Systems Perspectives in Fluid Dynamics

Maziar S. Hemati

Aerospace Engineering & Mechanics
University of Minnesota

UNH/IAM Workshop on
Advancing Wall-Turbulence Model Development and Implementation

November 20, 2015

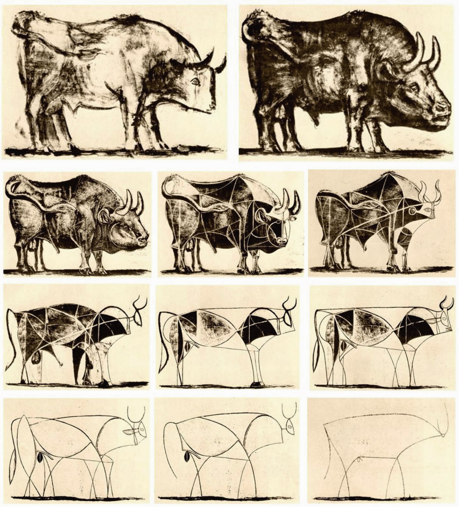
1. **Data-Informed Sparse (Dynamical) Representations**

2. Dynamic Mode Decomposition & the Koopman Operator

3. Practical Advances in Dynamic Mode Decomposition & Data-Driven Koopman Spectral Analysis

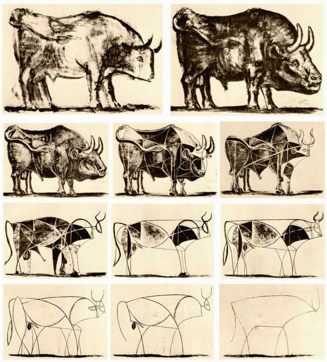
Sparse Representations

Pablo Picasso
"Bull"
(1945–1946)



Sparse Representation: Description based on a “minimal” set of “essential” features.

- Essential features can inform understanding.
- Significance of features to a description depends on context.
- Everything should be made as simple as possible, but not simpler.



Coherent Structures:

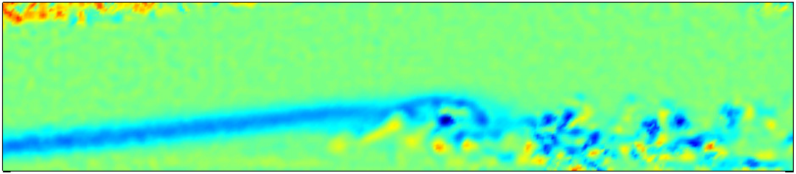
Fluid flows have large (infinite) number of degrees of freedom, but most are “inactive”.

Only a few interacting “active modes” dominate the complex evolution of the fluid flow.

Ruelle and Takens (Commun. Math. Phys., 1971); Hassan (Phys. Fluids, 1983);
Sirovich (Quarterly Appl. Math., 1987); Holmes, Lumley, Berkooz (1998).

Three broad classes of problems:

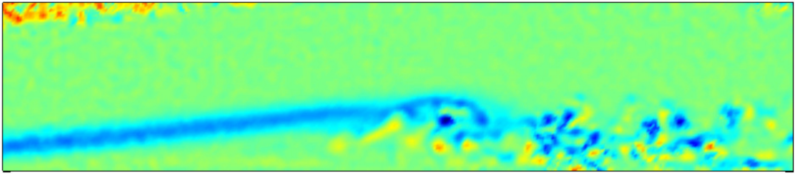
1. The study of dynamical systems for which the evolution law is given.
2. The extraction of qualitative and quantitative information from “data” collected in experiments/simulations.
3. A combination of (1) and (2).



Obtaining the data is one hurdle (i.e., experiments and numerics); interpreting and making sense of the data is another.

Three broad classes of problems:

1. The study of dynamical systems for which the evolution law is given.
2. The extraction of qualitative and quantitative information from “data” collected in experiments/simulations.
3. A combination of (1) and (2).



Obtaining the data is one hurdle (i.e., experiments and numerics); interpreting and making sense of the data is another.

Data-Informed Sparse (Dynamical) Representations

Understanding large amounts of multi-dimensional data requires synthesis into interpretable information.

Sparse representations address this issue through a reformulation of the data into fewer variables, while preserving the “essential features” of the original dataset.

Two key points:

1. Not all variables are relevant, so **identify the relevant variables**; ignore the rest.
   - e.g., (I/O systems) relevance of inputs can be quantified by influence on outputs (e.g., correlations between inputs/outputs); eliminate inputs with negligible influence.
2. Dimensionality of data may be larger than necessary (even if all data relevant), so **determine dependencies** and transform into a lower-order representation.
   - e.g., two highly correlated variables provide similar information; thus, knowing about one provides information about the other: instead of removing, find a transformed set.

“New dataset” should contain fewer variables, but should also preserve “interesting features” of the original dataset.

Example: Proper Orthogonal Decomposition (POD)

Consider a matrix of snapshot data $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{n \times m}$.

Use data covariance $C_X = \frac{1}{m-1} XX^*$ to identify

- relevant variables
  - by assumption, large variances (diagonal terms in C_X) correspond to *dynamics of interest*, whereas low variances are associated with *uninteresting dynamics*.
- redundant data
  - redundancy of variables quantified by covariances (off-diagonal terms in C_X): high covariance indicates high redundancy, whereas low covariance indicates statistical independence.

Diagonalizing C_X provides an ideal view of the data, since

- all redundancies will be removed, and
- directions with largest variance will be isolated and ordered.

Example: Proper Orthogonal Decomposition (POD) continued...

Method 1: Diagonalize covariance via eigendecomposition

Rewrite $XX^* = S\Lambda S^*$ with a unitary matrix S of eigenvectors arranged in columns, and a diagonal matrix Λ of eigenvalues.

Columns of S form a POD basis, which yields a transformed dataset $Y = S^* X$ with diagonal covariance $C_Y = \frac{1}{m-1} YY^* = \frac{1}{m-1}\Lambda$

Example: Proper Orthogonal Decomposition (POD) continued...

