LA-UR-20-26751
Approved for public release; distribution is unlimited.

Title: Survey of Dynamic Mode Decomposition Methods

Author(s): Hardy, Zachary Kenneth

Intended for: Report

Issued: 2020-08-31

Survey of Dynamic Mode Decomposition Methods

Zachary Hardy[†]
X-Theoretical Design Division, Primary Physics
Los Alamos National Laboratory
P.O. Box 1663, Los Alamos, NM 87545

August 24, 2020

Introduction

Dynamic mode decomposition (DMD) is a data-driven reduced order modeling (ROM) technique used for dynamic systems. The widely adopted algorithm was first introduced and demonstrated on fluid flow data by Schmid in [11]. In recent years, various other fields, such as nuclear engineering, have begun to adopt this method. For example, DMD has been used for estimating α-eigenvalues [8], as an ROM for pulsed neutron problems [6], for predicting isotopic composition in burnup calculations [1], as acceleration techniques for iterative methods [9] [10], and in capturing dynamic behaviors in molten salt reactor transients [3]. This report seeks to demonstrate the capabilities and limits of the standard DMD algorithm, and identify problem spaces where variants may be better suited. The primary variant this report considers is Multi-Resolution DMD (mrDMD). Because this serves as a survey, synthetically produced data is used in lieu of simulation results. The remainder of this report will go into detail on the DMD theory, outline the standard DMD and mrDMD algorithms, present test cases highlighting the applicability of each, and finally present a discussion on how to determine the best suited algorithm for a given problem. All calculations performed in this report are carried out using the open source DMD library, PyDMD [2].

Dynamic Mode Decomposition

DMD is an empirical method for dynamic systems that seeks to approximate a low-rank representation of the underlying evolution operator. Tu *et. al.* extended this notion to arbitrary operators in [12]. In the nuclear engineering community, this has been applied to approximating general operators in [10] and [9]. To use DMD, knowledge of the underlying operator is not required, only its action. For this reason, this method is applicable for both experimental and simulation data.

DMD assumes a mapping, $\mathcal{A}$, that maps one snapshot to the next, such that

$$\vec{x}_{n+1} = \mathcal{A}\vec{x}_n, \tag{1}$$

where $\vec{x} \in \mathbb{R}^M$ and $\mathcal{A} \in \mathbb{R}^{M\times M}$ with $M \gg N$. The method approximates the dynamics contained within $\mathcal{A}$ with a superposition of modes with associated decay/growth rates, given by

$$x_{\mathrm{DMD}}(t) = \sum_{k=0}^{K-1} b_k \vec{\phi}_k \exp\left(\omega_k t\right), \tag{2}$$

[†]This work was performed at Texas A&M University as a summer student for Los Alamos National Laboratory.

where $\vec{\phi}_k$ is a dynamic mode, ω_k is an eigenvalue associated with a dynamic mode, and b_k is an amplitude. While this is a linear representation of the dynamics, with the connection of DMD to the Koopman operator by Tu et. al. in [12], its use in representing non
dynamics became justified. The reader is referred to this paper for a more detailed discussion.

For dynamic systems, this can be much more physically insightful than its counterparts such as principal orthogonal decomposition (POD) in that one can gain insights into both the underlying spatial structures and time scales present within the snapshots. Additionally, this representation can be used for near future state predictions. The remainder of this section will outline the standard DMD algorithm and the mrDMD algorithm.

Standard Algorithm

As discussed before, DMD seeks to approximate the evolution operator, $\mathcal{A}$, which maps one snapshot to the next. This is given in matrix form by

$$X_1 = \mathcal{A}X_0, \tag{3}$$

where $X_0 = \{\vec{x}_0,\ \vec{x}_1,\ \ldots,\ \vec{x}_{N-1}\} \in \mathbb{R}^{M\times N-1}$, and X_1 is defined similarly. The first step of DMD is to take the singular value decomposition (SVD) of X_0. This gives $X_0 = U\Sigma V^*$, where $U \in \mathbb{C}^{M\times M}$ contains the left singular vectors, $\Sigma \in \mathbb{C}^{M\times N-1}$ is a diagonal rectangular matrix containing the singular values, and $V^* \in \mathbb{C}^{N-1\times N-1}$ is the complex conjugate of the right singular vectors. In practice, a truncation step occurs here and only r modes are kept. This leads to $U \in \mathbb{C}^{M\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$, and $V^* \in \mathbb{C}^{r\times N-1}$. The natural question that arises from this is how to choose r. There are many different methods for this. As suggested in Kutz [7], the error in a DMD representation of data is proportional to the largest truncated singular value such that

$$e_{\text{DMD}} = \|X - X_{DMD}\|_{L^2} \approx \sigma_{r+1}. \tag{4}$$

With this, one can truncate the SVD to a level representative of a desired level of accuracy. Another useful insight is that if the data set is produced by linear, time invariant dynamics, with a sufficient number of snapshots, a DMD representation can be formed that reproduces the original snapshots to round-off error. It is important to note that this holds only if linear dynamics produce the data set. The singular values in reality show how well the singular vectors (POD modes) span the data set. For this reason, when the similarity relationship is formed, if the POD modes span the snapshots, then the eigendecomposition of the underlying operator is exact to round-off. Otherwise, the eigendecomposition represents an approximation to the Koopman operator and is therefore not exact. Alternatives include the optimal hard threshold [5], using regularization to reduce the number of modes, and more.

Plugging the truncated SVD representation back into Eq. (3)

$$X_1 = \mathcal{A}U\Sigma V^*. \tag{5}$$

Multiplying on the left and right by U^* and $\Sigma^{-1}V$ respectively gives

$$\tilde{\mathcal{A}} = U^*AU = U^*X_1\Sigma^{-1}V. \tag{6}$$

This represents a similarity transformation to a lower-dimensional system where $\tilde{\mathcal{A}} \in \mathbb{C}^{r\times r}$.

An eigendecomposition of this system yields the low-rank eigenvectors $\vec{z} \in \mathbb{C}^r$ and eigenvectors λ. The dynamic modes are then given by a projection onto the POD basis, U, as $\vec{\phi} = U\vec{z} \in \mathbb{C}^M$. The continuous form of the eigenvalues are determined via $\omega = \log(\lambda)/\Delta t$ The last step of the DMD algorithm is to determine the amplitudes, b of each mode. While there are various methods to do this, generally, one will compute this via a least-squares fit to the initial condition, $\vec{x}_0$.

Multi-Resolution Algorithm

mrDMD is a variant of the standard DMD algorithm which aims to separate slow and fast modes. In this context, slow modes are those that are prevalent over the full sampling period and fast modes are those which are prevalent only in some slices of the sampling period. To the author's knowledge, little literature exists outside of [4] and [7] on physics applications of this algorithm. This algorithm is best suited for problems in which the fundamental assumptions of DMD do not hold over the full domain, but may approximately hold over sub-slices. This scenario could arise when there are strongly nonlinear dynamics, signals are only present for part of the sampling period, or signals are non-stationary.

