SINDy – Sparse Identification of Nonlinear Dynamics

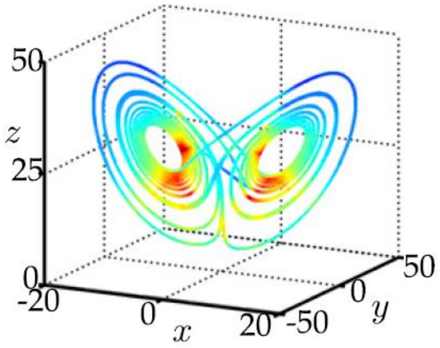
CWI Autumn School - Scientific Machine Learning and Dynamical Systems

Urban Fasel

Imperial College London

SINDy

Data → *Dynamics (assumptions)* → *Model structure & coefficients*



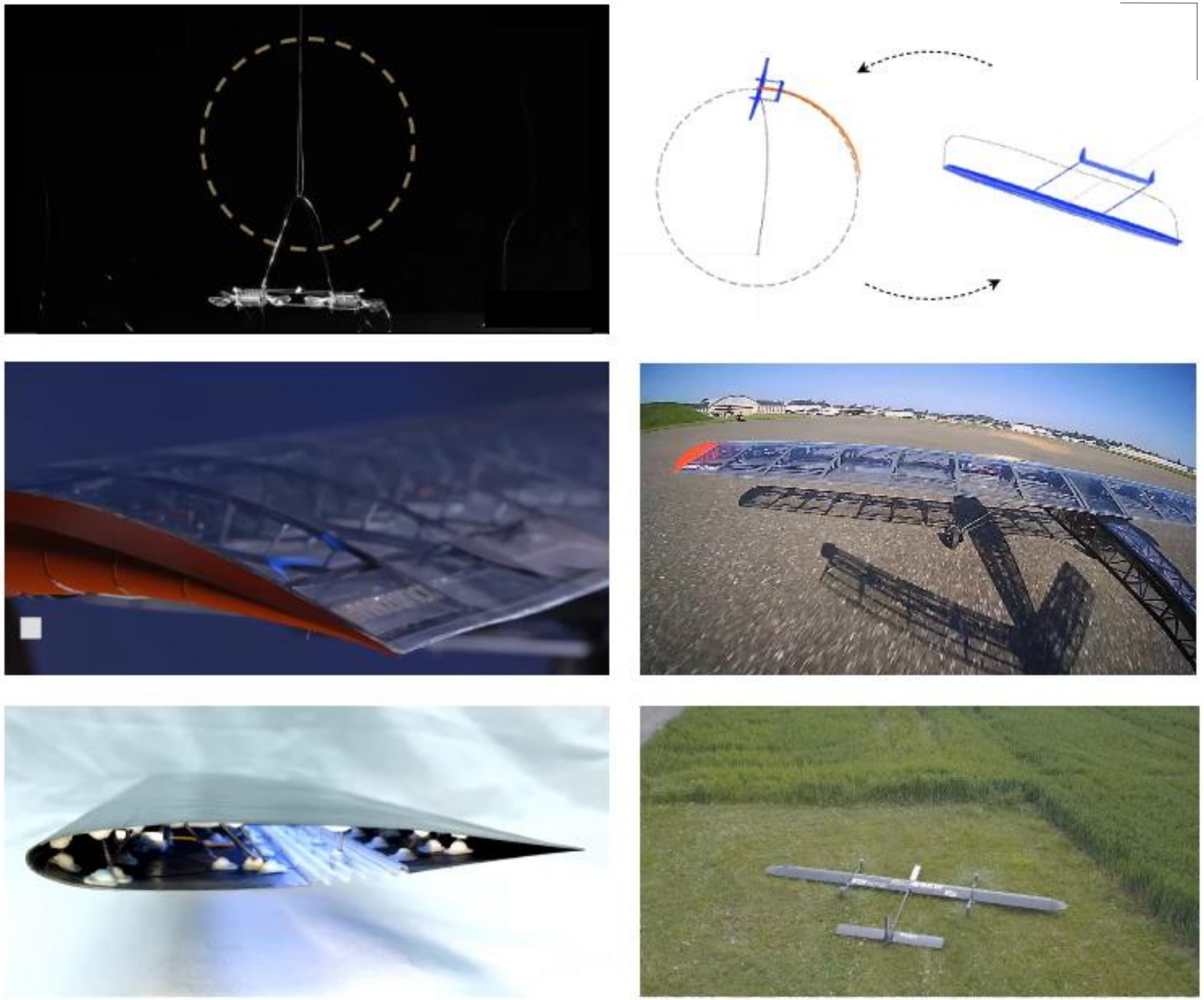
$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

