Data Fusion via Intrinsic Variables

M. O. Williams[1] I. G. Kevrekidis[1,2]

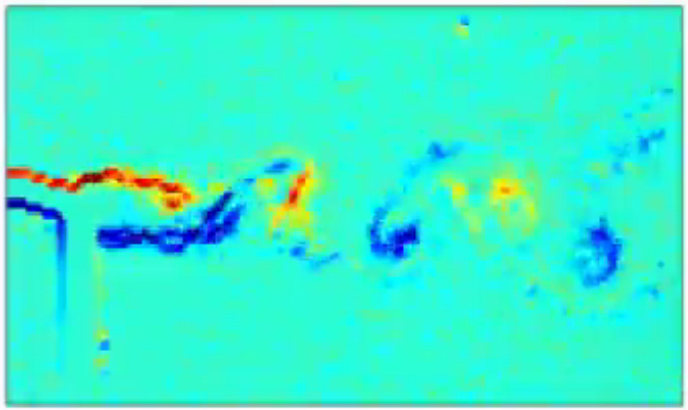
[1]Program in Applied and Computational Mathematics, Princeton University

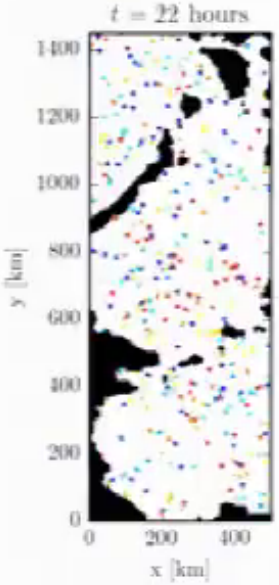
[2]Department of Chemical and Biological Engineering, Princeton University

May 21, 2015

Motivation: Data-Mining and Nonlinear Dynamics

- *High dimensional data* are becoming increasingly prevalent
- *Dimensionalilty reduction* has become a popular tool for uncovering structure in these types of data sets





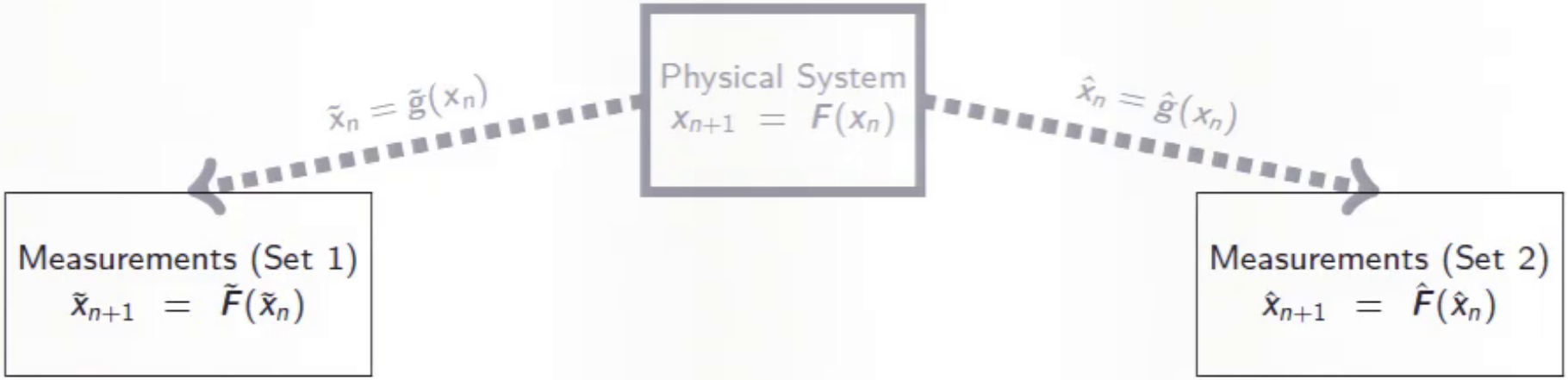
This talk focuses on data generated by multi-scale dynamical systems:

- High dimensional at first glance due to fast transients
- Effectively confined to a low-dimensional manifold afterwards

The Goal: Data Fusion or State Reconstruction

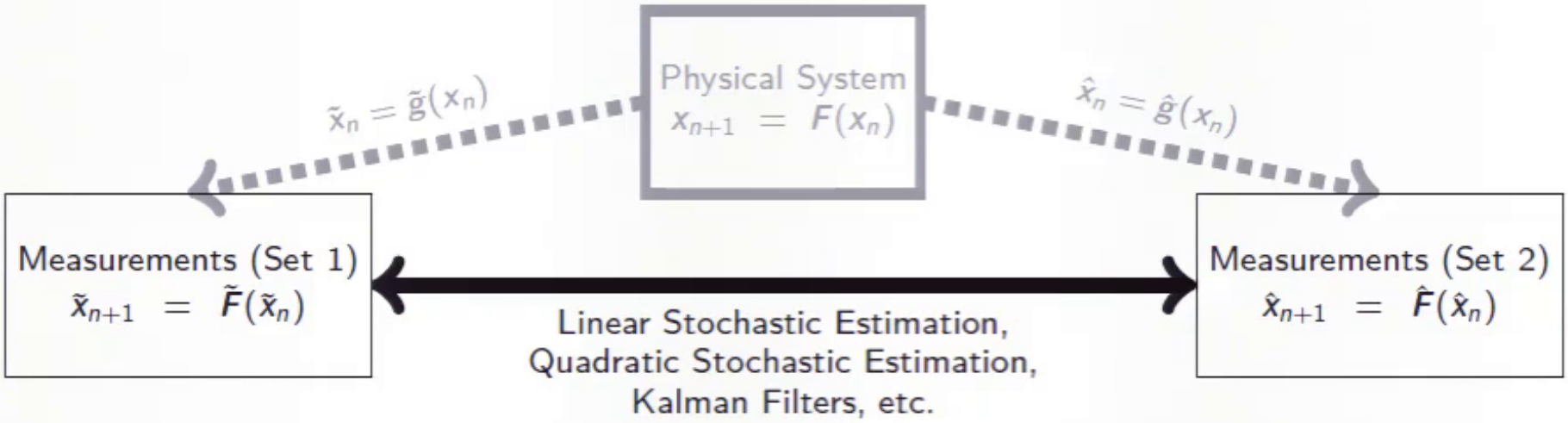
Data Fusion/State Reconstruction

Given two sets of measurements taken from the same underlying dynamics, find a mapping from one set to the other (e.g., approximating the velocity field in a fluid from pressure data on a body).



- Two types of sensors: $\tilde{g}$ and $\hat{g}$
- Two types of measurements: $\tilde{x}$ and $\hat{x}$
- Two effective evolution laws: $\tilde{F}$ and $\hat{F}$