Before outlining the algorithm, it is important to define some terms. The mrDMD algorithm is a recursive algorithm that performs standard DMD on increasingly small slices of the domain. Each subsequent refinement is defined as a level, and each split, or time level, on a given level is referred to as a node. This notation is adopted from [7]. On each level, slow and fast modes are defined. The distinction between slow and fast modes has flexibility. As an example, this can be defined by some measure such as the number of oscillations over the sampling period or perhaps by growth/decay rates relative to the period. At the end of the algorithm, the solution has the form

$$x_{\text{mrDMD}}(t) = \sum_{\ell=0}^{L-1} \sum_{j=0}^{J_\ell - 1} \sum_{k=0}^{K_{\ell,j}-1} f^{\ell,j}(t) b_k^{\ell,j} \phi_k^{\ell,j} \exp\left(\omega_k^{\ell,j} t\right), \tag{7}$$

where L is the number of levels, J_ℓ is the number nodes on level ℓ, and $K_{\ell,j}$ is the number of modes on node j of level ℓ. The function $f^{\ell,j}$ is a mapping function defined by

$$f^{\ell,j}(t) = \begin{cases} 1, & t \in [t_{\ell,j},\ t_{\ell,j+1}] \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

The algorithm starts by considering the full series of data. One performs a standard DMD on this data set and then defines fast and slow modes based upon the specified criterion. The domain is then split in half and the process repeated. This continues until either the maximum number of levels is reached or no slow modes exist remain on the finest level. While not explored in this paper, this algorithm can be easily expanded to be adaptive in which some regions are refined to more levels than others.

Oftentimes, to reduce the cost at each level, one may sparsely sample the snapshots such that the same number of snapshots are considered at each node. As an example, one could take every eighth snapshot on the zeroth level, every fourth on the first level, and so forth.

Pure Growth/Decay Problems

This section will discuss the application of DMD to growth and decay problems. In the nuclear engineering community, this comprises of the time dependent neutron diffusion or transport equations and the nuclide production-destruction equations characteristic of burnup calculations. These applications have been explored in [6], [8], and [1]. This section will explore the efficacy of standard DMD when various different time scales exist. First, an example of a simple linear problem with a small number of signals with large time scale separation is considered. Next, an example of many signals with time scales varying between the bounds of those before is explored. Lastly, the previous example is repeated, except a few signals are given out-sized amplitudes relative to the remainder of the signals.

Large Time Scale Separation

First, a 1D problem where time-scales that span many orders of magnitude is considered. For simplicity, no oscillatory behaviors in time are considered in this example. The signals in this example are of the form

$$f(x,t) = A\cos\left(\frac{n\pi x}{L}\right)\exp\left(\frac{\omega t}{T}\right),$$

where A is the amplitude, n is the frequency of the spatial component of the signal, L is the width of the spatial domain, ω is the time-scale that governs the signal, and T is the width of the temporal domain. As a first example, a three signal data-set is used. The spatial domain for this problem is $x \in [-1, 1]$ and the temporal domain is $t \in [0, 1]$. The data set is comprised of 41 snapshots of length 400. The parameters for the amplitudes, spatial frequencies and time-scales are given in Table 1, the profiles and time dynamics can be seen in Figure 1. The total signal can be seen in Figure 2. The total signal is normalized to its Frobenius-norm.

Table 1: Signal parameters for example 1.

Signal Number	A	n	ω
0	1	1	-1e-3
1	1	3	-1.0
2	1	5	-1e3

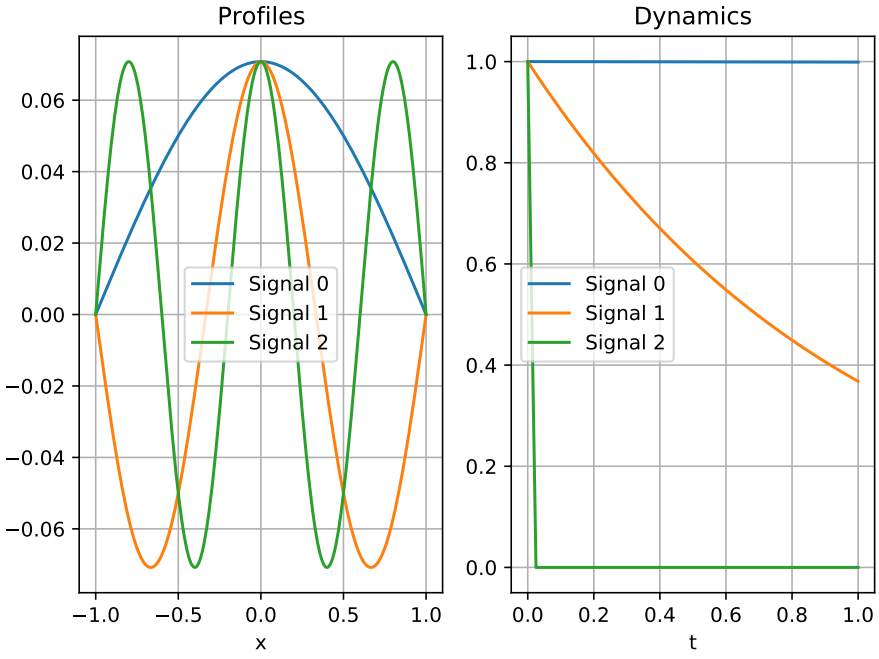

Figure 1: Signals for example 1.

The singular value spectrum for first 40 snapshots is shown in Figure 3.

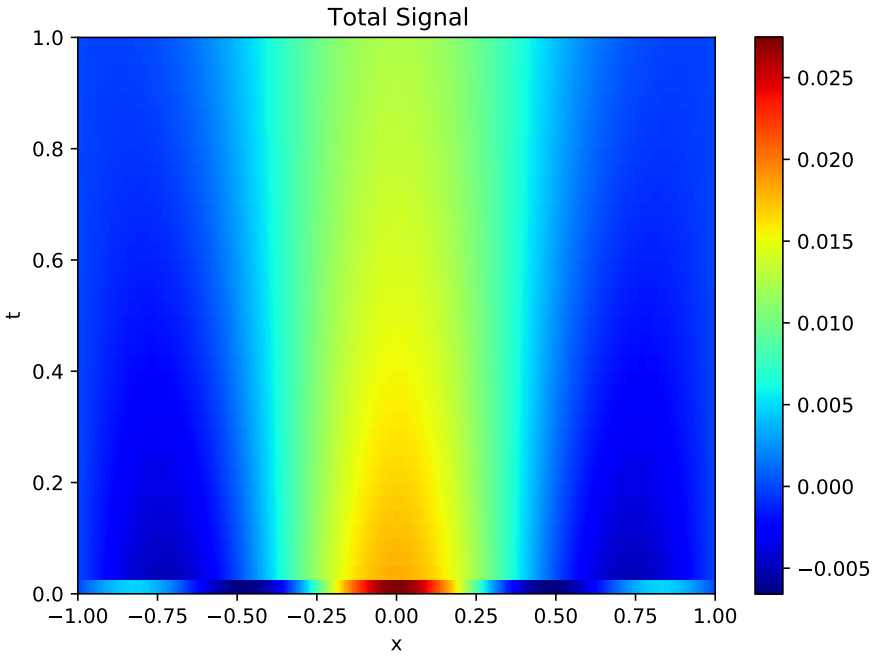

Figure 2: Total signal for example 1.

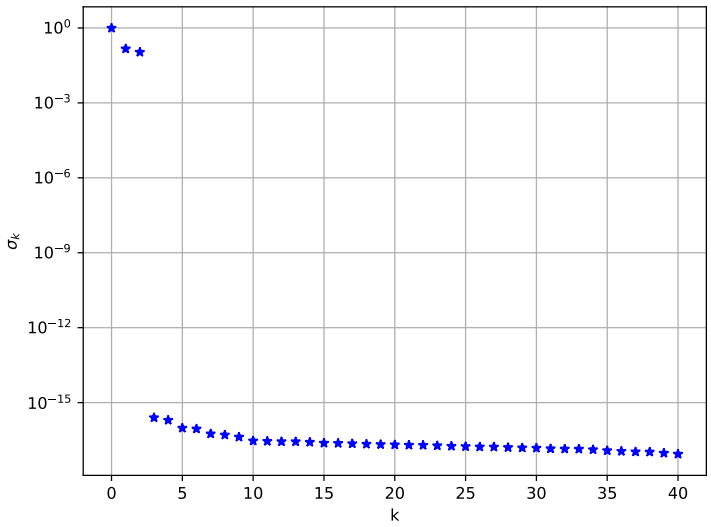

Figure 3: Singular value spectrum for example 1.

