Data-driven representation of dynamical systems

Tyrus Berry

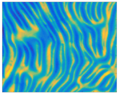
George Mason University

Sept. 20, 2019

Joint work with John Harlim, PSU and Dimitris Giannakis, NYU
Supported by NSF-DMS

Parametric Modeling

Data ⇄ **Model** ⇄ **Result**

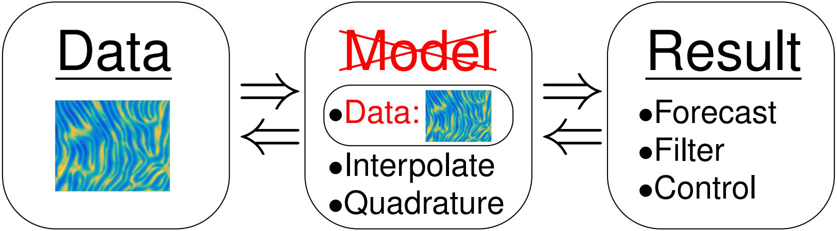


Model
- Equations
- Variables
- Parameters

Result
- Forecast
- Filter
- Control

- **Design Model:** Limited resolution and complexity
- **Assimilate Data:** Fit Parameters/Variables
  - Observability and noise
  - Model error
- **Study/Apply:** Ensemble Forecast

NONPARAMETRIC MODELING



- **Data IS the model:**
  - Assume a model exists
    - Data lies on/near an unknown sub-manifold
    - Data obeys an unknown dynamical system
  - Represent the model using training data

WHAT IS MANIFOLD LEARNING?

- **Manifold learning** $\Leftrightarrow$ **Estimating Laplace Operator**
- Euclidean space:
  - Eigenfunctions of Δ are $e^{i\vec{\omega}\cdot\vec{x}}$
  - Basis for Fourier transform
- Unit circle:
  - Eigenfunctions of Δ are $e^{ik\theta}$
  - Basis for Fourier series
- **Key Fact**: Eigenfunctions of Δ give the smoothest basis for square integrable functions on any manifold.

Why the Laplacian?

- **Manifold learning $\Leftrightarrow$ Estimating Laplace-Beltrami**
- Eigenfunctions $\Delta\varphi_i = \lambda_i\varphi_i$ orthonormal basis for $L^2(\mathcal{M})$
- Smoothest functions: φ_i minimizes the functional

$$\lambda_i = \min_{\substack{f\perp\varphi_k \\ k=1,\dots,i-1}} \left\{ \frac{\int_{\mathcal{M}} ||\nabla f||^2 \, dV}{\int_{\mathcal{M}} |f|^2 \, dV} \right\}$$

- Eigenfunctions of Δ are custom Fourier basis
  - Smoothest orthonormal basis for $L^2(\mathcal{M})$
  - Can be used to define wavelets
  - Define the Hilbert/Sobolev spaces on $\mathcal{M}$

DIFFUSION MAPS: GRAPH LAPLACIAN $\to$ MANIFOLD LAPLACIAN

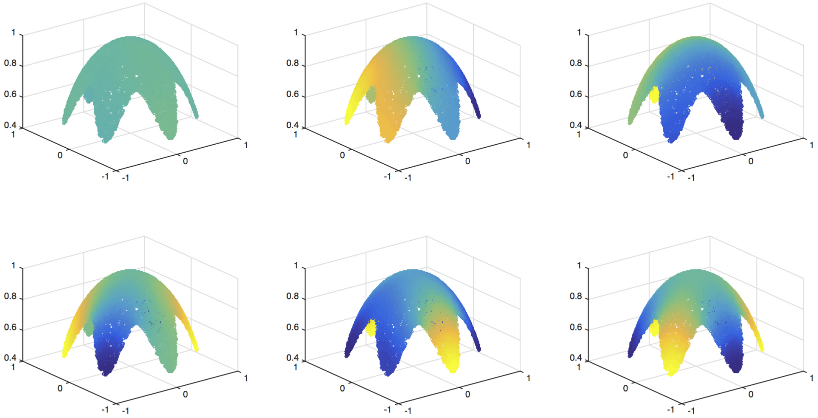
- For data points $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^n$
- Define $J_{ij} = J(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\delta^2}\right)$
- Define $D_{ii} = \sum_j J_{ij}$ (diagonal)
- Right normalization: $K = JD^{-1/2}$ and $\hat{D}_{ii} = \sum_j \hat{J}_{ij}$
- Left normalization: $\hat{K} = \hat{D}^{-1}K$
- Graph Laplacian: $L = \frac{1}{\delta^2}\left(I - \hat{K}\right)$
- **Theorem:** $L\vec{f} = \Delta_{p_{\rm eq}} + \mathcal{O}\left(\delta^2, N^{-1/2}\delta^{-1-d/2}\right)$

DIFFUSION MAPS: GRAPH LAPLACIAN $\to$ MANIFOLD LAPLACIAN

- For data points $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^n$
- Define $J_{ij} = J(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\delta^2}\right)$
- Define $D_{ii} = \sum_j J_{ij}$ (diagonal)
- Right normalization: $K = D^{-1/2} J D^{-1/2}$ and $\hat{D}_{ii} = \sum_j \hat{J}_{ij}$
- Left normalization: $\hat{K} = \hat{D}^{-1/2} K \hat{D}^{-1/2}$
- Graph Laplacian: $\hat{D}^{1/2} L \hat{D}^{-1/2} = \frac{1}{\delta^2}\left(I - \hat{K}\right)$
- **Theorem:** $L\vec{f} = \Delta_{p_{\mathrm{eq}}} + \mathcal{O}\left(\delta^2, N^{-1/2}\delta^{-1-d/2}\right)$