Standard Approaches

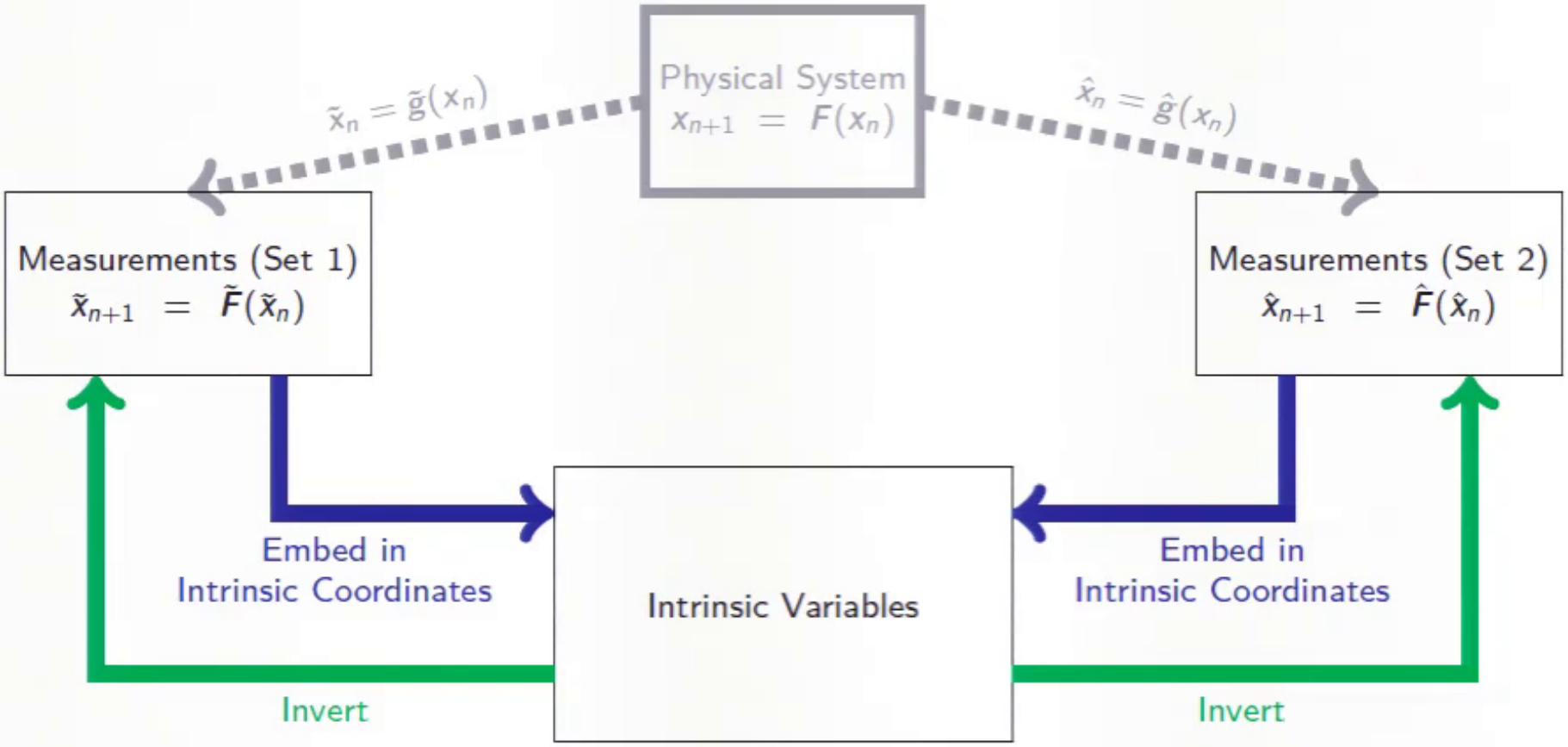


Standard Approaches: Find a mapping directly between $\hat{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}$. This requires:

- A large amount of joint data (for stochastic estimation)
- An accurate physical model (for Kalman filters)

Data Fusion via Intrinsic Variables

Our Approach: Map to and from a set of intrinsic variables



Question: How can intrinsic coordinates be defined?

Intrinsic Coordinates

We define our intrinsic coordinates based on **dynamics**.

- If they are *deterministic*, then we define our intrinsic variables so that:

$$\dot{x}_i = \mu_i x_i, \quad \text{for } i = 1, \ldots, N,$$

  using **Koopman spectral analysis**[1].

- If they are *stochastic*, then we define our intrinsic variables so that:

$$dx_i = a_i(\mathbf{x})dt + dw_i, \quad \text{for } i = 1, \ldots, N,$$

  using **Nonlinear Intrinsic Variables** (NIVs)[2].

---

[1] M.O.W. et al., "Data Fusion via Intrinsic Dynamic Variables: An Application of Data-Driven Koopman Spectral Analysis," *EPL* **109** 2015

[2] dSilva et al., "Nonlinear intrinsic variables and state reconstruction in multiscale simulations," *J. Chem. Phys.* **139** 2013

Outline

Finding the Linearity in Nonlinear Dynamics

Our Goal

Turn the system $\boldsymbol{x}_{n+1} = \boldsymbol{F}(\boldsymbol{x}_n)$ into

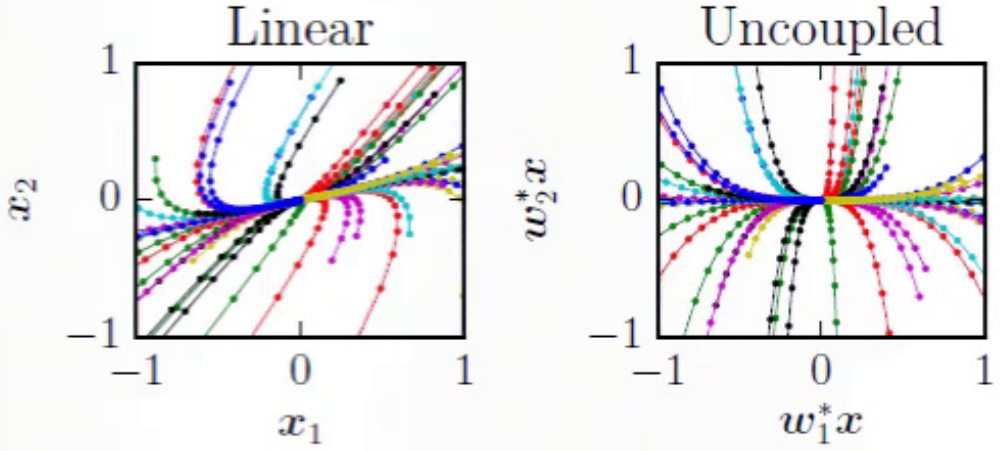
$$x_{n+1}^{(i)} = \mu_i x_n(i), \quad \text{for } i = 1, \ldots, N.$$

For the linear system $\boldsymbol{x}_{n+1} = \boldsymbol{A}\boldsymbol{x}_n$, this can be done via an **eigen-decomposition**.

$$\boldsymbol{x}_{n+m} = \sum_{k=1}^{N} \mu_k^m \boldsymbol{v}_k (\boldsymbol{w}_k^* \boldsymbol{x}_n)$$

where

- $\boldsymbol{v}_k \in \mathbb{R}^N$ the k-th eigenvector
- μ_k is the k-th eigenvalue
- $\boldsymbol{w}_k$ the k-th left eigenvector



- The $\boldsymbol{x}$ are dynamically coupled, but the $\boldsymbol{w}_n^* \boldsymbol{x}$ are not!

Uncoupling Nonlinear Systems: The Koopman Operator

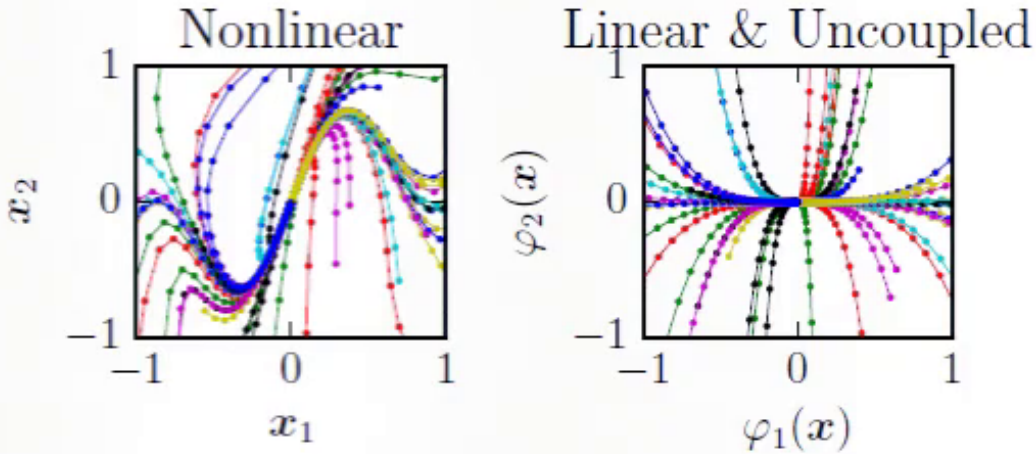
For the autonomous discrete time dynamical system $(n, \mathcal{M}, \boldsymbol{F})$, where $\boldsymbol{x}_{n+1} = \boldsymbol{F}(\boldsymbol{x}_n)$, we can define a **scalar observable** to be $\phi : \mathcal{M} \to \mathbb{R}$.