$$\dot{x} = \sigma(y - x)$$

$$\dot{y} = x(\rho - z) - y$$

$$\dot{z} = xy - \beta z$$

My work



(Flexible) flight systems

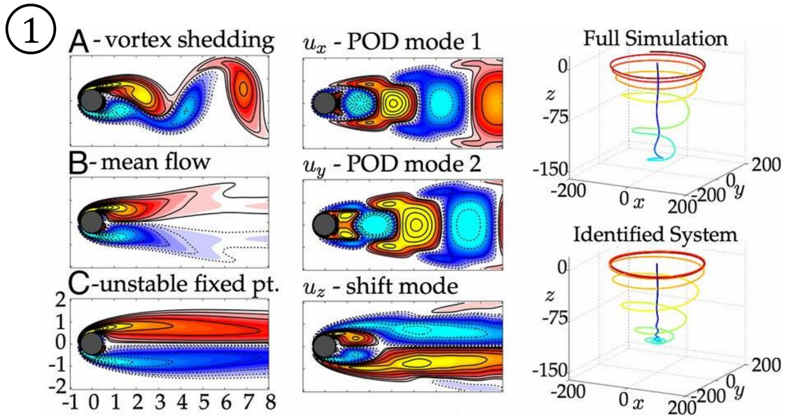
1. Flapping wing MAV
2. Renewable energy systems
3. Composite additive manufacturing morphing wing drones
4. Morphing wings
5. Airborne wind energy

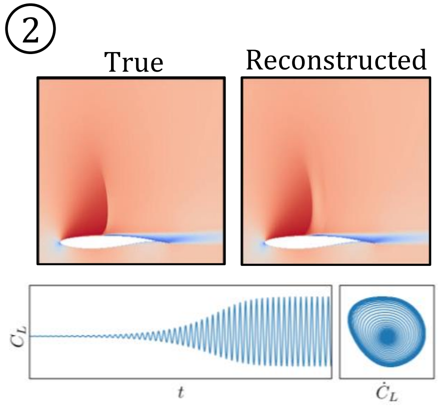
Methods

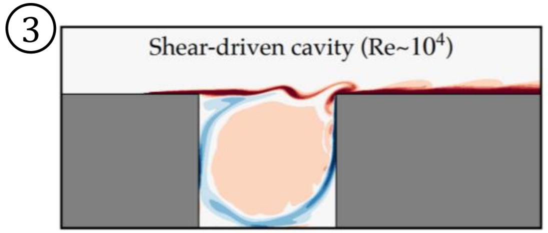
1. Co-design optimization
2. **Data driven modeling & control**
   - DMDc, SINDyC, E-SINDy

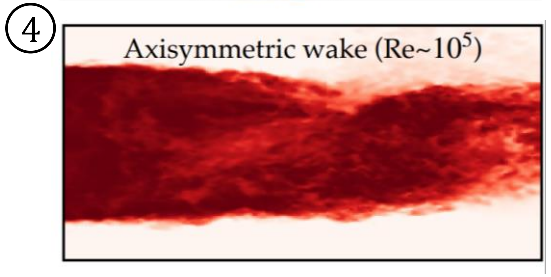
SINDy – applications

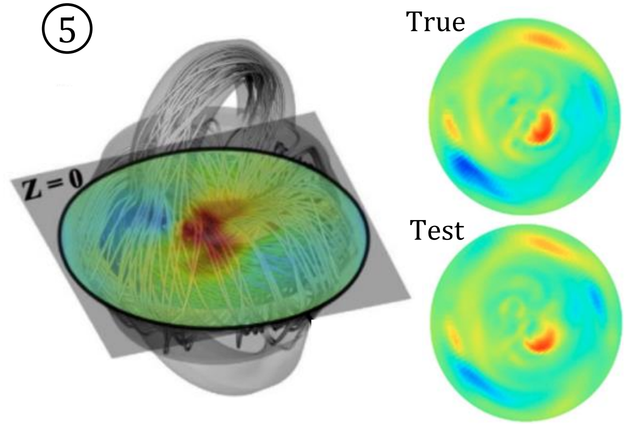
1. **[Vortex shedding past a cylinder]()**
   - *Time history of POD coefficients:*
     - $\dot{x} = \mu x - \omega y + Axz$
     - $\dot{y} = \omega x + \mu y + Ayz$
     - $\dot{z} = -\lambda(z - x^2 - y^2)$
2. **[Shock wave dynamics 2D airfoil transonic buffet conditions]()**
   - *Parametric c_L model for different α*
     - $c_L(r,\phi) = c_0 + c_1 r + c_2 r\cos(\varphi) + c_3 r\sin(\varphi) + c_4 r^2\cos(2\varphi) + c_5 r^2\sin(2\varphi)$
3. **[Cavity flow]()**
   - *Coefficients of 2 active DMD modes*
     - $\dot{\alpha}_1 = \lambda_1\alpha_1 - \mu_1\alpha_1|\alpha_1|^2$
     - $\dot{\alpha}_5 = \lambda_5\alpha_5 - \mu_5\alpha_5|\alpha_5|^2$
4. **[Experimental measurements turbulent bluff body wake]()**
   - *Statistical behavior of the CoP (learning drift and diffusion of SDE)*
     - $\dot{r} = \lambda r - \mu r^3 + \frac{\sigma^2}{2r} + (\sigma_0 + \sigma_1^2)w(t)$
5. **[Plasma dynamics (magnetohydrodynamics): 3D spheromak sim]()**
   - *Dominant POD coefficient dynamics*
     - $\dot{a}_1 = 0.091\mathrm{a}_2 + 0.009\mathrm{a}_5$
     - $\dot{a}_2 = -0.091\mathrm{a}_1 + 0.008\mathrm{a}_5 - 0.011\mathrm{a}_6$
     - …
6. **[Experimental weakly turbulent fluid flow in a thin electrolyte layer]()**
   - *Measured velocity field, identify PDE: form similar to N-S*
     - $\partial_t \mathbf{u} = c_1(\mathbf{u}\cdot\nabla)\mathbf{u} + c_2\nabla^2\mathbf{u} + c_3\mathbf{u} - \rho^{-1}\nabla p + \rho^{-1}\mathbf{f}$
7. **[Turbulent 3D channel flow ($Re = 1000$) Johns Hopkins database]()**
   - *Identify PDEs: N-S, continuity equation, boundary conditions*
     - $\partial_t \mathbf{u} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - 0.995\nabla p + 4.93\cdot 10^{-5}\nabla^2\mathbf{u}$

