^{(r)}$ *improves the reconstruction (bottom left plot), while backslashing the original* $\mathbb{V}_m$ *and the column scaled matrix* $\mathbb{V}_m^{(c)}$ *yields poor results (plots in the right column). A similar effect is observed when reconstructing the other* $\mathbf{f}_i$*'s.*

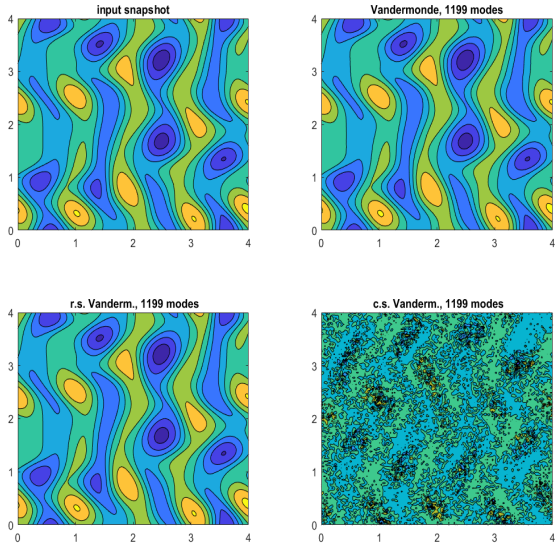


FIG. 4.2. *Reconstruction of* $\mathbf{f}_{321}$ *using* 1199 *dominant modes. The linear systems are solved in MATLAB using the backslash operator. Note that backslashing the original* $\mathbb{V}_m$ *and using a larger number of modes resulted in good reconstruction.*

The Björck–Pereyera method was not successful, and increasing the number of modes brings no improvement; see Figure 4.4.

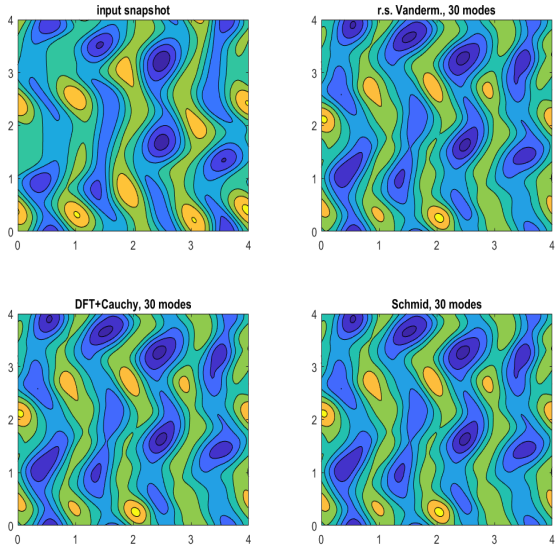


FIG. 4.3. *Reconstruction of* $\mathbf{f}_{321}$ *using* 30 *dominant modes. The row scaled Vandermonde and the DFT+Cauchy inversion, as well as Schmid's DMD reconstruction (second row), succeeded in reconstructing* $\mathbf{f}_{321}$ *using* 30 *modes with dominant amplitudes.*

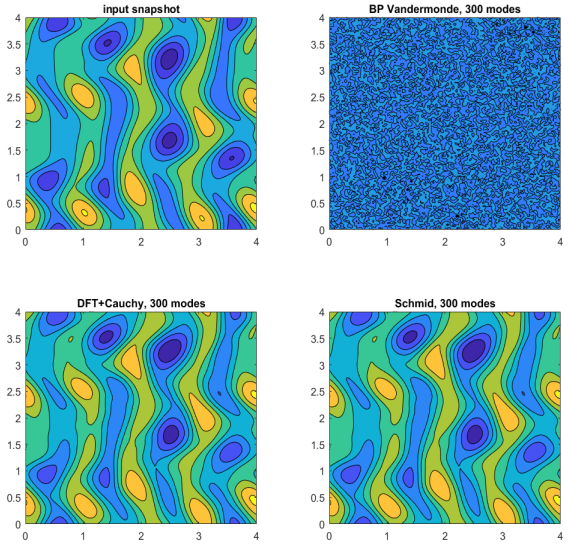


FIG. 4.4. *Reconstruction of* $\mathbf{f}_{321}$ *using* 300 *dominant modes. The Björck–Pereyera method (top right plot) failed to produce any useful data. The DFT+Cauchy inversion and Schmid's DMD reconstruction (second row) succeeded in reconstructing* $\mathbf{f}_{321}$ *pretty well, using* 300 *modes with dominant amplitudes.*

Increasing the number of modes in the reconstruction further may or may not bring improvement, as seen in Figure 4.5. At first, it may come as a surprise that the

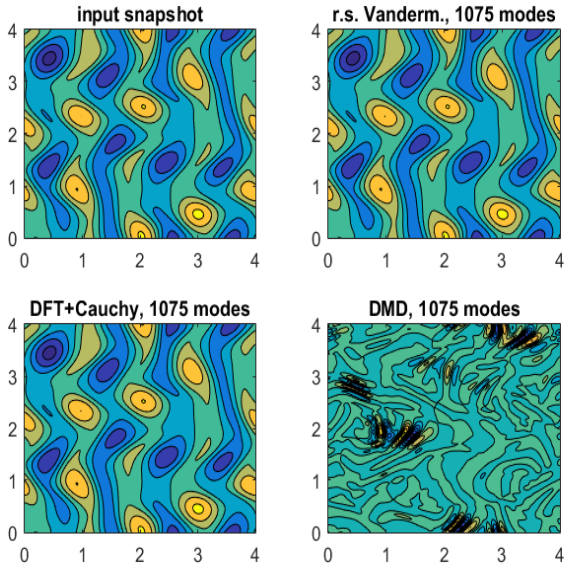


FIG. 4.5. *Reconstruction of* $\mathbf{f}_{1111}$ *using* 1075 *dominant modes. The DFT+Cauchy inversion did well, and the Schmid's DMD reconstruction (second row) unfortunately failed. (With* 1074 *modes DMD performed well, but starting with* 1075 *all reconstructions failed, including the one with all* 1200 *modes.)*

SVD based DMD starts failing after taking more and more nodes. On the other hand, we must take into account that the left singular vectors of the matrix $\mathbf{X}_m$ as well as the eigenvectors of the Rayleigh quotient S_k (lines 1 and 5 in the DMD Algorithm 1), may be computed inaccurately as their sensitivity depends on the condition numbers of $\mathbf{X}_m$ and S_k (line 4) and the gaps in the singular values and the eigenvalues, respectively.

Remark 4.1. The result shown in Figure 4.5, showing unsuccessful reconstruction by the SVD based DMD starting at mode 1075, seems surprising. On the other hand, in DMD, the modes are computed from the product of the left singular vector matrix of the snapshots and the eigenvectors of the Rayleigh quotient; both sets of vectors are sensitive to small gaps in the spectrum and may not be computed to high accuracy; recall Remark 2.5. For the sensitivity of the singular vectors see [26].

4.3. Reconstruction in the QR compressed space. An efficient method of reducing the dimension of the ambient space is the QR compressed scheme [14, sect. 5.2.1], which reduces n to $m+1$ via the QR factorization of $\mathbf{X}_{m+1} = (\mathbf{X}_m, \mathbf{f}_{m+1})$. We briefly review the main idea; for more details and the general case of nonsequential data $\mathbf{X}_m$ and $\mathbf{Y}_m = \mathbb{A}\mathbf{X}_m$ we refer the reader to [14].

In the QR factorization

$$\mathbf{X}_{m+1} = Q_f \begin{pmatrix} R_f \\ 0 \end{pmatrix} = \widehat{Q}_f R_f, \quad Q_f^* Q_f = I_n, \quad \widehat{Q}_f = Q_f(:, 1:m+1), \tag{4.2}$$

where R_f is $(m+1)\times(m+1)$ upper triangular, it holds that $\mathrm{range}(\widehat{Q}_f) \supseteq \mathrm{range}(\mathbf{X}_{m+1})$. (Clearly, $\mathrm{range}(\widehat{Q}_f) = \mathrm{range}(\mathbf{X}_{m+1})$ if and only if $\mathbf{X}_{m+1}$ is of full column rank.) Hence, we can reparametrize the data and work in the new representation in the basis defined by the columns of $\widehat{Q}_f$. To that end, set $\mathbf{Y}_m = \mathbb{A}\mathbf{X}_m = (\mathbf{f}_2, \ldots, f_{m+1})$,

$R_x = R_f(:, 1:m)$, $R_y = R_f(:, 2:m+1)$. Then

(4.3)

$$\mathbf{X}_m = \widehat{Q}_f R_x,\ \mathbf{Y}_m = \widehat{Q}_f R_y;\ R_f = \begin{pmatrix} \times & * & * & * & \div \\ & * & * & * & \div \\ & & * & * & \div \\ & & & * & \div \\ & & & & \div \end{pmatrix},\ R_x = \begin{pmatrix} \times & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \\ & & & 0 \end{pmatrix},\ R_y = \begin{pmatrix} * & * & * & \div \\ * & * & * & \div \\ & * & * & \div \\ & & * & \div \\ & & & \div \end{pmatrix},$$

and, in the basis of the columns of $\widehat{Q}_f$, we can identify $\mathbf{X}_m \equiv R_x$, $\mathbf{Y}_m \equiv R_y$; i.e., we can think of $\mathbf{X}_m$ and $\mathbf{Y}_m$ as m snapshots in an $(m+1)$ dimensional space.

Further, the coefficients $\mathbf{c} = (c_i)_{i=1}^m$ of the companion matrix C_m are computed as

$$\mathbf{c} = R_x^\dagger R_y(:, m) = R_x(1:m, 1:m)^{-1} R_y(1:m, m) \equiv R_f(1:m, 1:m)^{-1} R_f(1:m, m{+}1),$$

where the explicit use of R_x^{-1} assumes the full rank case. Numerically, we would solve the LS problem $\|R_x \mathbf{c} - R_y(:, m)\|_2 \to \min$ by an appropriate method (including refinement).