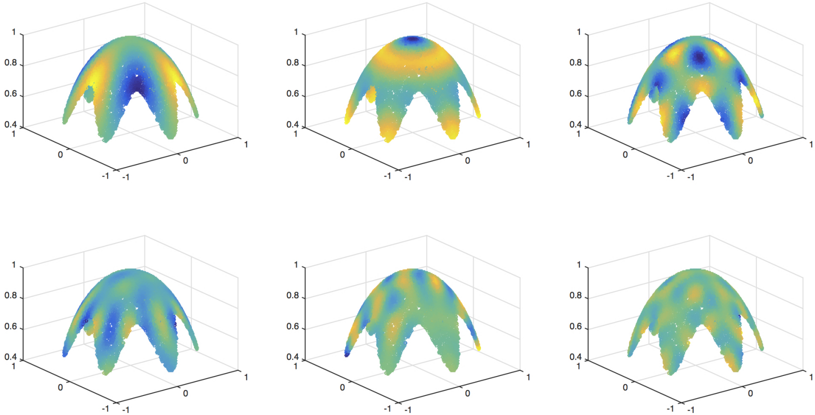
HARMONIC ANALYSIS ON MANIFOLDS/DATA SETS

- Unit circle: $\Delta = \frac{d^2}{d\theta^2}$ eigenfunctions are Fourier basis
- General manifold or data set $\Rightarrow$ Custom Fourier basis



Harmonic Analysis on Manifolds/Data Sets

- Unit circle: $\Delta = \frac{d^2}{d\theta^2}$ eigenfunctions are Fourier basis
- General manifold or data set $\Rightarrow$ Custom Fourier basis



FORECASTING WITH THE FOKKER-PLANK PDE

- Dynamical system: $dx = a(x)\,dt + b(x)\,dW_t$
- Uncertain initial state $x(0)$ with density $p(x,0)$
- Density solves Fokker-Planck PDE, $p_t = \mathcal{L}^* p$ where

$$\mathcal{L}^* f = -\nabla \circ (fa) + \frac{1}{2}\sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}\left(f \sum_k b_{ik} b_{jk} \right)$$

- Semigroup solution, $p(x,t) = e^{t\mathcal{L}^*} p(x,0)$

THE SHIFT MAP

- Given data samples $x_i = x(t_i)$ with $\tau = t_{i+1} - t_i$
- Define the *shift map* of a function by $Sf(x_i) = f(x_{i+1})$
- Using the Itô lemma we can show:

$$Sf(x_i) = f(x_{i+1}) = e^{\tau\mathcal{L}} f(x_i) + \int_{t_i}^{t_{i+1}} \nabla f^\top b\, dW_s + \int_{t_i}^{t_{i+1}} Bf\, ds$$

- Notice: $\mathbb{E}[S(f)] = e^{\tau\mathcal{L}} f$
- Need to minimize the stochastic integrand $\nabla f^\top b$

Representing the Shift Map

- Choose a basis $\{\varphi_j\}$ orthonormal with respect to $\langle\cdot,\cdot\rangle_{p_{\rm eq}}$
- The coefficients $c_l(t) = \langle p(x,t), \varphi_l\rangle$ have evolution:

$$\begin{aligned} c_l(t+\tau) &= \langle p(x,t+\tau), \varphi_l\rangle \\ &= \left\langle e^{\tau\mathcal{L}^*} p(x,t), \varphi_l \right\rangle = \left\langle p(x,t), e^{\tau\mathcal{L}}\varphi_l\right\rangle \\ &= \sum_j c_j(t) \left\langle \varphi_j, e^{\tau\mathcal{L}}\varphi_l\right\rangle_{p_{\rm eq}} = \sum_j A_{lj} c_j(t) \end{aligned}$$

- So $\vec{c}(t+\tau) = A\vec{c}(t)$
- Where $A_{lj} = \left\langle \varphi_j, e^{\tau\mathcal{L}}\varphi_l\right\rangle_{p_{\rm eq}} \approx \frac{1}{N}\sum_{i=1}^{N} \varphi_j(x_i)\varphi_l(x_{i+1})$

FORECASTING WITH THE SHIFT MAP

$$
\begin{array}{ccc}
p(x,t) & \xrightarrow{\text{Diffusion Forecast}} & p(x,t+\tau) \\
\downarrow \langle p, \varphi_j \rangle & & \uparrow \sum_j c_j \varphi_j p_{\mathrm{eq}} \\
\vec{c}(t) & \xrightarrow{A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_{p_{\mathrm{eq}}}]} & \vec{c}(t+\tau) = A\vec{c}(t).
\end{array}
$$

- We estimate $c_l(t) \approx \frac{1}{N}\sum_{i=1}^{N} \varphi_l(x_i) p(x_i,t)/p_{\mathrm{eq}}(x_i)$
- We estimate A_{lj} with $\hat{A}_{lj} = \frac{1}{N}\sum_{i=1}^{N} \varphi_j(x_i)\varphi_l(x_{i+1})$
- $\mathbb{E}[\hat{A}_{lj}] = A_{lj}$ with error $\mathcal{O}(||\nabla\varphi_l||_{p_{\mathrm{eq}}}\sqrt{\tau/N})$

