Planning of course:

- Dimensional reduction: singular value decomposition, principal component analysis, model reduction
- Regression, optimization, model selection
- Neural networks
- Physics informed neural networks, model discovery
- Projects

Model discovery https://www.youtube.com/watch?v=NxAn0oglMVw

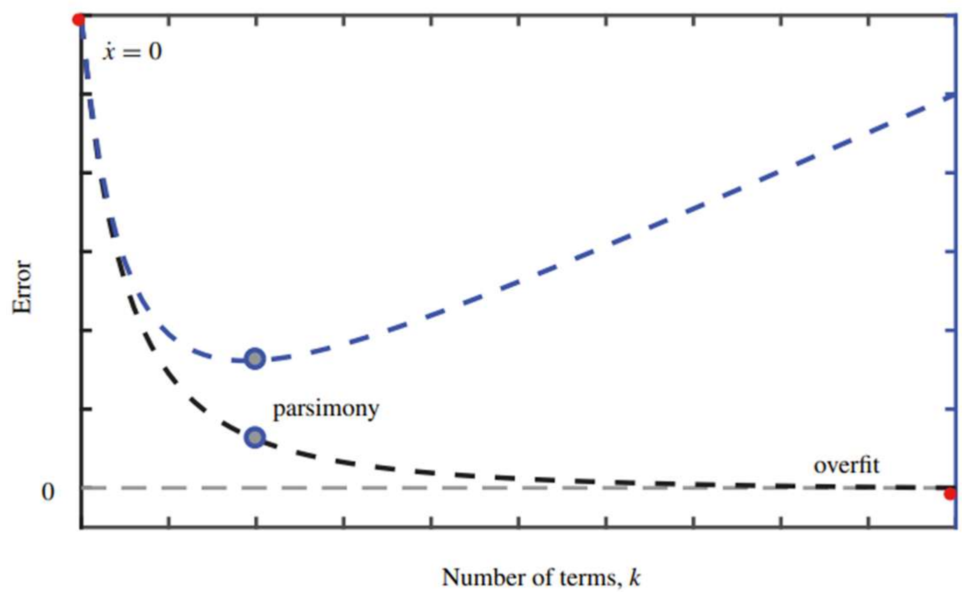
7.3 Sparse Identification of Nonlinear Dynamics (SINDy)

A parsimonious (as simple as possible) model defined by the sparse regression

$$\boldsymbol{\xi}_k = \underset{\boldsymbol{\xi}'_k}{\operatorname{argmin}} \|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\xi}'_k\|_2 + \lambda\|\boldsymbol{\xi}'_k\|_1 .$$

m x 1 m x dim(Theta) dim(Theta) x 1

where k=1,…,dim(x) (each column in X)



Code 7.4 [Python] Sequentially thresholded least-squares.

```
def sparsifyDynamics(Theta,dXdt,lamb,n):
    # Initial guess: Least-squares
    Xi = np.linalg.lstsq(Theta,dXdt,rcond=None)[0]

    for k in range(10):
        smallinds = np.abs(Xi) < lamb # Find small coeffs.
        Xi[smallinds] = 0       # and threshold
        for ind in range(n):    # n is state dimension
            biginds = smallinds[:,ind] == 0
            # Regress onto remaining terms to find sparse Xi
            Xi[biginds,ind] = np.linalg.lstsq(Theta[:,
                biginds],dXdt[:,ind],rcond=None)[0]

    return Xi
```

