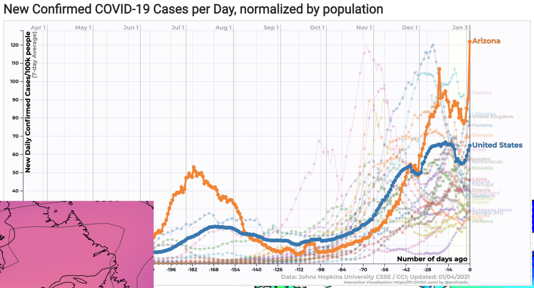
A Bayesian Approach for Data-Driven Dynamic Equation Discovery

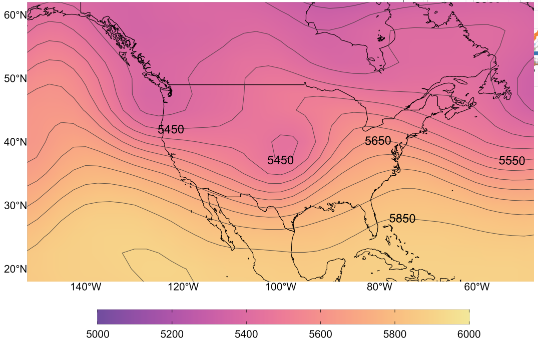
Joshua S. North, Christopher K. Wikle, and Erin M. Schliep

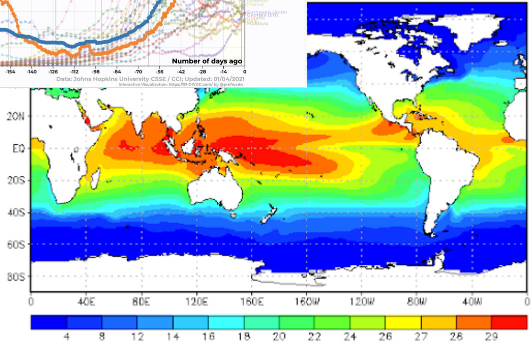
University of Missouri Columbia
Department of Statistics

August 10, 2022

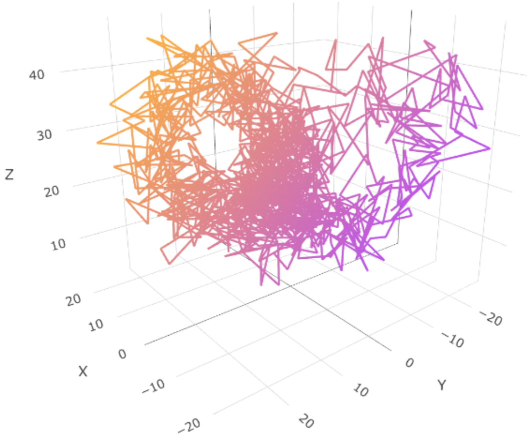
Motivation - Dynamics







Discover the governing equation(s) that define the dynamic system



$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

Who else has done this?

- **Symbolic Regression** (Bongard and Lipson, 2007; Schmidt and Lipson, 2009; Maslyaev et al., 2019)
  - Search over mathematical expressions (e.g., $+, -, \times$) to find the optimal model; very flexible; computationally expensive; doesn't scale well
- **Sparse Regression** (Brunton et al., 2016; Rudy et al., 2017; Champion et al., 2020)
  - Sparse Identification of Nonlinear Dynamics (SINDy): Three Steps; Requires a *library* of potential functions; very fast
  - Stochastic processes (Boninsegna et al., 2018); Numerical improvement (Schaeffer, 2017; Schaeffer et al., 2018); Python package (*pySINDy*; de Silva et al., 2020)
- **Deep Models**
  - Sparse Regression (Lagergren et al., 2020; Both et al., 2021)
  - Symbolic Regression (Long et al., 2019; Atkinson et al., 2019; Xu et al., 2021)

Statistical Approaches?