The action of the Koopman operator, $\mathcal{K}$, on ϕ is

$$(\mathcal{K}\phi)(\boldsymbol{x}) = \phi(\boldsymbol{F}(\boldsymbol{x})) = (\phi \circ \boldsymbol{F})(\boldsymbol{x}).$$

The Koopman operator maps functions to functions and not states to states.

- $\mathcal{K}$ is an *infinite* dimensional but *linear* operator
- The eigenfunctions, φ_k, linearize and uncouple the dynamics
- The eigenvalues, μ_k, determine the temporal evolution



Koopman Eigenvalues, Eigenfunctions, and Modes

Accompanying the Koopman eigenvalues and eigenfunctions are the **Koopman modes**, which depend on the dynamics *and* how they are observed.

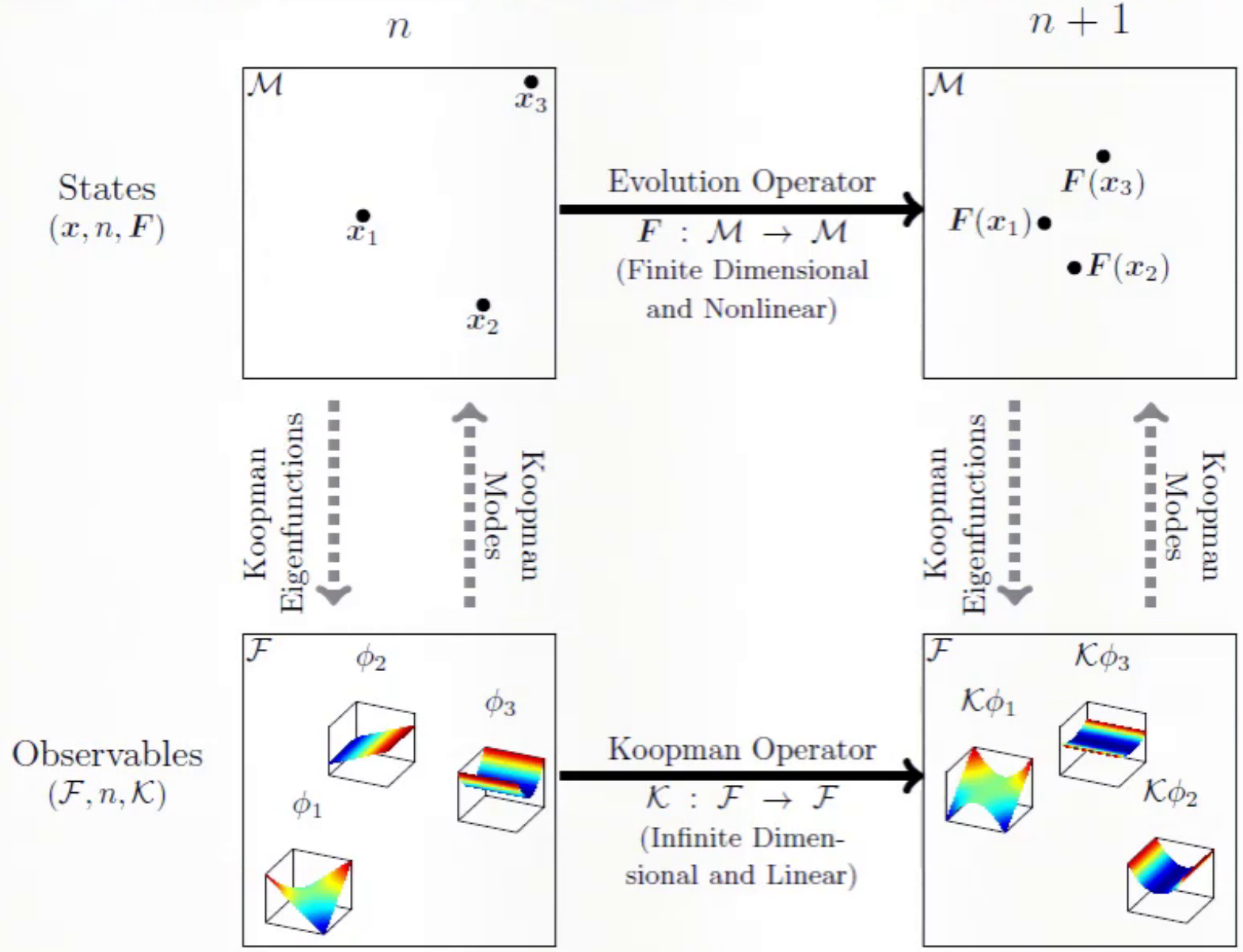
Often, vector valued observables (e.g., $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$) can be written as:

$$\boldsymbol{g}(\boldsymbol{x}_{n+m}) = \sum_{k=1}^{K} \mu_k^m \boldsymbol{v}_k \varphi_k(\boldsymbol{x}_n),$$

- $\boldsymbol{v}_k$ is the k-th **Koopman mode**, a fixed spatial profile
- μ_k is the k-th **Koopman eigenvalue**, and determines the temporal evolution of the k-th Koopman mode (often we also define $\lambda_k \triangleq \log(\mu_k)/\Delta t$)
- $\varphi_k : \mathcal{M} \to \mathbb{C}$ is the k-th **Koopman eigenfunction**, and determines the "weight" associated with the k-th Koopman mode

Nonlinear & Finite or Linear & Infinite?

The choice: Finite dimensional and nonlinear or infinite dimensional and linear?



Outline

Data-Driven Koopman Spectral Analysis

What do we do when we have data rather than equations?

- **Generalized Laplace Analysis:** Computes the Koopman modes and eigenfunctions, but requires Koopman eigenvalues
  - e.g., Mezić 2005; Budišić, Mohr, & Mezić 2012.
- **Ulam-Galerkin Method:** More commonly used to approximate the Perron-Frobenius operator, but could be used for computing Koopman eigenvalues and eigenfunctions
  - e.g., Dellnitz, Froyland, & Junge 2001; Froyland, Gottwald, & Hammerlindl 2014.
- **Dynamic Mode Decomposition:** Produces approximations of Koopman modes and eigenvalues
  - e.g., Schmid 2010; Rowley et al. 2009; Tu et al. 2014.
- **Extended Dynamic Mode Decomposition:** Produces approximations of Koopman modes, eigenvalues, and eigenfunctions
  - e.g., MOW, Kevrekidis, & Rowley (on arXiv).

Extended Dynamic Mode Decomposition

Data Rather Than Equations.

All we have access to is a data set $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$ where $\boldsymbol{y}_m = \boldsymbol{F}(\boldsymbol{x}_m)$.
However, $\boldsymbol{F}$ itself is unknown, and we cannot ask for more data.