From this, it is seen that the data set is spanned by 3 POD modes, implying that the system can be exactly characterized by 3 DMD modes.

Performing standard DMD yields the modes and dynamics observed in Figure 4.

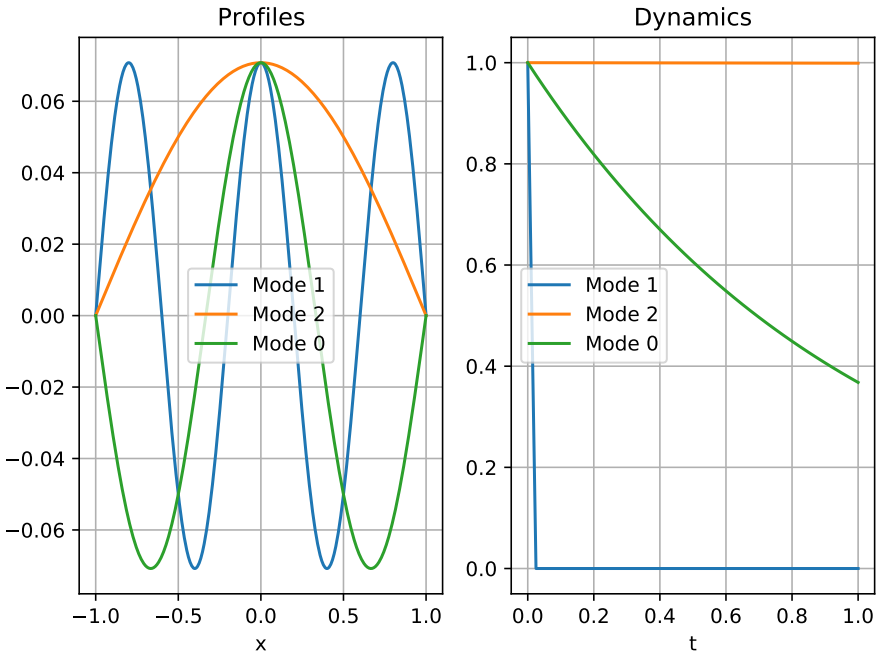

Figure 4: Dynamic modes for example 1.

Visually, the modes' profiles and dynamics are identical. Additionally, the extracted eigenvalues from DMD are identical to those of the signals. This is observed in Figure 5, where the discrete DMD eigenvalues are plotted against the signal eigenvalues. The discrete DMD eigenvalues are those obtained directly from the eigendecomposition of $\tilde{\mathcal{A}}$. The signal eigenvalues are obtained by the mapping $\lambda = \exp(\omega \Delta t)$.

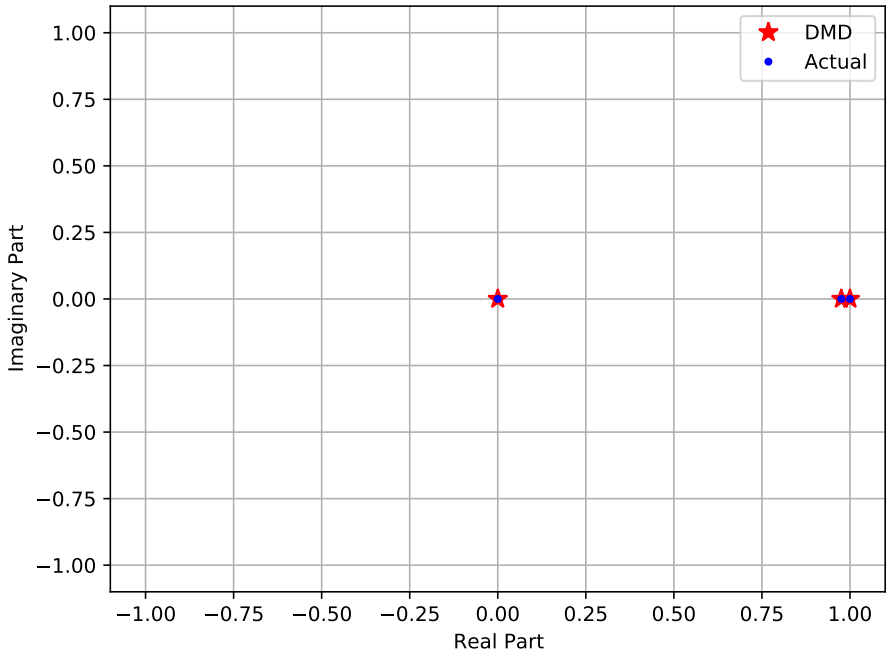

Figure 5: Eigenvalue comparison for example 1.

The absolute reconstruction error for this example is 7.0941e-15. This example highlights the ability of standard DMD to extract dynamics that operate on time scales orders of magnitude apart. The only caveat is that if time steps are large relative to the time scale of a fast mode, the reconstruction accuracy of the exact dynamics of the fast mode diminishes while the data reconstruction accuracy does not.

Many Time Scales

In this section, an example with many time scales is explored. Signals of the same form are used in this example on the same domain as before. 40 snapshots of length 400 are collected. 100 signals are considered with time scales bounded between the slowest and fastest signals of the previous example. The eigenvalues are log-spaced between -1e3 and -1e-3. The spatial frequencies are given by the signal number and are shifted by a random perturbation proportional to the spatial domain width. As before, all signals are given unit amplitudes and the total signal is normalized by its Frobenius-norm. The signal is shown in Figure 6.

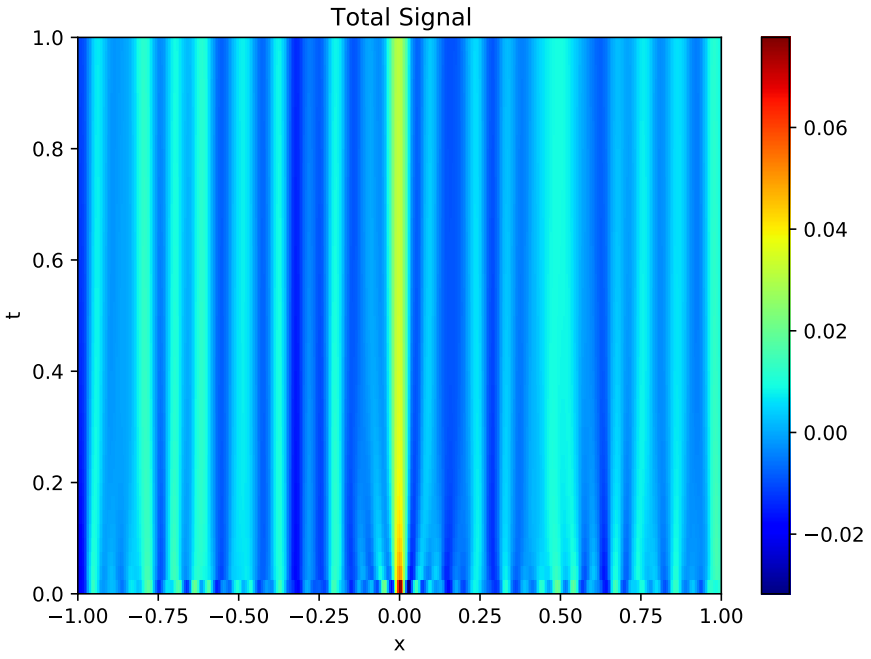

Figure 6: Total signal for example 2.