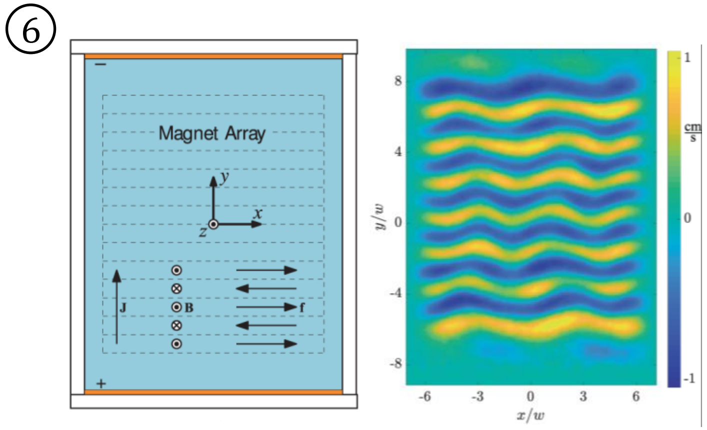


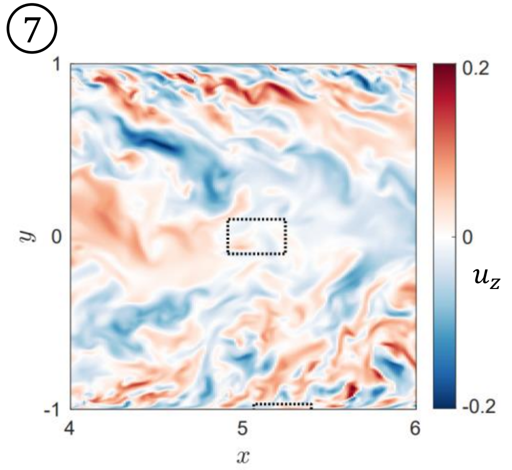












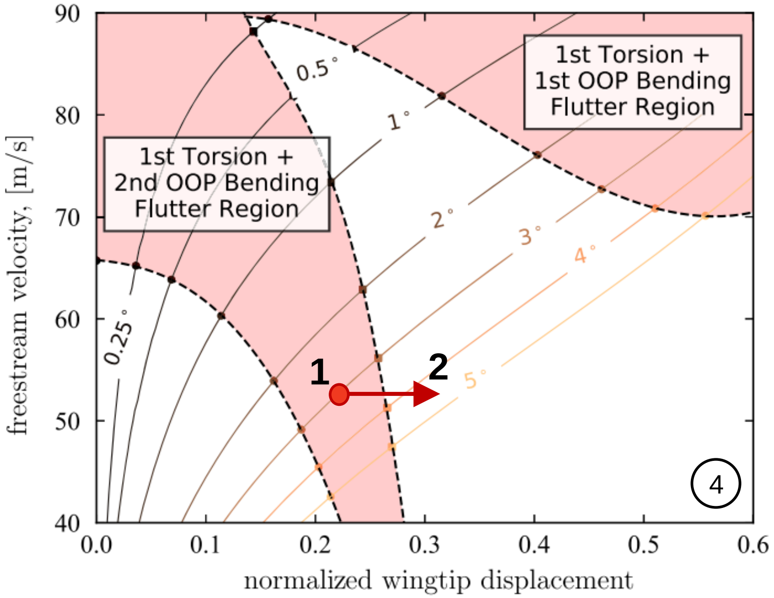
SINDy MPC – next generation transport aircraft control

AR20+ workshop last week at Imperial College London

Future aircraft: high aspect ratio wings

Wind tunnel flutter test: Pazy wing

Wing flutter **instability regions**



Nonlinear model predictive flutter control using SINDy models [5]

1. **Fast flutter suppression**: deflect flap synchronously to flutter vibration to counteract its effect
2. **Slow deflection flutter control**: move wing to different stability region

[1] AR20+ workshop (2023) https://cassyni.com/s/ar20plus
[2] Airbus ZEROe Concept https://www.airbus.com/en/innovation/low-carbon-aviation/hydrogen/zeroe
[3] Aeroelasticity Lab D Raveh (2023) Pazy wing flutter.
[4] M Artola, N Goizueta, A Wynn, R Palacios (2021) Aeroelastic Control and Estimation with a Minimal Nonlinear Modal Description.
[5] A Wynn, M Artola, R Palacios (2022) Nonlinear optimal control for gust load alleviation with a physics-constrained data-driven internal model.