- Choose $W_m(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_m)$. Then

$$\Psi_X = \begin{bmatrix} \psi_1(\boldsymbol{x}_1) & \cdots & \psi_K(\boldsymbol{x}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{x}_M) & \cdots & \psi_K(\boldsymbol{x}_M) \end{bmatrix}, \quad \Psi_Y = \begin{bmatrix} \psi_1(\boldsymbol{y}_1) & \cdots & \psi_K(\boldsymbol{y}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{y}_M) & \cdots & \psi_K(\boldsymbol{y}_M) \end{bmatrix}.$$

- The finite dimensional approximation of $\mathcal{K}$ is:

$$\boldsymbol{K} = \Psi_X^{+} \Psi_Y,$$

- The eigenvalues of $\boldsymbol{K}$ approximate the *eigenvalues* of $\mathcal{K}$
- The eigenvectors approximate the *eigenfunctions* of $\mathcal{K}$
- The left eigenvectors approximate the *modes* of $\mathcal{K}$

Extended Dynamic Mode Decomposition

Data Rather Than Equations.

All we have access to is a data set $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$ where $\boldsymbol{y}_m = \boldsymbol{F}(\boldsymbol{x}_m)$.
However, $\boldsymbol{F}$ itself is unknown, and we cannot ask for more data.

- Choose $W_m(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_m)$. Then

$$\Psi_X = \begin{bmatrix} \psi_1(\boldsymbol{x}_1) & \cdots & \psi_K(\boldsymbol{x}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{x}_M) & \cdots & \psi_K(\boldsymbol{x}_M) \end{bmatrix}, \quad \Psi_Y = \begin{bmatrix} \psi_1(\boldsymbol{y}_1) & \cdots & \psi_K(\boldsymbol{y}_1) \\ \vdots & & \vdots \\ \psi_1(\boldsymbol{y}_M) & \cdots & \psi_K(\boldsymbol{y}_M) \end{bmatrix}.$$

- The finite dimensional approximation of $\mathcal{K}$ is:

$$\boldsymbol{K} = \Psi_X^{+} \Psi_Y,$$

- The eigenvalues of $\boldsymbol{K}$ approximate the *eigenvalues* of $\mathcal{K}$
- The eigenvectors approximate the *eigenfunctions* of $\mathcal{K}$
- The left eigenvectors approximate the *modes* of $\mathcal{K}$

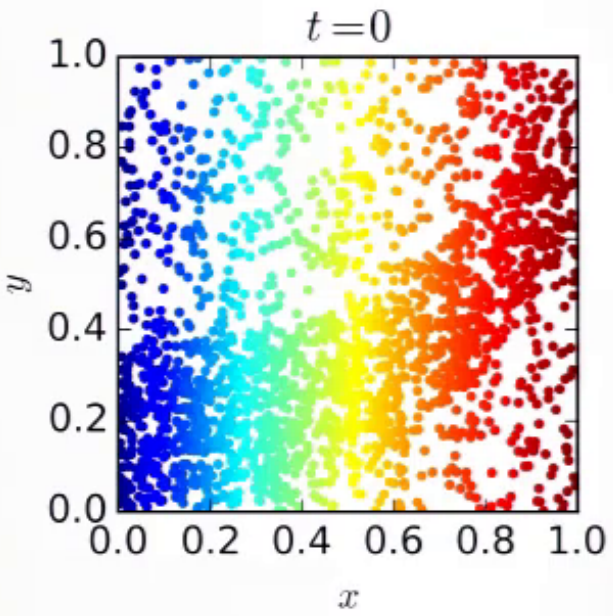
A Simple Example: A Singularly Perturbed System

Consider data from the system:

$$\dot{x} = -(x+y)$$
$$\epsilon\dot{y} = x^3 - y$$

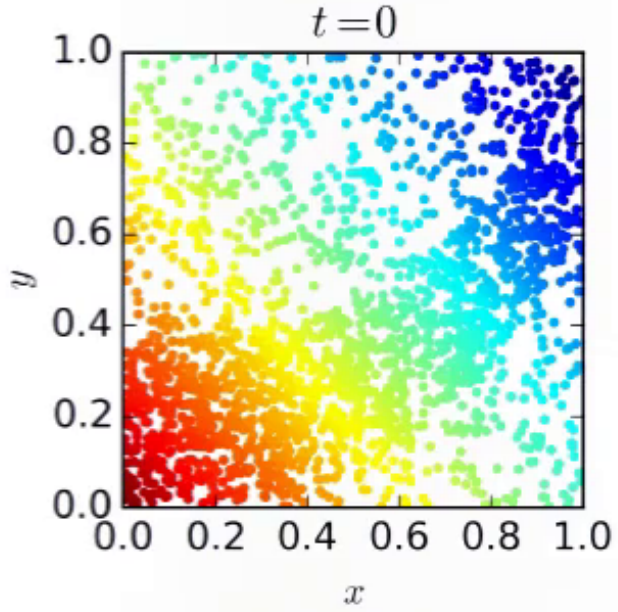
- x is "slow", y is "fast"
- 2000 snapshot pairs
- $\epsilon = 0.01$, $\Delta t = 0.001$
- Polynomial basis functions

Koopman



(Colored by slowest φ)

PCA



(Colors denote 1st PCA coeff.)

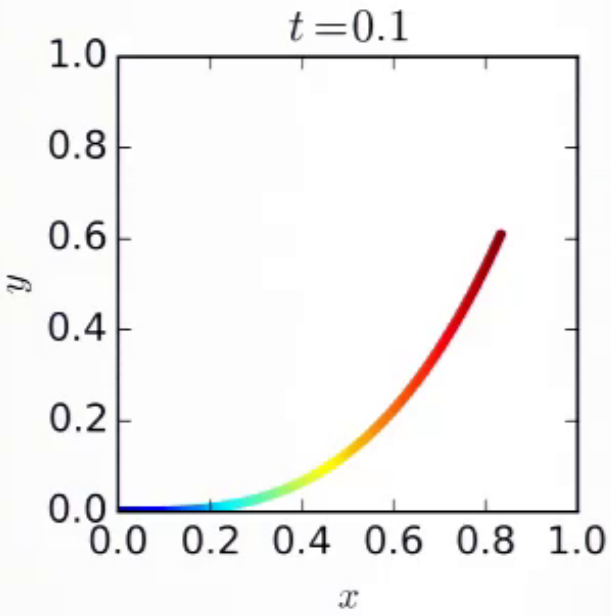
A Simple Example: A Singularly Perturbed System

Consider data from the system:

$$\dot{x} = -(x+y)$$
$$\epsilon\dot{y} = x^3 - y$$

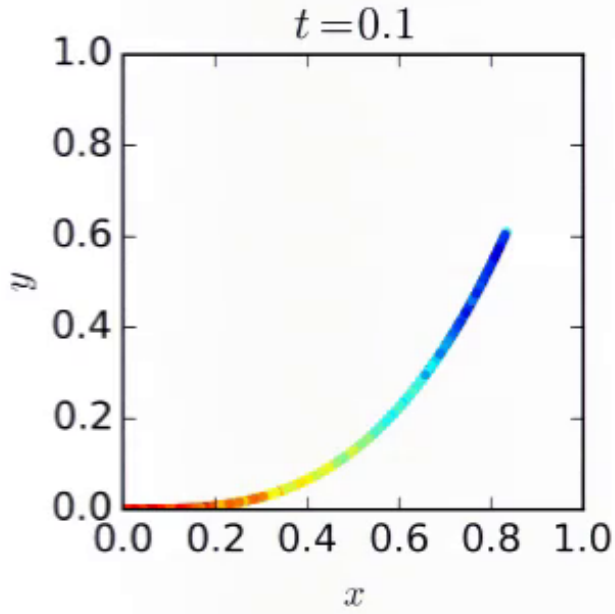
- x is "slow", y is "fast"
- 2000 snapshot pairs
- $\epsilon = 0.01$, $\Delta t = 0.001$
- Polynomial basis functions