- **Bayesian** (Zhang and Lin, 2018; Niven et al., 2020; Yang et al., 2020; Hirsh et al., 2021)
  - Prior on parameters
  - Bayesian differentiable programming
- **Bootstrap** (Fasel et al., 2021)
  - Observations
  - Library terms

- Requires derivatives (e.g., numerical differentiation)
- Observational uncertainty?
- Uncertainty in the dynamic system (e.g., derivatives)

Our Approach

- Bayesian hierarchical modeling: data, process, parameter models
  - **Consider dynamic process to be latent**
  - Accounts for measurement uncertainty and missing data in a probabilistic manner
  - Accounts for process dependence
  - Dynamic system as a random process
- Important components
  - Compute derivatives analytically via basis function representation
  - Feature library
  - Inclusion probability

Notation

Tensor

- $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ be a tensor of order N
- n-mode matricization is $\mathbf{X}_{(n)}$: arrange the mode-n fibers to be columns
  - For example, if $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$, then $\mathbf{Y}_{(3)} \in \mathbb{R}^{K \times (I \times J)}$
  - Generally only concerned with the mode-3 matricization of a tensor and use $\mathbf{Y}$ in place of $\mathbf{Y}_{(3)}$
- The n-mode product of the tensor $\mathcal{X}$ and matrix $\mathbf{B} \in \mathbb{R}^{I_n \times J}$ is $\mathcal{X} \times_n \mathbf{B}$ and is of size $I_1 \times I_2 \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$
  - $\mathcal{Z} = \mathcal{X} \times_n \mathbf{B} \Leftrightarrow \mathbf{Z}_{(n)} = \mathbf{B}\mathbf{X}_{(n)}$

Derivatives

- Denote partial derivatives using a subscript (e.g., $\frac{\partial u}{\partial t} = u_t$)
- Denote the ith order of a derivative generally as $\frac{\partial^{(i)} u}{\partial t^{(i)}} = u_{t^{(i)}}$

Dynamic System

Consider the dynamic system describing the evolution of a continuous process $\{\mathbf{u}(\mathbf{s}, t) : \mathbf{s} \in D_s, t \in D_t\}$:

$$\mathbf{u}_{t^{(J)}}(\mathbf{s}, t) = M\left(\mathbf{u}(\mathbf{s}, t), \mathbf{u}_x(\mathbf{s}, t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s}, t), ..., \mathbf{u}_{t^{(J-1)}}(\mathbf{s}, t), \boldsymbol{\omega}(\mathbf{s}, t)\right), \ \mathbf{u}(\mathbf{s}, t) \in \mathbb{R}^N$$

Re-parameterizing and accounting for potential stochastic forcing:

$$\mathbf{u}_{t^{(J)}}(\mathbf{s}, t) = \mathbf{Mf}\left(\mathbf{u}(\mathbf{s}, t), \mathbf{u}_x(\mathbf{s}, t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s}, t), ..., \boldsymbol{\omega}(\mathbf{s}, t)\right) + \boldsymbol{\eta}(\mathbf{s}, t),$$

- $\mathbf{M}$ is a $N \times D$ *sparse* matrix of coefficients
- $\mathbf{f}(\cdot)$ is a length-D vector-valued nonlinear transformation function
- $\boldsymbol{\eta}(\mathbf{s}, t) \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_U)$ is a mean zero Gaussian process with variance/covariance matrix $\boldsymbol{\Sigma}_U$

Tensor Formulation of the Dynamic System

$$\mathcal{U}_{t(J)} = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}$$

- Tensor of the dynamic process:
  $\mathcal{U} = \{u(\mathbf{s}, t, n) : \mathbf{s} \in D_s, t = 1, ..., T, n = 1, ..., N\} \in \mathbb{R}^{S \times T \times N}$
- Tensor of the function: $\mathcal{F} \in \mathbb{R}^{S \times T \times D}$ (note, still a function of the state process $\mathcal{U}$)
- Space-time uncertainty tensor: $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^{S \times T \times N}$