Choosing a basis

- Need to minimize the error term $\mathcal{O}(||\nabla \varphi_l||_{p_{\rm eq}} \sqrt{\tau/N})$
- The minimizers of $||\nabla \varphi_l||_{p_{\rm eq}}$ are a generalized Fourier basis
- Let $\Delta_{p_{\rm eq}} = \Delta + \frac{\nabla p_{\rm eq}}{p_{\rm eq}} \cdot \nabla$ be the Laplacian weighted by $p_{\rm eq}$
- The eigenfunctions $\Delta_{p_{\rm eq}} \varphi_j = \lambda_j \varphi_j$ minimize $||\nabla \varphi_j||_{p_{\rm eq}} = \lambda_j$
- How do we find φ_j? Manifold Learning: Diffusion Maps

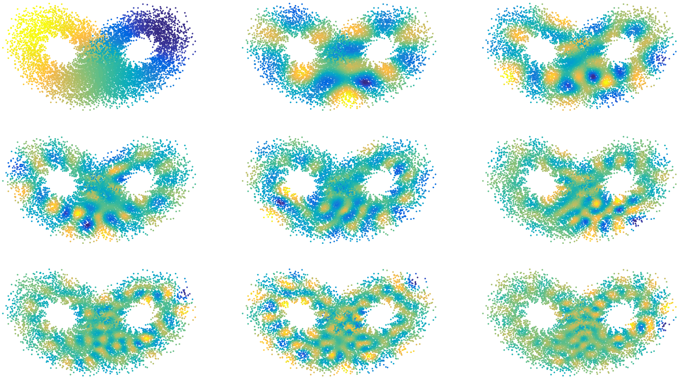
Diffusion Forecast

- Autonomous SDE: $dx = a(x)\,dt + b(x)\,dW_t$
- Density solves Fokker-Planck PDE: $\frac{\partial}{\partial t} p = \mathcal{L}^* p$
- Shift map: $S(p)(x_i) = p(x_{i+1})$ estimates: $\mathbb{E}[S(p)] = e^{\tau \mathcal{L}} p$
- Project onto custom Fourier basis (spectral method)

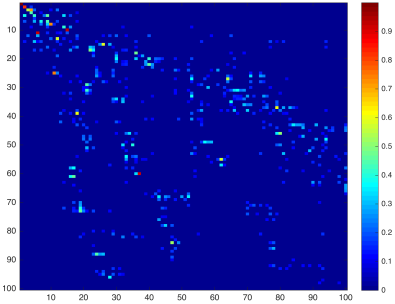
$$
\begin{array}{ccc}
p(x,t) & \xrightarrow{\text{Diffusion Forecast}} & p(x,t+\tau) = e^{\tau\mathcal{L}^*} p(x,t) \\
\Big\downarrow \langle p, \varphi_j \rangle & & \Big\uparrow \sum_j c_j \varphi_j q \\
\vec{c}(t) & \xrightarrow{A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_q]} & \vec{c}(t+\tau) = A\vec{c}(t).
\end{array}
$$

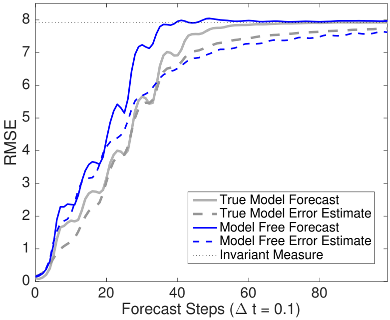
MANIFOLD LEARNING ⇒ CUSTOM 'FOURIER' BASIS

- **Optimal basis:** Minimum variance $A_{lj} \equiv \mathbb{E}[\langle \varphi_j, S\varphi_l \rangle_q]$



Shift map ⇒ Markov matrix





DIFFUSION FORECAST EXAMPLE

(Loading Video...)

RELATIONSHIP TO CLASSICAL METHODS

- For partial observations, use Takens' reconstruction
- Local linear representations
  - Based on nearest neighbor interpolation
  - Kernel regression also interpolates from neighbors ($\approx$ linear for large data set near manifold)
  - Diffusion forecast extends the map to distributions
- Partition state space $\Rightarrow$ Markov matrix
  - Also uses the shift map, just a different basis
  - Diffusion forecast is optimal basis for estimation