The singular value spectrum for this example is seen in Figure 7

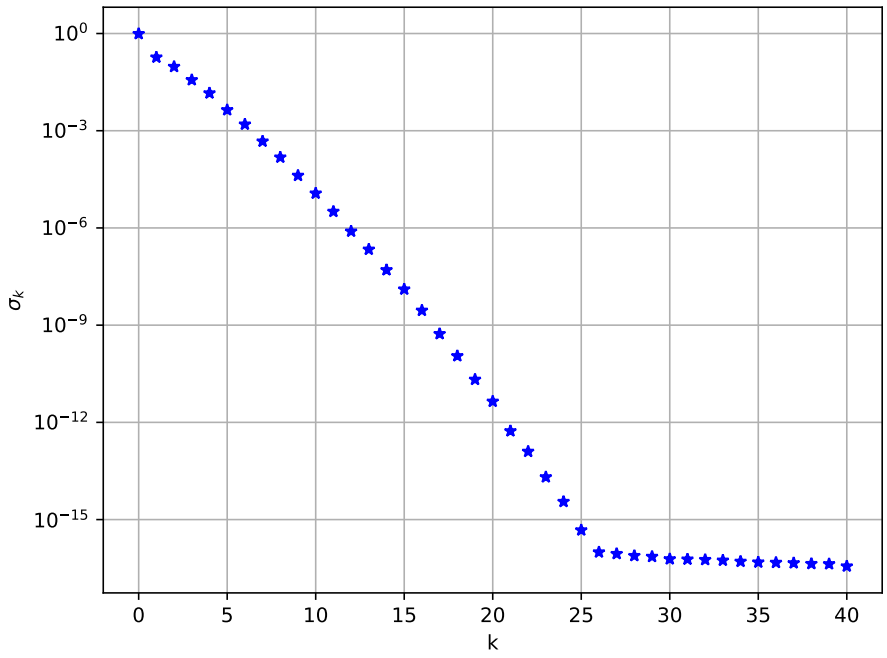

Figure 7: Singular value spectrum for example 2.

From this, it is observed that the signal can be represented to double precision about 25 modes. The dynamic mode with the largest amplitude is shown in Figure 8.

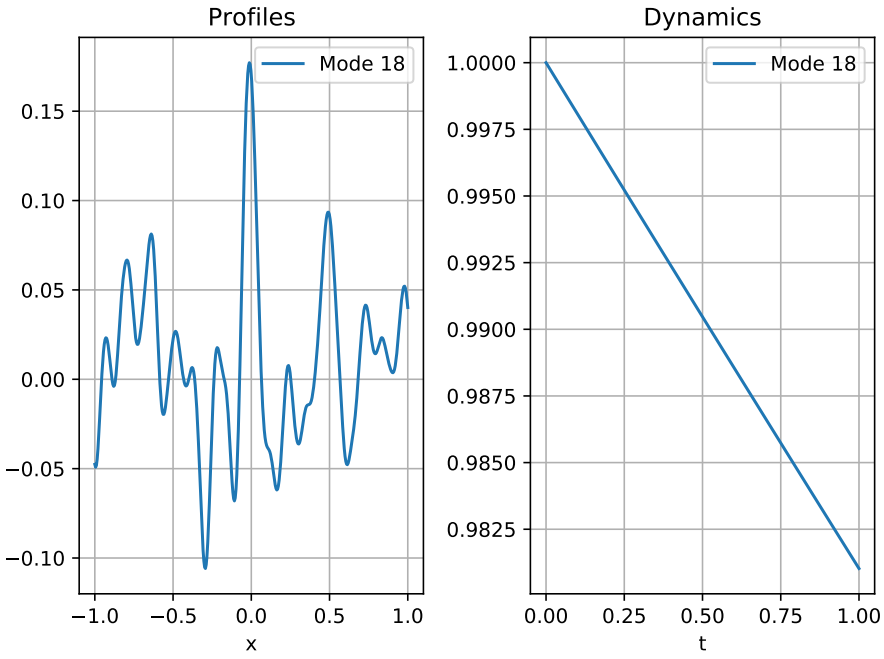

Figure 8: Dynamic mode with the largest magnitude for example 2.

The mode appears to represent several of the signals in the original snapshots squashed together. The DMD and signal eigenvalues are presented in Figure 9.

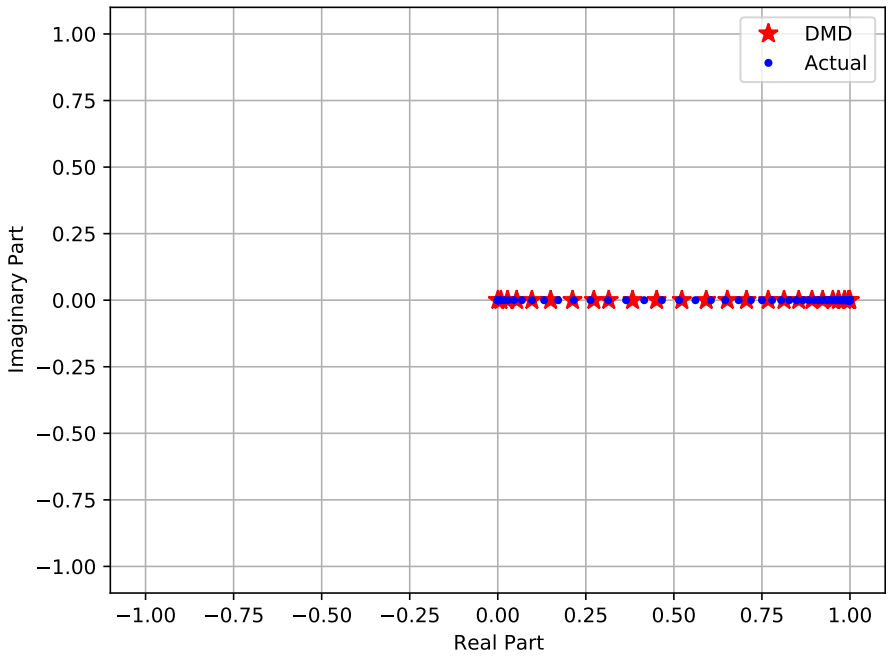

Figure 9: DMD and signal eigenvalues for example 2.

The eigenvalues of the dynamic modes lie between those of the actual signals, further demon-

strating DMD compress a data set comprised of many signals.

In this example, it is shown how DMD can compress many signals of different time scales into a reduced set of signals without loss of accuracy in reconstruction of the total signal. If one seeks to extract specific dynamics information, a smaller number of modes could be used at the expense of reconstruction accuracy.

A Few Dominant Modes Among Many

In some instances, there may be many different signals, but only a few may be relevant. This is the precise type of problem investigated in [6]. This section uses the same setup as the previous example, but the time scales are shuffled and the first three signals are promoted in amplitude three to four orders of magnitude far above the remainder.

To avoid redundancy, plots of the total signal and the singular value spectrum are omitted. The important result from this exercise is comparing the top three dynamic modes by amplitude to those of the top three signals by amplitude. This is shown in Figure 10.