Method 1: Diagonalize covariance via eigendecomposition

Rewrite $XX^* = S\Lambda S^*$ with a unitary matrix S of eigenvectors arranged in columns, and a diagonal matrix Λ of eigenvalues.

Columns of S form a POD basis, which yields a transformed dataset $Y = S^* X$ with diagonal covariance $C_Y = \frac{1}{m-1} YY^* = \frac{1}{m-1}\Lambda$

Method 2: Diagonalize covariance via singular value decomposition (SVD)

Singular Value Decomposition (SVD)

SVD is a factorization of an $n \times m$ matrix A (assume $m \geq n$):

$$A = U \Sigma V^*$$

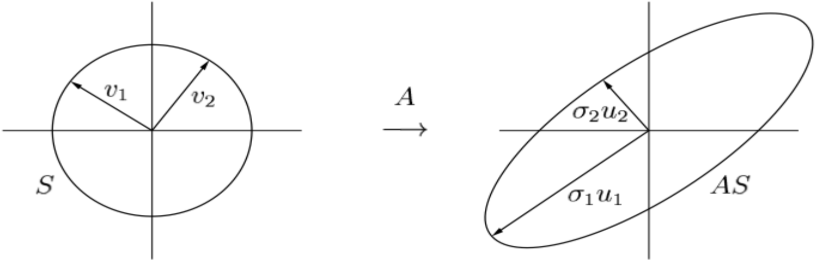
U	$n \times n$ unitary matrix
Σ	$n \times m$ rectangular diagonal matrix (sorted entries, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$)
V^*	$m \times m$ unitary matrix

Useful Properties:

- Guaranteed existence for *any* A (not true of eig. decomp., even for square A).
- u_i and v_i form orthonormal bases for the four fundamental subspaces of A.
- Pseudo-inverse of A can be expressed as $A^+ = V \Sigma^+ U^*$.
- If $\mathrm{rank}(A) = r$, then Σ will have exactly r non-zero diagonal entries $(\sigma_1, \ldots, \sigma_r)$.

An intuitive view of the SVD

Since V is unitary, $AV = U\Sigma$ or $Av_i = \sigma_i u_i$.



Courtesy of Trefethen and Bau (1997)

Consider the action of $A \in \mathbb{R}^{n \times m}$ on the unit sphere S:

- Transforms the unit sphere $S \in \mathbb{R}^m$ into a hyper-ellipse $AS \in \mathbb{R}^n$.
- S is stretched by σ_i along the orthogonal directions u_i (i.e., in the direction of the principle semi-axes of the hyper-ellipse AS.

Example: Proper Orthogonal Decomposition (POD) continued...

Method 1: Diagonalize covariance via eigendecomposition

Rewrite $XX^* = S\Lambda S^*$ with a unitary matrix S of eigenvectors arranged in columns, and a diagonal matrix Λ of eigenvalues.

Columns of S form a POD basis, which yields a transformed dataset $Y = S^* X$ with diagonal covariance $C_Y = \frac{1}{m-1} YY^* = \frac{1}{m-1}\Lambda$

Method 2: Diagonalize covariance via singular value decomposition (SVD)

Rewrite $X = U\Sigma V^*$, then columns of U form a POD basis.

Transformed dataset $Y = U^* X$ will have diagonal covariance $C_Y = \frac{1}{m-1} YY^* = \frac{1}{m-1}\Sigma^2$.

Note: there is an explicit connection between SVD/eigendecomposition methods.

Another view of relevance and redundancy in data: Low-rank approximation

Given: a matrix of snapshot data $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{n \times m}$.

Find: a low-rank matrix $\hat{X} \in \mathbb{R}^{n \times m}$ that approximates X “optimally” in some sense.

Another view of relevance and redundancy in data: Low-rank approximation

Given: a matrix of snapshot data $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{n \times m}$.

Find: a low-rank matrix $\hat{X} \in \mathbb{R}^{n \times m}$ that approximates X "optimally" in some sense.

Example: Optimal in Frobenius Norm (Eckart-Young-Mirsky Theorem)

$$\underset{\hat{X}}{\text{minimize}} \; \|X - \hat{X}\|_F \qquad \text{such that} \qquad r = \operatorname{rank}(\hat{X}) \leq \operatorname{rank}(X)$$

Another view of relevance and redundancy in data: Low-rank approximation

Given: a matrix of snapshot data $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \in \mathbb{R}^{n \times m}$.

Find: a low-rank matrix $\hat{X} \in \mathbb{R}^{n \times m}$ that approximates X "optimally" in some sense.

Example: Optimal in Frobenius Norm (Eckart-Young-Mirsky Theorem)

$$\underset{\hat{X}}{\text{minimize}} \; \|X - \hat{X}\|_F \qquad \text{such that} \qquad r = \mathrm{rank}(\hat{X}) \leq \mathrm{rank}(X)$$

Analytical solution in terms of the (truncated) SVD:

$$\hat{X}_{\text{optimal}} = U_r \Sigma_r V_r^*$$

where Σ_r, U_r, V_r correspond to the r largest singular values of $X = U \Sigma V^*$.

Coming up this session...

Systems perspectives to be discussed this session:

1. Dynamic Modes and the Koopman Operator (Hemati)
2. Input-Output Models (Illingworth)
3. Resolvent Models and Passivity-Based Control (Sharma)

1. Data-Informed Sparse (Dynamical) Representations

2. **Dynamic Mode Decomposition & the Koopman Operator**

3. Practical Advances in Dynamic Mode Decomposition & Data-Driven Koopman Spectral Analysis

The Koopman Operator

Definition (Koopman, PNAS 1931)

For a discrete-time dynamical system

$$x \mapsto F(x)$$

where $x \in \mathcal{M}$, the Koopman operator $\mathcal{K}$ acts on scalar functions $g : \mathcal{M} \to \mathbb{C}$, as

