

Numerical linear algebra for data driven analysis of nonlinear dynamics: Koopman-Schur Decomposition

Zlatko Drmač, Igor Mezić

Abstract

The Extended Dynamic Mode Decomposition (DMD/EDMD) has become a tool of trade in computational data driven analysis of complex nonlinear dynamical systems, e.g. fluid flows, where it can be used to reveal coherent structures by decomposing the flow field into component fluid structures, called DMD modes, that describe the evolution of the flow. The theoretical underpinning of the EDMD is the Koopman composition operator that can be used for spectral analysis of nonlinear dynamical system [6]. The numerical realization and software implementation pose several challenges to numerical linear algebra, and this contribution discusses few selected ones.

To set the stage, consider the initial value problem

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t)) \equiv \begin{pmatrix} F_1(\mathbf{x}(t)) \\ \vdots \\ F_N(\mathbf{x}(t)) \end{pmatrix}, \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (1)$$

with state space $\mathcal{X} \subset \mathbb{R}^N$ and vector-valued nonlinear function $F : \mathcal{X} \rightarrow \mathbb{R}^N$. The corresponding flow map ϕ^t is defined as $\phi^t(\mathbf{x}(t_0)) = \mathbf{x}(t_0 + t) = \mathbf{x}(t_0) + \int_{t_0}^{t_0+t} F(\mathbf{x}(\tau))d\tau$.

Instead of the states, consider observables (functions of the states) $f : \mathcal{X} \rightarrow \mathbb{C}$, $f \in \mathcal{F}$; e.g. $\mathcal{F} = L^p(\mathcal{X}, \mu)$ ($1 \leq p \leq \infty$). Koopman operator semigroup $(\mathcal{U}_{\phi^t})_{t \geq 0}$ is defined by $\mathcal{U}_{\phi^t} f = f \circ \phi^t$, $f \in \mathcal{F}$. The Koopman (composition) operator \mathcal{U}_{ϕ^t} is linear operator that can be considered an infinite dimensional linearization of (1) that takes the action into the space \mathcal{F} of observables.

In the case of discrete dynamical system $\mathbf{x}_{i+1} = T(\mathbf{x}_i)$, the Koopman operator $\mathcal{U} \equiv \mathcal{U}_T$ is defined on a space of observables \mathcal{F} by $\mathcal{U}f = f \circ T$, $f \in \mathcal{F}$. In practical computation, one always works with discrete systems. If we run a numerical simulation of the ODE's (1) in a time interval $[t_0, t_*]$, the numerical solution is obtained on a discrete equidistant grid with fixed time lag Δt : $t_0, t_1 = t_0 + \Delta t, \dots, t_{i-1} = t_{i-2} + \Delta t, t_i = t_{i-1} + \Delta t, \dots$. In this case, a black-box software toolbox acts as a discrete dynamical system $\mathbf{z}_i = T(\mathbf{z}_{i-1})$ that produces the discrete sequence of $\mathbf{z}_i \approx \mathbf{x}(t_i)$. For $t_i = t_0 + i\Delta t$ we have $f(\mathbf{x}(t_0 + i\Delta t)) = (f \circ \phi^{i\Delta t})(\mathbf{x}(t_0)) = (\mathcal{U}_{\phi^{i\Delta t}} f)(\mathbf{x}(t_0)) = (\mathcal{U}_{\phi^{\Delta t}}^i f)(\mathbf{x}(t_0))$, where $\mathcal{U}_{\phi^{\Delta t}}^i = \mathcal{U}_{\phi^{\Delta t}} \circ \dots \circ \mathcal{U}_{\phi^{\Delta t}}$.