RELATIONSHIP TO RESERVOIR COMPUTERS

- Create a random (recurrent) network $v_k \in \mathbb{R}^N$

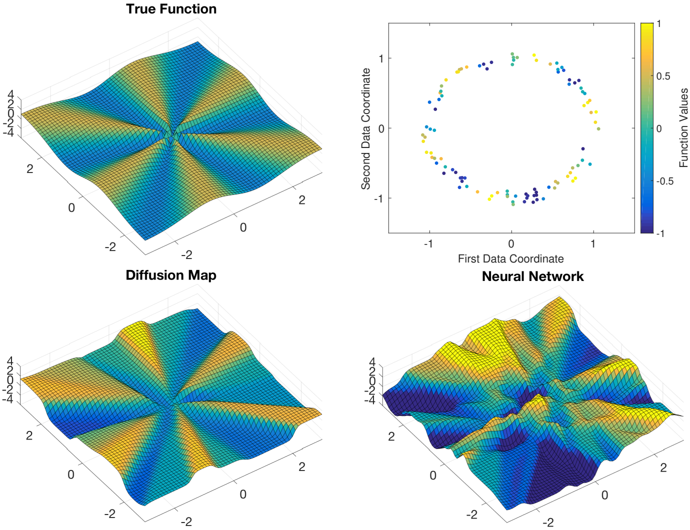
$$v_{k+1} = f(Av_k + Bx_k)$$

- Continuously feed in the time series x_k

$$\begin{aligned} v_{k+1} &= f(Af(Av_{k-1} + Bx_{k-1}) + Bx_k) = \cdots \\ &= f(Af(A \cdots f(Av_{k-\tau} + Bx_{k-\tau}) + \cdots) + Bx_k) \\ &= g(x_k, x_{k-1}, ..., x_{k-\tau}) \end{aligned}$$

- Predict: $x_{k+1} = Wv_k = Wg(x_k, ..., x_{k-\tau})$
- Since $\lambda_{\max}(A) < 1$ network forgets distant past
- Effectively a random diffeomorphism of a delay embedding
- Effectively uses a linear combination W of random basis!

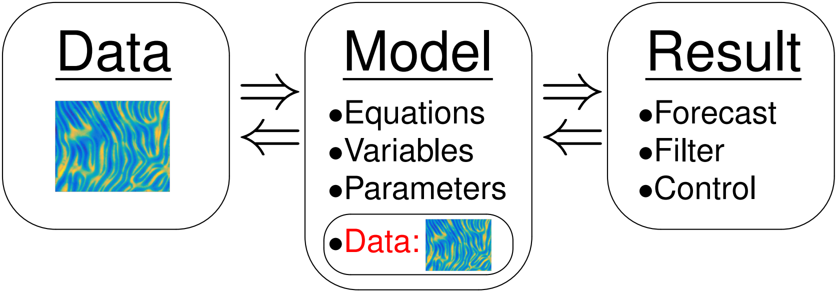
Choose your basis: Nyström vs. Neural net



PROBLEM: CURSE OF DIMENSIONALITY

- Nonparametric methods $\rightarrow$ Data required grows like a^{dim}
- Maybe we shouldn't throw out the model...
- Use diffusion forecast to fix model error!

SEMIPARAMETRIC MODELING



- **Data becomes part of the model:**
  - Start with imperfect parametric model
  - Fit training data with time-varying parameters
  - Query data as part of running model
- **Compensate for model error:**
  - Truncated resolution and complexity
  - Non-analytic expressions
  - Non-stationarity/Inhomogeneity

SEMIPARAMETRIC FORECAST MODEL

- Partially known model $\dot{x} = f(x, \theta)$
- Unknown: $d\theta = a(\theta)\,dt + b(\theta)\,dW_t$
- Apply the Diffusion Forecast to $p(\theta, t)$
- Sample $\theta^k(t) \sim p(\theta, t)$ and pair with ensemble $x^k(t)$

$$(x^k(t), \theta^k(t)) \xrightarrow{\dot{x}=f(x,\theta)} (x^k(t+\tau), \theta^k(t+\tau))$$

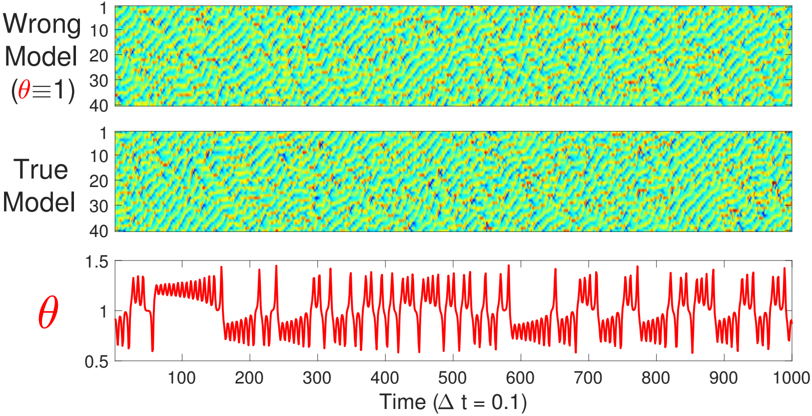
$$\uparrow \theta^k(t) \qquad\qquad \uparrow \theta^k(t+\tau)$$

$$p(\theta, t) \dashrightarrow p(\theta, t+\tau)$$

Diffusion Forecast

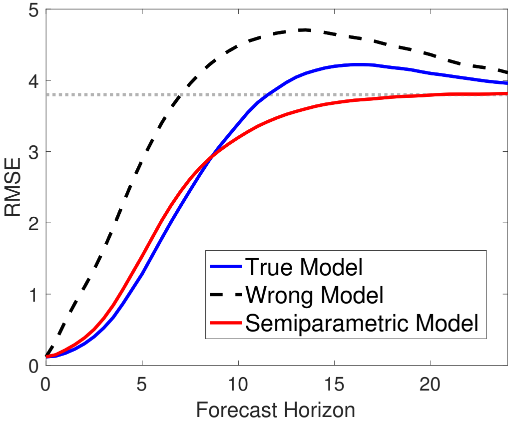
Example: 40-dimensional Lorenz-96 system

$$\dot{x}_i = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



EXAMPLE: 40-DIMENSIONAL LORENZ-96 SYSTEM

$$\dot{x}_i = \theta x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$



Projections of High Dimensional Dynamics

- Consider the 40-dimensional Lorenz-96 system:

$$\dot{x}_i = x_{i-1}x_{i+1} - x_{i-1}x_{i-2} - x_i + 8$$

- Assume we only observe a projection of this system

$$y = h(x_1, ..., x_{40})$$

- Example: Spatial Fourier mode $y = \hat{x}_\omega = \sum_{k=1}^{40} x_i e^{-k\omega}$
- Evolution of y is not closed, sometimes modeled by SDEs