Tutorial outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- ODEs and PDEs
- MATLAB and Python (PySINDy) examples
- SINDy limitations

"Vanilla" SINDy

Part 2: SINDy with control & parametric models

SINDy extensions

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Part 5: Your PhD project data sets

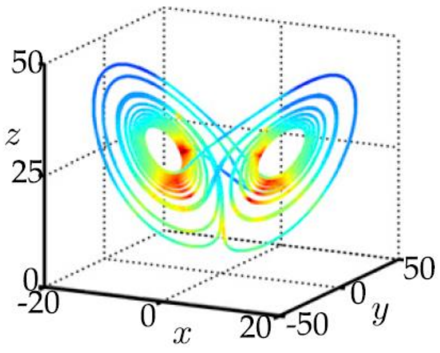
Applications

Part 1

- SINDy: Learning ODEs and PDEs from time series data
- **ODEs**
  1. Collect time series data & compute time derivatives
  2. Build library of nonlinear terms
  3. Sparse regression
- ODE MATLAB & Python examples
- **PDEs**
- Challenges & limitations of “Vanilla” SINDy

Learning ODEs and PDEs from data – SINDy

Data → **Dynamics (assumptions)** → **Model structure & coefficients**

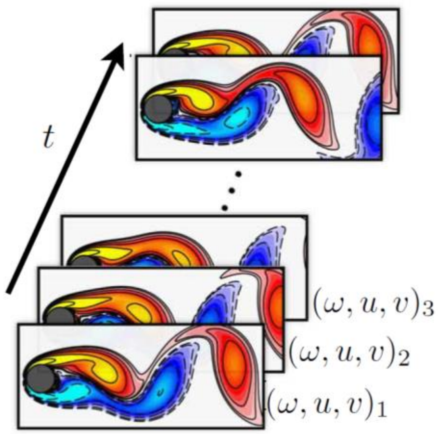


$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

$$\dot{x} = \sigma(y - x)$$

$$\dot{y} = x(\rho - z) - y$$

$$\dot{z} = xy - \beta z$$

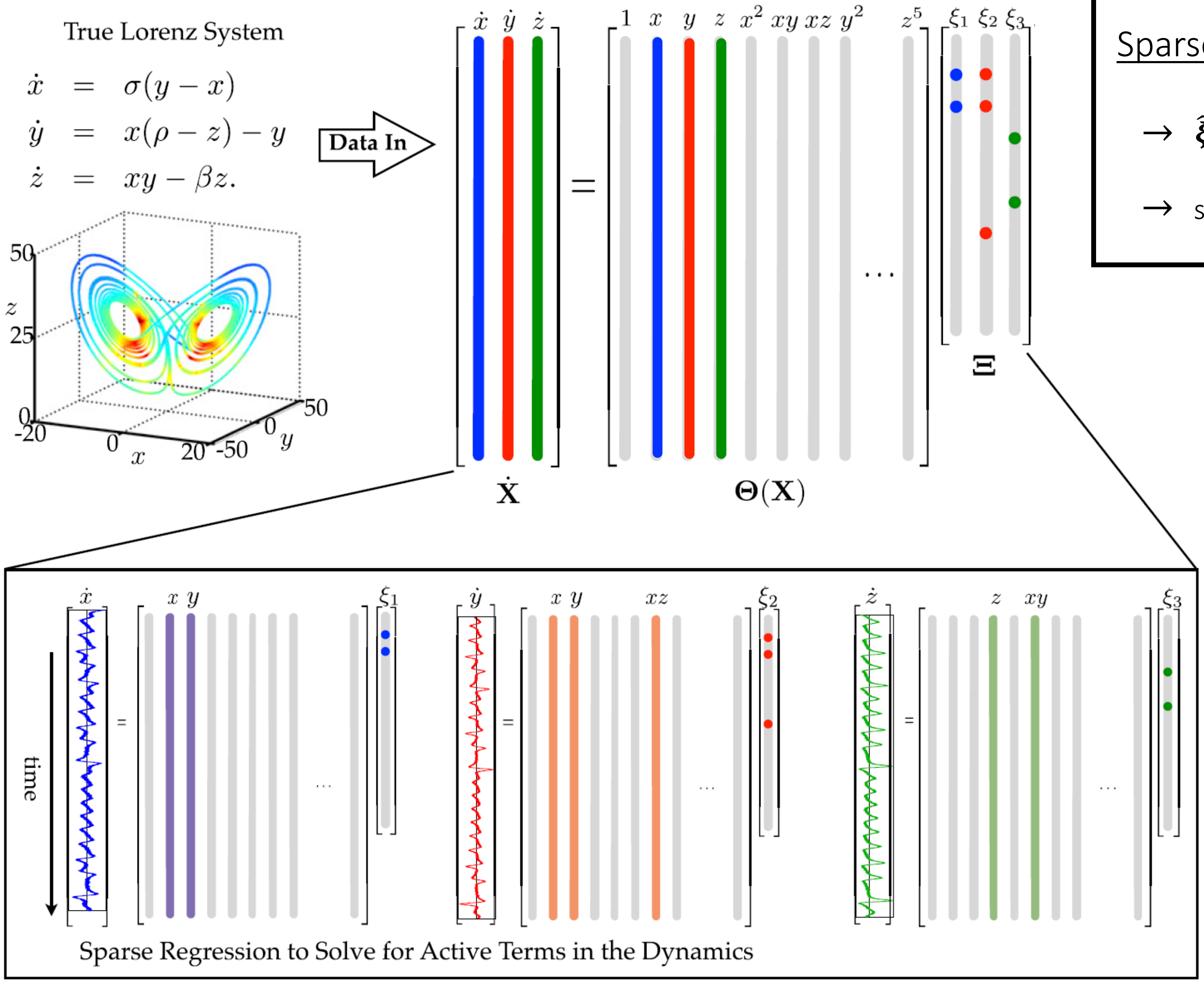


$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u})$$

$$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$$

SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

SINDy