On the other hand, using $\mathcal{U}f = f \circ T$, $\mathbf{z}_i \approx \mathbf{x}(t_i)$, $T^2 = T \circ T$, $T^i = T \circ T^{i-1}$, $f(\mathbf{z}_i) = f(T(\mathbf{z}_{i-1})) = \dots = f(T^i(\mathbf{z}_0)) = (\mathcal{U}^i f)(\mathbf{z}_0)$. Hence, in a software simulation of (1) with the initial condition $\mathbf{z}_0 = \mathbf{x}(t_0)$, we have an approximation $(\mathcal{U}^i f)(\mathbf{z}_0) \approx (\mathcal{U}_{\phi^{\Delta t}}^i f)(\mathbf{z}_0)$, $f \in \mathcal{F}$, $\mathbf{z}_0 \in \mathcal{X}$, $i = 0, 1, 2, \dots, m$. Such a sequence of values from the trajectory of the system may be also obtained by experimental measurements (e.g. high speed camera recording, wind tunnel measurements, particle image velocimetry/thermometry), without using/knowning the governing equations. In general, the observables are vector valued, $\mathbf{f} = (f_1, \dots, f_n)^T$, and $\mathcal{U}\mathbf{f} = (\mathcal{U}f_1, \dots, \mathcal{U}f_n)^T$ is defined component-wise. Often $n \gg m$. The acquired numerical values of the observables (data snapshots) are assembled column-wise into the matrix (column index corresponds to discrete time step)

$$F = (\mathbf{f}(\mathbf{x}_0) \dots \mathbf{f}(\mathbf{x}_{m-1}) \mathbf{f}(\mathbf{x}_m)) = \begin{pmatrix} f_1(\mathbf{x}_0) & \dots & f_1(\mathbf{x}_{m-1}) & f_1(\mathbf{x}_m) \\ f_2(\mathbf{x}_0) & \dots & f_2(\mathbf{x}_{m-1}) & f_2(\mathbf{x}_m) \\ \vdots & & \vdots & \vdots \\ f_n(\mathbf{x}_0) & \dots & f_n(\mathbf{x}_{m-1}) & f_n(\mathbf{x}_m) \end{pmatrix}, \quad \mathbf{f}(\mathbf{x}_{i+1}) = \mathbf{f}(T(\mathbf{x}_i)) = (\mathcal{U}\mathbf{f})(\mathbf{x}_i) = \mathbf{f}(T(T(\mathbf{x}_{i-1}))).$$

The columns of the data snapshot matrix F can be interpreted as the Krylov sequence $\mathbf{f}, \mathcal{U}\mathbf{f}, \dots, \mathcal{U}^m\mathbf{f}$, evaluated at \mathbf{x}_0 , which motivates looking for approximate eigenvalues and eigenvectors of \mathcal{U} .

Why is spectral data of \mathcal{U} interesting? If $(\mathcal{U}\phi_i)(s) \approx \lambda_i\phi_i(s)$, $i = 1, \dots, m$, and if for some carefully and judiciously selected $\lambda_{i_1}, \dots, \lambda_{i_\ell}$ and vector coefficients \mathbf{z}_{i_j} (that must be computed)

$$\mathbf{f}(s) \approx \sum_{j=1}^{\ell} \mathbf{z}_{i_j} \phi_{i_j}(s), \text{ then } (\mathcal{U}^k \mathbf{f})(s) = \begin{pmatrix} (\mathcal{U}^k f_1)(s) \\ \vdots \\ (\mathcal{U}^k f_n)(s) \end{pmatrix} \approx \sum_{j=1}^{\ell} \mathbf{z}_{i_j} \phi_{i_j}(s) \lambda_{i_j}^k, \quad k = 0, 1, \dots \quad (2)$$

This decomposition (called the Kopman Mode Decomposition, KMD) reveals the latent structure of the dynamics (in particular when ℓ is relatively small) and allows for forecasting future values, because applying the powers of \mathcal{U} in (2) means pushing the dynamics forward in time.

In this decomposition, the vector coefficients \mathbf{z}_i 's are approximate eigenvectors of a matrix A such that $A\mathbf{f}(\mathbf{x}_i) = \mathbf{f}(T(\mathbf{x}_i))$ for all i . The matrix A is the DMD matrix – it may not be uniquely determined by the data and only certain Ritz pairs can be computed by a data driven version of the classical Rayleigh-Ritz extraction procedure, as in the DMD algorithm [8] and its enhancement [5] that provides computable residuals and uses them to select physically meaningful eigenvalues and modes, and to guide sparse representation of the snapshot in the KMD.