Basis Function Representation

$$\mathcal{U} \approx \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a(p,q,r)\boldsymbol{\psi}(p) \circ \boldsymbol{\phi}(q) \circ \boldsymbol{\theta}(r)$$
$$= \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} := [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!]$$

- Basis Coefficients: $\mathcal{A} \in \mathbb{R}^{P \times Q \times R}$
- Spatial Basis Functions (fixed, known, differentiable): $\boldsymbol{\Psi} \in \mathbb{R}^{S \times P}$
- Temporal Basis Functions (fixed, known, differentiable): $\boldsymbol{\Phi} \in \mathbb{R}^{T \times Q}$
- Component Basis Functions (fixed, known): $\boldsymbol{\Theta} \in \mathbb{R}^{N \times R}$

$$\mathcal{U}_t = \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi}_t \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_t, \boldsymbol{\Theta}]\!]$$
$$\mathcal{U}_x = \mathcal{A} \times_1 \boldsymbol{\Psi}_x \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_x, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!]$$
$$\mathcal{U}_{xyt} = \mathcal{A} \times_1 \boldsymbol{\Psi}_{xy} \times_2 \boldsymbol{\Phi}_t \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_{xy}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!]$$

Basis Function Representation

Proposition 1

The mode-3 decomposition of $[\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_{t(J)}, \boldsymbol{\Theta}]\!] = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}$ where $\boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$ in space and time at location $\mathbf{s}$ and time t is

$$\boldsymbol{\Theta}\mathbf{A}(\phi_{t(J)}(t) \otimes \psi(\mathbf{s}))' = \mathbf{M}\mathbf{f}(\mathbf{A}, \psi(\mathbf{s}), \psi_x(\mathbf{s}), ..., \phi_{t(0)}(t), ..., \boldsymbol{\omega}(\mathbf{s}, t)) + \boldsymbol{\eta}(\mathbf{s}, t),$$

where $\mathbf{A}$ is a $R \times PQ$ matrix of basis coefficients, $\psi(\mathbf{s})$ is a length-P vector of spatial basis functions, $\phi(t)$ is a length-Q vector of temporal basis functions, and $\boldsymbol{\Theta}$ is a $N \times R$ matrix of component basis functions.

Space-Time Response

What about equations with a spatio-temporal derivative of $\mathbf{u}$?

$$\nabla^2 \mathbf{u}_t(\mathbf{s}, t) = \mathbf{u}_{xxt}(\mathbf{s}, t) + \mathbf{u}_{yyt}(\mathbf{s}, t)$$

General PDE

$$g(\mathbf{u}_{t^{(J)}}(\mathbf{s}, t)) = M\left(\mathbf{u}(\mathbf{s}, t), \mathbf{u}_x(\mathbf{s}, t), \mathbf{u}_y(\mathbf{s}, t), ..., \mathbf{u}_{t^{(1)}}(\mathbf{s}, t), ..., \boldsymbol{\omega}(\mathbf{s}, t)\right)$$

- $g(\cdot)$ is some linear differential operator
- Fluid dynamics

General Basis Function Representation

Proposition 2

Let $g(\cdot)$ be a linear differential operator. The basis formulation of a PDE with a space-time response $g(\mathbf{u}_{t^{(J)}}(\mathbf{s}, t))$ is

$$\boldsymbol{\Theta}\mathbf{A}(\phi_{t^{(J)}}(t) \otimes g(\psi(\mathbf{s})))'.$$