<u>Sparse regression</u>: penalised least squares

→ $\hat{\boldsymbol{\xi}}_k = \text{argmin}_{\boldsymbol{\xi}_k} \left\|\dot{\mathbf{X}}_k - \mathbf{\Theta}(\mathbf{X})\boldsymbol{\xi}_k\right\|_2^2 + \lambda\|\boldsymbol{\xi}_k\|_0$

→ sequential thresholded least squares algorithm

SL Brunton, JL Proctor, JN Kutz (2016) [Discovering governing equations from data by sparse identification of nonlinear dynamical systems]().

Sequential thresholded least squares algorithm

Sparse regression: penalised least squares

$$\rightarrow \hat{\xi}_k = \mathrm{argmin}_{\xi_k} \left\|\dot{\mathbf{X}}_k - \mathbf{\Theta}(\mathbf{X})\xi_k\right\|_2^2 + \lambda\|\xi_k\|_0$$

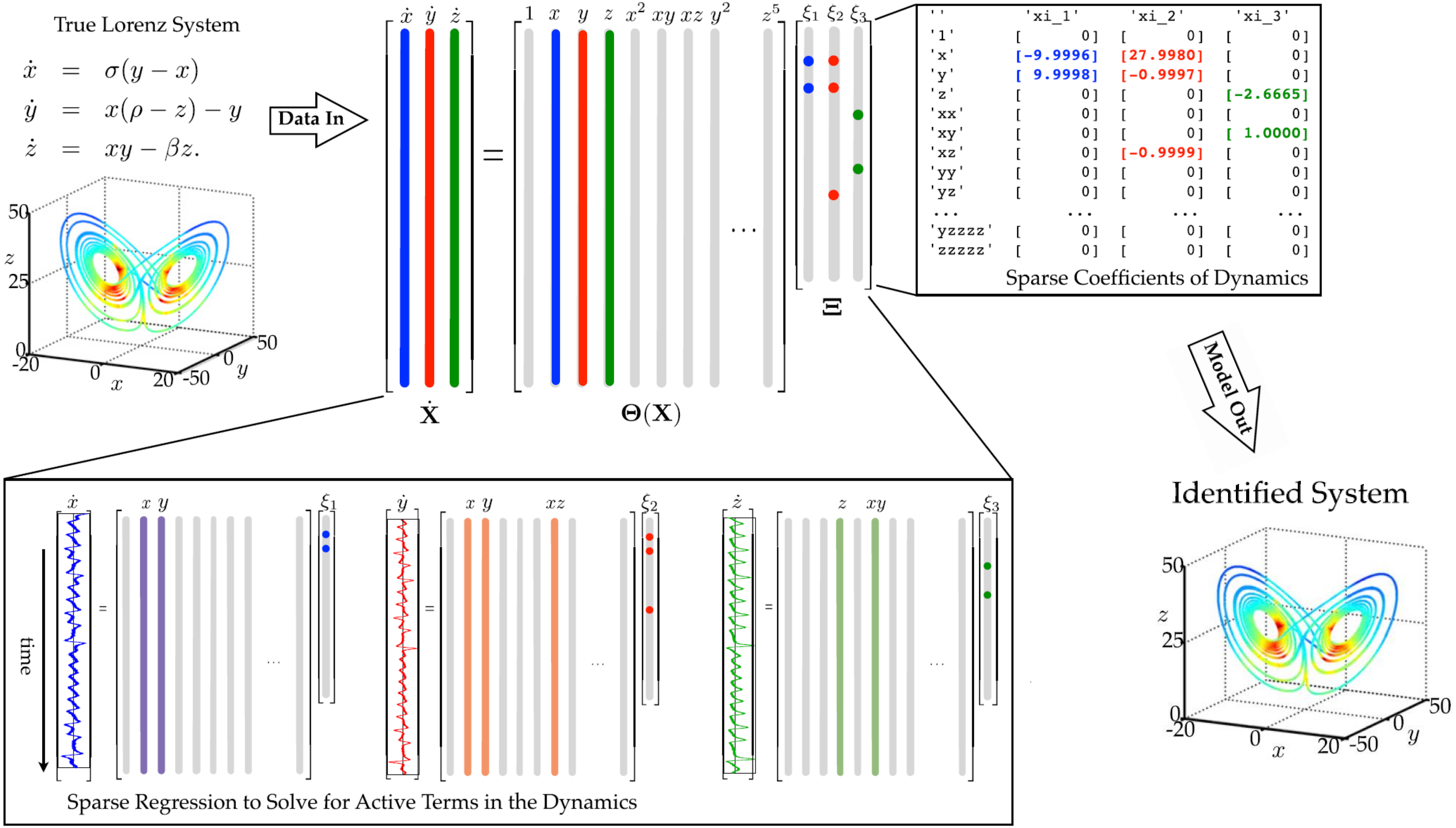
```
function Xi = sparsifyDynamics(Theta,dXdt,lambda,n)
% Compute Sparse regression: sequential least squares
Xi = Theta\dXdt;  % Initial guess: Least-squares

% Lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda);   % Find small coefficients
    Xi(smallinds)=0;                % and threshold
    for ind = 1:n                   % n is state dimension
        biginds = ~smallinds(:,ind);
% Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXdt(:,ind);
    end
end
```