But, there is a problem. One of the computational/numerical challenges in the Koopman/DMD framework is the case of non-normal operators, when the computed (Ritz) eigenvectors of the DMD matrix become severely ill-conditioned. High non-normality of the eigenmodes is not just a mathematically manufactured and for the sake of academic exercise contrived misfortune. It does occur in important applications. For instance, Schmid [7] discusses examples of non-normal operators in fluid flows, and the impact of non-normality on treatment of stability of such flows. A well-known and intensively studied example is the formulation of the viscous stability problem for parallel shear flows, in which linearization of the Navier-Stokes equation leads to the Orr-Sommerfeld linear partial differential equation whose solutions exhibit highly non-normal behavior.

The severity of the problem can be easily demonstrated by running a numerical simulation and visualizing the pseudospectrum and computing the angles between the eigenvectors. This issue is mostly ignored in the DMD literature, but the practitioners have experienced the problem in applications, and it is listed in [10] as one of the challenges in applications of the DMD.

To alleviate the problem of ill-conditioned eigenvectors in the existing implementations of the Dynamic Mode Decomposition (DMD) and the Extended Dynamic Mode Decomposition (EDMD, [9]), in [4] we introduce a Koopman-Schur decomposition – Schur decomposition of a data driven compression of the Koopman operator onto a subspace \mathcal{F}_N defined by a given dictionary ψ_1, \dots, ψ_N .

The first step in this approach is as follows. As in the EDMD, compute the data driven compression

$$P_{\mathcal{F}_N} \mathcal{U}|_{\mathcal{F}_N} \left(\begin{pmatrix} \psi_1(x) & \dots & \psi_N(x) \end{pmatrix} \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{pmatrix} \right) = \begin{pmatrix} \psi_1(x) & \dots & \psi_N(x) \end{pmatrix} \left(U_N \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{pmatrix} \right),$$

where $P_{\mathcal{F}_N}$ stands for the algebraic least squares projection using the available data, and U_N is the matrix of the compression. With the notation $\Psi(x) = (\psi_1(x), \dots, \psi_N(x))^T \in \mathbb{C}^N$, the action of \mathcal{U} on \mathcal{F}_N can be compactly written as $\mathcal{U}(\Psi(\mathbf{x})^T \mathbf{z}) = \Psi(\mathbf{x})^T U_N \mathbf{z} + R(\mathbf{x})^T \mathbf{z}$, $\mathbf{x} \in \mathbb{C}^n$, $\mathbf{z} \in \mathbb{C}^N$, where $R(x) = (\rho_1(x), \dots, \rho_N(x))^T \in \mathbb{C}^N$ is the residual that has been minimized over the available data. In this setting, a well defined object is, instead of U_N , another compression – U_N is compressed

onto a $N \times r$ POD basis V (computed using the SVD of a data snapshot matrix, with $r < N$) and replaced with the $r \times r$ Rayleigh quotient \widehat{U} . In the proposed approach [4], we compute a Schur decomposition of \widehat{U} , $\widehat{U} = QTQ^*$, and $U_N Z = ZT$ becomes a partial Schur form with $Z = VQ$. On the operator level, this becomes

$$\begin{aligned} \mathcal{U}((\psi_1(x) \dots \psi_N(x)) Z) &= (\psi_1(x) \dots \psi_N(x)) U_N Z + R(x)^T Z \\ &= (\psi_1(x) \dots \psi_N(x)) ZT + R(x)^T Z \simeq (\psi_1(x) \dots \psi_N(x)) ZT. \end{aligned} \quad (3)$$

If we define a new sequence $(\zeta_1(x) \dots \zeta_r(x)) = (\psi_1(x) \dots \psi_N(x)) Z$, then (3) reads

$$\mathcal{U}(\zeta_1(x) \dots \zeta_r(x)) = (\zeta_1(x) \dots \zeta_r(x)) T + R(x)^T Z \simeq (\zeta_1(x) \dots \zeta_r(x)) T. \quad (4)$$

Since T is upper triangular, (4) contains a nested sequence of partial triangulations

$$\mathcal{U}(\zeta_1 \quad \zeta_2 \quad \dots \quad \zeta_i) \simeq (\zeta_1 \quad \zeta_2 \quad \dots \quad \zeta_i) T(1:i, 1:i), \quad i = 1, \dots, r. \quad (5)$$