The general form of a PDE with a spatio-temporal response:

$$\boldsymbol{\Theta}\mathbf{A}(\phi_{t^{(J)}}(t) \otimes g(\psi(\mathbf{s})))' = \mathbf{M}\mathbf{f}(\mathbf{A}, \psi(\mathbf{s}), \psi_x(\mathbf{s}), \phi_{t^{(0)}}(t), \dots) + \boldsymbol{\eta}(\mathbf{s}, t), \quad \boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$$

$$\nabla^2 \mathbf{u}_t(\mathbf{s}, t) = \mathbf{u}_{xxt}(\mathbf{s}, t) + \mathbf{u}_{yyt}(\mathbf{s}, t) \Rightarrow \boldsymbol{\Theta}\mathbf{A}(\phi_{t^{(1)}}(t) \otimes (\psi_{xx}(\mathbf{s}) + \psi_{yy}(\mathbf{s})))'$$

Hierarchical Model

For location $\mathbf{s}$ and time t, the general model is

$$\begin{aligned}\mathbf{v}(\mathbf{s}, t) &= \mathbf{H}(\mathbf{s}, t)\mathbf{\Theta A}(\phi_{t^{(0)}}(t) \otimes \psi(\mathbf{s}))' + \epsilon(\mathbf{s}, t) \\ \mathbf{\Theta A}(\phi_{t^{(J)}}(t) \otimes g(\psi(\mathbf{s})))' &= \mathbf{Mf}(\mathbf{A}, \psi(\mathbf{s}), \psi_x(\mathbf{s}), \phi_{t^{(0)}}(t), \ldots) + \boldsymbol{\eta}(\mathbf{s}, t)\end{aligned}$$

- $\mathbf{v}(\mathbf{s}, t) \in \mathbb{R}^{L(\mathbf{s},t)}$ is the observed process
- $\mathbf{H}(\mathbf{s}, t) \in \mathbb{R}^{L(\mathbf{s},t)\times N}$ is the incidence matrix
- $\mathbf{M}$ is the $N \times D$ coefficient matrix - Spike and Slab (George et al., 1993)
- $\mathbf{A}$ is the matrix (tensor) of basis coefficients - Elastic Net (Li and Lin, 2010)
- $\epsilon(\mathbf{s}, t) \overset{indep.}{\sim} N_{L(\mathbf{s},t)}(\mathbf{0}, \mathbf{\Sigma}_V(\mathbf{s}, t))$ is the measurement error
- $\boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \mathbf{\Sigma}_U)$ is the process error

Simulations and Examples - PDE

Simulations

1. Burgers' Equation
2. Reaction-Diffusion Equation

Examples

1. Barotropic Vorticity Equation

$$\mathbf{v}(\mathbf{s}, t) = \mathbf{u}(\mathbf{s}, t) + \zeta \boldsymbol{\epsilon}(\mathbf{s}, t), \quad \boldsymbol{\epsilon}(\mathbf{s}, t) \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_N)$$

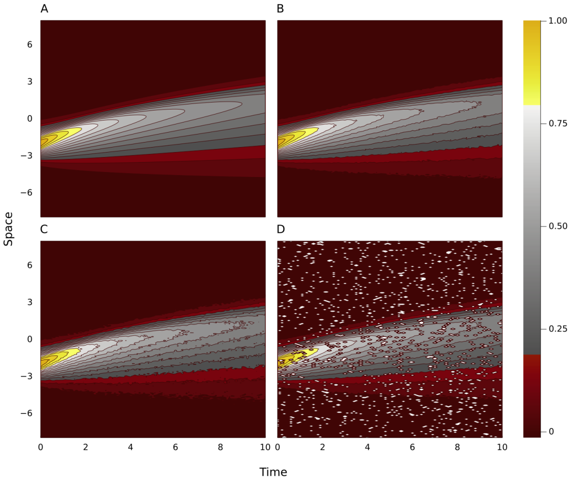
- $\mathbf{u}(\mathbf{s}, t)$ is the simulated data
- σ is the standard deviation of $\mathbf{u}(\mathbf{s}, t)$
- $\zeta \in [0, 1]$ is the percent of noise