Sparse Regression to Solve for Active Terms in the Dynamics

SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

SINDy

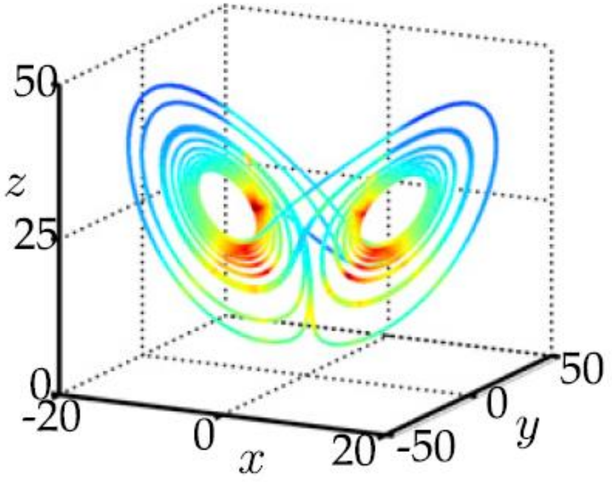


SL Brunton, JL Proctor, JN Kutz (2016) [Discovering governing equations from data by sparse identification of nonlinear dynamical systems](#).

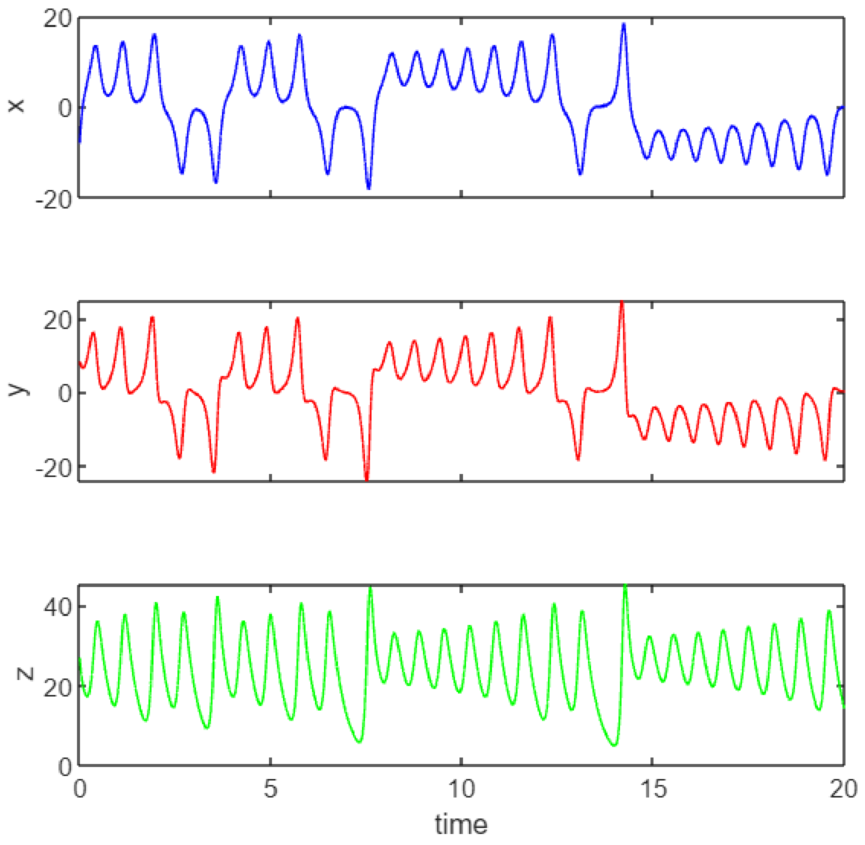
MATLAB tutorial: identify ODE → https://github.com/urban-fasel

Lorenz system

$$
\begin{aligned}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= x(\rho - z) - y \\
\dot{z} &= xy - \beta z.
\end{aligned}
$$



Data: time series x, y, z



No MATLAB installed?