In this representation, if we set $\widehat{\mathbf{f}}_i \equiv R_x(:, i)$, and if we have a computed reconstruction $\widetilde{\widehat{\mathbf{f}}}_i \approx \widehat{\mathbf{f}}_i$, then $\widehat{Q}_f \widetilde{\widehat{\mathbf{f}}}_i \approx \mathbf{f}_i$, with the error $\|\widehat{Q}_f \widetilde{\widehat{\mathbf{f}}}_i - \widehat{Q}_f \widehat{\mathbf{f}}_i\|_2 = \|\widetilde{\widehat{\mathbf{f}}}_i - \widehat{\mathbf{f}}_i\|_2$. Hence we can run the entire process (including the initial DMD) in the basis $\widehat{Q}_f$ and switch to the original representation at the very end. Since the factorization (4.2) is available in high performance software implementations, the overall procedure is more efficient in particular in the cases when $m \ll n$.

Let us now try this scheme using the test data from sections 4.1 and 4.2. As expected, the compressed scheme computes considerably faster. The numerical results are similar, with some minor variations. To illustrate, we again reconstruct $\mathbf{f}_{321}$ and $\mathbf{f}_{1111}$.

The results shown in Figure 4.6, when compared to Figure 4.1, seem to indicate that pure Vandermonde inversion, while still performing poorly, shows some improvement. This is probably due to the fact that the Vandermonde inverse is applied to a triangular matrix; we omit the discussion for the sake of brevity. (The DMD and the DFT+Cauchy algorithms produced results similar to those in Figure 4.3.)

With the snapshot $\mathbf{f}_{1111}$ (see Figure 4.5), the QR compressed DMD started failing at using 1073 dominant modes; the DFT+Cauchy method performed well, as before.

5. Snapshot reconstruction: Theoretical insights. It is desirable to have a representation analogous to (2.10) but with a smaller number of eigenmodes and with representation error as small as possible. Suppose that we use ℓ modes, $\ell < m$, and that the Ritz pairs are so enumerated that the selected ones are $(\lambda_1, \mathbf{z}_1), \ldots, (\lambda_\ell, \mathbf{z}_\ell)$. We do not specify how the pairs $(\lambda_i, \mathbf{z}_i)$ have been computed—the $\mathbf{z}_i$'s can be the Ritz vectors or, e.g., the refined Ritz vectors [14]. The selection of the particular ℓ pairs can be guided, e.g., by the sparsity promoting DMD [25]. The goal is to determine the coefficients $\alpha_1, \ldots, \alpha_\ell$ that minimize the following LS reconstruction error over all snapshots:

$$\sum_{i=1}^{m} \|\mathbf{f}_i - \sum_{j=1}^{\ell} \mathbf{z}_j \alpha_j \lambda_j^{i-1}\|_2^2 \longrightarrow \min. \tag{5.1}$$

If we set $Z_\ell = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_\ell)^T$, $\Delta_{\boldsymbol{\alpha}} = \mathrm{diag}(\boldsymbol{\alpha})$, $\Lambda_j = (\lambda_1^{j-1}, \ldots, \lambda_\ell^{j-1})^T$, and $\Delta_{\Lambda_j} = \mathrm{diag}(\Lambda_j)$, then the objective (5.1) can be written as

$$\Omega^2(\boldsymbol{\alpha}) \equiv \|\mathbf{X}_m - Z_\ell \Delta_{\boldsymbol{\alpha}} \begin{pmatrix} \Lambda_1 & \Lambda_2 & \ldots & \Lambda_m \end{pmatrix} \|_F^2 \longrightarrow \min. \tag{5.2}$$

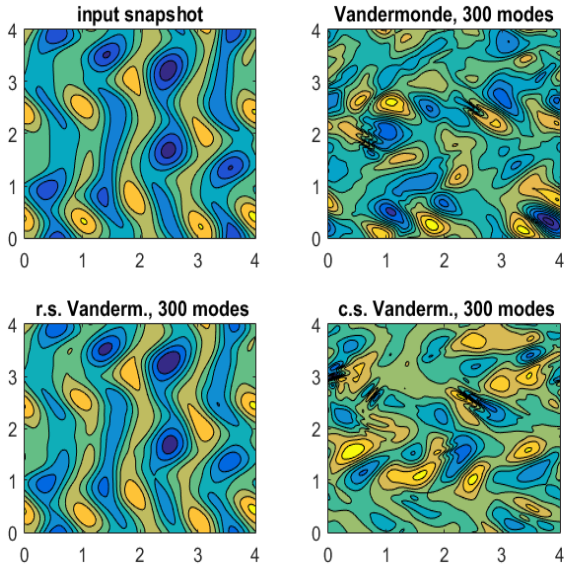


FIG. 4.6. *Reconstruction of* $\mathbf{f}_{321}$ *using* 300 *dominant modes in the basis* $\widehat{Q}_f$. *Compare with Figure* 4.1. *A similar effect is observed when reconstructing the other* $\mathbf{f}_i$*'s.*

If we compute the economy size (tall) QR factorization $Z_\ell = QR$ and define projected snapshots $\mathbf{g}_i = Q^*\mathbf{f}_i$, then the LS problem can be compactly written using the Kronecker product $\otimes$ as

(5.3)
$$\|\mathbf{g} - S\boldsymbol{\alpha}\|_2 \longrightarrow \min, \quad \text{where } \mathbf{g} = \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_m \end{pmatrix}, \quad S = (I_m \otimes R) \begin{pmatrix} \Delta_{\Lambda_1} \\ \vdots \\ \Delta_{\Lambda_m} \end{pmatrix} \equiv \begin{pmatrix} R\Delta_{\Lambda_1} \\ \vdots \\ R\Delta_{\Lambda_m} \end{pmatrix}.$$

The normal equations approach [25] allows efficient computation of $\boldsymbol{\alpha} = S^\dagger\mathbf{g}$ due to the particular structure of S. Recently, [15] proposed a corrected seminormal solution and a QR factorization based method.

In this section, our interests are of a theoretical nature. We explore other choices of the pseudoinverse based solution, different from the Moore–Penrose solution $S^\dagger\mathbf{g}$, in an attempt to establish a connection between the numerical linear algebra framework and the GLA.

5.1. Minimization in a Z_ℓ-induced norm. The reflexive g-inverse [34, Definition 2.4] of S,

$$S^- = \begin{pmatrix} \Delta_{\Lambda_1} \\ \vdots \\ \Delta_{\Lambda_m} \end{pmatrix}^\dagger (I \otimes R^{-1}),$$

allows interesting explicit formulas that reveal a relation to the GLA [31],[29]. Indeed, if we define $\boldsymbol{\alpha}_\star = S^- \mathbf{g}$, then

$$\boldsymbol{\alpha}_\star = \begin{pmatrix} \Delta_{\Lambda_1} \\ \vdots \\ \Delta_{\Lambda_m} \end{pmatrix}^\dagger (I \otimes R^{-1}) \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_m \end{pmatrix} = \left(\sum_{k=1}^m \Delta_{\Lambda_k}^* \Delta_{\Lambda_k} \right)^{-1} \sum_{i=1}^m \Delta_{\Lambda_i}^* (R^{-1}\mathbf{g}_i) \tag{5.4}$$

$$= \begin{pmatrix} \sum_{k=1}^m |\lambda_1|^{2(k-1)} & 0 & \dots & 0 \\ 0 & \sum_{k=1}^m |\lambda_2|^{2(k-1)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sum_{k=1}^m |\lambda_\ell|^{2(k-1)} \end{pmatrix}^{-1} \sum_{i=1}^m \Delta_{\Lambda_i}^* (R^{-1}\mathbf{g}_i) \tag{5.5}$$

$$= \sum_{i=1}^m \begin{pmatrix} \frac{\overline{\lambda}_1^{i-1}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} & 0 & \cdot & 0 \\ 0 & \frac{\overline{\lambda}_2^{i-1}}{\sum_{k=1}^m |\lambda_2|^{2(k-1)}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & \frac{\overline{\lambda}_\ell^{i-1}}{\sum_{k=1}^m |\lambda_\ell|^{2(k-1)}} \end{pmatrix} \begin{pmatrix} (R^{-1}\mathbf{g}_i)_1 \\ (R^{-1}\mathbf{g}_i)_2 \\ \vdots \\ (R^{-1}\mathbf{g}_i)_\ell \end{pmatrix} \tag{5.6}$$

$$= \sum_{i=1}^m \begin{pmatrix} \frac{\overline{\lambda}_1^{i-1}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_1 \\ \frac{\overline{\lambda}_2^{i-1}}{\sum_{k=1}^m |\lambda_2|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_2 \\ \vdots \\ \frac{\overline{\lambda}_\ell^{i-1}}{\sum_{k=1}^m |\lambda_\ell|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_\ell \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m \frac{\overline{\lambda}_1^{i-1}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_1 \\ \sum_{i=1}^m \frac{\overline{\lambda}_2^{i-1}}{\sum_{k=1}^m |\lambda_2|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_2 \\ \vdots \\ \sum_{i=1}^m \frac{\overline{\lambda}_\ell^{i-1}}{\sum_{k=1}^m |\lambda_\ell|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_\ell \end{pmatrix}. \tag{5.7}$$

Although this $\boldsymbol{\alpha}_\star$ does not minimize (5.2) or (5.3), it follows from the theory of generalized inverses that it does have some interesting properties.

Remark 5.1. Recall that, by definition, S^- satisfies $SS^-S = S$, $S^-SS^- = S^-$, and $S^-S = (S^-S)^*$. In fact, since $SS^- \neq (SS^-)^*$, we have in general that $S^- \neq S^\dagger$, so $\boldsymbol{\alpha}_\star \neq S^\dagger \mathbf{g}$.