Burgers' Equation

$$u_t(x,t) = -u(x,t)u_x(x,t) + \nu u_{xx}(x,t)$$
$$u(x,0) = exp\{-(x+2)^2\}$$

- $\nu = 0.1$ is the viscocity
- (A) no measurement noise
  (B) 2% measurement noise
  (C) 5% measurement noise
  (D) 2% measurement noise with 5% of data missing at random

$$[u, u^2, u^3, u_x, uu_x, u^2u_x, u^3u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u^3u_{xx}, u_{xxx}, uu_{xxx}, u^2u_{xxx}, u^3u_{xxx}]$$



Burgers' Equation

Noise	Missing Data	Statistic	Discovered Equation
0%	0%	Mean	$u_t = -\mathbf{0.994}uu_x + \mathbf{0.098}u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.092u_{xx}$
		Upper HPD	$u_t = -0.964uu_x + 0.103u_{xx}$
2%	0%	Mean	$u_t = -\mathbf{0.990}uu_x + \mathbf{0.096}u_{xx}$
		Lower HPD	$u_t = -1.033uu_x + 0.086u_{xx}$
		Upper HPD	$u_t = -0.954uu_x + 0.103u_{xx}$
5%	0%	Mean	$u_t = -\mathbf{0.981}uu_x + 0.094u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.087u_{xx}$
		Upper HPD	$u_t = -0.951uu_x + 0.099u_{xx}$
2%	5%	Mean	$u_t = -\mathbf{0.957}uu_x + 0.087u_{xx}$
		Lower HPD	$u_t = -1.003uu_x + 0.078u_{xx}$
		Upper HPD	$u_t = -0.931uu_x + 0.095u_{xx}$

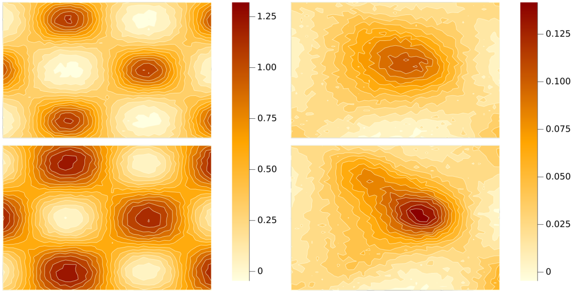
$$u_t = -uu_x + 0.1u_{xx}$$

Reaction-Diffusion Equation

$$u_t = D_u u_{xx} + D_u u_{yy} + \gamma_0 u - \frac{\gamma_0}{\gamma_1} u^2 - \beta uv \qquad u(\mathbf{s}, 0) = \exp\{cos(2\pi x/15) sin(2\pi y/15)\}$$

$$v_t = D_v v_{xx} + D_v v_{yy} + \mu uv - \eta v \qquad v(\mathbf{s}, 0) = 0.1 \exp\{cos(2\pi y/30) sin(2\pi x/30 - 5)\}$$

- $D = 0.1$ is the diffusivity constant
- Predator-prey reaction component
- Prey (left) and predator (right) with 5% measurement noise



$$[u, u^2, u^3, v, v^2, v^3, uv, u^2v, uv^2, uu_x, uu_y, vv_x,$$
$$vv_y, u_x, u_y, u_{xx}, u_{yy}, u_{xy}, v_x, v_y, v_{xx}, v_{yy}, v_{xy}]$$