→ Run the tutorials on **MATLAB online**: https://matlab.mathworks.com/

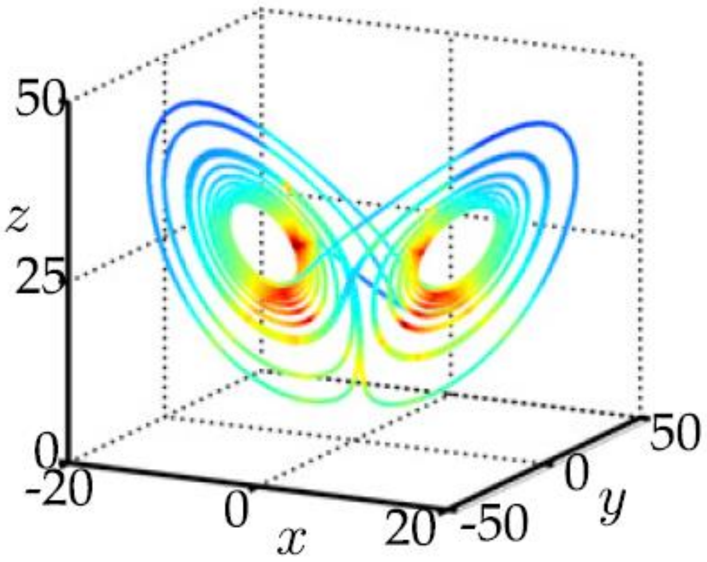
→ Or use **PySINDy** (next slide): https://github.com/dynamicslab/pysindy

→ Or **Julia** SciML: https://docs.sciml.ai/DataDrivenDiffEq/stable/#Package-Overview

Python tutorial – identify ODE

1. PySINDy

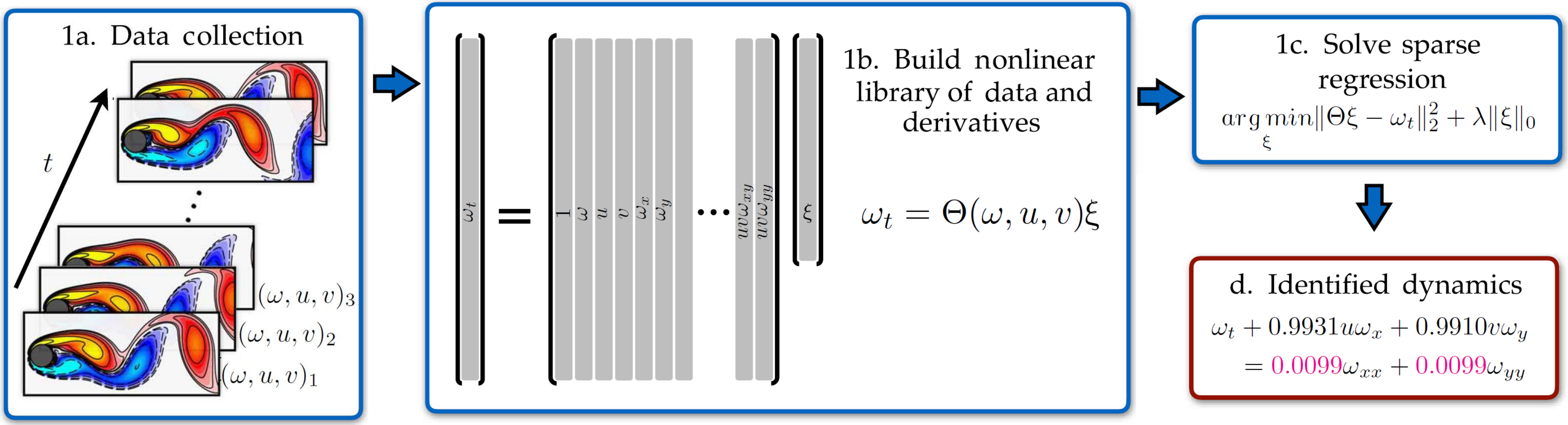
- SINDy python package: [JOSS article]
- GitHub: https://github.com/dynamicslab/pysindy
    - Check out the **binder notebook** examples!
- PySINDy lectures: notebooks and YouTube videos
    - GitHub interactive notebook
    - Tutorial videos Alan Kaptanoglu