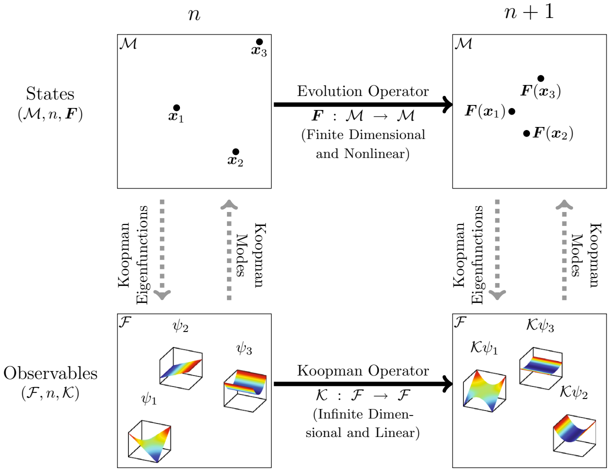
$$\mathcal{K}g(x) := g(F(x)).$$

- $\mathcal{K}$ is linear and acts on functions on $\mathcal{M}$.
- Analyze dynamical system via the spectral properties of $\mathcal{K}$ (Mezić, 2005)

$$F(x) = \sum_{k=1}^{\infty} \lambda_k \varphi_k \xi_k(x)$$

where λ_k, φ_k, and ξ_k are the Koopman eigenvalues, modes, and eigenfunctions, respectively.

The Koopman Operator



Courtesy of Williams, Kevrekidis, Rowley (2015)

Data-Driven Koopman Spectral Analysis

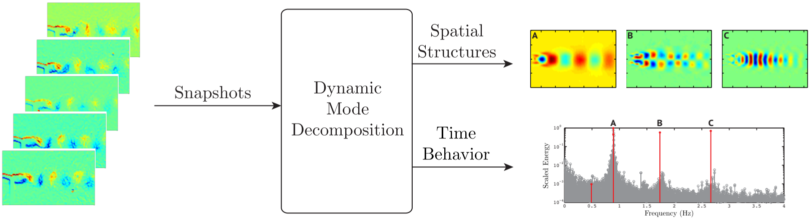
- The Koopman operator is an infinite-dimensional linear operator that captures everything about a *nonlinear* dynamical system.
  - Koopman eigenvalues $\longrightarrow$ "temporal" description
  - Koopman modes $\longrightarrow$ "spatial" description
  - Koopman eigenfunctions $\longrightarrow$ linear dynamics (via nonlinear change of coordinates)

- Dynamic Mode Decomposition (DMD) is an algorithm for computing Koopman eigenvalues, modes, and eigenfunctions from snapshot data. . . (sometimes)

Dynamic Mode Decomposition (DMD)

DMD first introduced in the fluids community (Schmid & Sesterhenn, 2008).

Process snapshot data to extract dynamically relevant spatial structures and associated temporal characteristics (i.e., growth/decay rates and frequencies).



DMD analysis is relevant for nonlinear systems owing to connections with Koopman spectral analysis (Rowley et al., 2009; Tu et al., 2014; Williams et al., 2015).

Dynamic Mode Decomposition (DMD)

Consider a discrete-time system

$$x \mapsto f(x) \in \mathbb{R}^n$$

with snapshot data matrices

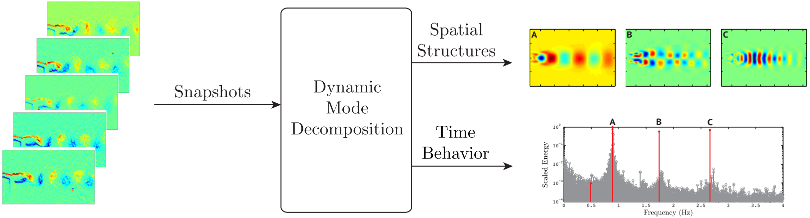
$$\begin{aligned} X &:= \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \\ Y &:= \begin{bmatrix} f(x_1) & f(x_2) & \cdots & f(x_m) \end{bmatrix} \end{aligned}$$

DMD modes and eigenvalues correspond to eigenvectors and eigenvalues of the DMD operator

$$K := YX^+ \in \mathbb{R}^{n \times n}.$$

(Tu et al., 2014)

$$\begin{aligned} \tilde{K} &= U_r^* Y V_r \Sigma_r^{-1} \in \mathbb{R}^{r \times r}, \\ K &= U_r \tilde{K} U_r^* \in \mathbb{R}^{n \times n} \end{aligned}$$



1. Data-Informed Sparse (Dynamical) Representations

2. Dynamic Mode Decomposition & the Koopman Operator

3. **Practical Advances in Dynamic Mode Decomposition & Data-Driven Koopman Spectral Analysis**

DMD and the Nature of "Data"

While DMD has been used in a number of areas, the standard framework can be "upgraded" to accommodate practical challenges that arise due to the "nature of data".

Data scientists characterized data by a set of V's:

- Volume $\longrightarrow$ how large?
- Velocity$\longrightarrow$how fast?
- Veracity$\longrightarrow$ how trustworthy?
- Variety $\longrightarrow$ how many types?

Ultimately, we only care about one V:

- Value$\longrightarrow$what is learned?

What to do when faced with lots of snapshots?

A Streaming Dynamic Mode Decomposition

What is DMD really doing?

1. Compute an orthonormal basis for the image of X.
2. Construct a “small” proxy system to solve the eigenproblem.
3. Relate the eigenvectors and eigenvalues of the small problem to those of the full problem (i.e., $K = Q_X \tilde{K} Q_X^*$).

Standard DMD

$$\underbrace{\tilde{K}}_{r_X \times r_X} := \underbrace{Q_X^*}_{r_X \times n} \underbrace{Y}_{n \times m} \underbrace{X^+}_{m \times n} \underbrace{Q_X}_{n \times r_X},$$

Dynamic Mode Decomposition (DMD)

Accommodating Large and Streaming Datasets

What is DMD really doing?

1. Compute an orthonormal basis for the image of X.
2. Construct a “small” proxy system to solve the eigenproblem.
3. Relate the eigenvectors and eigenvalues of the small problem to those of the full problem (i.e., $K = Q_X \tilde{K} Q_X^*$).