PROPOSITION 5.2. *Let* $M = I \otimes (RR^*)^{-1}$, $(\mathbf{x}, \mathbf{y})_M = \mathbf{y}^* M \mathbf{x}$, *and let* $\|\mathbf{x}\|_M = \sqrt{\mathbf{x}^* M \mathbf{x}}$. *Then* $\boldsymbol{\alpha}_\star$ *is the minimum* $\|\cdot\|_2$*-norm solution of the weighted LS problem*

$$\|\mathbf{g} - S\boldsymbol{\alpha}\|_M \longrightarrow \min. \tag{5.8}$$

Proof. The matrix M is obviously positive definite, and $\|\cdot\|_M$ is a well defined norm. The claim is a direct corollary of [34, Theorem 3.3], because one can easily establish that

$$(SS^-)^* = (I \otimes (RR^*)^{-1})(SS^-)(I \otimes (RR^*)) = M(SS^-)M^{-1}.$$

The claim can be derived also by brute force calculation. Since

$$M = (I \otimes R^{-*})(I \otimes R^{-1}) \equiv LL^*, \;\; L \equiv I \otimes R^{-*},$$

is the Cholesky factorization and $\|x\|_M = \|(I \otimes R^{-1})x\|_2$, we have

$$\left\| \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_m \end{pmatrix} - \begin{pmatrix} R\Delta_{\Lambda_1} \\ \vdots \\ R\Delta_{\Lambda_m} \end{pmatrix} \boldsymbol{\alpha} \right\|_M^2 = \left\| (I \otimes R^{-1}) \left[\begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_m \end{pmatrix} - \begin{pmatrix} R\Delta_{\Lambda_1} \\ \vdots \\ R\Delta_{\Lambda_m} \end{pmatrix} \boldsymbol{\alpha} \right] \right\|_2^2 \tag{5.9}$$

$$= \sum_{j=1}^m \|R^{-1}\mathbf{g}_j - \Delta_{\Lambda_j}\boldsymbol{\alpha}\|_2^2 = \sum_{j=1}^m \|\mathbf{g}_j - R\Delta_{\Lambda_j}\boldsymbol{\alpha}\|_M^2. \tag{5.10}$$

However, the framework of the theory of matrix generalized inverses gives deeper insight. ∎

Remark 5.3. If $Z_\ell^* Z_\ell = I_\ell$ (e.g., if $\mathbb{A}$ is normal, and we compute an orthonormal set of Ritz vectors), then R is diagonal and unitary, and $S^\dagger = S^-$.

5.2. LS reconstruction in the frequency domain. We now repeat the reconstruction procedure, but in the frequency domain, following the methodology of section 3. That is, instead of n time observable trajectories sampled at m time-equidistant points (the rows of $\mathbf{X}_m$), we have their DFT transforms, i.e., $\mathbf{X}_m\mathbb{F}$, as the raw input. We assume that the eigenvalues are simple and are enumerated so that the selected ones are $\lambda_1, \ldots, \lambda_\ell$; the QR factorization of the selected modes is $Z_\ell = QR$.

If we postmultiply the expression in (5.2) by the unitary DFT matrix $\mathbb{F}$, the objective becomes

(5.11)
$$\Omega(\boldsymbol{\alpha}) \equiv \|\mathbf{X}_m\mathbb{F} - Z_\ell \Delta_{\boldsymbol{\alpha}} \begin{pmatrix} \Lambda_1 & \Lambda_2 & \ldots & \Lambda_m \end{pmatrix} \mathbb{F}\|_F^2 \longrightarrow \min \ \text{ as the function of } \boldsymbol{\alpha},$$

where $\begin{pmatrix} \Lambda_1 & \Lambda_2 & \ldots & \Lambda_m \end{pmatrix} \mathbb{F}$ is the DFT of the $\ell \times m$ submatrix $\mathbb{V}_{\ell,m}$ of $\mathbb{V}_m$. Because of the structure of $\mathbb{V}_{\ell,m}\mathbb{F}$, we distinguish two cases.

5.2.1. Case 1: $\prod_{i=1}^{\ell}(\lambda_i^m - 1) \neq 0$. If none of the λ_i's equals an mth root of unity, then (see (3.1))

$$\begin{pmatrix} \Lambda_1 & \Lambda_2 & \ldots & \Lambda_m \end{pmatrix} \mathbb{F} = \widehat{\mathcal{D}}_1 \widehat{\mathcal{C}} \mathcal{D}_2,$$

where $\widehat{\cdot}$ denotes an $\ell \times m$ submatrix from the relation (3.1). In particular, $(\widehat{\mathcal{D}}_1)_{ii} = (\lambda_i^m - 1)/\sqrt{m} \neq 0$, $(\widehat{\mathcal{C}})_{ij} = 1/(\lambda_i - \overline{\omega}^{j-1})$, and $\mathcal{D}_2$ is a unitary diagonal matrix. Introduce column partition

(5.12)
$$\widehat{\mathcal{C}} = \begin{pmatrix} \widehat{\mathcal{C}}_1 & \ldots & \widehat{\mathcal{C}}_m \end{pmatrix}, \ \widehat{\mathcal{C}}_j = \begin{pmatrix} \dfrac{1}{\lambda_1 - \overline{\omega}^{j-1}} \\ \vdots \\ \dfrac{1}{\lambda_\ell - \overline{\omega}^{j-1}} \end{pmatrix},$$

$$\Delta_{\mathcal{C}_j} = \begin{pmatrix} \dfrac{1}{\lambda_1 - \overline{\omega}^{j-1}} & 0 & \cdot & 0 \\ 0 & \dfrac{1}{\lambda_2 - \overline{\omega}^{j-1}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & \dfrac{1}{\lambda_\ell - \overline{\omega}^{j-1}} \end{pmatrix}.$$

Let $\widehat{\mathbf{X}}_m = \mathbf{X}_m\mathbb{F}\mathcal{D}_2^* = (\widehat{\mathbf{f}}_1, \ldots, \widehat{\mathbf{f}}_m)$, $\Delta_{\boldsymbol{\beta}} = \Delta_{\boldsymbol{\alpha}}\widehat{\mathcal{D}}_1$. Then

(5.13)
$$\Omega(\boldsymbol{\alpha}) = \|\widehat{\mathbf{X}}_m - Z_\ell \Delta_{\boldsymbol{\beta}} \widehat{\mathcal{C}}\|_F^2 = \|(\widehat{\mathbf{f}}_1, \ldots, \widehat{\mathbf{f}}_m) - Z_\ell \Delta_{\boldsymbol{\beta}} \begin{pmatrix} \widehat{\mathcal{C}}_1 & \ldots & \widehat{\mathcal{C}}_m \end{pmatrix} \|_F^2$$
$$= \left\| \begin{pmatrix} \widehat{\mathbf{f}}_1 \\ \vdots \\ \widehat{\mathbf{f}}_m \end{pmatrix} - \begin{pmatrix} Z_\ell \Delta_{\boldsymbol{\beta}} \mathcal{C}_1 \\ \vdots \\ Z_\ell \Delta_{\boldsymbol{\beta}} \mathcal{C}_m \end{pmatrix} \right\|_2^2 = \boxed{\text{use: } \Delta_{\boldsymbol{\beta}} \mathcal{C}_i = \Delta_{\mathcal{C}_i} \boldsymbol{\beta}} = \left\| \begin{pmatrix} \widehat{\mathbf{f}}_1 \\ \vdots \\ \widehat{\mathbf{f}}_m \end{pmatrix} - \begin{pmatrix} Z_\ell \Delta_{\mathcal{C}_1} \\ \vdots \\ Z_\ell \Delta_{\mathcal{C}_m} \end{pmatrix} \boldsymbol{\beta} \right\|_2^2.$$

If we set $Q^*\widehat{\mathbf{f}}_i = \widehat{\mathbf{g}}_i$, then, using the reflexive g-inverse as before,

$$\boldsymbol{\alpha} = \widehat{\mathcal{D}}_1^{-1} \begin{pmatrix} \Delta_{\mathcal{C}_1} \\ \vdots \\ \Delta_{\mathcal{C}_m} \end{pmatrix}^{\dagger} (I \otimes R^{-1}) \begin{pmatrix} \widehat{\mathbf{g}}_1 \\ \vdots \\ \widehat{\mathbf{g}}_m \end{pmatrix} = \widehat{\mathcal{D}}_1^{-1} \left(\sum_{k=1}^{m} \Delta_{\mathcal{C}_k}^* \Delta_{\mathcal{C}_k} \right)^{-1} \sum_{j=1}^{m} \Delta_{\mathcal{C}_j}^* (R^{-1}\widehat{\mathbf{g}}_j)$$

$$= \widehat{\mathcal{D}}_1^{-1} \begin{pmatrix} \sum_{k=1}^{m} \frac{1}{|\lambda_1 - \overline{\omega}^{k-1}|^2} & 0 & \dots & 0 \\ 0 & \sum_{k=1}^{m} \frac{1}{|\lambda_2 - \overline{\omega}^{k-1}|^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sum_{k=1}^{m} \frac{1}{|\lambda_\ell - \overline{\omega}^{k-1}|^2} \end{pmatrix}^{-1} \sum_{j=1}^{m} \Delta_{\mathcal{C}_j}^* (R^{-1}\widehat{\mathbf{g}}_j)$$

$$= \widehat{\mathcal{D}}_1^{-1} \sum_{j=1}^{m} \begin{pmatrix} \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_1 - \omega^{j-1}}{|\lambda_1 - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_1 \\ \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_2 - \omega^{j-1}}{|\lambda_2 - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_2 \\ \vdots \\ \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_\ell - \omega^{j-1}}{|\lambda_\ell - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_\ell \end{pmatrix} = \widehat{\mathcal{D}}_1^{-1} \begin{pmatrix} \sum_{j=1}^{m} \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_1 - \omega^{j-1}}{|\lambda_1 - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_1 \\ \sum_{j=1}^{m} \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_2 - \omega^{j-1}}{|\lambda_2 - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_2 \\ \vdots \\ \sum_{j=1}^{m} \frac{1}{\sum_{k=1}^{m} \frac{\overline{\lambda}_\ell - \omega^{j-1}}{|\lambda_\ell - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_\ell \end{pmatrix}.$$

Hence, the components α_i of the LS solution via the reflexive g-inverse are

$$\alpha_i = \frac{\sqrt{m}}{\lambda_i^m - 1} \sum_{j=1}^{m} \frac{\dfrac{1}{\overline{\overline{\lambda}_i - \omega^{j-1}}}}{\displaystyle\sum_{k=1}^{m} \frac{1}{|\lambda_i - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_i$$

$$= \frac{\sqrt{m}}{\displaystyle\prod_{p=1}^{m} (\lambda_i - \overline{\omega}^{p-1})} \sum_{j=1}^{m} \frac{\dfrac{1}{\overline{\overline{\lambda}_i - \omega^{j-1}}}}{\displaystyle\sum_{k=1}^{m} \frac{1}{|\lambda_i - \overline{\omega}^{k-1}|^2}} (R^{-1}\widehat{\mathbf{g}}_j)_i.$$

Remark 5.4. It remains an interesting open question whether these formulas have an appropriate interpretation in a wider context. Also, their ingredients, such as $R^{-1}\widehat{\mathbf{g}}_j$ or the harmonic mean of squared distances from λ_i to the mth roots of unity, may be of independent interest.