ATTRACTOR RECONSTRUCTION

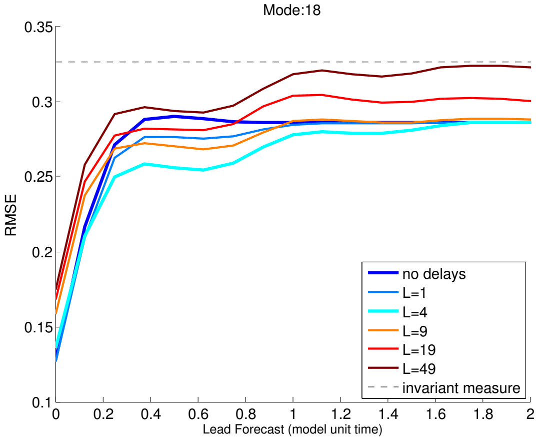
- Evolution of $y = h(x)$ is not closed (missing information)
- Idea: Use delay-embedding to recover the missing info
- Problem 1: Delay embeddings are biased towards stable directions

$$\tilde{y}_t \equiv (y_t, y_{t-\tau}, ..., y_{t-L\tau}) = (h(x_t), h(F_{-\tau}(x_t), ..., h(F_{-L\tau}(x_t))$$

- Problem 2: Curse-of-dimensionality prevents learning the full attractor
- Adding some delays helps, but adding too many hurts

Attractor Reconstruction

- Evolution of $y = h(x)$ is not closed
- Adding some delays helps, but adding too many hurts



NEXT STEPS: MORI-ZWANZIG FORMALISM

- Evolution of $y = h(x)$ is not closed
- Delay-embedding, $\tilde{y}_t$ only yeilds partial reconstruction
- Projections of dynamical systems can be closed as

  Mori-Zwanzig formalism: $$\frac{d}{dt}\tilde{y} = V + K + R$$

- Diffusion Forecast includes: V (Markovian), R (stochastic)
- Missing the memory term: $K = \int_{-\infty}^{t} K(s, \tilde{y}_t, \tilde{y}_s)\tilde{y}_s \, ds$

WHAT ABOUT THE OTHER RIEMANNIAN STRUCTURE?

- A Riemannian manifold has an exterior calculus:
  - Calculus of tensors and differential forms
  - Built entirely from the Riemannian metric $g \Leftrightarrow \Delta$
  - Formulates the generalization of the FTC (Stokes' Thm)
  - Can construct Laplacians on k-forms, Δ_k
  - Eigenforms of Δ_k are smoothest basis for k-forms
- **Question:** Given only the eigenfunctions of the Laplacian how can we construct the rest of the exterior calculus?

WHAT ABOUT THE OTHER RIEMANNIAN STRUCTURE?

- **Good News:** Laplacian $\Leftrightarrow$ Riemannian metric

$$g(\nabla f, \nabla h) = \nabla f \cdot \nabla h = \frac{1}{2}(f\Delta h + h\Delta f - \Delta(fh))$$

- Let $v, w \in T_x\mathcal{M}$, there exists $f_1, ..., f_d$ such that $\nabla f_1, ..., \nabla f_d$ span $T_x\mathcal{M}$ and

$$g(v, w) = v \cdot w = \sum_{ij} v_i w_j \nabla f_i \cdot \nabla f_j$$

- **Bad News:** There may be no $f_1, ..., f_d$ that work for all x
- Hairy Ball Thm: Every smooth vector field on S^2 must vanish: at these points the gradients do not span $T_x\mathcal{M}$.

How can we use the Laplacian eigenfunctions?

- Cannot find $\nabla f_1, ..., \nabla f_d$ basis for all $T_x\mathcal{M}$
- **Whitney:** We can find $\nabla f_1, ..., \nabla f_{2d}$ span all $T_x\mathcal{M}$
- **Thm**[1]**:** $\exists J$ such that $\nabla \varphi_1, ..., \nabla \varphi_J$ span all $T_x\mathcal{M}$
- Representing vector fields in a frame (overcomplete set)
  - Let $v(x) \in T_x\mathcal{M}$ be a smooth vector field
  - Then $v(x) = \sum_{j=1}^{J} c_j(x) \nabla \varphi_j(x)$ where $c_j(x)$ are smooth
  - So $c_j(x) = \sum_{i=1}^{\infty} c_{ij} \varphi_i(x)$
  - Finally $v = \sum_{i,j} c_{ij} \varphi_i \nabla \varphi_j$ (not uniquely)

[1] J. Portegies, Embeddings of Riemannian Manifolds with Heat Kernels and Eigenfunctions. (2014).

How can we use the Laplacian eigenfunctions?