Standard DMD

$$\underbrace{\tilde{K}}_{r_X \times r_X} := \underbrace{Q_X^*}_{r_X \times n} \underbrace{Y}_{n \times m} \underbrace{X^+}_{m \times n} \underbrace{Q_X}_{n \times r_X},$$

To design a streaming DMD method, assume:

1. Only one snapshot pair (x_i, y_i) can be stored at a given time (i.e., “single-pass”).
2. The data in X and Y are low-rank.

Standard DMD

$$\underbrace{\tilde{K}}_{r_X \times r_X} := \underbrace{Q_X^*}_{r_X \times n} \underbrace{Y}_{n \times m} \underbrace{X^+}_{m \times n} \underbrace{Q_X}_{n \times r_X},$$

Re-write DMD as

$$\underbrace{\tilde{K}}_{r_X \times r_X} = \underbrace{Q_X^*}_{r_X \times n} \underbrace{Q_Y}_{n \times r_Y} \underbrace{A}_{r_Y \times r_X} \underbrace{G_X^+}_{r_X \times r_X}.$$

- Q_X, Q_Y can be computed via a Gram-Schmidt procedure.
- $A := \tilde{Y}\tilde{X}^*$, $G_X := \tilde{X}\tilde{X}^*$, $\tilde{X} := Q_X^* X$, $\tilde{Y} := Q_Y^* Y$ can be dynamically updated.

Dynamic Mode Decomposition (DMD)

Accommodating Large and Streaming Datasets

Standard DMD

$$\underbrace{\tilde{K}}_{r_X \times r_X} := \underbrace{Q_X^*}_{r_X \times n} \underbrace{Y}_{n \times m} \underbrace{X^+}_{m \times n} \underbrace{Q_X}_{n \times r_X},$$

Re-write DMD as

$$\underbrace{\tilde{K}}_{r_X \times r_X} = \underbrace{Q_X^*}_{r_X \times n} \underbrace{Q_Y}_{n \times r_Y} \underbrace{A}_{r_Y \times r_X} \underbrace{G_X^+}_{r_X \times r_X}.$$

- (Optional) Maintain low-rank via POD compression.
  - Define $G_Y := \tilde{Y}\tilde{Y}^*$ and make use of leading eigenvectors of G_X, G_Y.

Standard DMD

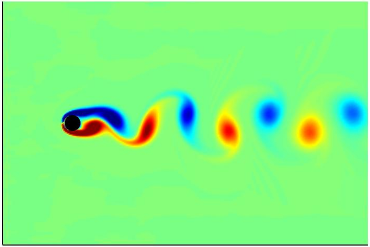
$$\underbrace{\tilde{K}}_{r_X \times r_X} := \underbrace{Q_X^*}_{r_X \times n} \underbrace{Y}_{n \times m} \underbrace{X^+}_{m \times n} \underbrace{Q_X}_{n \times r_X},$$

Re-write DMD as

$$\underbrace{\tilde{K}}_{r_X \times r_X} = \underbrace{Q_X^*}_{r_X \times n} \underbrace{Q_Y}_{n \times r_Y} \underbrace{A}_{r_Y \times r_X} \underbrace{G_X^+}_{r_X \times r_X}.$$

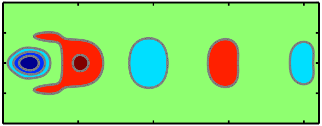
- $\mathcal{O}(nr^2)$ operations per iterate with mode computations.
- $\mathcal{O}(nr)$ operations per iterate without mode computations.
- $\mathcal{O}(nr)$ storage of matrix entries (single-pass method).

Example #1: Numerical simulation data for laminar flow past a cylinder (Re=100)

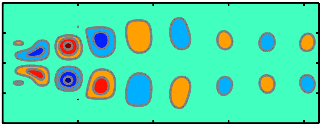


No compression: Data is already low-rank.

DMD Mode 1
($\lambda_1 = 0.998 + 0.0531\imath$)

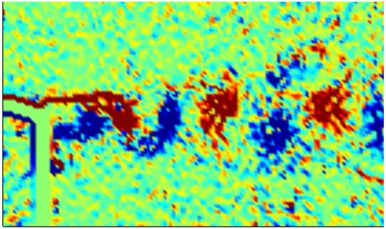


DMD Mode 2
($\lambda_2 = 0.994 + 0.106\imath$)



- Filled contours are DMD modes from batch-processed DMD.
- Gray curves are DMD modes from streaming DMD.

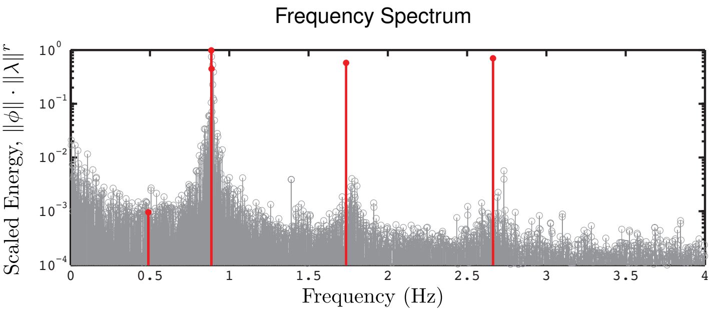
Example #2: PIV experiment data for laminar flow past a cylinder (Re=413)



PIV data courtesy of Jessica Shang (Stanford).

Noise makes the data full-rank, regardless of the nature of the underlying dynamics.