5.2.2. Case 2: $\prod_{i=1}^{\ell}(\lambda_i^m - 1) = 0$. We now consider the case when some of the λ_i's are among the mth roots of unity. Here we recall (3.2); i.e., if $\lambda_i = \omega^{1-j}$, then the ith row of $\begin{pmatrix} \Lambda_1 & \Lambda_2 & \dots & \Lambda_m \end{pmatrix} \mathbb{F}$ reads

$$\left(\begin{pmatrix} \Lambda_1 & \Lambda_2 & \dots & \Lambda_m \end{pmatrix} \mathbb{F} \right)_{ij} = \frac{1}{\sqrt{m}} \prod_{\substack{k=1 \\ k \neq j}}^{m} (\lambda_i - \omega^{1-k}) \omega^{1-j},$$

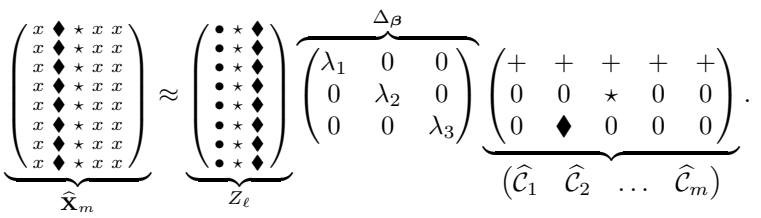
$$\left(\begin{pmatrix} \Lambda_1 & \Lambda_2 & \dots & \Lambda_m \end{pmatrix} \mathbb{F} \right)_{ik} = 0 \ \text{ for } k \neq j.$$

We set $(\mathcal{D}_1)_{ii} = 1/\sqrt{m}$, $\mathcal{C}_{ij} = \prod_{\substack{k=1 \\ k \neq j}}^{m} (\lambda_i - \omega^{1-k})$, and $\mathcal{C}_{ik} = 0$ for $k \neq j$. Hence, if, e.g., $\lambda_i = \omega^{1-p}$, $\lambda_{i+1} = \omega^{1-q}$ are the only two eigenvalues that match some mth root of

unity, we have the following version of the partition (5.12):

$$\widehat{\mathcal{C}}_p = \begin{pmatrix} \frac{1}{\lambda_1 - \overline{\omega}^{p-1}} \\ \vdots \\ \frac{1}{\lambda_{i-1} - \overline{\omega}^{p-1}} \\ \frac{\prod_{k=1}^{m}(\lambda_i - \omega^{1-k})}{k\neq p} \\ 0 \\ \frac{1}{\lambda_{i+2} - \overline{\omega}^{p-1}} \\ \vdots \\ \frac{1}{\lambda_\ell - \overline{\omega}^{p-1}} \end{pmatrix}, \quad \widehat{\mathcal{C}}_q = \begin{pmatrix} \frac{1}{\lambda_1 - \overline{\omega}^{q-1}} \\ \vdots \\ \frac{1}{\lambda_{i-1} - \overline{\omega}^{q-1}} \\ 0 \\ \frac{\prod_{k=1}^{m}(\lambda_i - \omega^{1-k})}{k\neq q} \\ \frac{1}{\lambda_{i+2} - \overline{\omega}^{p-1}} \\ \vdots \\ \frac{1}{\lambda_\ell - \overline{\omega}^{q-1}} \end{pmatrix}, \quad \widehat{\mathcal{C}}_j = \begin{pmatrix} \frac{1}{\lambda_1 - \overline{\omega}^{j-1}} \\ \vdots \\ \frac{1}{\lambda_{i-1} - \overline{\omega}^{j-1}} \\ 0 \\ 0 \\ \frac{1}{\lambda_{i+2} - \overline{\omega}^{p-1}} \\ \vdots \\ \frac{1}{\lambda_\ell - \overline{\omega}^{j-1}} \end{pmatrix}, \quad k \notin \{p,q\}.$$

As an illustration of the reconstruction by the product $Z_\ell \Delta_{\boldsymbol{\beta}} \widehat{\mathcal{C}}$ when two of the λ_i's are among the roots of unity, consider the following schematic representation of (5.13) with $m = 5$, $\ell = 3$:



Here λ_2 and λ_3 are roots of unity, and thus $Z_\ell(:,2)$ ($\star$) and $Z_\ell(:,3)$ ($\blacklozenge$) participate in the representation of only $\widehat{\mathbf{X}}_m(:,3)$ and $\widehat{\mathbf{X}}_m(:,2)$, respectively. That is, if $\lambda_i = \omega^{1-j}$, then $Z_\ell(:,i)$ participates in the reconstruction of only one transformed snapshot, namely $\widehat{\mathbf{f}}_j$.

Remark 5.5. Any system with a (quasi-)periodic attractor will have a subset of eigenvalues that are mth roots of unity. If the system is dissipative (i.e., the off-attractor spectrum of the Koopman operator is present), then, besides mth roots of unity, the spectrum would have a "lattice" structure. A complete description of these cases is given in [30].

6. Reconstructions based on GLA theory. We now wish to compare and contrast our above calculations with the generalized Laplace analysis (GLA) theorem [31, 29] adapted to this matrix setting. The GLA theorem is a theoretical result for a Koopman operator acting on infinite dimensional function spaces which constructs projections of functions onto Koopman eigenfunctions via ergodic-type averages. The classical formulation begins with merely knowing the spectrum of $\mathbb{A}$ and constructs projections onto the eigenspaces using infinite-time averages.

6.1. Matrix GLA. We call an eigenvalue λ of $\mathbb{A}$ a dominant eigenvalue if for all other eigenvalues μ, $|\lambda| \geq |\mu|$. We call λ strictly dominant if for all μ, $|\lambda| > |\mu|$. We tacitly assume that $\mathbb{A}$ is diagonalizable.

PROPOSITION 6.1 (matrix GLA). *Let $\lambda \in \sigma(\mathbb{A})$ be a dominant eigenvalue, and*

let $\Pi_\lambda : \mathbb{C}^n \to \mathbb{C}^n$ *be the (skew) projection onto* $N(\mathbb{A} - \lambda I)$*. Then for any* $\mathbf{f} \in \mathbb{C}^n$

$$\Pi_\lambda \mathbf{f} = \lim_{m\to\infty} \frac{1}{m} \sum_{i=0}^{m-1} \lambda^{-i} \mathbb{A}^i \mathbf{f}. \tag{6.1}$$

Remark 6.2. If we let $\mathbf{f}_i = \mathbb{A}^{i-1}\mathbf{f}$, the above average is $\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} \lambda^{-i+1}\mathbf{f}_i$.

This is easily shown by expanding $\mathbf{f}$ in the eigenvector basis. The above sum is then composed of n terms of the form $\frac{1}{m}\sum_{i=0}^{m-1} \left(\frac{\mu}{\lambda}\right)^i$. Since λ is a dominant eigenvalue, for any $\mu \neq \lambda$ the term $\frac{1}{m}\sum_{i=0}^{m-1} \left(\frac{\mu}{\lambda}\right)^i$ converges to 0. Projection onto another eigenspace requires subtracting from $\mathbf{f}$ all projections $\Pi_\lambda \mathbf{f}$ corresponding to all eigenvalues strictly dominating the one of interest and then running the above average for the new vector.

Remark 6.3. The GLA is an inherently infinitary result. A straightforward application using finite data quickly leads to numerical instability.

The projection operators Π_λ are skew projections along the other eigenvectors of $\mathbb{A}$. We can define a weighted inner product in which these spectral projections become orthogonal. Let $Z_n = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ be the normalized eigenvectors of $\mathbb{A}$, and let $Z_n = QR$ be the QR decomposition. Let $L = R^{-1}Q^* (\equiv Z_n^{-1} \equiv Z_n^\dagger)$ and $P = L^*L$. Define the Hermitian form

$$\langle \mathbf{v}\,,\, \mathbf{w} \rangle_P = \mathbf{w}^* P \mathbf{v}. \tag{6.2}$$

Since Z_n is full rank, this form is nondegenerate and defines an inner product. Noting that L is just Z_n^{-1}, it can easily be shown that $\langle \mathbf{z}_i\,,\, \mathbf{z}_j \rangle_P = \delta_{i,j}$ which implies that the projections are orthogonal projections with respect to this inner product.

We can formulate a "coordinate" version of the GLA theorem. Since we know the eigenvectors, we can reduce the infinitary result to a finitary one.

PROPOSITION 6.4. *Let* $\mathbf{f} = Z_n \boldsymbol{\alpha} = \sum_{i=1}^n \mathbf{z}_i \alpha_i$*, and define* $\mathbf{f}_i = \mathbb{A}^{i-1}\mathbf{f} = Z_n \Lambda_n^{i-1} \boldsymbol{\alpha}$*, where* Λ_n *is the diagonal matrix of eigenvalues of* $\mathbb{A}$*. Then*

$$\boldsymbol{\alpha} = \frac{1}{m}\sum_{i=1}^{m} \Lambda_n^{-(i-1)} R^{-1} Q^* \mathbf{f}_i = \frac{1}{m}\sum_{i=1}^{m} \Lambda_n^{-(i-1)} L \mathbf{f}_i = \frac{1}{m}\sum_{i=1}^{m} \Lambda_n^{-(i-1)} Z_n^{-1} \mathbf{f}_i. \tag{6.3}$$

Proof. Since Z_n^{-1} exists,

$$\frac{1}{m}\sum_{i=1}^{m} \Lambda_n^{-i+1} Z_n^{-1} \mathbf{f}_i = \frac{1}{m}\sum_{i=1}^{m} \Lambda_n^{-i+1} Z_n^\dagger Z_n \Lambda_n^{i-1} \boldsymbol{\alpha} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\alpha} = \boldsymbol{\alpha}.$$

By definition, $Z_n^{-1} = L = R^{-1}Q^*$. This gives the equivalent formulations. ∎

6.2. GLA formulas for reconstruction weights. As in section 5, we wish to reconstruct the evolution $(\mathbf{f}_1, \ldots, \mathbf{f}_m)$ with a smaller number of Ritz vectors $Z_\ell = [\mathbf{z}_1, \ldots, \mathbf{z}_\ell]$, $\ell \leq m$. Given that the matrix version of the GLA (Proposition 6.4) gives the exact reconstruction weights, we would like to investigate how well GLA reconstructs the weights without full knowledge of the eigenvectors. Since the GLA gives a skew projection onto the eigenfunctions rather than an orthogonal one, the reconstructed weights will not be optimal in minimizing the error's 2-norm.