Koopman



(Colored by slowest φ)

PCA



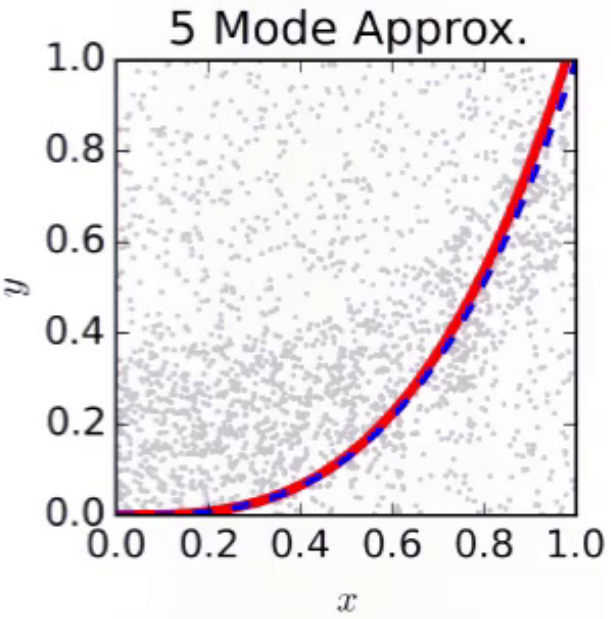
(Colors denote 1st PCA coeff.)

State Reconstruction with Koopman Modes

Truncated Koopman mode expansions reconstruct "slow" dynamics:

$$\boldsymbol{x} \approx \sum_{m=0}^{5} \boldsymbol{v}_m \varphi_m(\boldsymbol{x})$$

- Grey: original points
- Red: projected points
- Blue: slow manifold



In Summary:

- The **Koopman eigenfunctions** can parameterize state space in such a way that the dynamics look linear
- The **Koopman modes** allow the state to be approximated
- Both can be computed from data **if a sufficiently rich set of basis functions are chosen**

Outline

Large State Spaces and the Curse of Dimensionality

A Practical Problem: Size

Choosing a "rich" basis quickly results in unreasonably large problems as N, the dimension of state space, grows.

- Example: Approximating all polynomials on $\mathbb{R}^{400}$ up to 10th order requires $K \sim 10^{19}$ basis functions, and an $\mathcal{O}(MK^2)$ computation
- The cost can be reduced to $\mathcal{O}(KM^2)$ by using the matrix

$$\hat{\boldsymbol{K}} = \hat{\boldsymbol{G}}^{+}\hat{\boldsymbol{A}},$$

- The ij-th entries of these matrices are

$$\hat{\boldsymbol{G}}^{(ij)} = \boldsymbol{\Psi}(\boldsymbol{x}_i)^T \boldsymbol{\Psi}(\boldsymbol{x}_j) \qquad \hat{\boldsymbol{A}}^{(ij)} = \boldsymbol{\Psi}(\boldsymbol{y}_i)^T \boldsymbol{\Psi}(\boldsymbol{x}_i),$$

where $\boldsymbol{\Psi}(\boldsymbol{x}) = [\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \ldots, \psi_K(\boldsymbol{x})]^T$.

Extended DMD and the Kernel Trick

The Kernel Trick[3]

Rather than explicitly computing inner products,
we define a **kernel function** $f : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, and let

$$\hat{\boldsymbol{G}}^{(ij)} = f(\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad \hat{\boldsymbol{A}}^{(ij)} = f(\boldsymbol{y}_i, \boldsymbol{x}_j).$$

- If $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$, then $\boldsymbol{\Psi}(\boldsymbol{x}) = \boldsymbol{x}$ and this is DMD
- If $f(\boldsymbol{x}, \boldsymbol{y}) = (1 + \boldsymbol{x}^T \boldsymbol{y})^\alpha$, then $\boldsymbol{\Psi}$ spans all polynomials up to order α.
- A simple example:

$$f(x,y) = (1+xy)^2 = (1+2xy+x^2y^2) = \begin{bmatrix} 1 & \sqrt{2}x & x^2 \end{bmatrix} \cdot \begin{bmatrix} 1 & \sqrt{2}y & y^2 \end{bmatrix},$$

  which uses the basis functions $\boldsymbol{\Psi}(x) = \begin{bmatrix} 1 & \sqrt{2}x & x^2 \end{bmatrix}$.

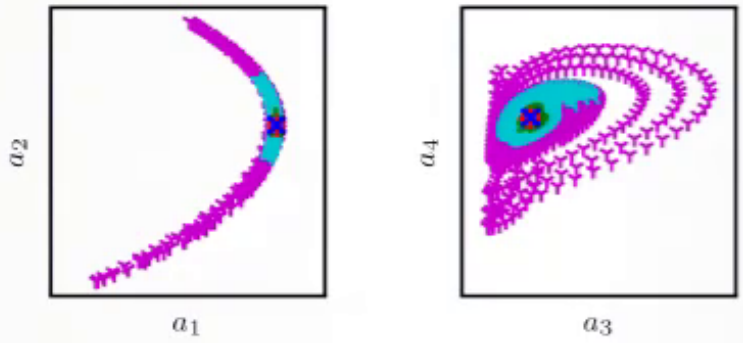
- $\hat{\boldsymbol{K}}$ can be computed in $\mathcal{O}(M^2 N)$ time; *the same as DMD* if $N > M$.

[3] C.M. Bishop, *Pattern recognition and machine learning*, 2006.

A Synthetic Example: The FitzHugh-Nagumo PDE

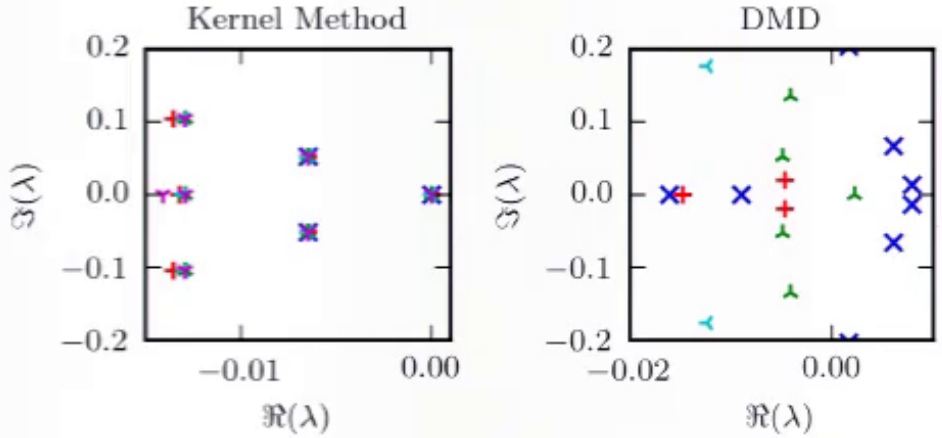
Using five sets of numerical data from the FitzHugh-Nagumo PDE:

The Data (in $\mathbb{R}^{400}$)