$\rightarrow$ Apply POD Compression ($r = 25$)



Batch-Processed DMD: 3 Cores; Wall-clock $\sim \mathcal{O}$(hours)
Streaming DMD: My laptop; Wall-clock $\sim \mathcal{O}$(minutes)

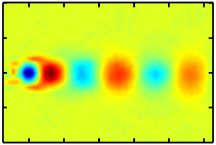
$n = 10800$, $m = 8000$

Dynamic Mode Decomposition (DMD)

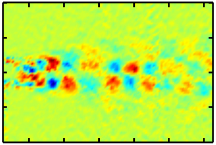
Accommodating Large and Streaming Datasets

Batch-Processed DMD

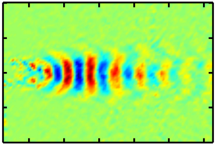
$f = 0.888$ Hz



$f = 1.744$ Hz

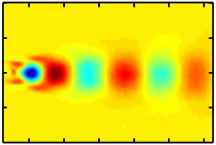


$f = 2.732$ Hz

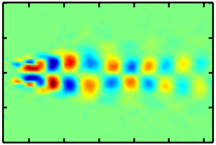


Streaming DMD

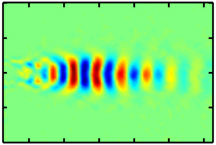
$f = 0.887$ Hz



$f = 1.737$ Hz



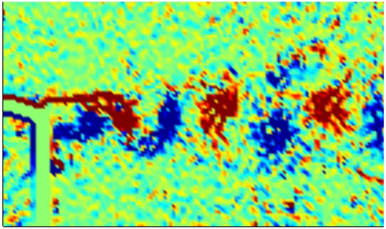
$f = 2.664$ Hz



What to do when faced with uncertain/noisy data?

A Noise-Aware Dynamic Mode Decomposition

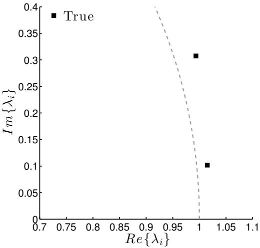
Measurement noise and data uncertainty are common characteristics of many datasets.



Current practice is to apply DMD to noisy datasets directly, but to incorporate some form of truncation, ensemble averaging, and/or cross-validation for “de-noising” the results.

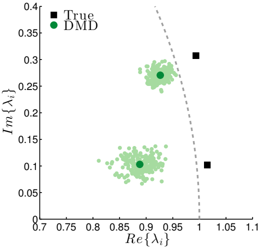
- How does measurement noise influence DMD analyses?
- Are such analyses representative of the “true” system dynamics?

Example: A complex-valued linear system ($n = 250$, $r = 2$)



- Additive measurement noise $(\Delta X, \Delta Y) \sim \mathcal{CN}(0, 0.05)$.
- Computations repeated for 200 independent noise realizations.

Example: A complex-valued linear system ($n = 250$, $r = 2$)

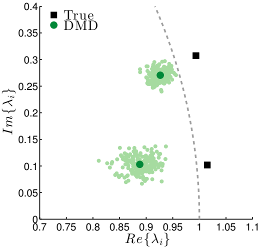


$m = 100$

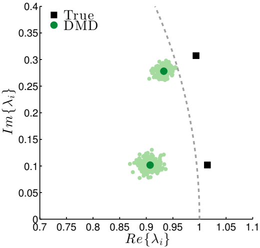
- Additive measurement noise $(\Delta X, \Delta Y) \sim \mathcal{CN}(0, 0.05)$.
- Computations repeated for 200 independent noise realizations.

Here, DMD identifies unstable eigenvalues as stable and decaying!

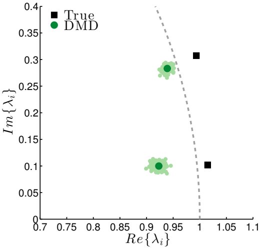
Example: A complex-valued linear system ($n = 250$, $r = 2$)



$m = 100$



$m = 200$



$m = 500$

- Additive measurement noise $(\Delta X, \Delta Y) \sim \mathcal{CN}(0, 0.05)$.
- Computations repeated for 200 independent noise realizations.

Here, DMD identifies unstable eigenvalues as stable and decaying!

Assume additive zero-mean i.i.d. noise with variance σ^2 on all snapshots (X, Y), and recall that $K = YX^+$ (or, $\tilde{K} = Q_X^* YX^+ Q_X$).

For small noise, noise-induced error has an approximate closed-form solution, so correct for this error by considering the eigendecomposition of

$$\tilde{K}_{\text{corrected}} = \tilde{K} \left(I - m\sigma^2 \Sigma^{-2} \right)$$

$$\begin{aligned} m &:= \text{\# snapshots} \\ \Sigma &:= \text{matrix of non-zero singular values of } X \\ \sigma^2 &:= \text{measurement noise variance} \end{aligned}$$

Instead of relying on knowledge of the noise distribution, let's directly consider the *interpretation* of DMD as

$$K = YX^{+}.$$

Instead of relying on knowledge of the noise distribution, let's directly consider the *interpretation* of DMD as

$$K = YX^{+}.$$

In the over-constrained case ($i.e.$, $m > n$), this can be re-written as

$$\min_{K,\Delta Y} \|\Delta Y\|_F, \quad \text{subject to} \quad Y + \Delta Y = KX.$$

Instead of relying on knowledge of the noise distribution, let's directly consider the *interpretation* of DMD as

$$K = YX^{+}.$$

In the over-constrained case ($i.e.$, $m > n$), this can be re-written as

$$\min_{K,\Delta Y} \|\Delta Y\|_F, \quad \text{subject to} \quad Y + \Delta Y = KX.$$

When snapshots are noisy, the residual ΔY can be interpreted as a "noise-correction."

Instead of relying on knowledge of the noise distribution, let's directly consider the *interpretation* of DMD as

$$K = YX^{+}.$$

In the over-constrained case ($i.e.$, $m > n$), this can be re-written as

$$\min_{K,\Delta Y} \|\Delta Y\|_F, \quad \text{subject to} \quad Y + \Delta Y = KX.$$

When snapshots are noisy, the residual ΔY can be interpreted as a "noise-correction."

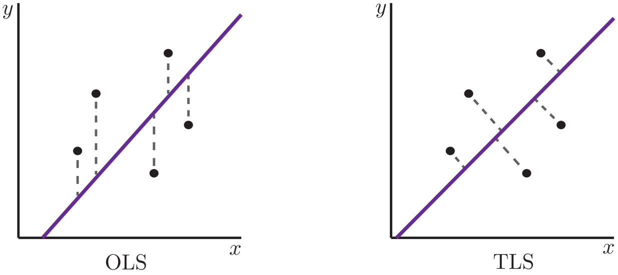
What about ΔX? $\longrightarrow$ Asymmetric treatment of noise!