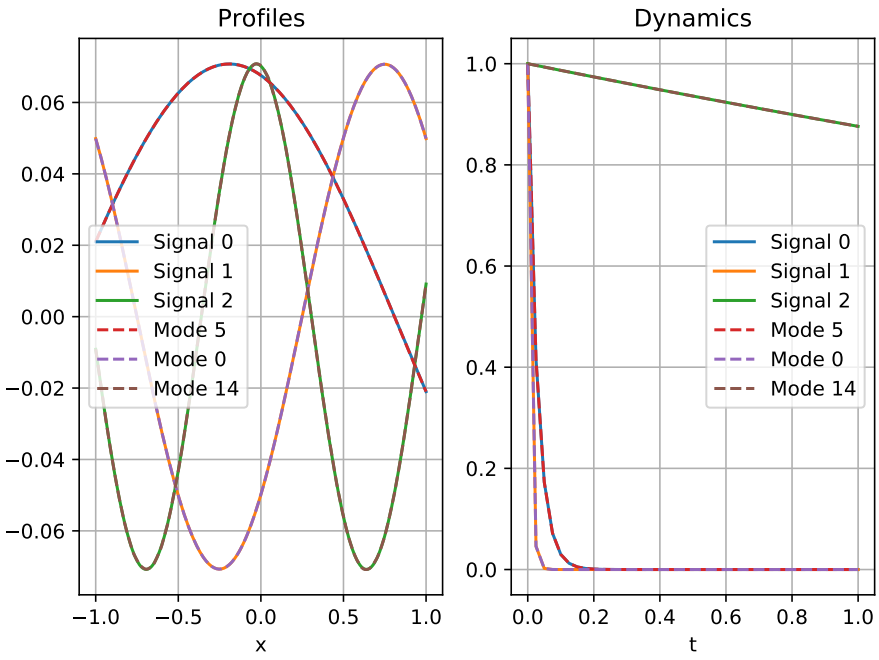

Figure 10: Dynamic mode comparison to the underlying signals for example 3.

This plot was produced using 25 dynamic modes as in the previous example. The important takeaway from this is that if some underlying behaviors are dominant compared to others, those behaviors are seen in the underlying structures of DMD. The dominant behaviors are returned as they exist in the data set, and the other behaviors are squashed into a more compact form as observed in the previous example.

Remarks

In pure linear decay/growth problems, standard DMD is more than sufficient in a variety of different cases. First, it was shown that standard DMD can extract time scales that vary many orders of magnitude. DMD can also compress many time scales into a smaller number of signals without loss

of accuracy. Lastly, DMD can extract the most relevant dynamics from a series of snapshots while compressing the rest into a smaller number of modes. These results are easily explained because the examples considered fit the fundamental assumption of DMD, a linear mapping from one snapshot to the next. While cyclic behaviors are not considered in these examples, their inclusion would not change these results.

Localized Signals

In this section, problems with either localized profiles or dynamics are explored. As outlined by Kutz in [7], DMD breaks down when there are translational or rotational invariances and when there are transient time dynamics. In this section, problems that fit this description are explored. The first example of this section will explore a problem with localized profiles that translate across the domain. The next will explore a problem with localized dynamics, where each signal's contribution to the data-set is only relevant for a portion of the sampling period. This section aims to show that standard DMD does not hold in these instances and that mrDMD may be a useful alternative.

Localized, Translating Profiles

The first example is modeled after an example given in Kutz [7]. In this example, two Gaussian signal translate across a one-dimensional domain at different speeds. SVD-based methods extract underlying behaviors that act over the full span of snapshots. In the case of a traveling waves, there are no underlying structures over the full domain. One would not expect that two modes could be used to represent the two translating signals. Rather, many different behaviors must be aggregated to produce the behavior, assuming it can be represented at all. In some cases, this aggregation can produce quite accurate reconstructions, but in others, it may do a poor job. The remainder of this section will present the form of the signal, the application of standard DMD, and the application or mrDMD.

The signal used in this example is of the form

$$f(x,t) = \exp\left[-\left(\frac{(x-vt)-x_0}{\sigma}\right)^2\right]\exp\left[(a+bi)\,t\right], \tag{9}$$

where v is the velocity of the signal, x_0 is the original offset of the signal, σ is the standard deviation of the Gaussian, a is the decay/growth rate, and b represents the temporal frequency of the signal. With this, the signal is a translating Gaussian that is undergoing damped oscillations. The standard values used for these parameters are $v = 0.3,\ 1.1$, $x_0 = 0.25,\ 0.1$, $\sigma = 0.2,\ 0.05$, $a = -0.1,\ -0.5$, and $b = 0.5,\ 1.5$ The domain considered for this example is a unit box with $x,\ t \in [0,\ 1]$. 400 spatial points are used and 200 snapshots are taken. The signal is shown in Figure 11.

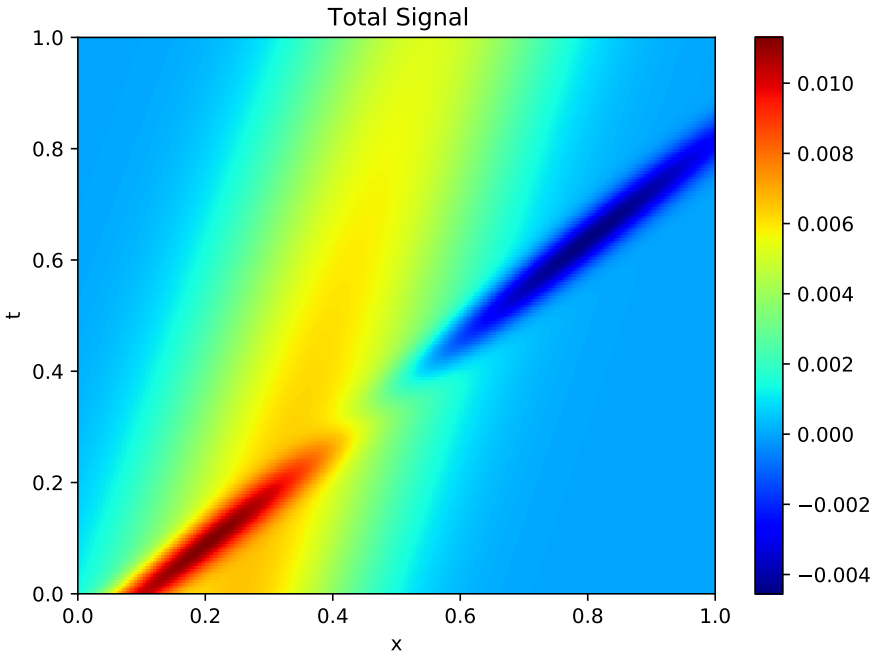

Figure 11: Signal for example 4.

Figure 12 shows the singular value spectrum for this data set.

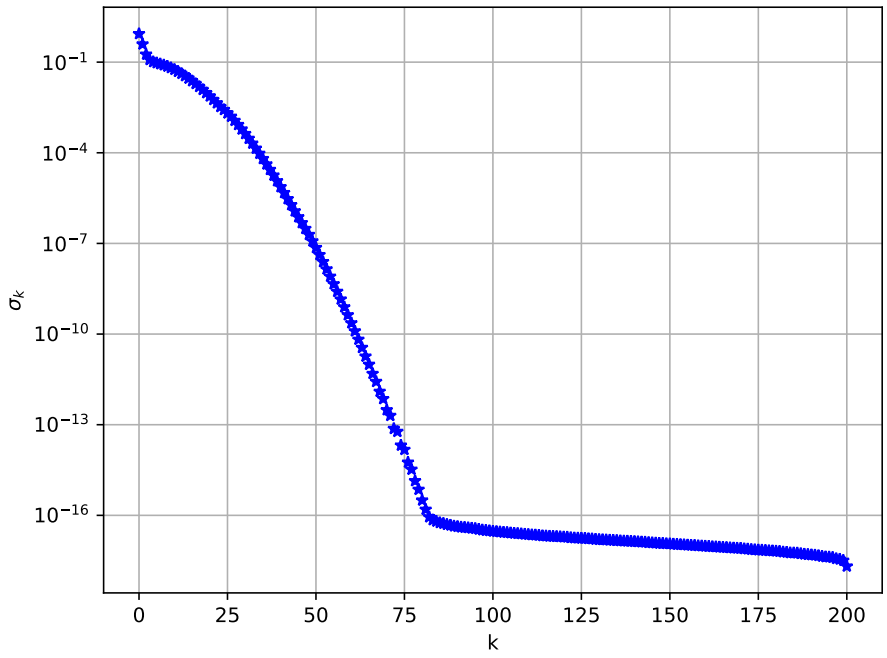

Figure 12: Singular value spectrum for example 4.