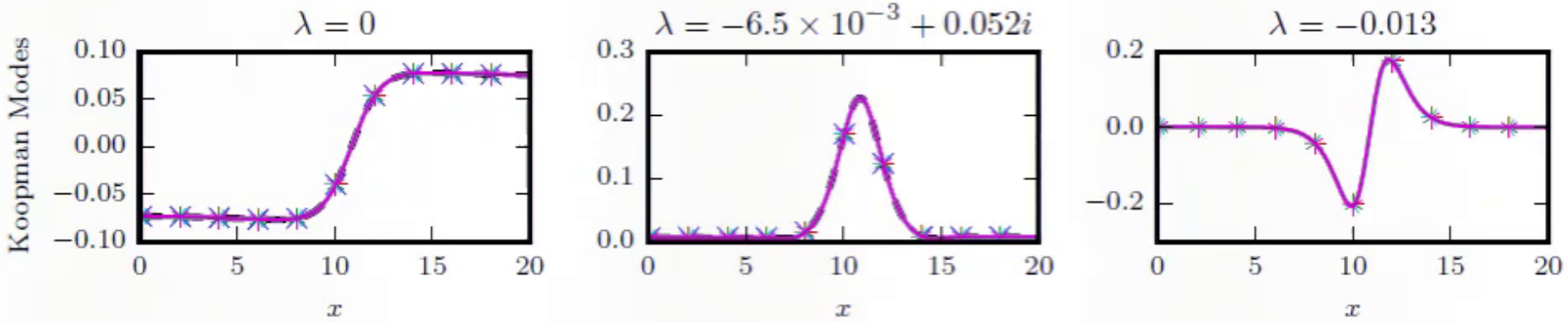


(Only first 4 PCA coeffs. shown)

The Spectra



The Koopman Modes



Outline

Data Fusion with Koopman Eigenfunctions

Our Approach

Uses the properties of the Koopman operator to perform data fusion on data produced by nonlinear systems in a way that minimizes the number of "joint" measurements

Treat $\tilde{\boldsymbol{x}}_{n+1} = \tilde{\boldsymbol{F}}(\tilde{\boldsymbol{x}}_n)$ and $\hat{\boldsymbol{x}}_{n+1} = \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}_n)$ as two separate systems.

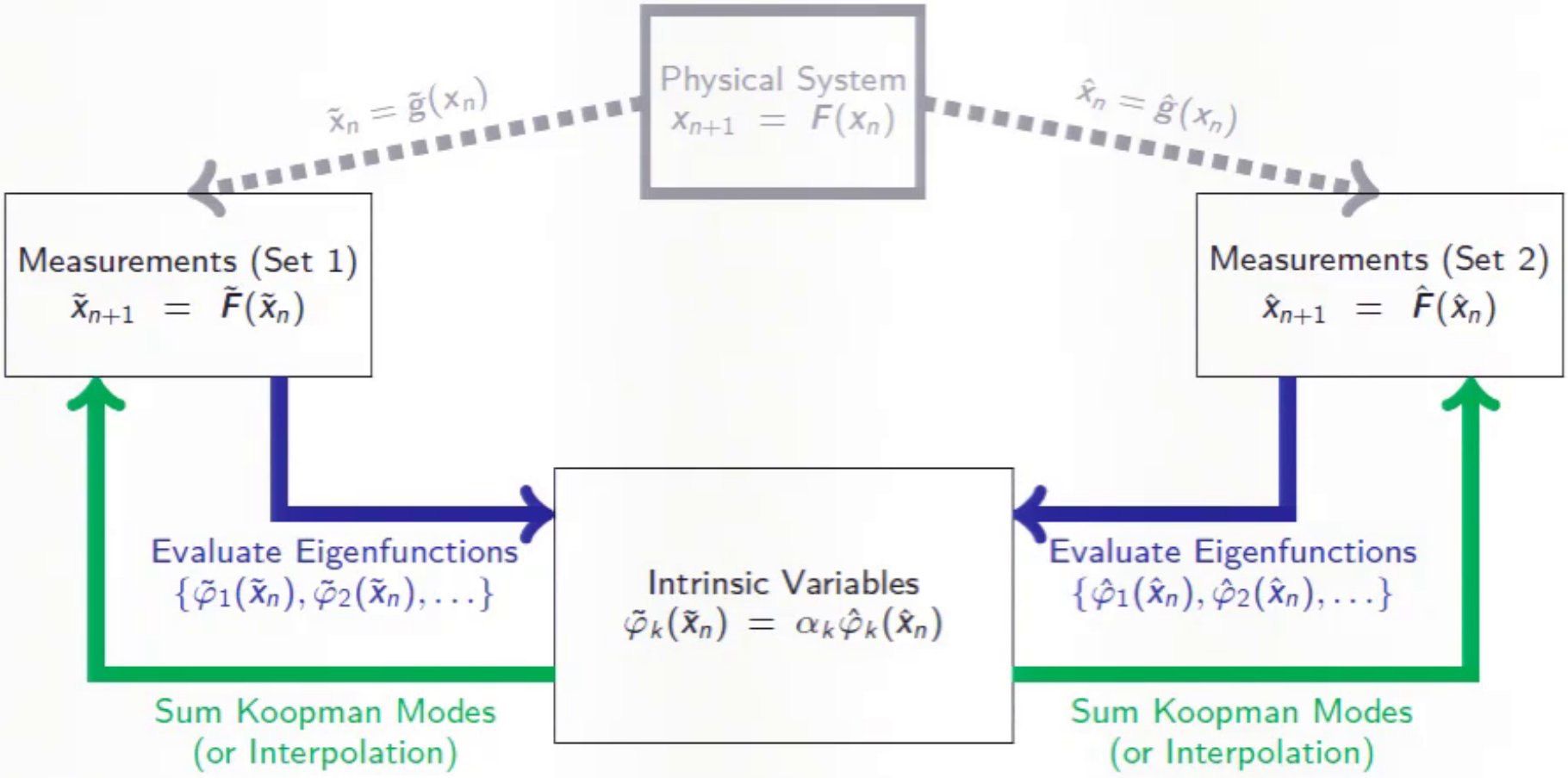
There are three key properties of the Koopman operator we need:

1. Koopman eigenfunctions can be used to parameterize state space[4]
2. $\tilde{\mathcal{K}}$ and $\hat{\mathcal{K}}$ have the same eigenvalues
   - Eigenfunctions can be "matched" using their associated eigenvalues
3. If $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{g}}(\boldsymbol{x})$ and $\hat{\boldsymbol{x}} = \hat{\boldsymbol{g}}(\boldsymbol{x})$, then $\tilde{\varphi}_k(\tilde{\boldsymbol{x}}) = \alpha_k \hat{\varphi}_k(\hat{\boldsymbol{x}})$.
   - The same parameterization can be obtained, but we must fix a normalization constant

[4] A. Mauroy, I. Mezic, & J. Moehlis, *Physica D* **261** (2013).

Data Fusion with Koopman Eigenfunctions

The Idea: Use the Koopman eigenfunctions as intrinsic variables



This approach requires:

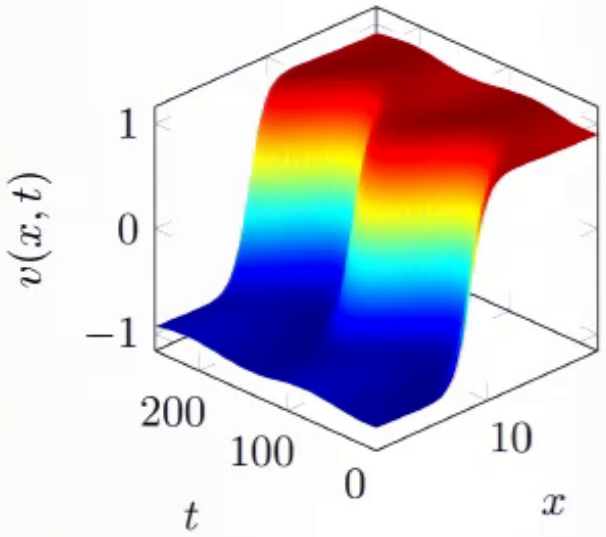
1. Koopman eigenfunctions and eigenvalues for $\tilde{x}$
2. Koopman eigenfunctions and eigenvalues for $\hat{x}$
3. The normalization constants α_k