The sparse vectors $\boldsymbol{\xi}_k$ may be synthesized into a dynamical system:

$$\dot{x}_k = \boldsymbol{\Theta}(\mathbf{x})\boldsymbol{\xi}_k.$$

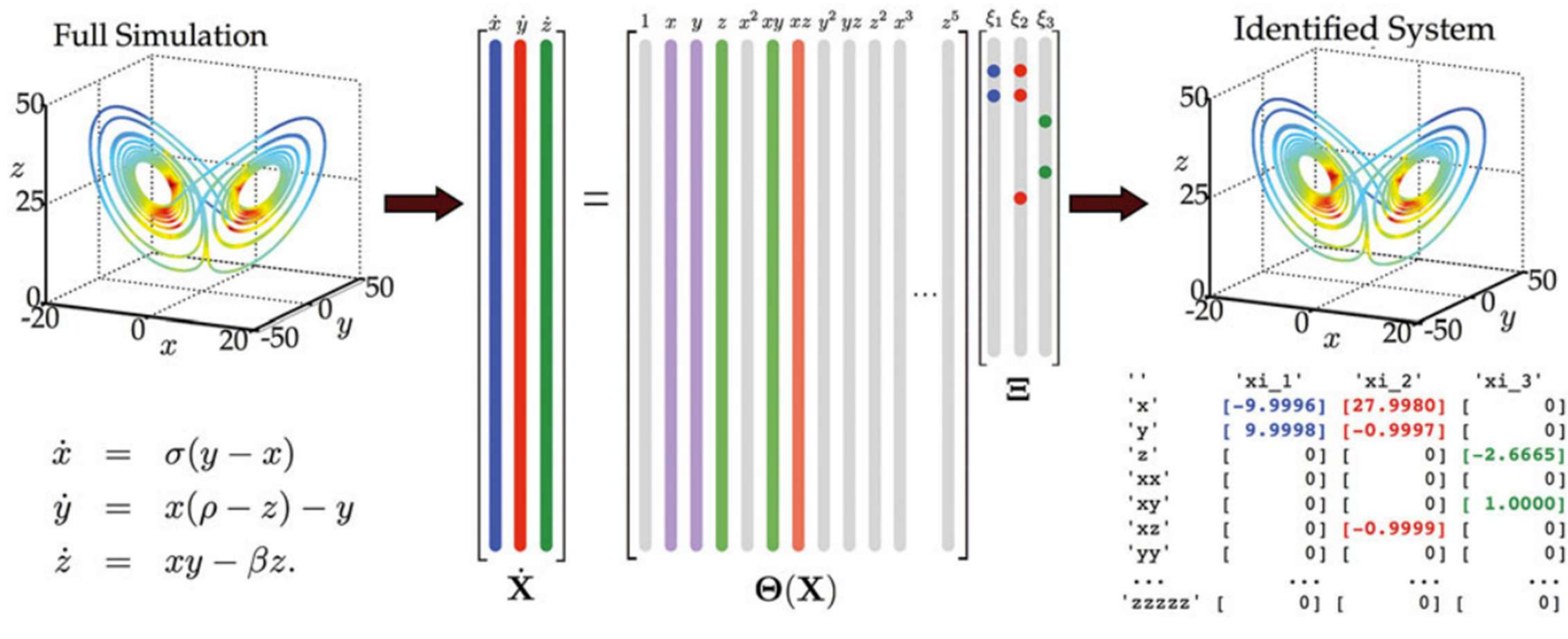


Figure 7.5 Schematic of the sparse identification of nonlinear dynamics (SINDy) algorithm [132].

PySINDy

CI passing | docs passing | pypi package 1.7.5 | codecov 95% | JOSS 10.21105/joss.02104 | JOSS 10.21105/joss.03994 | DOI 10.5281/zenodo.7808834

PySINDy is a sparse regression package with several implementations for the Sparse Identification of Nonlinear Dynamical systems (SINDy) method introduced in Brunton et al. (2016a), including the unified optimization approach of Champion et al. (2019), SINDy with control from Brunton et al. (2016b), Trapping SINDy from Kaptanoglu et al. (2021), SINDy-PI from Kaheman et al. (2020), PDE-FIND from Rudy et al. (2017), and so on. A comprehensive literature review is given in de Silva et al. (2020) and Kaptanoglu, de Silva et al. (2021).

https://github.com/dynamicslab/pysindy

The SINDy algorithm has recently been applied to identify high-dimensional dynamical systems, such as fluid flows, based on POD coefficients [132, 454, 455]. Figure 7.6