To represent this signal, theory states one must use over 75 modes to represent the signal. This is an example of where theory breaks down. When 75 modes are used, while that many POD

modes may span the data set, the DMD representation is unstable and the reconstruction blows up. Instead, a stable representation with four modes are used because it was found to give the most accurate and concise representation. The modes, sorted by amplitude, are shown in Figure 13.

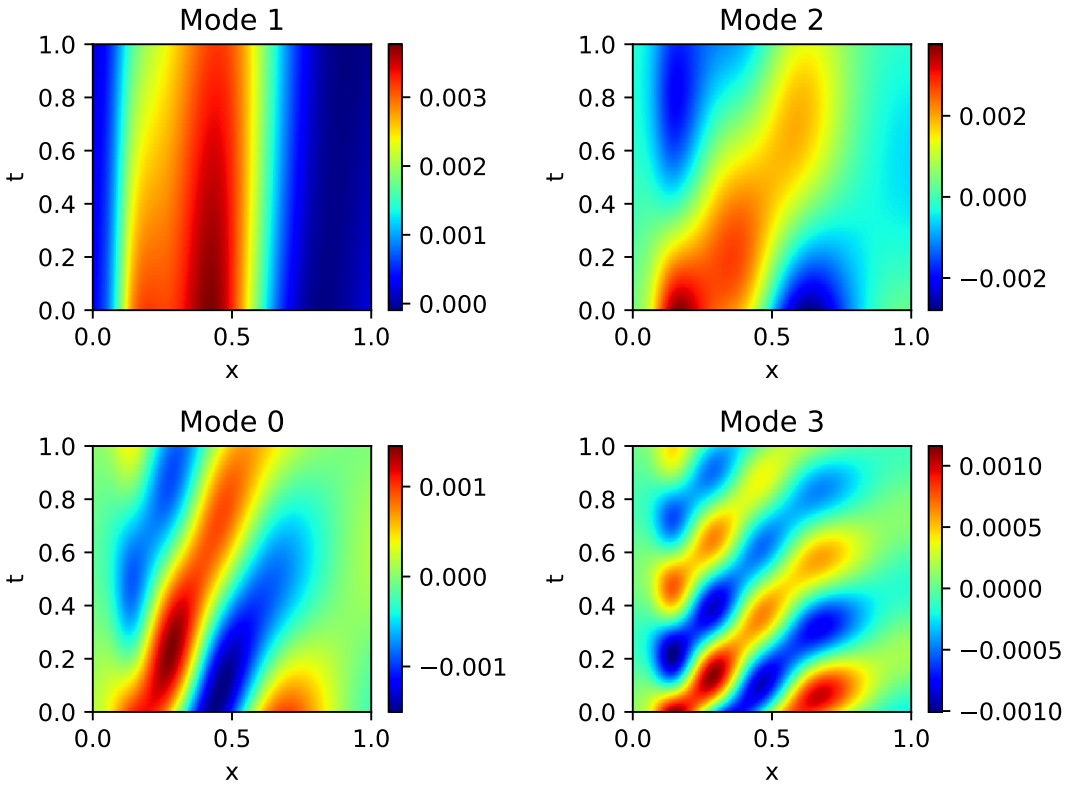

Figure 13: Dynamic modes for example 4.

This representation gives a reconstruction that is only accurate to one digit. The modes in the upper half of Figure 13 somewhat resemble the underlying structures, while the others do not. This result is most likely due to the small number of modes used in the reconstruction. One would expect to see more familiar structures with fewer modes at the expense of reconstruction accuracy.

A few hyper-parameters must be discussed before proceeding with mrDMD results. Within the PyDMD framework, when mrDMD is applied, levels are sub-sampled to reduce computational cost, and to enhance the detection of slow modes. This sub-sampling is based on the maximum number of cycles in a window. A cycle is defined as a period of temporal oscillations. The default of 1 is used in this example. As the maximum number of cycles is increased, the decomposition more closely resembles standard DMD because more modes are included in first level. Each level represents a halving of the domain. The number of levels simply means the recursion depth for the decomposition. With 200 snapshots, the maximum number of levels that can be achieved with the sub-sampling is 5. This is used in this example.

mrDMD is performed using pyDMD's optimal rank SVD capabilities [2]. Ultimately, this improves the reconstruction accuracy of standard DMD by two orders of magnitude to three digits. Further optimization could be performed to increase this accuracy by adding levels (and therefore using more snapshots) or tuning other hyper-parameters. Rather than showing the many modes at various levels of the decomposition, it is more instructive to show the conglomeration of modes

at each level. The zeroth level, or the slow modes over the full domain are shown in Figure 14.

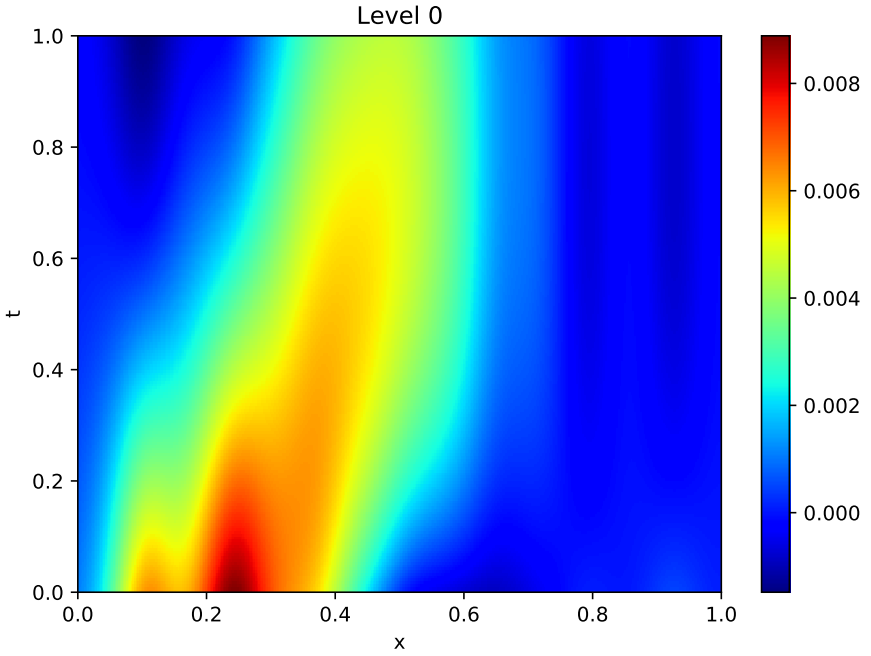

Figure 14: Level 0 reconstruction of the example 4 data set.