Instead, consider a problem of total least-squares:

$$\min_{K,\Delta X,\Delta Y} \left\| \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \right\|_F, \quad \text{subject to} \quad Y + \Delta Y = K(X + \Delta X).$$

Instead, consider a problem of total least-squares:

$$\min_{K,\Delta X,\Delta Y} \left\| \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \right\|_F, \quad \text{subject to} \quad Y + \Delta Y = K(X + \Delta X).$$



A two-stage method for noise-aware “total” DMD (TDMD) analysis:

Stage 1: Subspace Projection

Define an augmented snapshot matrix $Z := \begin{bmatrix} X \\ Y \end{bmatrix}$,

then $\bar{Y} = Y\mathbb{P}_{Z_n^*}$, $\bar{X} = X\mathbb{P}_{Z_n^*}$,

where Z_n is the best rank-n approximation of Z.

*When the underlying dynamics are r-dimensional, replace n with r.
Results are “best” when $r \ll m$.

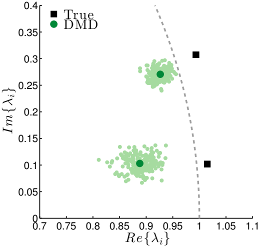
Stage 2: Operator Identification

Perform DMD on the projected snapshots $\bar{X}$, $\bar{Y}$.

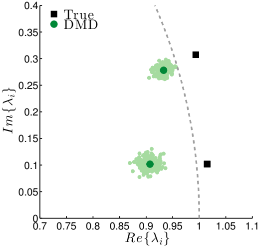
*Any variant of DMD can be used here (e.g., streaming DMD).

- the “de-biasing” occurs in the subspace projection stage.

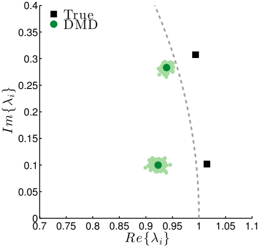
Example #1: A complex-valued linear system ($n = 250$, $r = 2$)



$m = 100$



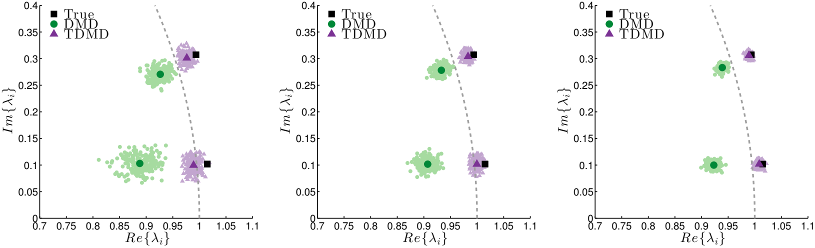
$m = 200$



$m = 500$

- Additive measurement noise $(\Delta X, \Delta Y) \sim \mathcal{CN}(0, 0.05)$.
- Computations repeated for 200 independent noise realizations.

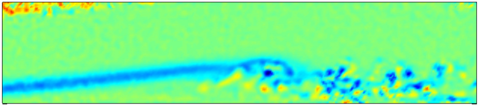
Example #1: A complex-valued linear system ($n = 250$, $r = 2$)



$m = 100$ $m = 200$ $m = 500$

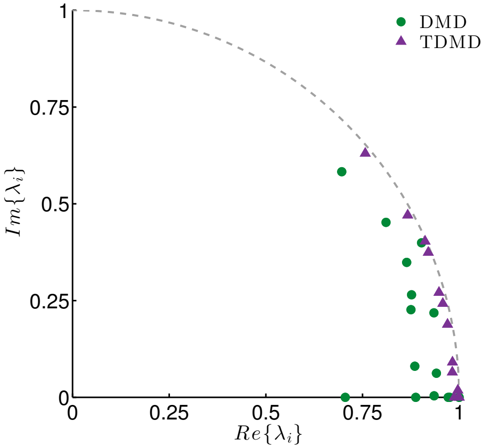
- Additive measurement noise $(\Delta X, \Delta Y) \sim \mathcal{CN}(0, 0.05)$.
- Computations repeated for 200 independent noise realizations.

Example #2: PIV data from a separated flow over a flat plate (Re=10,000)

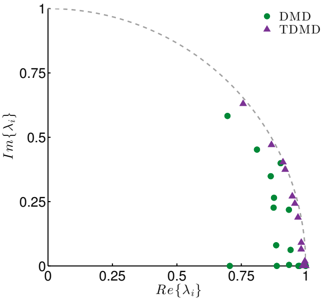


video slowed 40 $\times$

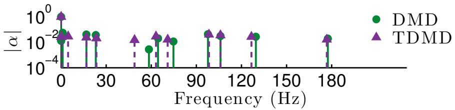
$n = 42\,976$, $m = 3000$



DMD and TDMD spectra for $r = 25$.



DMD
TDMD
1
0.75
0.5
0.25
0
0
0.25
0.5
0.75
1
$Im\{\lambda_i\}$
$Re\{\lambda_i\}$

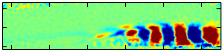


DMD
TDMD
10^{0}
10^{-2}
10^{-4}
$|\alpha|$
0
30
60
90
120
150
180
Frequency (Hz)

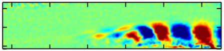
Dynamic Mode Decomposition (DMD)

Accommodating Measurement Noise

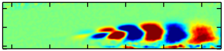
DMD ($r = 25$)



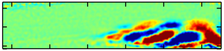
$f = 177$ Hz, $|\lambda| = 0.91$, $|\alpha| = 0.02$



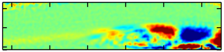
$f = 130$ Hz, $|\lambda| = 0.93$, $|\alpha| = 0.03$