Reaction-Diffusion Equation

Noise	Statistic	Recovered Equation
5%	Mean	$u_t = \mathbf{0.401}u - 0.269u^2 - \mathbf{0.493}uv + 0.090u_{xx} + \mathbf{0.099}u_{yy}$
	Lower HPD	$u_t = 0.400u - 0.270u^2 - 0.504uv + 0.085u_{xx} + 0.091u_{yy}$
	Upper HPD	$u_t = 0.403u - 0.267u^2 - 0.487uv + 0.093u_{xx} + 0.102u_{yy}$
5%	Mean	$v_t = -\mathbf{0.101}v + \mathbf{0.300}uv + 0.092v_{xx} + \mathbf{0.106}v_{yy}$
	Lower HPD	$v_t = -0.103v + 0.297uv + 0.086v_{xx} + 0.097v_{yy}$
	Upper HPD	$v_t = -0.099v + 0.302uv + 0.095v_{xx} + 0.151v_{yy}$

$$u_t = 0.4u - 0.266u^2 - 0.5uv + 0.1u_{xx} + 0.1u_{yy}$$
$$v_t = -0.1v + 0.3uv + 0.1v_{xx} + 0.1v_{yy}$$

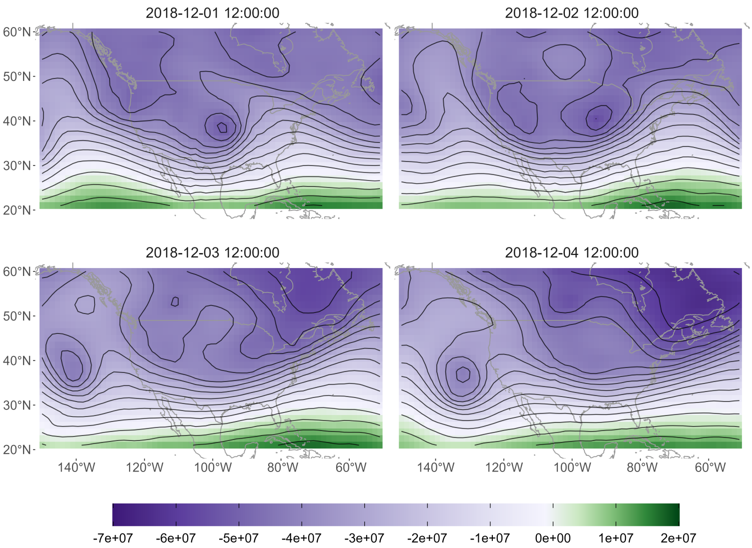
Barotropic Vorticity Equation

$$\xi_t(\mathbf{s}, t) = -\mathbf{v}(\mathbf{s}, t) \cdot \nabla(\xi(\mathbf{s}, t) + f(\phi(\mathbf{s})))$$

- $\xi = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \nabla^2 \psi$ is the relative vorticity
- $\mathbf{v} = (u, v)$ is the non-divergent horizontal wind
- $u = -\frac{\partial \psi}{\partial y}$ is the Zonal wind, $v = \frac{\partial \psi}{\partial x}$ is the Meridional wind
- $f = 2\Omega sin(\phi)$ is the Coriolis parameter with $\Omega = 7.292 \times 10^{-5}$ rad s^{-1} the angular speed of rotation of the Earth
- ψ is the streamfunction

$$\nabla^2 \psi_t = \psi_{xxt} + \psi_{yyt} = \psi_y \psi_{xxx} + \psi_y \psi_{xyy} - \psi_x \psi_{xxy} - \psi_x \psi_{yyy} - \psi_x f_y$$

Barotropic Vorticity Equation



Streamfunction is a computed quantity based on the observed geopotential height, Φ

$$\psi(\mathbf{s}, t) = \Phi^*(\mathbf{s}, t)/f(\phi(\mathbf{s}))$$

$$\Phi^*(\mathbf{s}, t) = \Phi(\mathbf{s}, t) - \int_D \Phi(\mathbf{s}, t)$$

Barotropic Vorticity Equation

$$[\psi, \psi_x, \psi_{xx}, \psi_y, \psi_{yy}, \psi_{xy}, \psi_x\psi_{xxx}, \psi_y\psi_{xxx}, \psi_x\psi_{yyy}, \psi_y\psi_{yyy},$$
$$\psi_x\psi_{xxy}, \psi_y\psi_{xxy}, \psi_x\psi_{xyy}, \psi_y\psi_{xyy}, \psi_x f_y, \psi_y f_y]$$