The underlying structure here is quite apparent. Around $t = 0$, the Gaussian profiles at their corresponding centroids are observed. Further, relative to one another, they appear to decay at the appropriate rates relative to one another. This feature is among the primary advantages of mrDMD. Figure 15 presents the build-up of the reconstruction level by level.

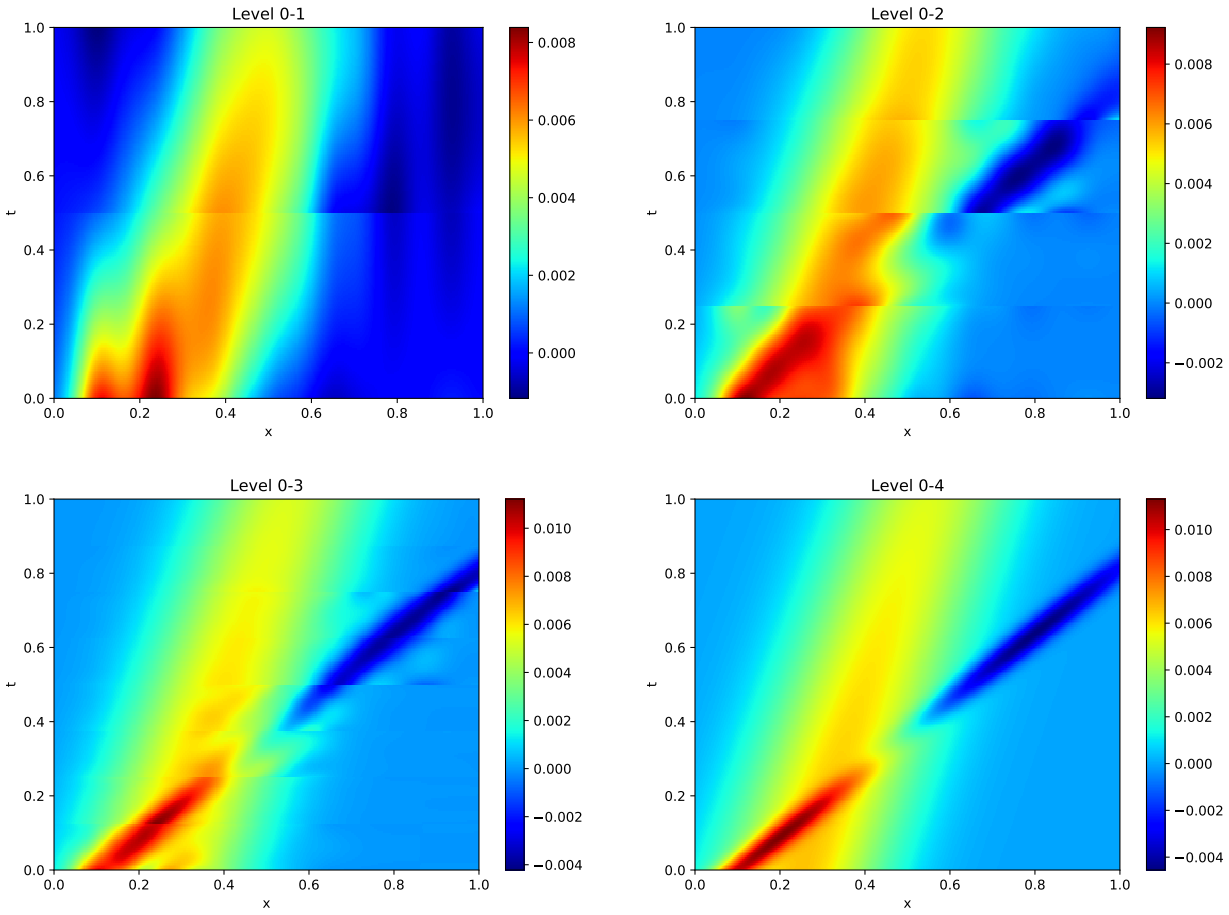

Figure 15: Level by level reconstruction of the example 4 data set.

From this, it is observed that the slow moving Gaussian signal is built up in levels 0-2 while the fast moving signal is extracted in primarily in levels 3-4.

In this example, the fundamental assumptions of DMD are violated because each signal propagates across the domain. It is important to note that translating signals can be reconstructed with relatively high accuracy. This is demonstrated in a Kutz Chapter 1 example highlighting the deficiencies of DMD using one translating and one stationary signal [7]. In that example, good reconstruction accuracy can be achieved, however, the structure of the modes do not represent the underlying structures. In fact, if one uses only a single Gaussian signal in this example, standard DMD can give a reconstruction accuracy up to 8 digits. Overall, using mrDMD is similar to using a sliding window across the domain in that one does not decompose signals until they are approximately stationary within the window.

Localized Dynamics

This subsection discusses the application of DMD to localized dynamics. In particular, pulse-like (Gaussian) dynamics will be explored. Similar to the previous example, two signals are used with different parameters governing their profiles and dynamics. The form of the signal is

$$f(x,t) = \cos\left(\frac{n\pi x}{L}\right)\exp\left[-\left(\frac{t-t_0}{\sigma}\right)^2\right], \tag{10}$$

where n is the frequency of the spatial profile, L is the spatial domain width, t_0 is the centroid of the Gaussian dynamics, and σ is the standard deviation of the Gaussian dynamics. In this example, $n = 2.0,\ 4.0$, $t_0 = 0.3,\ 0.6$, and $\sigma = 0.3,\ 0.1$. The total signal is seen in Figure 16

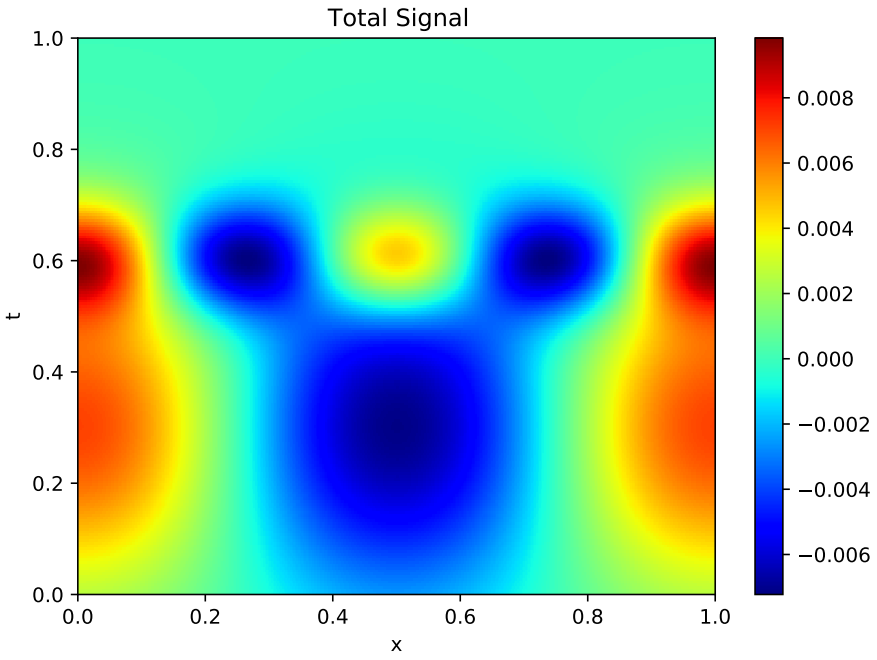

Figure 16: Total signal for example 5.

The singular value spectrum decays to round-off by the third entry, however, as before, this does not yield an accurate reconstruction as that the dynamics cannot be characterized exactly by the eigendecomposition representation given by DMD. The reconstruction is shown in Figure 17.