$f = 106$ Hz, $|\lambda| = 0.99$, $|\alpha| = 0.03$

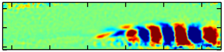


$f = 98$ Hz, $|\lambda| = 0.93$, $|\alpha| = 0.04$

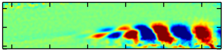


$f = 75$ Hz, $|\lambda| = 0.92$, $|\alpha| = 0.01$

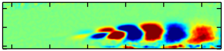
TDMD ($r = 25$)



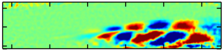
$f = 177$ Hz, $|\lambda| = 0.99$, $|\alpha| = 0.02$



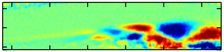
$f = 127$ Hz, $|\lambda| = 0.99$, $|\alpha| = 0.03$



$f = 106$ Hz, $|\lambda| = 1.0$, $|\alpha| = 0.04$



$f = 98$ Hz, $|\lambda| = 0.99$, $|\alpha| = 0.04$

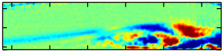


$f = 71$ Hz, $|\lambda| = 0.99$, $|\alpha| = 0.02$

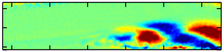
Dynamic Mode Decomposition (DMD)

Accommodating Measurement Noise

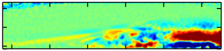
DMD ($r = 25$)



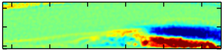
$f = 65\,\mathrm{Hz}, |\lambda| = 0.90, |\alpha| = 0.02$



$f = 58\,\mathrm{Hz}, |\lambda| = 0.96, |\alpha| = 0.003$

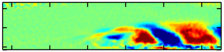


$f = 23\,\mathrm{Hz}, |\lambda| = 0.89, |\alpha| = 0.04$

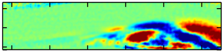


$f = 17\,\mathrm{Hz}, |\lambda| = 0.94, |\alpha| = 0.04$

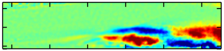
TDMD ($r = 25$)



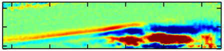
$f = 63\,\mathrm{Hz}, |\lambda| = 0.99, |\alpha| = 0.03$



$f = 49\,\mathrm{Hz}, |\lambda| = 0.99, |\alpha| = 0.02$

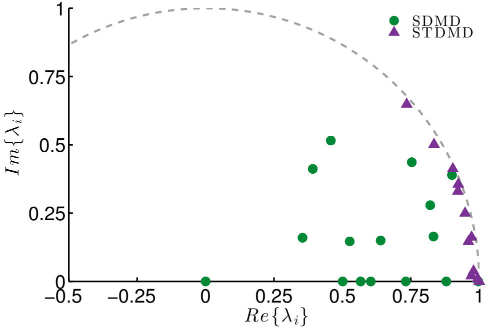


$f = 23\,\mathrm{Hz}, |\lambda| = 0.99, |\alpha| = 0.02$

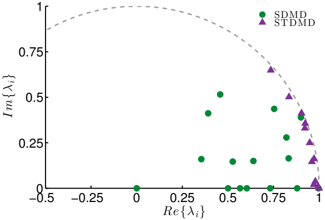


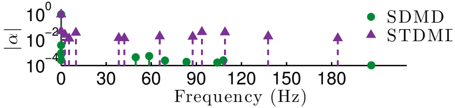
$f = 17\,\mathrm{Hz}, |\lambda| = 0.98, |\alpha| = 0.02$

Example #3: PIV data from a separated flow over a flat plate (Re=10,000), now using the streaming formulations



Example #3: PIV data from a separated flow over a flat plate (Re=10,000), now using the streaming formulations

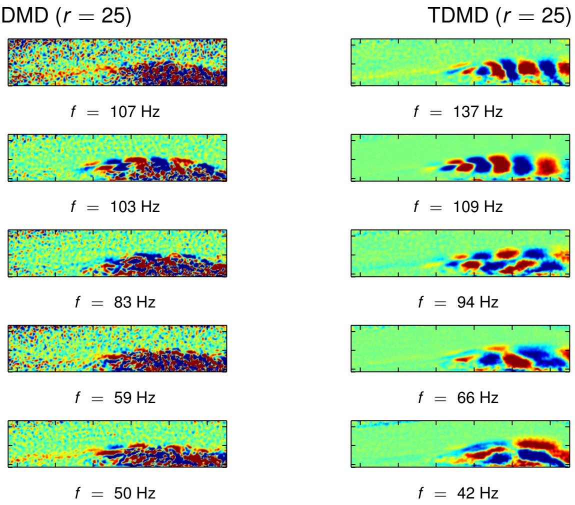




Dynamic Mode Decomposition (DMD)

Accommodating Measurement Noise

Example #3: PIV data from a separated flow over a flat plate (Re=10,000), now using the streaming formulations



What to do when the “right” observables are not available?

An Extended Dynamic Mode Decomposition

Definition (Koopman, PNAS 1931)

For a discrete-time dynamical system

$$x \mapsto F(x)$$

where $x \in \mathcal{M}$, the Koopman operator $\mathcal{K}$ acts on scalar functions $g : \mathcal{M} \to \mathbb{C}$, as

$$\mathcal{K}g(x) := g(F(x)).$$

- $\mathcal{K}$ is linear and acts on functions on $\mathcal{M}$.
- Analyze dynamical system via the spectral properties of $\mathcal{K}$ (Mezić, 2005)

$$F(x) = \sum_{k=1}^{\infty} \lambda_k \varphi_k \xi_k(x)$$

where λ_k, φ_k, and ξ_k are the Koopman eigenvalues, modes, and eigenfunctions, respectively.

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) c x_1^2 \end{bmatrix}.$$

This system has a stable equilibrium at the origin.

The Koopman eigenvalues are $\lambda, \mu, \lambda\mu, \lambda^2, \mu^2, \ldots$

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) c x_1^2 \end{bmatrix}.$$

This system has a stable equilibrium at the origin.

The Koopman eigenvalues are $\lambda, \mu, \lambda\mu, \lambda^2, \mu^2, \ldots$

A few examples:
Set $\lambda = 0.9$, $\mu = 0.5$.