This Schur form can be reordered – with a suitable unitary matrix Θ , $\widetilde{T} = \Theta^* T \Theta$ is again upper triangular with diagonal entries corresponding to the eigenvalues in any given order. The new partial Schur form of U_N becomes $U_N(VQ\Theta) = (VQ\Theta)\widetilde{T}$, $\widetilde{T} = \Theta^* T \Theta$, and we replace (4) with

$$\mathcal{U}(\zeta_1(x) \quad \zeta_2(x) \quad \dots \quad \zeta_r(x)) \Theta \simeq (\zeta_1(x) \quad \zeta_2(x) \quad \dots \quad \zeta_r(x)) \Theta(\Theta^* T \Theta), \quad (6)$$

i.e. the new functions are generated using $\widetilde{Z} = Z\Theta = VQ\Theta$ ($\widetilde{Z}^* \widetilde{Z} = I_r$),

$$\left(\widetilde{\zeta}_1(x) \quad \dots \quad \widetilde{\zeta}_r(x) \right) = (\zeta_1(x) \quad \dots \quad \zeta_r(x)) \Theta = (\psi_1(x) \quad \dots \quad \psi_N(x)) \widetilde{Z}. \quad (7)$$

In this framework, the spectral analysis of the snapshots, including the KMD (2), is entirely based on unitary transformations. The analysis in terms of the eigenvectors as modes of a Koopman operator compression is replaced with a modal decomposition in terms of a flag of invariant subspaces that correspond to selected eigenvalues – the partial ordered Schur decomposition provides convenient orthonormal bases for subspaces determined by any given selection $\lambda_{i_1}, \dots, \lambda_{i_\ell}$ of the eigenvalues.

From this point, we proceed in two directions. First, we analyze the convergence (as the size of the dictionary and the number of data snapshots become infinite) to obtain results analogous to the EDMD. Then, to have the same functionality as the existing EDMD (snapshot reconstruction using selected eigenvalues, dynamically changed data window in online applications, forecasting, formulation with the kernel trick etc.), many technical (algorithms and software related) details have to be worked out. For instance, in the case of real data, a real ordered partial Schur form is used and the computation is based on real orthogonal transformations, even when the spectrum is complex. Other details include e.g. streaming data with dynamically changing data windows. Numerical experiments show superior performances in the numerically difficult non-normal cases.

The second topic that will be discussed is numerical implementation of the Hermitian case of the physics informed DMD [1], [3] – if it is a priori known that the underlying operator is Hermitian, how to ensure that the numerical implementation of the DMD guarantees real spectrum and orthonormal eigenvectors? What are the other issues when it comes to software implementation [2] e.g. in the framework of the LAPACK library?

These themes are excellent case study examples that demonstrate the importance of numerical linear algebra and of the power of numerical analysis of an algorithm – it precisely predicts in what way it may fail and indicates what has to be done to provably fix the problem.

References

- [1] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. N. Kutz, and S. L. Brunton. Physics-informed dynamic mode decomposition (piDMD). *arXiv e-prints*, page arXiv:2112.04307, December 2021.
- [2] Zlatko Drmač. A LAPACK implementation of the Dynamic Mode Decomposition. *ACM Trans. Math. Softw.*, jan 2024.
- [3] Zlatko Drmač. Hermitian Dynamic Mode Decomposition – numerical analysis and software solution. *ACM Trans. Math. Softw.*, jan 2024.
- [4] Zlatko Drmač and Igor Mezić. A data driven Koopman–Schur decomposition for computational analysis of nonlinear dynamics. *arXiv e-prints*, page arXiv:2312.15837, December 2023.
- [5] Zlatko Drmač, Igor Mezić, and Ryan Mohr. Data driven modal decompositions: Analysis and enhancements. *SIAM Journal on Scientific Computing*, 40(4):A2253–A2285, 2018.
- [6] Clarence W. Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.
- [7] Peter J Schmid. Nonmodal stability theory. *Annual review of fluid mechanics*, 39(1):129–162, 2007.
- [8] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [9] M.O. Williams, I.G. Kevrekidis, and C.W. Rowley. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.
- [10] Ziyou Wu, Steven L. Brunton, and Shai Revzen. Challenges in dynamic mode decomposition. *Journal of The Royal Society Interface*, 18(185):20210686, 2021.