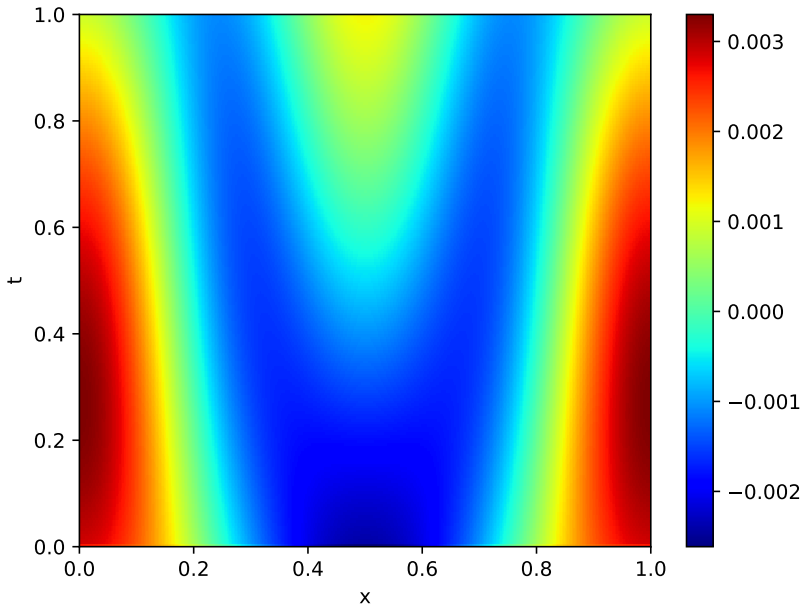

Figure 17: DMD reconstruction of the example 5 data set.

The reconstruction bears slight resemblance to the original signal, but smeared across the domain. With mrDMD, the reconstruction accuracy is improved by two digits of accuracy. The mrDMD reconstruction is seen in Figure 18.

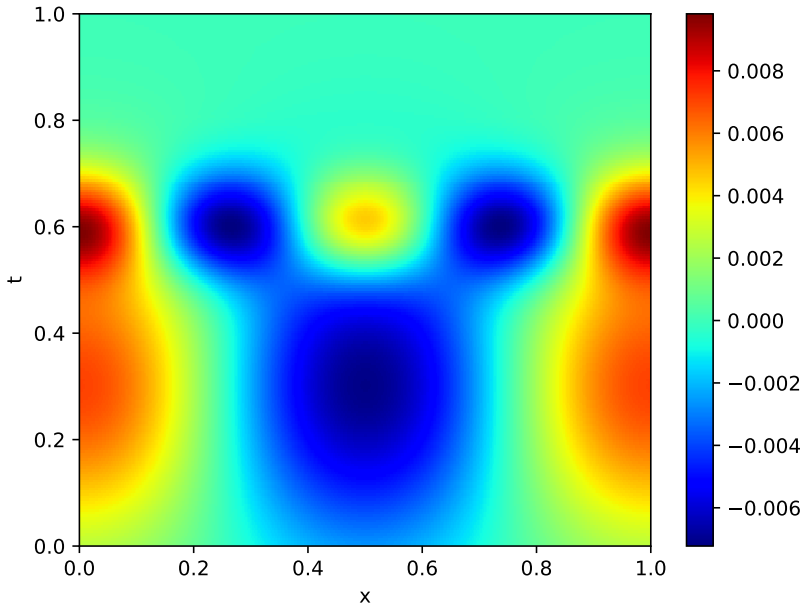

Figure 18: mrDMD reconstruction of the example 5 data set.

mrDMD resolves the finer features within the original signals and provides a more crisp reconstruction.

In this example, it was shown that mrDMD can help provide additional resolution in the reconstruction of a signal produced by localized dynamics, where standard DMD generally degrades. Using mrDMD, one recursively performs standard DMD on simplified signals on shorter time scales. This, in essence, keeps only the most important features at a given time scale, disregarding those likely do not represent an underlying structure. An alternative approach would be to use a true sliding window approach. With this, rather than recursively removing underlying structures and refining the time scale, one could choose a partitioning to construct a piece-wise DMD representation of the signal.

Conclusions

In this report, DMD was explored on five different examples. The first section of examples explored linear problems with time invariant dynamics and highlighted three key features of DMD. These features were the ability of DMD to extract time scales varying many orders of magnitude, the compression achieved when data comprised of many signals is decomposed, and the ability to isolate important modes among many, while compressing the rest. In the last section of examples, the application of mrDMD on problems with localized signals was explored. The first example focused on two translating spatially Gaussian signals. In this example, it was shown that standard DMD fails to extract the underlying signals, however, mrDMD extracts each signal on the time scale most representative of its the signal speed. In the second example, two signals with Gaussian dynamics are explored. Standard DMD fails to extract representative profiles or dynamics while mrDMD constructs a much more representative and crisp reconstruction.

If one seeks to use DMD, any knowledge of the underlying structures should be used to decide which method is most appropriate. For linear dynamics, standard DMD is generally sufficient unless there is a compelling reason otherwise. With nonlinear or localized dynamics, the choice is less obvious. If reconstruction accuracy is the only concern, then in some cases, standard DMD can provide quite accurate reconstruction. However, if one seeks to gain insight into underlying structures, then generally speaking, mrDMD is likely the best option. In any case, the question of accuracy versus insight must be considered. The more modes one uses, generally speaking, the better the reconstruction accuracy, however, the more underlying structures may be blurred. If one seeks to extract the dominant structures in a data set, then a small number of modes should be used in the decomposition at the expense of reconstruction accuracy. As a final note, DMD is developing technique and by no means are these methods the only variants available. For a more thorough review of existing methods, the reader is encouraged to visit the Kutz textbook [7].

References

[1] M. Abdo, R. Elzohery, and J. Roberts. Data-driven surrogate model to predict isotropic composition using dynamic mode deomposition. In *Proceedings of the PHYSOR 2018, Cancun, Mexico, July 2-5, 2018*, Cancun, Mexico, April 2018. American Nuclear Society.

[2] N. Demo, M. Tezzele, and G. Rozza. PyDMD: Python Dynamic Mode Decomposition. *The Journal of Open Source Software*, 3(22):530, 2018.

[3] A. Di Ronco, C. Introini, E. Cervi, S. Lorenzi, Y. S. Jeong, S. B. Seo, I. C. Bang, F. Giacobbo, and A. Cammi. Dynamic mode decomposition for the stability analysis of the molten salt fast reactor core. *Nuclear Engineering and Design*, 362:110529, 2020.

[4] D. Dylewsky, M. Tao, and J. N. Kutz. Dynamic mode decomposition for multiscale nonlinear physics. *Physical Review E*, 99(6):063311, 2019.

[5] M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[6] Z. K. Hardy, J. E. Morel, and C. Ahrens. Dynamic mode decomposition for subcritical metal systems. *Nuclear Science and Engineering*, 193(11):1173–1185, 2019.

[7] J. N. Kutz, S. L. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data driven modeling for complex systems*. Society for Industrial and Applied Mathematics Philadelphia, 2016.

[8] R. G. McClarren. Calculating time eigenvalues of the neutron transport equation with dynamic mode decomposition. *Nuclear Science and Engineering*, pages 1–14, 2019.

[9] R. G. McClarren and T. S. Haut. Acceleration of source iteration using the dynamic mode decomposition. *arXiv preprint arXiv:1812.05241*, 2018.

[10] J. A. Roberts, L. Xu, R. Elzohery, and M. Abdo. Acceleration of the power method with dynamic mode decomposition. *Nuclear Science and Engineering*, 193(12):1371–1378, 2019.

[11] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[12] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.