The objective is to find $\boldsymbol{\alpha} \in \mathbb{C}^{\ell\times 1}$ such that

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\beta}\in\mathbb{C}^{\ell\times 1}}{\arg\min} \sum_{i=1}^{m} \left\| \mathbf{f}_i - \sum_{j=1}^{\ell} \mathbf{z}_j \lambda_j^{i-1} \beta_j \right\|_2^2 . \tag{6.4}$$

In general, it will not be possible to perfectly reconstruct the data. Define for $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_\ell)^T$, the functionals

$$\Omega_i(\boldsymbol{\beta}) = \left\| \mathbf{f}_i - \sum_{j=1}^{\ell} \mathbf{z}_j \lambda_j^{i-1} \beta_j \right\|_2^2, \quad \Omega(\boldsymbol{\beta}) = \sum_{i=1}^{m} \Omega_i(\boldsymbol{\beta}).$$

First, consider the case of a particular $\mathbf{f}_i$. We look for $\boldsymbol{\alpha}^{(i)} \in \mathbb{C}^{\ell\times 1}$ that optimally reconstructs $\mathbf{f}_i$, i.e., the $\boldsymbol{\alpha}^{(i)}$ satisfying $\boldsymbol{\alpha}^{(i)} = \arg\min_{\boldsymbol{\beta}\in\mathbb{C}^\ell} \Omega_i(\boldsymbol{\beta})$.

Write $Z_\ell = (\mathbf{z}_1 \cdots \mathbf{z}_\ell) \in \mathbb{C}^{n\times\ell}$. The QR factorization of this matrix is written as $Z_\ell = \begin{pmatrix} Q_\ell & Q_\ell^\perp \end{pmatrix} \binom{R}{\mathbf{0}}$, where $\operatorname{span} Z_\ell = \operatorname{span} Q_\ell$, and $R \in \mathbb{C}^{\ell\times\ell}$. The pseudoinverse of Z_ℓ is $Z_\ell^\dagger = (R^{-1} \,|\, \mathbf{0})Q^*$. Let $\mathbf{g}_i \in \mathbb{C}^\ell$ and $\mathbf{h}_i \in \mathbb{C}^{n-\ell}$ such that $\mathbf{f}_i = Q \binom{\mathbf{g}_i}{h_i}$. Now,

$$\begin{aligned} \Omega_i(\boldsymbol{\beta}) &= \|\mathbf{f}_i - Z_\ell \Lambda_\ell^{i-1}\boldsymbol{\beta}\|_2^2 = \|Q^*(\mathbf{f}_i - Z_\ell\Lambda_\ell^{i-1}\boldsymbol{\beta})\|_2^2 = \left\| \begin{pmatrix} \mathbf{g}_i \\ \mathbf{h}_i \end{pmatrix} - \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix} \Lambda_\ell^{i-1}\boldsymbol{\beta} \right\|_2^2 \\ &= \|\mathbf{h}_i\|_2^2 + \|\mathbf{g}_i - R\Lambda_\ell^{i-1}\boldsymbol{\beta}\|_2^2. \end{aligned}$$

Thus $\Omega_i(\cdot)$ is minimized with the choice $\boldsymbol{\alpha}^{(i)} = \arg\min_{\boldsymbol{\beta}} \Omega_i(\boldsymbol{\beta}) = \Lambda_\ell^{-i+1} R^{-1}\mathbf{g}_i$. Componentwise this reads

$$(\boldsymbol{\alpha}^{(i)})_j = \frac{1}{\lambda_j^{i-1}} (R_\ell^{-1}\mathbf{g}_i)_j. \tag{6.5}$$

It is unlikely that $\boldsymbol{\alpha}^{(i)}$ will be the minimizer for every $\Omega_j(\cdot)$, and it is therefore unlikely to be a minimizer for $\Omega = \Omega_1 + \cdots + \Omega_m$. However, Jensen's inequality will allow us to construct an $\boldsymbol{\alpha}$ which does better than the average of the errors induced by each $\boldsymbol{\alpha}^{(i)}$. Let $\Omega = \{1, \ldots, m\}$, and define the probability measure $p : \Omega \to [0, 1]$ as $p(i) = \frac{1}{m}$. Define the random vectors $X : \Omega \to \mathbb{C}^n$ as $X(i) = \boldsymbol{\alpha}^{(i)}$. Then, since Ω is a convex function, Jensen's inequality gives us

$$\sum_{i=1}^{m} p(i)\Omega(X(i)) \geq \Omega\Big(\sum_{i=1}^{m} p(i)X(i) \Big), \quad \text{i.e.,} \quad \frac{1}{m}\sum_{i=1}^{m} \Omega(\boldsymbol{\alpha}^{(i)}) \geq \Omega\Big(\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\alpha}^{(i)} \Big). \tag{6.6}$$

We can rewrite $\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\alpha}^{(i)}$ as

$$\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\alpha}^{(i)} = \frac{1}{m}\sum_{i=1}^{m} \Lambda_\ell^{-i+1} R^{-1}\mathbf{g}_i. \tag{6.7}$$

Compare this formula with (6.3) in Proposition 6.4. This is a GLA formula, except now we can only project onto the selected ℓ eigenfunctions. We call the components of this vector the GLA reconstruction weights $\vec{\boldsymbol{\alpha}}_{(GLA)}$. This can be written in a form similar to $\boldsymbol{\alpha}_\star$ (equation (5.7)):

$$\boldsymbol{\alpha}_{(GLA)} = \frac{1}{m}\sum_{i=1}^{m} \Lambda_\ell^{-i+1} R^{-1}\mathbf{g}_i = \begin{pmatrix} \frac{1}{m}\sum_{i=1}^{m} \lambda_1^{-i+1}(R^{-1}\mathbf{g}_i)_1 \\ \frac{1}{m}\sum_{i=1}^{m} \lambda_2^{-i+1}(R^{-1}\mathbf{g}_i)_2 \\ \vdots \\ \frac{1}{m}\sum_{i=1}^{m} \lambda_\ell^{-i+1}(R^{-1}\mathbf{g}_i)_\ell \end{pmatrix}. \tag{6.8}$$

6.2.1. Optimal reconstruction formula as a generalized ergodic average. Using a reflexive g-inverse, we recall formula (5.7) giving the optimal reconstruction weights using ℓ,

$$\text{(5.7 revisited)} \qquad \boldsymbol{\alpha}_\star = \sum_{i=1}^{m} \begin{pmatrix} \frac{\bar{\lambda}_1^{i-1}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} & & \\ & \ddots & \\ & & \frac{\bar{\lambda}_\ell^{i-1}}{\sum_{k=1}^m |\lambda_\ell|^{2(k-1)}} \end{pmatrix} \begin{pmatrix} (R^{-1}\mathbf{g}_i)_1 \\ \vdots \\ (R^{-1}\mathbf{g}_i)_\ell \end{pmatrix}.$$

In the case of a unimodular spectrum, the weights given via the reflexive g-inverse reduce exactly to the GLA weights.

PROPOSITION 6.5. *When the spectrum of* $\mathbb{A}$ *lies on the unit circle,* $\boldsymbol{\alpha}_\star = \boldsymbol{\alpha}_{(GLA)}$.

Proof. For λ_j with $|\lambda_j| = 1$, we have $\overline{\lambda_j} = \lambda_j^{-1}$. Then for each j,

$$\begin{aligned}(\boldsymbol{\alpha}_\star)_j &= \sum_{i=1}^{m} \frac{\overline{\lambda}_j^{i-1}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} (R^{-1}\mathbf{g}_i)_j = \sum_{i=1}^{m} \frac{(\lambda_j^{-1})^{i-1}}{\sum_{k=1}^m 1^{k-1}} (R^{-1}\mathbf{g}_i)_j \\ &= \sum_{i=1}^{m} \frac{\lambda_j^{-i+1}}{m} (R^{-1}\mathbf{g}_i)_j = (\boldsymbol{\alpha}_{(GLA)})_j. \end{aligned}$$

∎

When we do not have a unimodular spectrum, we can interpret the optimal reconstruction weights as the result of a weighted GLA average. Indeed, the formula for each component $(\vec{\boldsymbol{\alpha}}_\star)_j$ can be manipulated as follows:

$$\begin{aligned}(\boldsymbol{\alpha}_\star)_j &= \sum_{i=1}^{m} \frac{\bar{\lambda}_j^{i-1}(R^{-1}\mathbf{g}_i)_j}{\sum_{k=1}^m |\lambda_j|^{2(k-1)}} = \sum_{i=1}^{m} \frac{\bar{\lambda}_j^{i-1}}{\sum_{k=1}^m |\lambda_j|^{2(k-1)}} \frac{\lambda_j^{i-1}}{\lambda_j^{i-1}} (R^{-1}\mathbf{g}_i)_j \\ &= \sum_{i=1}^{m} \frac{|\lambda_j|^{2(i-1)}}{\sum_{k=1}^m |\lambda_j|^{2(k-1)}} \frac{1}{\lambda_j^{i-1}} (R^{-1}\mathbf{g}_i)_j. \end{aligned} \tag{6.9}$$

Define $w_{m,j}(i) = \frac{|\lambda_j|^{2(i-1)}}{\sum_{k=1}^m |\lambda_j|^{2(k-1)}}$. Then $w_{m,j}(i) > 0$ and $\sum_{i=1}^m w_{m,j}(i) = 1$. Then, for each component $(\boldsymbol{\alpha}_\star)_j$, we have the (nonuniformly) weighted average

$$\boldsymbol{\alpha}_j = \sum_{i=1}^{m} w_{m,j}(i) \lambda_j^{-i+1} (R^{-1}\mathbf{g}_i)_j. \tag{6.10}$$

Let us define the set of diagonal matrices $W_m^{(i)}$ as

$$W_m^{(i)} = \begin{pmatrix} w_{m,1}(i) & & \\ & \ddots & \\ & & w_{m,\ell}(i) \end{pmatrix} = \begin{pmatrix} \frac{|\lambda_1|^{2(i-1)}}{\sum_{k=1}^m |\lambda_1|^{2(k-1)}} & & \\ & \ddots & \\ & & \frac{|\lambda_\ell|^{2(i-1)}}{\sum_{k=1}^m |\lambda_\ell|^{2(k-1)}} \end{pmatrix}. \tag{6.11}$$

Then, the optimal reconstruction weights are given by the weighted GLA formula

$$\boldsymbol{\alpha}_\star = \sum_{i=1}^{m} W_m^{(i)} \Lambda_\ell^{-i+1} R^{-1}\mathbf{g}_i \equiv \sum_{i=1}^{m} W_m^{(i)} \Lambda_\ell^{-i+1} R^{-1}\mathbf{g}_i. \tag{6.12}$$

Remark 6.6. Compare this with the formula for $\boldsymbol{\alpha}_{(GLA)}$ (equation (6.8)), where the weight matrix $W_{m,j}$ for the GLA formula is the uniform weight matrix $W_m^{(i)} = (1/m)I_m$.