- **Thm (Berry & Giannakis)** Let φ_i be the eigenfunctions of the Laplacian then $\{\varphi_i \nabla \varphi_j \,:\, j = 1, ..., J, i = 1, ..., \infty\}$ is a **frame** for the L^2 space of vector fields on $\mathcal{M}$.
- A **frame** is an overcomplete spanning set commonly used in Harmonic analysis, must satisfy the frame inequalities:

$$A||v||^2 \leq \sum_{i,j} \left\langle v, \varphi_i \nabla \varphi_j \right\rangle^2 \leq B||v||^2$$

where $A, B > 0$ and $|| \cdot ||^2 = \langle \cdot, \cdot \rangle$ is the Hodge inner prod.

T. Berry & D. Giannakis, Spectral exterior calculus. (Preprint available on arXiv)

The Spectral Exterior Calculus (SEC)

- We extend Thm to frames for Sobolev spaces of tensors
- SEC formulates the entire exterior calculus in these frames
- Key accomplishment: Representation of the 1-Laplacian

$$\Delta_1 = d\delta + \delta d$$

- Key challenge: Frame representations are not unique, requires Sobolev regularizations for numerical stability

T. Berry & D. Giannakis, Spectral exterior calculus. (Preprint available on arXiv)

A CALCULUS NEEDS FORMULAS!

Object	Symbolic	Spectral
Function	f	$\hat{f}_k = \langle \phi_k, f \rangle_{L^2}$
Laplacian	Δf	$\langle \phi_k, \Delta f \rangle_{L^2} = \lambda_k \hat{f}_k$
L^2 Inner Product	$\langle f, h \rangle_{L^2}$	$\sum_i \hat{f}_i^* \hat{h}_i$
Dirichlet Energy	$\langle f, \Delta f \rangle_{L^2}$	$\sum_i \lambda_i \lvert \hat{f}_i \rvert^2$
Multiplication	$\phi_i \phi_j$	$c_{ijk} = \left\langle \phi_i \phi_j, \phi_k \right\rangle_{L^2}$
Function Product	fh	$\sum_{ij} c_{kij} \hat{f}_i \hat{h}_j$
Riemannian Metric	$\nabla \phi_i \cdot \nabla \phi_j$	$g_{kij} \equiv \left\langle \nabla \phi_i \cdot \nabla \phi_j, \phi_k \right\rangle_{L^2}$ $= \frac{1}{2}(\lambda_i + \lambda_j - \lambda_k) c_{kij}$
Gradient Field	$\nabla f(h) = \nabla f^* \cdot \nabla h$	$\langle \phi_k, \nabla f(h) \rangle_{L^2} = \sum_{ij} g_{kij} \hat{f}_i \hat{h}_j$
Exterior Derivative	$df(\nabla h) = df^* \cdot dh$	$\sum_{ij} g_{kij} \hat{f}_i \hat{h}_j$
Vector Field (basis)	$v(f) = v^* \cdot \nabla f$	$\sum_j v_{ij} \hat{f}_j$
Divergence	$\mathrm{div} v$	$\langle \phi_i, \mathrm{div} v \rangle_{L^2} = -v_{0i}$
Frame Elements	$b_{ij}(\phi_l) = \phi_i \nabla \phi_j(\phi_l)$	$G_{ijkl} \equiv \left\langle b_{ij}(\phi_l), \phi_k \right\rangle_{L^2} = \sum_m c_{mik} g_{mjl}$
Vector Field (frame)	$v(f) = \sum_{ij} v^{ij} b_{ij}(f)$	$\langle \phi_k, v(f) \rangle_{L^2} = \sum_{ijl} G_{ijkl} v^{ij} \hat{f}_l$
Frame Elements	$b^{ij}(v) = b^i \, db^j(v)$	$\left\langle \phi_k, b^{ij}(v) \right\rangle_{L^2} = \sum_{nlm} c_{kmi} G_{nlmj} v^{nl}$
1-Forms (frame)	$\omega = \sum_{ij} \omega_{ij} b^{ij}$	$\langle \phi_k, \omega(v) \rangle_{L^2} = \sum_{ij} \omega_{ij} \left\langle \phi_k, b^{ij}(v) \right\rangle_{L^2}$

Operator	Tensor	Symmetries
Quadruple Product	$c^0_{ijkl} = \left\langle \phi_i\phi_j, \phi_k\phi_l \right\rangle_{L^2} = \sum_s c_{ijs}c_{skl}$	Fully symmetric
Product Energy	$c^p_{ijkl} = \left\langle \Delta^p(\phi_i\phi_j), \phi_k\phi_l \right\rangle_{L^2} = \sum_s \lambda^p_s c_{ijs}c_{skl}$	(1,2), (3,4), (1,3), (2,4)
Hodge Grammian	$G_{ijkl} = \left\langle b^{ij}, b^{kl} \right\rangle_{L^2_1} = \frac{1}{2}\left[(\lambda_j+\lambda_l)c^0_{ijkl} - c^1_{ijkl}\right]$	(1,3), (2,4)
Antisymmetric	$\hat{G}_{ijkl} = \langle \hat{b}^{ij}, \hat{b}^{kl} \rangle_{L^2_1} = G_{ijkl} + G_{jilk} - G_{jikl} - G_{ijlk}$	(1,3), (2,4)
Dirichlet Energy $E_{ijkl} = \left\langle b^{ij}, \Delta_1(b^{kl}) \right\rangle_{L^2_1}$	$E_{ijkl} = \frac{1}{4}\left[(\lambda_i+\lambda_j+\lambda_k+\lambda_l)(c^1_{iljk} - c^1_{ikjl})\right.$ $\left. +(\lambda_j+\lambda_l-\lambda_i-\lambda_k)c^1_{ijkl} + (c^2_{ijkl} + c^2_{ikjl} - c^2_{iljk})\right]$	(1,3), (2,4)
Antisymmetric	$\hat{E}_{ijkl} = \langle \hat{b}^{ij}, \Delta_1 \hat{b}^{kl} \rangle_{L^2_1}$ $= (\lambda_i+\lambda_j+\lambda_k+\lambda_l)(c^1_{iljk} - c^1_{ikjl}) + (c^2_{ikjl} - c^2_{iljk})$	(1,3), (2,4)
Sobolev H^1 Grammian	$G^1_{ijkl} = E_{ijkl} + G_{ijkl}, \quad \hat{G}^1_{ijkl} = \hat{E}_{ijkl} + \hat{G}_{ijkl}$	(1,3), (2,4)