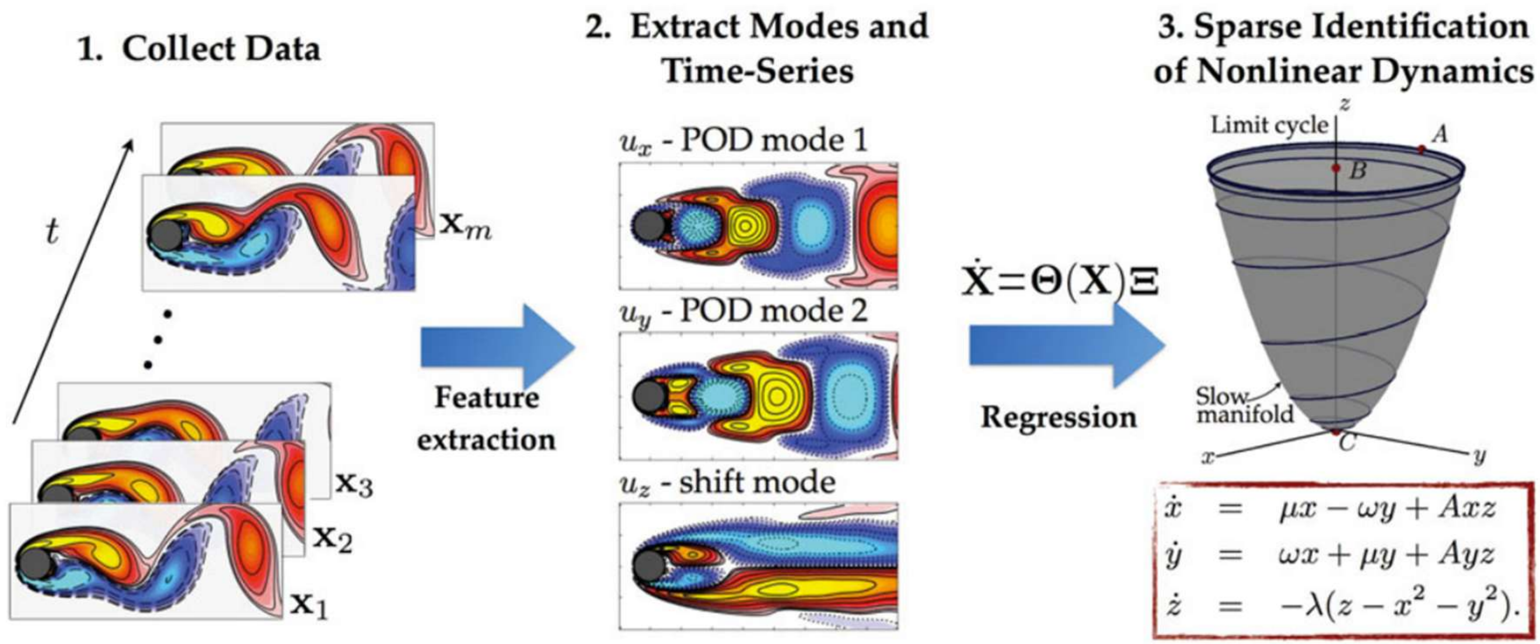
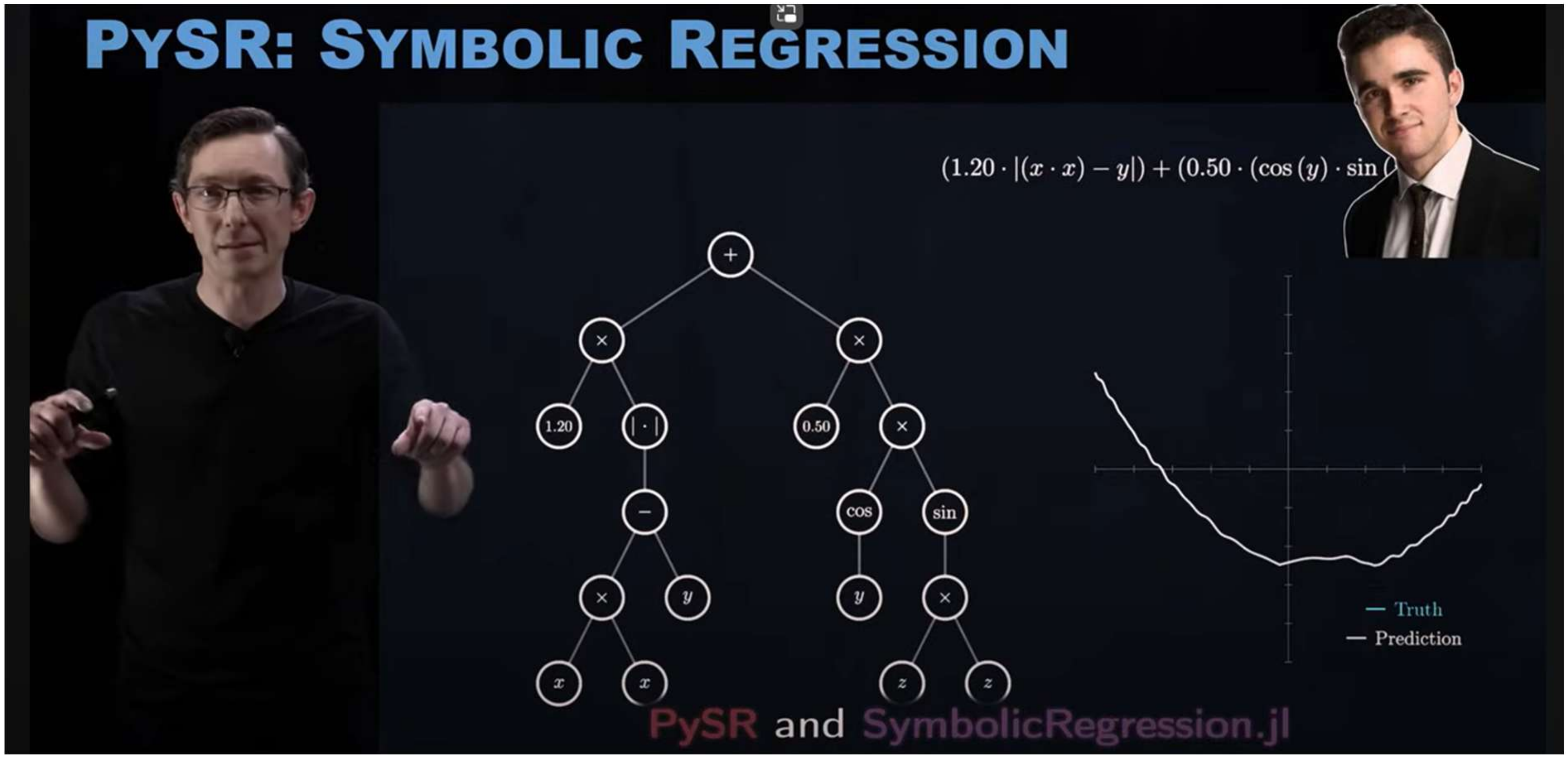
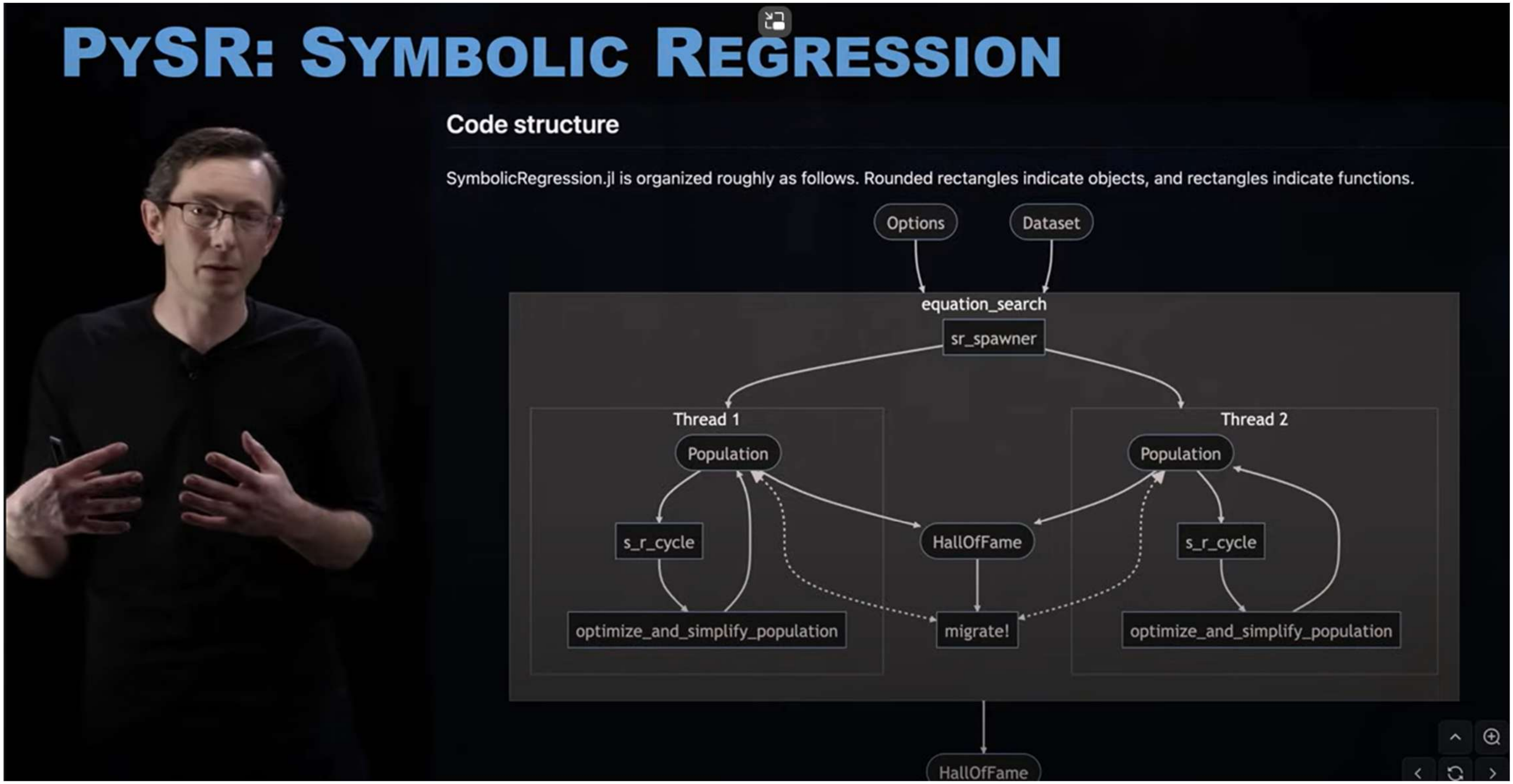


Figure 7.6 Schematic overview of nonlinear model identification from high-dimensional data using the sparse identification of nonlinear dynamics (SINDy) [132]. This procedure is modular, so that different techniques can be used for the feature extraction and regression steps. In this example of flow past a cylinder, SINDy discovers the model of Noack et al. [523]. Modified from Brunton et al. [132].

Symbolic regression https://github.com/MilesCranmer/PySR




PySR: Symbolic Regression
Code structure
SymbolicRegression.jl is organized roughly as follows. Rounded rectangles indicate objects, and rectangles indicate functions.
Options
Dataset
equation_search
sr_spawner
Thread 1
Population
s_r_cycle
optimize_and_simplify_population
HallOfFame
migrate!
Thread 2
Population
s_r_cycle
optimize_and_simplify_population
HallOfFame