6.2.2. Distance between the GLA weights and the optimal reconstruction weights. We have two expressions for reconstructing m snapshots using ℓ eigenfunctions, namely (5.7) and (6.8). We have already shown that when the spectrum is on the unit circle, these formulas are equivalent. In the case when the spectrum is not contained in the unit circle, the question of their equivalence remains. Clearly, for any fixed number m of snapshots, these formulas give different weights. But what is the limit when trying to reconstruct an increasing number of snapshots while still only using ℓ eigenfunctions? If we write $\boldsymbol{\alpha}_\star^{(m)}$ and $\boldsymbol{\alpha}_{(GLA)}^{(m)}$ for the optimal and GLA weights reconstructing m snapshots, does $\|\boldsymbol{\alpha}_\star^{(m)} - \boldsymbol{\alpha}_{(GLA)}^{(m)}\|_2 \to 0$ as $m \to \infty$?

In general, the answer is no. We consider the simple case when $\mathbb{A} \in \mathbb{C}^{3\times 3}$ with eigenvectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$, where $\boldsymbol{z}_1 \perp \boldsymbol{z}_2$ and $\boldsymbol{z}_3$ is not orthogonal to either of the other eigenvectors. We also assume that the associated eigenvalues satisfy $1 > |\lambda_1| \geq |\lambda_2| > |\lambda_3|$. The evolution $\mathbf{f}_i = \mathbb{A}^{i-1}\boldsymbol{v}$ satisfies $\mathbf{f}_i = \sum_{j=1}^{3} \beta_j \mathbf{z}_j \lambda_j^{i-1}$. We reconstruct with $\mathbf{z}_1, \mathbf{z}_2$. In this case, the $R^{-1}\mathbf{g}_i$ term in the weights' formulas is

$$R^{-1}\mathbf{g}_i = R^{-1}Q_2^*\mathbf{f}_i = \begin{pmatrix} \beta_1\lambda_1^{i-1} + \beta_3\lambda_3^{i-1}\langle \mathbf{z}_3\,,\,\mathbf{z}_1\rangle \\ \beta_2\lambda_2^{i-1} + \beta_3\lambda_3^{i-1}\langle \mathbf{z}_3\,,\,\mathbf{z}_2\rangle \end{pmatrix}. \tag{6.13}$$

For $j = 1, 2$, and with $s_j(i) = \beta_j + \beta_3(\frac{\lambda_3}{\lambda_j})^{i-1}\langle \mathbf{z}_3\,,\,\mathbf{z}_j\rangle$, we have

$$(\boldsymbol{\alpha}_\star^{(m)})_j - (\boldsymbol{\alpha}_{(GLA)}^{(m)})_j = \sum_{i=1}^{m}\left(w_{j,m}(i) - \frac{1}{m}\right)s_j(i) = \sum_{i=1}^{m}\left(\frac{w_j(i)}{A_m} - \frac{1}{m}\right)s_j(i),$$

where $w_j(i) = |\lambda_j|^{2(i-1)}$ and $A_m = \sum_{i=1}^{m} w_j(i)$. Note that A_m is summable:

$$A_m = \sum_{i=1}^{m}|\lambda_j|^{2(i-1)} = \frac{1 - |\lambda_j|^{2m}}{1 - |\lambda_j|}. \tag{6.14}$$

We can define the limit of the optimal weights as

$$\bar{w}_j(i) = \lim_{m\to\infty}\frac{w_j(i)}{A_m} = |\lambda_j|^{2(i-1)}(1 - |\lambda_j|). \tag{6.15}$$

Clearly, since $\bar{w}_j(i) \to 0$ exponentially fast in i, for m large enough we have that $\bar{w}_j(1) > m^{-1}$ and $\bar{w}_j(m) < m^{-1}$. Since $\frac{w_j(i)}{A_m} \to \bar{w}_j(i)$ exponentially fast in m, for all m large enough $\frac{w_j(1)}{A_m} > m^{-1}$ and $\frac{w_j(m)}{A_m} < m^{-1}$. Let t_m be the largest integer k such that $\bar{w}_j(k) > m^{-1}$. Then

$$(\boldsymbol{\alpha}_\star^{(m)})_j - (\boldsymbol{\alpha}_{(GLA)}^{(m)})_j = \sum_{i=1}^{t_m}\left|\frac{w_j(i)}{A_m} - \frac{1}{m}\right|s_j(i) - \sum_{i=t_m+1}^{m}\left|\frac{w_j(i)}{A_m} - \frac{1}{m}\right|s_j(i). \tag{6.16}$$

Since $s_j(i) = \beta_j + \beta_3(\frac{\lambda_3}{\lambda_j})^{i-1}\langle \mathbf{z}_3\,,\,\mathbf{z}_j\rangle$ and we can vary β_3, $\mathbf{z}_3$, and $\langle \mathbf{z}_3\,,\,\mathbf{z}_j\rangle$ (subject to the above assumptions on the last two), it is clear that we can find signals $\{s_j(\cdot)\}$ such that $\lim_{m\to\infty}(\boldsymbol{\alpha}_\star^{(m)})_j - (\boldsymbol{\alpha}_{(GLA)}^{(m)})_j \not\to 0$ as $m \to \infty$.

What is interesting is that in the limit $\boldsymbol{\alpha}_{(GLA)}^{(m)} \to (\beta_1, \beta_2)^T$; i.e., $\boldsymbol{\alpha}_{(GLA)}^{(m)}$ is asymptotically correct or is a consistent estimator in statistical language. To see this, fix $\epsilon > 0$, and let $M \in \mathbb{N}$ be such that $|s_j(i) - \beta_j| < \frac{\epsilon}{2}$ for $m \geq M$. Then

$$\left|(\boldsymbol{\alpha}_{(GLA)}^{(m)})_j - \beta_j\right| = \left|\left(\frac{1}{m}\sum_{i=1}^{M}s_j(i) + \frac{1}{m}\sum_{i=M+1}^{m}s_j(i)\right) - \beta_j\right| \leq \frac{1}{m}\left|\sum_{i=1}^{M}s_j(i)\right| + \frac{m-M}{m}\frac{\epsilon}{2}. \tag{6.17}$$

For all m large enough, we have for each $j = 1, 2$, $\left|(\boldsymbol{\alpha}_{(GLA)}^{(m)})_j - \beta_j\right| < \epsilon$.

It is interesting that while each $\boldsymbol{\alpha}_\star{}^{(m)}$ gives the optimal reconstruction of m snapshots (i.e., it is an efficient estimator in statistical language), these weights are not asymptotically correct, whereas the GLA weights are suboptimal for any finite m but are asymptotically correct. This is summarized in Table 6.1.

TABLE 6.1
Comparison of estimators $\boldsymbol{\alpha}_\star^{(m)}$ and $\boldsymbol{\alpha}_{(GLA)}^{(m)}$. Efficiency is with respect to the loss function (6.4). *Consistency/asymptotic correctness is with respect to whether they converge to the correct eigenvector coefficients.*

Estimator	Consistent/Asymptotically Correct	Efficient/Optimal
$\boldsymbol{\alpha}_\star^{(m)}$	no	yes
$\boldsymbol{\alpha}_{(GLA)}^{(m)}$	yes	no

Remark 6.7. The reason that GLA weights were asymptotically correct is due to the fact eigenvalues λ_1, λ_2 associated with the reconstruction vectors $\mathbf{z}_1, \mathbf{z}_2$ dominated the eigenvalue of the unresolved eigenvector. This causes the signal $s_j(i)$ to converge exponentially fast to the correct weight β_j. This result will continue to hold if $|\lambda_3| = |\lambda_2|$. However, the result will be *false* if any eigenvalue of an unresolved eigenvector has a larger modulus than the reconstruction eigenvalues, i.e., if $|\lambda_2| > |\lambda_j|$ for $j = 1$ or 2.

6.2.3. Eigenvector-adapted Hermitian forms and reflexive g-inverses. The weight matrix $M \in \mathbb{C}^{m\ell \times m\ell}$ in Proposition 5.2 and the minimizer $\boldsymbol{\alpha}_\star$ can be recovered from a Hermitian form on $\mathbb{C}^n$ which is adapted to the eigenvector basis. This gives a connection with the abstract GLA theorem [31] in which appropriate spaces of observables for the Koopman operator were constructed by adapting the norm in order to orthogonalize the principal eigenfunctions and their products.

Let $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ be the eigenvectors of $\mathbb{A}$. These are not necessarily orthogonal with respect to the standard inner product, $\langle \cdot\,, \cdot \rangle_{\mathbb{C}^n}$. We can construct a weighted Hermitian form in which the first ℓ eigenvectors are orthogonal. Let $Z_\ell = \begin{pmatrix} Q_\ell & Q_\ell^\perp \end{pmatrix} \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix}$ be the QR factorization of Z_ℓ, where $\operatorname{Ran} Z_\ell = \operatorname{Ran} Q_\ell$, $R \in \mathbb{C}^{\ell \times \ell}$, and $Q = \begin{pmatrix} Q_\ell & Q_\ell^\perp \end{pmatrix}$ is unitary. Define the matrix $P \in \mathbb{C}^{n \times n}$ as

$$P = Q_\ell R^{-*} R^{-1} Q_\ell^* + Q_\ell^\perp Q_\ell^{\perp *}, \tag{6.18}$$

and define the bilinear form $\langle \cdot\,, \cdot \rangle_P : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ as $\langle \mathbf{x}\,, \mathbf{y} \rangle_P = \mathbf{y}^* P \mathbf{x}$.

PROPOSITION 6.8. *Let $1 \le i, j \le \ell$ and $\mathbf{y} \in \operatorname{Ran} Q_\ell^\perp$. Then*
(a) $\langle \mathbf{z}_i\,, \mathbf{z}_j \rangle_P = \delta_{i,j}$, *and*
(b) $\langle \mathbf{z}_i\,, \mathbf{y} \rangle_P = 0$.