	Discovered Equation
Mean	$\nabla^2\psi_t = \mathbf{0.289}\psi_y\psi_{xxx} + \mathbf{0.277}\psi_y\psi_{xyy} - \mathbf{0.280}\psi_x\psi_{xxy} - \mathbf{0.185}\psi_x\psi_{yyy} - 6.354\psi_x f_y$
Lower HPD	$\nabla^2\psi_t = 0.235\psi_y\psi_{xxx} + 0.267\psi_y\psi_{xyy} - 0.343\psi_x\psi_{xxy} - 0.215\psi_x\psi_{yyy} - 7.570\psi_x f_y$
Upper HPD	$\nabla^2\psi_t = 0.317\psi_y\psi_{xxx} + 0.286\psi_y\psi_{xyy} - 0.223\psi_x\psi_{xxy} - 0.160\psi_x\psi_{yyy} + 1.491\psi_x f_y$

$$\nabla^2\psi_t = \psi_y\psi_{xxx} + \psi_y\psi_{xyy} - \psi_x\psi_{xxy} - \psi_x\psi_{yyy} - \psi_x f_y$$

Contributions

- Bayesian hierarchical method to learn complex nonlinear dynamic equations using a data-driven approach
- Robust to measurement noise and missing data
- Provide uncertainty quantification and inclusion probabilities to the terms in the library
- Bypass the need for numerical differentiation

References I

Atkinson, S., Subber, W., Wang, L., Khan, G., Hawi, P., and Ghanem, R. (2019). Data-driven discovery of free-form governing differential equations. *arXiv preprint arXiv:1910.05117*, pages 1–7.

Bongard, J. and Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948.

Boninsegna, L., Nüske, F., and Clementi, C. (2018). Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723.

Both, G.-J., Choudhury, S., Sens, P., and Kusters, R. (2021). DeepMoD: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428(1):109985.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937.

References II

Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L., and Kutz, J. N. (2020). A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271.

de Silva, B., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J., and Brunton, S. (2020). PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5(49):2104.

Fasel, U., Kutz, J. N., Brunton, B. W., and Brunton, S. L. (2021). Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *arXiv preprint arXiv:2111.10992*, pages 1–18.

George, E. I., Mcculloch, R. E., George, E. I., and Mcculloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

Hirsh, S. M., Barajas-Solano, D. A., and Kutz, J. N. (2021). Sparsifying priors for Bayesian uncertainty quantification in model discovery. *arXiv preprint arXiv:2107.02107*, pages 1–22.

References III

Lagergren, J. H., Nardini, J. T., Michael Lavigne, G., Rutter, E. M., and Flores, K. B. (2020). Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2234):20190800.

Li, Q. and Lin, N. (2010). The Bayesian elastic net. *Bayesian Analysis*, 5(1):151–170.

Long, Z., Lu, Y., and Dong, B. (2019). PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925.

Maslyaev, M., Hvatov, A., and Kalyuzhnaya, A. (2019). Data-driven partial derivative equations discovery with evolutionary approach. In *Computational Science – ICCS 2019*, pages 635–641. Springer International Publishing.

Niven, R., Mohammad-Djafari, A., Cordier, L., Abel, M., and Quade, M. (2020). Bayesian identification of dynamical systems. *Proceedings*, 33(1):33.

References IV

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614.

Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446.

Schaeffer, H., Tran, G., and Ward, R. (2018). Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295.

Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85.

Xu, H., Zhang, D., and Zeng, J. (2021). Deep-learning of parametric partial differential equations from sparse and noisy data. *Physics of Fluids*, 33(3):037132.

Yang, Y., Aziz Bhouri, M., and Perdikaris, P. (2020). Bayesian differential programming for robust systems identification under uncertainty. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2243):20200290.

References V

Zhang, S. and Lin, G. (2018). Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217):20180305.