Apply DMD with initial states x given by (1,1), (5,5), (-1,1), (-5,5).

Case 1: $c = 0$ (linear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5]$ correspond to Koopman eigenvalues

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) c x_1^2 \end{bmatrix}.$$

This system has a stable equilibrium at the origin.

The Koopman eigenvalues are $\lambda, \mu, \lambda\mu, \lambda^2, \mu^2, \ldots$

A few examples:

Set $\lambda = 0.9$, $\mu = 0.5$.

Apply DMD with initial states x given by (1,1), (5,5), (-1,1), (-5,5).

Case 1: $c = 0$ (linear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5]$ correspond to Koopman eigenvalues

Case 2: $c = 1$ (nonlinear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 2.002]$ do not correspond to Koopman eigenvalues, and the equilibrium appears unstable!

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) c x_1^2 \end{bmatrix}.$$

This system has a stable equilibrium at the origin.

The Koopman eigenvalues are $\lambda, \mu, \lambda\mu, \lambda^2, \mu^2, \ldots$

A few examples:
Set $\lambda = 0.9$, $\mu = 0.5$.

Apply DMD with initial states x given by (1,1), (5,5), (-1,1), (-5,5).

Case 1: $c = 0$ (linear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5]$ correspond to Koopman eigenvalues

Case 2: $c = 1$ (nonlinear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 2.002]$ do not correspond to Koopman eigenvalues, and the equilibrium appears unstable!

Case 3: $c = 1$ (nonlinear map, extended set of observables),
now assume access to more observables $g = (x_1, x_2, x_1^2)$, then DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5, 0.81]$ correspond to Koopman eigenvalues.

Consider the map

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \lambda x_1 \\ \mu x_2 + (\lambda^2 - \mu) c x_1^2 \end{bmatrix}.$$

This system has a stable equilibrium at the origin.

The Koopman eigenvalues are $\lambda, \mu, \lambda\mu, \lambda^2, \mu^2, \ldots$

A few examples:
Set $\lambda = 0.9$, $\mu = 0.5$.

Apply DMD with initial states x given by (1,1), (5,5), (-1,1), (-5,5).

Case 1: $c = 0$ (linear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5]$ correspond to Koopman eigenvalues

Case 2: $c = 1$ (nonlinear map),
DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 2.002]$ do not correspond to Koopman eigenvalues, and the equilibrium appears unstable!

Case 3: $c = 1$ (nonlinear map, extended set of observables),
now assume access to more observables $g = (x_1, x_2, x_1^2)$, then DMD eigenvalues $\lambda^{\text{DMD}} = [0.9, 0.5, 0.81]$ correspond to Koopman eigenvalues.

Note, in Case 3, the extended set of observables $g(x)$ spans the set of Koopman eigenfunctions, $\xi_\lambda(x) = x_1$, $\xi_\mu(x) = x_2 - cx_1^2$.

Theorem: Koopman and DMD (Tu et al., 2014; Williams et al., 2015)

Let ξ be an eigenfunction of $\mathcal{K}$ with eigenvalue λ, and suppose $\xi \in \text{span}\{g_j\}$, so that

$$\xi(x) = w_1 g_1(x) + w_2 g_2(x) + \cdots + w_n g_n(x)$$

for some $w = (w_1, \ldots, w_n) \in \mathbb{C}^n$. If $w \in \mathcal{R}(X)$, then w is a left eigenvector of K with eigenvalue λ: $w^* K = \lambda w^*$

Thus, DMD eigenvalues are Koopman eigenvalues, provided:

1. the set of observables is sufficiently large (i.e., $\xi \in \text{span}\{g_j\}$)
2. the dataset is sufficiently rich (i.e., $w \in \mathcal{R}(X)$)

Furthermore, Koopman eigenfunctions can be computed from the left eigenvectors of the DMD matrix K, as $\xi(x) = w^* g(x)$.

Theorem: Koopman and DMD (Tu et al., 2014; Williams et al., 2015)

Let ξ be an eigenfunction of $\mathcal{K}$ with eigenvalue λ, and suppose $\xi \in \mathrm{span}\{g_j\}$, so that

$$\xi(x) = w_1 g_1(x) + w_2 g_2(x) + \cdots + w_n g_n(x)$$

for some $w = (w_1, \ldots, w_n) \in \mathbb{C}^n$. If $w \in \mathcal{R}(X)$, then w is a left eigenvector of K with eigenvalue λ: $w^* K = \lambda w^*$

Thus, DMD eigenvalues are Koopman eigenvalues, provided:

1. the set of observables is sufficiently large (i.e., $\xi \in \mathrm{span}\{g_j\}$)
2. the dataset is sufficiently rich (i.e., $w \in \mathcal{R}(X)$)

Furthermore, Koopman eigenfunctions can be computed from the left eigenvectors of the DMD matrix K, as $\xi(x) = w^* g(x)$.

- Note: for a linear system, $\mathcal{K}$ has linear eigenfunctions, so the full-state observable $g(x) = x$ is sufficient to capture these; for nonlinear systems, however, linear observables are typically insufficient$\rightarrow$ transform the available data to construct an extended set of observables.

Conclusions

- Streaming DMD for handling large and streaming datasets.
  - Hemati, Williams, & Rowley (2014),
    "Dynamic Mode Decomposition for Large and Streaming Datasets,"
    Physics of Fluids.
- Noise-Corrected and Noise-Aware/Total DMD for handling noisy data.
  - Hemati, Rowley, Deem, Cattafesta (2015),
    "De-Biasing the Dynamic Mode Decomposition for Applied Koopman Spectral Analysis,"
    [pre-print, arXiv:1502.03854].
  - Dawson, Hemati, Williams, Rowley (2015),
    "Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition"
    [pre-print, arXiv: 1507.02264].
- Extended DMD for handling insufficiently rich sets of observables.
  - Williams, Kevrekidis, Rowley (2015),
    "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition,"
    J. Nonlinear Science.

Python and Matlab packages available for download at **http://z.umn.edu/dmdtools**