Proof. For $i \le \ell$, $\mathbf{z}_i = Z_\ell \mathbf{e}_i = Q_\ell R \mathbf{e}_i$. Then

$$\langle \mathbf{z}_i\,, \mathbf{z}_j \rangle_P = \mathbf{z}_j^* P \mathbf{z}_i = \mathbf{z}_j^* Q_\ell R^{-*} R^{-1} Q_\ell^* \mathbf{z}_i = (R^{-1} Q_\ell^* \mathbf{z}_j)^* (R^{-1} Q_\ell^* \mathbf{z}_i) = (\mathbf{e}_j)^* \mathbf{e}_i = \delta_{i,j}. \tag{6.19}$$

Write $\mathbf{y} = Q_\ell^\perp \mathbf{c}$. Then since $Q_\ell^{\perp *} Q_\ell = \mathbf{0}$,

$$\langle \mathbf{z}_i\,, \mathbf{y} \rangle_P = \mathbf{y}^* P \mathbf{z}_i = \mathbf{c}^* Q_\ell^{\perp *} (Q_\ell R^{-*} R^{-1} Q_\ell^* + Q_\ell^\perp Q_\ell^{\perp *}) Q_\ell R \mathbf{e}_i = \mathbf{0}. \tag{6.20}$$

∎

With this inner product, for each $i \le \ell$, $\Pi_{\mathbf{z}_i}(\cdot) = \langle \cdot\,,\, \mathbf{z}_i\rangle_P \mathbf{z}_i$ is the *orthogonal* projection onto $\operatorname{span}\{\mathbf{z}_i\}$. It is easy to see that $\langle \cdot\,,\,\cdot\rangle_P$ is a positive, semidefinite Hermitian form. It is also nondegenerate; for any $\mathbf{y} \in \mathbb{C}^n$, if $\langle \mathbf{y}\,,\,\mathbf{x}\rangle_P = 0$ for all $\mathbf{x}$, then $\mathbf{y} = \mathbf{0}$. Therefore the Hermitian form generates a norm $\|\cdot\|_P = \sqrt{\langle \cdot\,,\,\cdot\rangle_P}$.

Remark 6.9. When the first ℓ eigenfunctions are orthonormal, the P-norm reduces to the canonical norm on the space. Indeed, if $Z_\ell = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$ are orthonormal, then the QR factorization satisfies $Z_\ell = Q\binom{I_\ell}{\mathbf{0}}$ where $Q = (Z_\ell\ Z_\ell^\perp)$ is unitary. Then for any $\mathbf{x} \in \mathbb{C}^n$

$$\begin{aligned}\|\mathbf{x}\|_P^2 &= \langle \mathbf{x}\,,\,\mathbf{x}\rangle_P = \mathbf{x}^*(Z_\ell I_\ell^{-*} I_\ell^{-1} Z_\ell^* + Z_\ell^\perp Z_\ell^{\perp *})\mathbf{x} = \mathbf{x}^*(Z_\ell Z_\ell^* + Z_\ell^\perp Z_\ell^{\perp *})\mathbf{x}\\ &= \mathbf{x}^*(QQ^*)\mathbf{x} = \|Q^*\mathbf{x}\|_2^2.\end{aligned}$$

Since Q^* is unitary, $\|Q^*\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$, which gives the result $\|\mathbf{x}\|_P = \|\mathbf{x}\|_2$.

We reformulate (6.4) using this P-norm:

$$\boldsymbol{\alpha}_P = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{C}^{\ell\times 1}} \sum_{i=1}^{m} \|\boldsymbol{f}_i - \sum_{j=1}^{\ell} \mathbf{z}_j \lambda_j^{i-1}\beta_j\|_P^2. \tag{6.21}$$

We write each $\mathbf{f}_i$ as $\mathbf{f}_i = \sum_{j=1}^{\ell}\langle \mathbf{f}_i\,,\,\mathbf{z}_j\rangle_P \mathbf{z}_j + \mathbf{h}_i$, where $\mathbf{h}_i \perp_P \operatorname{span}\{\mathbf{z}_j \,|\, j = 1, \ldots, \ell\}$.

Using Proposition 6.8, we get

(6.22)

$$\sum_{i=1}^{m}\left\|\mathbf{f}_i - \sum_{j=1}^{\ell}\mathbf{z}_j\lambda_j^{i-1}\beta_j\right\|_P^2 = \sum_{i=1}^{m}\|\mathbf{h}_i\|_P^2 + \left\|\sum_{j=1}^{\ell}\langle \mathbf{f}_i\,,\,\mathbf{z}_j\rangle_P\mathbf{z}_j - \sum_{j=1}^{\ell}\mathbf{z}_j\lambda_j^{i-1}\beta_j\right\|_P^2$$

$$= \sum_{i=1}^{m}\|\mathbf{h}_i\|_P^2 + \left\|\begin{pmatrix}\langle \mathbf{f}_i\,,\,\mathbf{z}_1\rangle_P\\ \vdots\\ \langle \mathbf{f}_i\,,\,\mathbf{z}_\ell\rangle_P\end{pmatrix} - \begin{pmatrix}\lambda_1^{i-1} & & \\ & \ddots & \\ & & \lambda_\ell^{i-1}\end{pmatrix}\boldsymbol{\beta}\right\|_2^2. \tag{6.23}$$

Using definition (6.18) of P and noting that $R^{-1}Q_\ell^*\mathbf{z}_j = \mathbf{e}_j$ and $\mathbf{z}_j^* Q_\ell^\perp = (Q_\ell^{\perp *}\mathbf{z}_j)^* = \mathbf{0}$, we have

$$\langle \mathbf{f}_i\,,\,\mathbf{z}_j\rangle_P = (\mathbf{z}_j)^*(Q_\ell R^{-*}R^{-1}Q_\ell^* + Q_\ell^\perp Q_\ell^{\perp *})\mathbf{f}_i = (R^{-1}Q_\ell^*\mathbf{z}_j)^*(R^{-1}Q_\ell^*\mathbf{f}_i) = (R^{-1}Q_\ell^*\mathbf{f}_i)_j,$$

and therefore,

(6.24)

$$\sum_{i=1}^{m}\left\|\mathbf{f}_i - \sum_{j=1}^{\ell}\mathbf{z}_j\lambda_j^{i-1}\beta_j\right\|_P^2 = \sum_{i=1}^{m}\|\mathbf{h}_i\|_P^2 + \left\|R^{-1}Q_\ell^*\mathbf{f}_i - \Delta_{\Lambda_i}\boldsymbol{\beta}\right\|_2^2$$

$$= \left\|(I_m \otimes R^{-1}Q_\ell^*)\begin{pmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_m\end{pmatrix} - \begin{pmatrix}\Delta_{\Lambda_1}\\ \vdots\\ \Delta_{\Lambda_m}\end{pmatrix}\boldsymbol{\beta}\right\|_2^2 + \sum_{i=1}^{m}\|\mathbf{h}_i\|_P^2. \tag{6.25}$$

If we define

(6.26)

$$T = \begin{pmatrix}\Delta_{\Lambda_1}\\ \vdots\\ \Delta_{\Lambda_m}\end{pmatrix}, \text{ then } T^\dagger = (T^*T)^{-1}T^* = \left(\sum_{k=1}^{m}\Delta_{\Lambda_i}^*\Delta_{\Lambda_i}\right)^{-1}\begin{pmatrix}\Delta_{\Lambda_1}^* & \cdots & \Delta_{\Lambda_\ell}^*\end{pmatrix}.$$

The coefficient vector that solves (6.21) is

$$\boldsymbol{\alpha}_P = T^{\dagger}(I_m \otimes R^{-1}Q_\ell^*)\begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_m \end{pmatrix}.$$

Noting that $Q_\ell^*\mathbf{f}_i = \boldsymbol{g}_i$ and expanding the above formula for $T^{\dagger}$, we recover formula (5.7) for the optimal reconstruction weights, i.e., $\boldsymbol{\alpha}_P = \boldsymbol{\alpha}_\star$.

7. Conclusions. We have presented a new, SVD-free variant of the dynamic mode decomposition (DMD) that follows the natural formulation in terms of Krylov bases. Using high accuracy numerical linear algebra techniques, we were able to curb the ill-conditioning of the companion matrix's associated Vandermonde matrix, allowing us to invert it and find the DMD modes. In addition to the inherent elegance in terms of the companion matrix formulation of DMD, this framework has a close connection to Koopman operator theory as well as to the result originating from generalized Laplace analysis theory. Existing theoretical analysis of high accuracy computations with Vandermonde and Cauchy matrices, backward error interpretability of the companion matrix based approach, together with numerical evidence, show that this approach is competitive with the SVD based DMD both theoretically and in practical computation. Furthermore, this scheme can be incorporated in a meta-algorithm which reconstructs data snapshots. We take up this line of enquiry in a companion paper [15].

8. Appendix. MATLAB codes.

8.1. Vand_DFT_LDU.

Algorithm 2. MATLAB implementation of (3.6).

```
function [XL, D, YU, P1, P2 ] = Vand_DFT_LDU( x, my, LDUorXDY )
% Vand_FFT_LDU computes entry-wise forward stable LDU decomposition of
% the matrix G=V(x)*DFT, where V(x)=fliplr(vander(x)) is the Vandermonde
% matrix defined by the real or complex vector x, and DFT is the discrete
% Fourier transform. V(x) is in  general rectangular with [length(x)] rows
% and [my] columns, V(x)_{ij} = x(i)^(j-1).  The LDU is computed with full
% pivoting. The code uses explicit formulas for the  Schur complement update.
% It is written for clarity, and not for optimality.
% On input:
% ~~~~~~~~~~
% x         :: real or complex vector that defines the Vandermonde matrix V(x).
% my        :: number of columns of V(x) and the dimension of the DFT matrix
% LDUorXDY :: job descrption; defines the factors on the output
%              If 'LDU' , then G(P1,P2) = XL * diag(D) * YU
%              If 'LU'  , then G(P1,P2) = XL * YU
%              If 'XDYT', then G = XL * diag(D) * YU'
%              If 'XYT' , then G = XL * YU'
% On exit:
% ~~~~~~~~
% XL, YU, D :: The computed factors. XL and YU are matrices and D is column
%              vector that defines diagonal matrix diag(D).
%              See the descripton of LDUorXDY.
% P1, P2    :: permutation matrices used in the pivoted LDU.
%              See the descripton of LDUorXDY.
% Coded by Zlatko Drmac, drmac@math.hr.
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

(Continued on next page.)