2. Lorenz system ODE tutorial

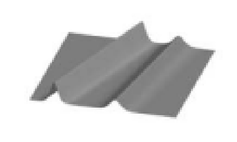
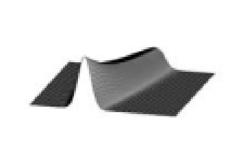
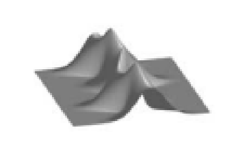
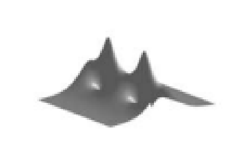
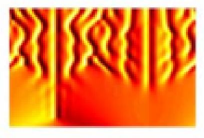
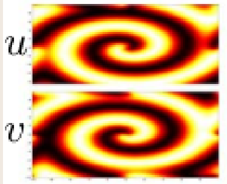
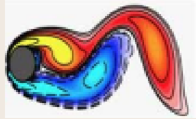
1. Start with the feature overview tutorial in PySINDy to **identify the Lorenz system**
2. Try to identify the **Rossler attractor**: `from pysindy.utils import rossler`
    - Generate data: `x_train = solve_ivp(rossler, ...`
3. Test other data sets generated from different ODEs → ODEs in PySINDy
    - Large library of chaotic systems: https://github.com/williamgilpin/dysts

PDEs



SH Rudy, SL Brunton, JL Proctor, JN Kutz (2017) [Data-driven discovery of partial differential equations](#).

PDEs

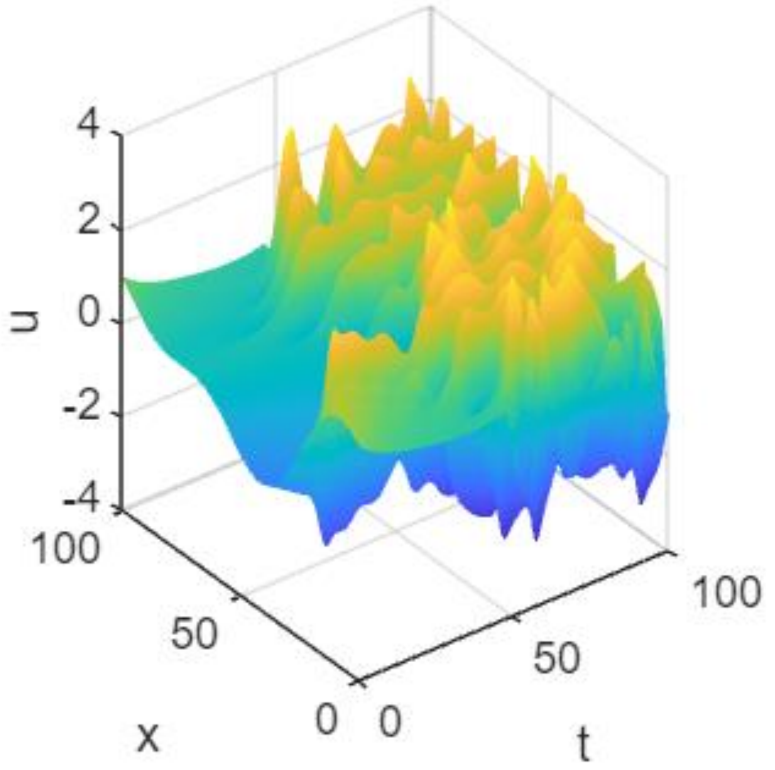
PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512,\ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256,\ t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512,\ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \|u\|^2 u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512,\ t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024,\ t \in [0, 100], m = 251$
 Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02 \pm 0.01\%,\ 3.8 \pm 2.4\%$	$x, y \in [-10, 10], n = 256,\ t \in [0, 10], m = 201$ subsample 1.14%
 Navier-Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$	$1 \pm 0.2\%\ ,\ 7 \pm 6\%$	$x \in [0, 9], n_x = 449,\ y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151,$ subsample 2.22%

SH Rudy, SL Brunton, JL Proctor, JN Kutz (2017) Data-driven discovery of partial differential equations.

PDE tutorial – Kuramoto-Sivashinsky equation

- **Kuramoto-Sivashinsky equation**: $u_t = -uu_x - u_{xx} - u_{xxxx}$
  - **Describes** e.g. chaotic dynamics of laminar flame fronts (Sivashinsky 1977) or reaction-diffusion systems (Kuramoto and Tsuzuki 1076).
  - **Challenging** PDE to identify, because it involves higher order partial derivatives.

- **MATLAB tutorial** (or PySINDy PDE-FIND)
  - **Library $\boldsymbol{\Theta}$**: 14 terms in the library
    - Polynomials: u, u^2, u^3
    - Partial derivatives: $u_x, u_{xx}, u_{xxx},\ u_{xxxx}$
    - Mixed terms: $u \cdot u_x,\ u \cdot u_{xx},\ u^2 \cdot u_x,\ \ldots$
  - **Derivatives**: finite difference
  - **Optimizer**: sequentially thresholded least squares



Challenges / limitations "vanilla" SINDy

Data → how much and what quality is needed?

Library → how to choose an effective library of candidate terms?

Optimization → what algorithm/regularization to use?

Model selection → how to select models / tune hyperparameters?

MATLAB tutorial – challenges / limitations of SINDy

Additional MATLAB tutorials

- Different optimizers: STLS vs LASSO
- Different ODEs and PDEs (Roessler, Burger's)
- …

Tutorial outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- ODEs and PDEs
- MATLAB and Python (PySINDy) examples
- SINDy limitations

"Vanilla" SINDy

Part 2: SINDy with control & parametric models

SINDy extensions

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Part 5: Your PhD project data sets

Applications