Object	Symbolic	Spectral
Multiple Product	$c^0_I = \left\langle b^{i_0}\cdots b^{i_k}, 1 \right\rangle_H$	$c^0_I = \sum_s c_{i_0 i_1 s} c^0_{s i_2 \cdots i_k}$
Tensor Evaluation	$H^{\tilde{I}\tilde{J}} = (db^{i_1}\cdot db^{j_1})\cdots(db^{i_k}\cdot db^{j_k})$ $= \nabla b^{i_1} \otimes \cdots \otimes \nabla b^{i_k}(b^{j_1}, \ldots, b^{j_k})$	$\hat{H}^{\tilde{I}\tilde{J}}_l \equiv \langle H^{\tilde{I}\tilde{J}}, b^l \rangle_H$ $= \sum_{n=1}^{k^2} \prod_{s,r=1}^{k} g_{i_s j_r m_n} c_{l m_1 \cdots m_{k^2}}$
Tensor Product	$b_J = b^{j_0}\nabla b^{j_1} \otimes \cdots \otimes \nabla b^{j_k}$	$\left\langle b_J(b^{i_1}, \ldots, b^{i_k}), b^l \right\rangle = \sum_s \hat{H}^{\tilde{J}\tilde{I}}_s c_{s j_0 l}$
Frame Elements	$b^I = b^{i_0}db^{i_1}\wedge\cdots\wedge db^{i_k}$	$\left\langle b^I(b_J), b^l \right\rangle_H = \left\langle b^I\cdot b^J, b^l \right\rangle_H$
Riemannian Metric	$b^I \cdot b^J = b^{i_0}b^{j_0}\det([db^{i_q}\cdot db^{j_n}])$	$\left\langle b^I\cdot b^J, b_l \right\rangle_H = \sum_s \sum_{\sigma\in S_k} \operatorname{sgn}(\sigma) c_{s i_0 j_0 l} \hat{H}^{\tilde{I}\sigma(\tilde{J})}_s$
Hodge Grammian	$G_{IJ} = \left\langle b^I, b^J \right\rangle_{H_k} = \left\langle b^I\cdot b^J, 1 \right\rangle_H$	$\sum_s \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) c_{s i_0 j_0} \hat{H}^{\tilde{I}\sigma(\tilde{J})}_s$
d-Energy	$E^d_{IJ} = \left\langle db^I, db^J \right\rangle_{H_{k+1}}$	$\left\langle db^I\cdot db^J, 1 \right\rangle_{H_{k+1}} = \hat{H}^{IJ}_0$

Back to Basis

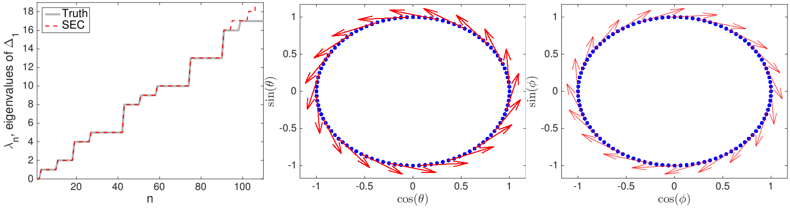
- We need the frame representation to build the 1-Laplacian

$$\Delta_1 = d\delta + \delta d$$

- Once we have Δ_1, the eigenfields form the smoothest possible basis for vector fields
- Can use to smooth vector fields and represent operators

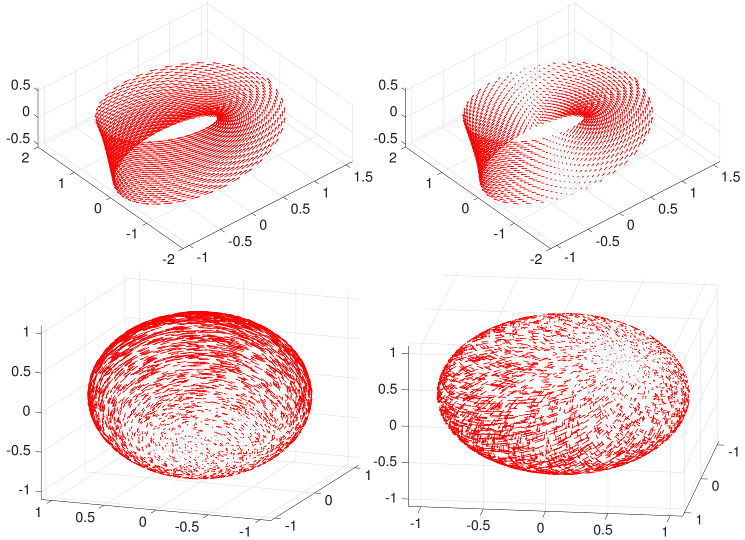
Numerical Verification on Flat Torus

Captures the true spectrum of the Hodge Laplacian.