```
mx = max(size(x)) ; y = (exp(-2*pi*1i/my).^(0:my-1)).' ;
G  = zeros(mx,my) ; s = 1/sqrt(my) ; tol = sqrt(my)*eps ;
for r = 1 : mx, for c = 1 : my
if ( abs( x(r) - y(c) ) > tol )
G(r,c) = (s*(x(r)^my - 1)*y(c)) / (x(r)-y(c)) ;
else G(r,c) = prod(x(r)-y(1:c-1))*prod(x(r)-y(c+1:my)) * y(c) * s ; end
end; end
P1 = 1:mx ; P2 = 1:my ;
for k = 1 : min(mx,my)
[ colmax, jcm] = max( abs( G(k:mx,k:my) ), [] , 1 )        ;
[ ~, jm ] = max( colmax ) ; im = jcm(jm)+k-1 ; jm = jm+k-1 ;
if ( k ~= im )
itmp = P1(k)  ; P1(k)  = P1(im)  ; P1(im)  = itmp ;
tmp  = x(k)   ; x(k)   = x(im)   ; x(im)   = tmp  ;
vtmp = G(k,:) ; G(k,:) = G(im,:) ; G(im,:) = vtmp ;
end
if ( k~= jm )
itmp = P2(k)  ; P2(k)  = P2(jm)  ; P2(jm)  = itmp ;
tmp  = y(k)   ; y(k)   = y(jm)   ; y(jm)   = tmp  ;
vtmp = G(:,k) ; G(:,k) = G(:,jm) ; G(:,jm) = vtmp ;
end
if ( abs(G(k,k)) > realmin )
for r = k + 1 : mx, for c = k + 1 : my
if ( G(r,c) ~= 0 )
G(r,c) = G(r,c) * (x(r)-x(k))*(y(c)-y(k)) / ((y(c)-x(k))*(x(r)-y(k))) ;
else G(r,c) = -G(r,k)*G(k,c) / G(k,k) ; end
end; end
else G(k:mx,k:my) = 0 ; end
end
D  = diag(G) ; nD = find(abs(D) >= realmin , 1, 'last' ) ;
Dinv = diag([ 1./D(1:nD) ; zeros(min(mx,my)-nD,1)]) ;
XL = tril(G(1:mx,1:min(mx,my)),-1)*Dinv + eye(mx,min(mx,my)) ;
if ( strcmp( LDUorXDY, 'LDU') || strcmp( LDUorXDY, 'XDYT') )
YU = Dinv*triu(G(1:min(mx,my),1:my),1) + eye(min(mx,my),my) ;
else YU = triu(G(1:mx,1:my),1) + diag(D)*eye(min(mx,my),my) ; end
if ( strcmp( LDUorXDY, 'XDYT' ) || strcmp( LDUorXDY, 'XYT' ) )
rowpinv(P1) = 1 : mx ;  XL = XL(rowpinv,:)  ; colpinv(P2) = 1 : my ;
YU = YU(:,colpinv)' ; end
end
```

8.2. X_inv_Vandermonde.

8.3. Software license. The following license applies to the software in this section.

3-Clause BSD License.

Additional copyrights may follow.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors

Algorithm 3. MATLAB implementation of the formula (3.8).

```
function Y = X_inv_Vandermonde( z, X  )
% X_inv_Vandermonde computes Y = X*inv(V(z)), where X has m columns and
% V(z)=fliplr(vander(z)) is the m x m Vandermonde matrix defined by the
% m x 1 vector z; V(z)_{ij} = z(i)^(j-1), i,j=1,...,m.
%.....................................................................
% Coded by Zlatko Drmac, drmac@math.hr.
%.....................................................................
%
m = length(z) ;
[ L, D, U, p1, p2 ] = Vand_DFT_LDU( z, m, 'LDU' ) ;
Y = ifft(X,[],2)  ;
Y = ( ( Y(:,p2) / U ) * diag(sqrt(m)./D) ) / L ;
p1i(p1) = 1:m ; Y = Y(:,p1i) ; % p1i is the inverse of the permutation p1
end
```


REFERENCES

[1] H. Arbabi and I. Mezić, *Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator*, SIAM J. Appl. Dyn. Syst., 16 (2017), pp. 2096–2126, https://doi.org/10.1137/17M1125236.

[2] J. L. Aurentz, T. Mach, R. Vandebril, and D. S. Watkins, *Fast and backward stable computation of roots of polynomials*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 942–973, https://doi.org/10.1137/140983434.

[3] F. S. V. Bazán, *Conditioning of rectangular Vandermonde matrices with nodes in the unit disk*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 679–693, https://doi.org/10.1137/S0895479898336021.

[4] L. Berman and A. Feuer, *On perfect conditioning of Vandermonde matrices on the unit circle*, Electron. J. Linear Algebra, 16 (2007), pp. 157–161.

[5] D. A. Bini, P. Boito, Y. Eidelman, L. Gemignani, and I. Gohberg, *A fast implicit QR eigenvalue algorithm for companion matrices*, Linear Algebra Appl., 432 (2010), pp. 2006–2031.

[6] Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.

[7] Å. Björck, *Numerical Methods in Matrix Computations*, Springer, 2015.

[8] S. C. Eisenstat and I. Ipsen, *Relative perturbation results for eigenvalues and eigenvectors of diagonalisable matrices*, BIT, 38 (1998), pp. 502–509.

[9] K. K. Chen, J. H. Tu, and C. W. Rowley, *Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses*, J. Nonlinear Sci., 22 (2012), pp. 887–915.
[10] J. Demmel, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 562–580, https://doi.org/10.1137/S0895479897328716.
[11] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
[12] J. Demmel and P. Koev, *Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials*, Linear Algebra Appl., 417 (2006), pp. 382–396.
[13] F. M. Dopico and J. M. Molera, *Accurate solution of structured linear systems via rank-revealing decompositions*, IMA J. Numer. Anal., 32 (2012), pp. 1096–1116.
[14] Z. Drmač, I. Mezić, and R. Mohr, *Data driven modal decompositions: Analysis and enhancements*, SIAM J. Sci. Comput., 40 (2018), pp. A2253–A2285, https://doi.org/10.1137/17M1144155.
[15] Z. Drmač, I. Mezić, and R. Mohr, *On Least Squares Problems with Certain Vandermonde–Khatri–Rao Structure with Applications to DMD*, preprint, https://arxiv.org/abs/1811.12562, 2018.
[16] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994, https://doi.org/10.1137/S0036142995292633.
[17] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm:* I, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1322–1342, https://doi.org/10.1137/050639193.
[18] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm:* II, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1343–1362, https://doi.org/10.1137/05063920X.
[19] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Math. Comp., 64 (1995), pp. 763–776.
[20] W. Gautschi, *How (un)stable are Vandermonde systems?*, in Asymptotic and Computational Analysis, Lecture Notes Pure Appl. Math. 124, R. Wong, ed., CRC Press, 1990, pp. 193–210.
[21] W. Gautschi, *Optimally conditioned Vandermonde matrices*, Numer. Math., 24 (1975), pp. 1–12.
[22] W. Gautschi, *Optimally scaled and optimally conditioned Vandermonde and Vandermonde-like matrices*, BIT, 51 (2011), pp. 103–125.
[23] D. J. Higham, *Condition numbers and their condition numbers*, Linear Algebra Appl., 214 (1995), pp. 193–213.
[24] N. J. Higham, *Error analysis of the Björck-Pereyra algorithms for solving Vandermonde systems*, Numer. Math., 50 (1987), pp. 613–632.
[25] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, *Sparsity-promoting dynamic mode decomposition*, Phys. Fluids, 26 (2014), 024103.
[26] R.-C. Li, *Relative perturbation theory:* II. *Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492, https://doi.org/10.1137/S0895479896298506.
[27] U. Luther and K. Rost, *Matrix exponentials and inversion of confluent Vandermonde matrices*, Electron. Trans. Numer. Anal., 18 (12004), pp. 91–100.
[28] I. Mezić, *Spectral properties of dynamical systems, model reduction and decompositions*, Nonlinear Dynam., 41 (2005), pp. 309–325.
[29] I. Mezić, *Analysis of fluid flows via spectral properties of the Koopman operator*, Ann. Rev. Fluid Mech., 45 (2013), pp. 357–378.
[30] I. Mezic, *Koopman Operator Spectrum and Data Analysis*, preprint, https://arxiv.org/abs/1702.07597, 2017.
[31] R. Mohr and I. Mezić, *Construction of Eigenfunctions for Scalar-Type Operators via Laplace Averages with Connections to the Koopman Operator*, preprint, http://arxiv.org/1403.6559, 2014.
[32] V. Y. Pan, *How bad are Vandermonde matrices?*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 676–694, https://doi.org/10.1137/15M1030170.
[33] V. Y. Pan and A.-L. Zheng, *New progress in real and complex polynomial root-finding*, Comput. Math. Appl., 61 (2011), pp. 1305–1334.
[34] C. R. Rao and S. K. Mitra, *Generalized inverse of a matrix and its applications*, in Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics, University of California Press, 1972, pp. 601–620.
[35] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, J. Fluid Mech., 641 (2009), pp. 115–127.

[36] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, *Applications of the dynamic mode decomposition*, Theoret. Comput. Fluid Dynam., 25 (2011), pp. 249–259.
[37] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech., 656 (2010), pp. 5–28.
[38] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.
[39] B. Suri, J. Tithof, R. Mitchell, R. O. Grigoriev, and M. F. Schatz, *Velocity profile in a two-layer Kolmogorov-like flow*, Phys. Fluids, 26 (2014), 053601.
[40] J. Tithof, B. Suri, R. K. Pallantla, R. O. Grigoriev, and M. F. Schatz, *Bifurcations in a quasi-two-dimensional Kolmogorov-like flow*, J. Fluid Mech., 828 (2017), pp. 837–866.
[41] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications*, J. Comput. Dynam., 1 (2014), pp. 391–421.
[42] L. R. Turner, *Inverse of the Vandermonde Matrix with Applications*, NASA Technical Note D–3547, Lewis Research Center, Cleveland, 1966.
[43] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[44] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition*, J. Nonlinear Sci., 25 (2015), pp. 1307–1346.