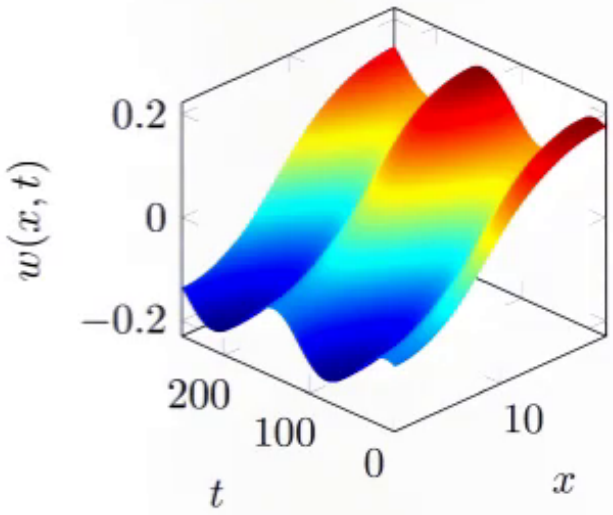
Data Fusion with Koopman Eigenfunctions

To compute these needed quantities, we need three sets of data:

1. A data set of snapshots pairs $\{(\tilde{\boldsymbol{x}}_m, \tilde{\boldsymbol{y}}_m)\}_{m=1}^{\tilde{M}}$ where $\tilde{\boldsymbol{y}}_m = \tilde{\boldsymbol{F}}(\tilde{\boldsymbol{x}}_m)$
2. Another data set of snapshots pairs $\{(\hat{\boldsymbol{x}}_m, \hat{\boldsymbol{y}}_m)\}_{m=1}^{\hat{M}}$ where $\hat{\boldsymbol{y}}_m = \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}_m)$
3. A set of "joint data," $\{(\hat{\boldsymbol{x}}_m, \tilde{\boldsymbol{x}}_m)\}_{m=1}^{M_{joint}}$, that are known to correspond to the same underlying state

We demonstrate this approach on a "master" set of data taken from the FitzHugh-Nagumo PDE in 1D:



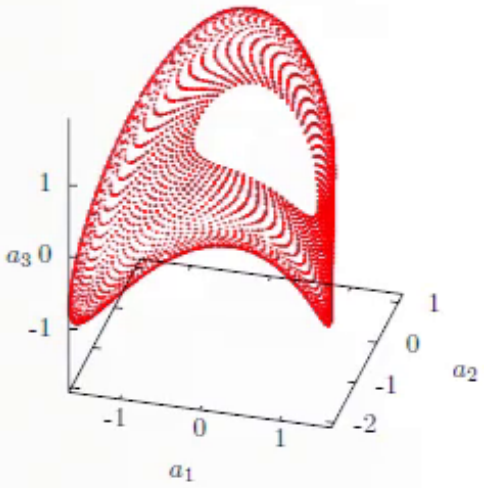


Data Fusion: FitzHugh-Nagumo PDE

The master data set:

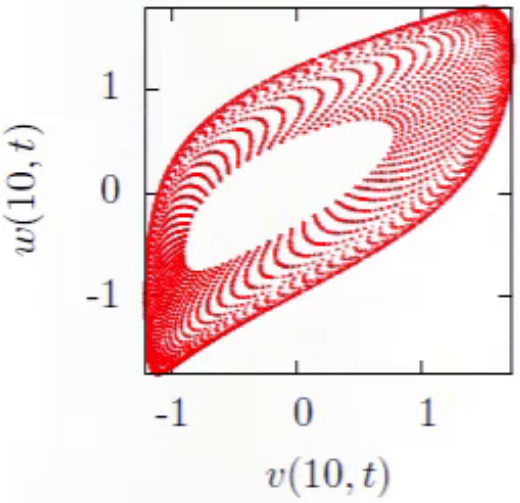
- contains 4×10^4 snapshots of v and w
- created by perturbing the unstable fixed point, and recording the approach to the limit cycle
- split into two data sets with 2×10^4 points:
  - Principal component coefficients
  - Pointwise measurements

PCA Data



(PCA performed on 2×10^4 data points)

Pointwise Data



(Pointwise measurements of v and w)

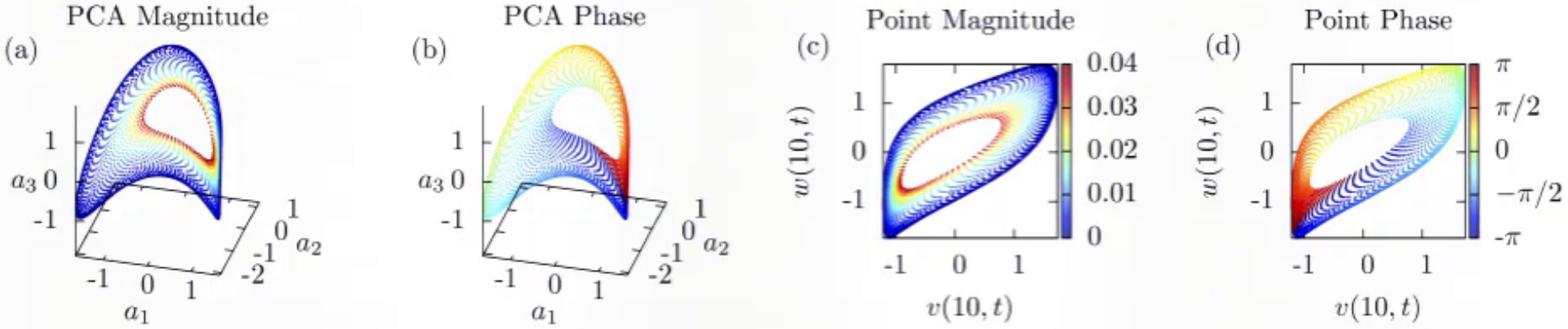
Joint Data

$$a_1 = -1.52$$
$$a_2 = -1.60$$
$$a_3 = -1.02$$
$$v(10) = 1.65$$
$$w(10) = 0.85$$

The Data Fusion Procedure

1. Compute the Koopman eigenfunctions and eigenvalues using the data in $\{(\tilde{\boldsymbol{x}}_m, \tilde{\boldsymbol{y}}_m)\}_{m=1}^{\tilde{M}}$. Use them to parameterize state space.
2. Repeat for the data in $\{(\hat{\boldsymbol{x}}_m, \hat{\boldsymbol{y}}_m)\}_{m=1}^{\hat{M}}$.
3. Ensure an consistent ordering by "matching" eigenfunctions based on their eigenvalues
4. Using the one "joint" measurement, pick α_k so that $\hat{\varphi}_k(\hat{\boldsymbol{x}}) = \alpha_k \tilde{\varphi}_k(\tilde{\boldsymbol{x}})$