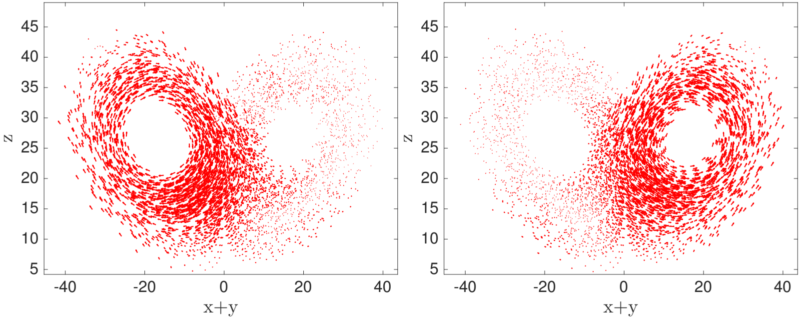


Harmonic forms correspond to unique homology classes.

SMOOTHEST VECTOR FIELDS ON THE MANIFOLD



SEC IS APPLICABLE TO ANY DATA SET



Matlab Code: http://math.gmu.edu/˜berry/

Applying the SEC to Dynamical Systems

- Smooth/Denoise vector fields using SEC basis
- Compute Lyapunov vector fields in the SEC basis
- Next Step: Hodge decomposition

$$v = \nabla U + \delta A + v^{\perp}$$

- U is a potential, A is a tensor field, and $\Delta_1 v^{\perp} = 0$

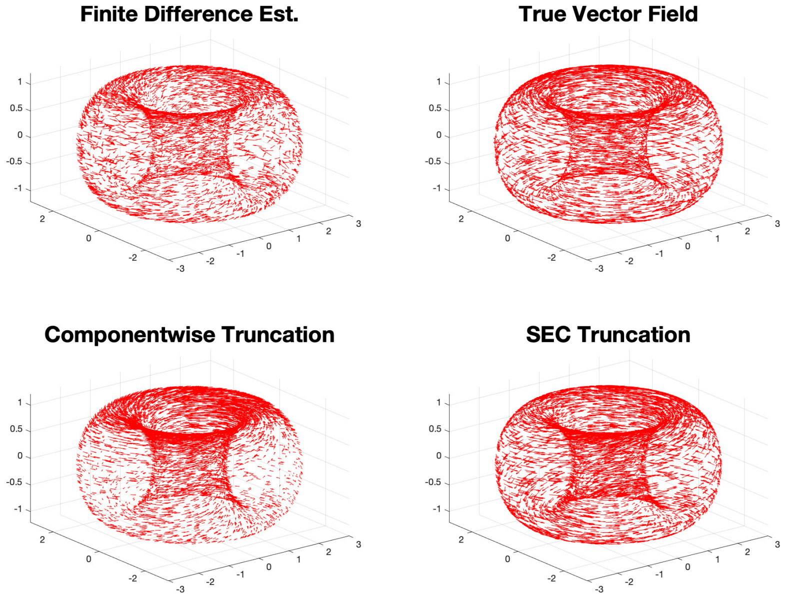
DECOMPOSING SDE COMPONENTS

- Given a realization of an SDE on a manifold:

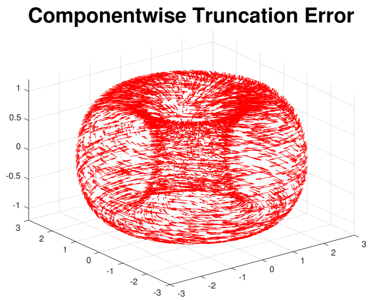
$$dx = f(x)\,dt + B(x)\,dW_t$$

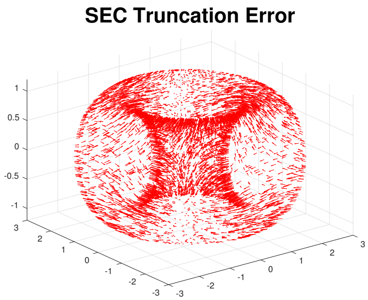
- Want to extract the deterministic component, $f(x)$
- Finite differences $x(t+\tau) - x(t) \approx f(x(t))$ but noisy
- Can smooth component functions using DM basis
- Better to smooth with SEC eigenvectorfields

Decomposing SDE Components



Decomposing SDE Components





For more information: http://math.gmu.edu/˜berry/

Building the basis

- B. and Giannakis, *Spectral Exterior Calclulus.*
- Coifman and Lafon, *Diffusion maps.*
- B. and Harlim, *Variable Bandwidth Diffusion Kernels.*
- B. and Sauer, *Local Kernels and Geometric Structure of Data.*

Nonparametric forecast

- B., Giannakis, and Harlim, *Nonparametric forecasting of low-dimensional dynamical systems.*
- B. and Harlim, *Forecasting Turbulent Modes with Nonparametric Diffusion Models.*

Semiparametric forecast

- B. and Harlim, *Semiparametric forecasting and filtering: correcting low-dimensional model error in parametric models.*