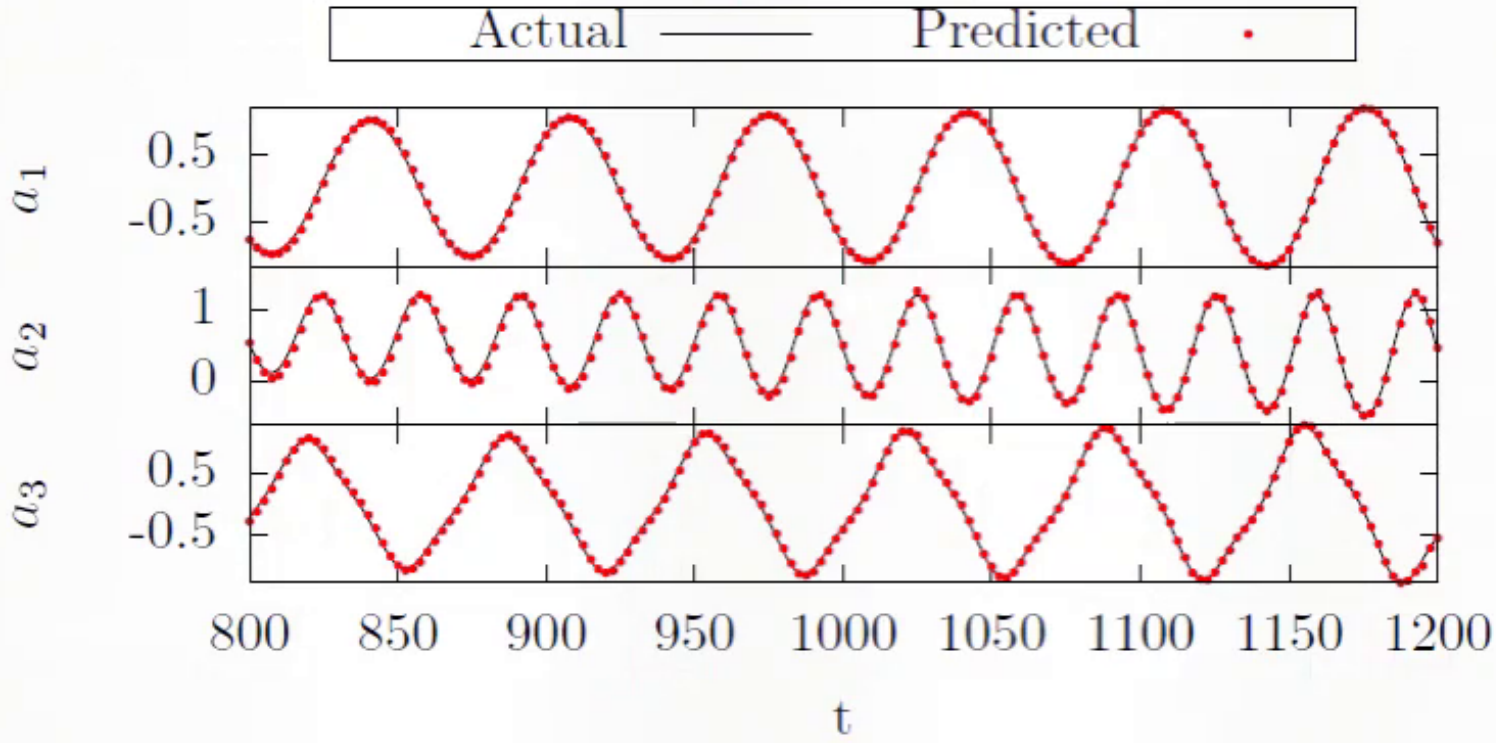
The Result



Embedding New Data

Test Case: a trajectory of 10^4 new pointwise measurements that were not used in the computation

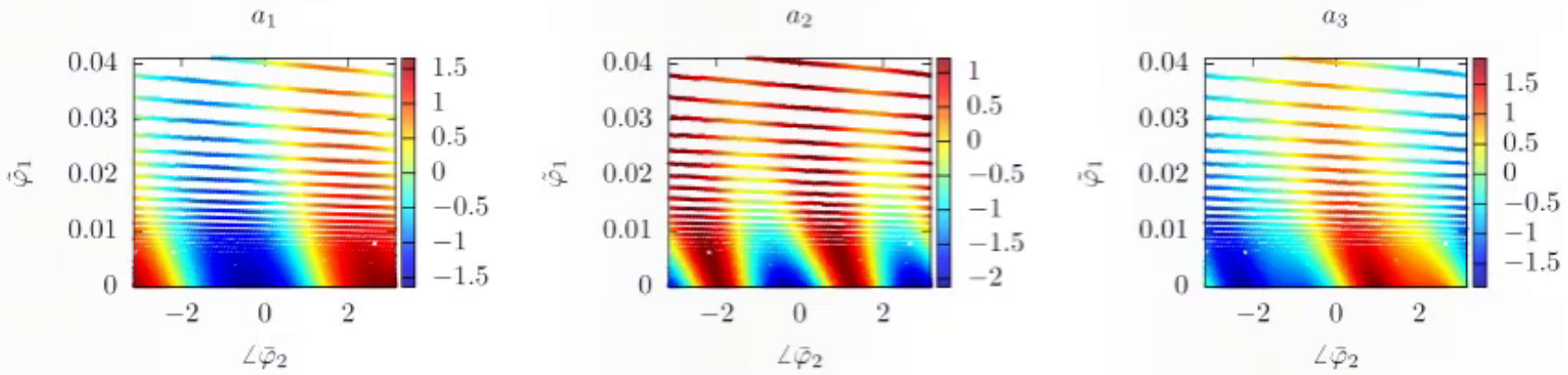
1. For each data point, evaluate the pointwise eigenfunctions to find its coordinates in the Koopman-based parameterization
2. Use interpolation to predict the corresponding PCA coeffs.



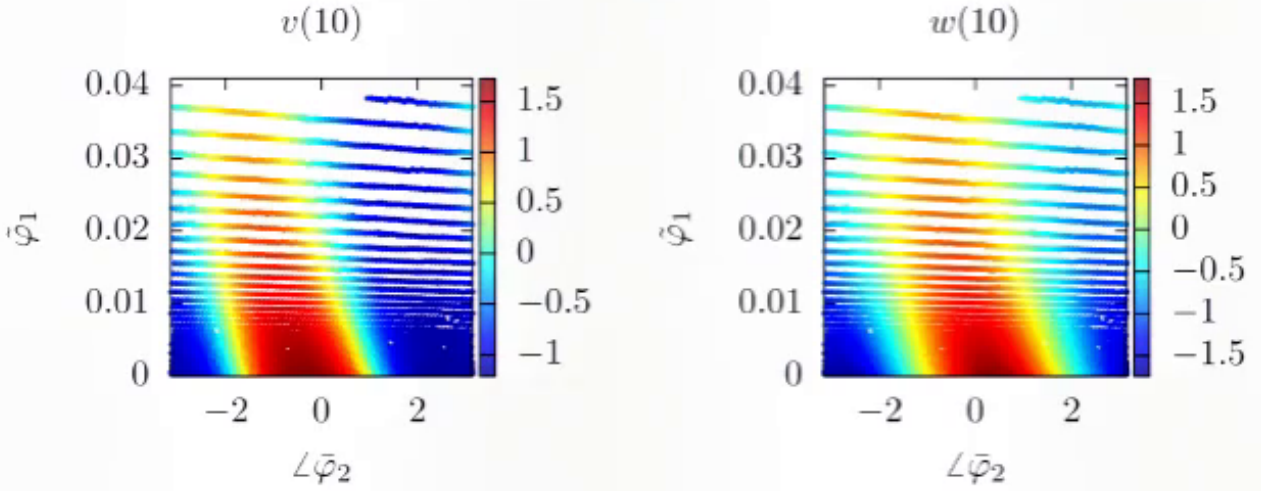
Eigenfunctions as Intrinsic Variables

The data displayed using a common set of coordinates:

Principal Component Coefficients



Pointwise Measurements



Embedding New Data

Test Case: a trajectory of 10^4 new pointwise measurements that were not used in the computation

1. For each data point, evaluate the pointwise eigenfunctions to find its coordinates in the Koopman-based parameterization
2. Use interpolation to predict the corresponding PCA coeffs.