An Introduction to Mathematical Climate Modelling with Applications

Erica Han, **Joanna Jiang** and **Katrina Wohlgemut**

Mentor: **Yusuf Aydogdu**

Faculty of Mathematics

University of Waterloo, Ontario, Canada

Women in Math Directed Reading Program

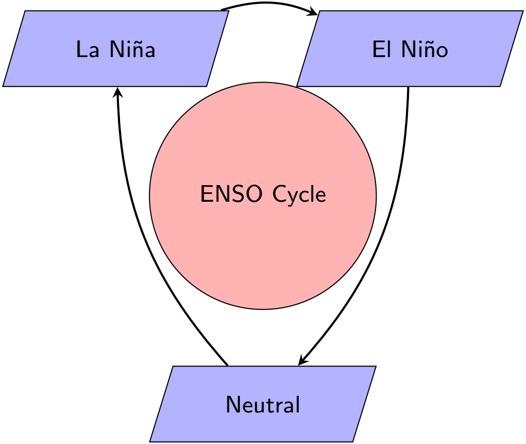
April 7, 2025

Contents

El Niño–Southern Oscillation

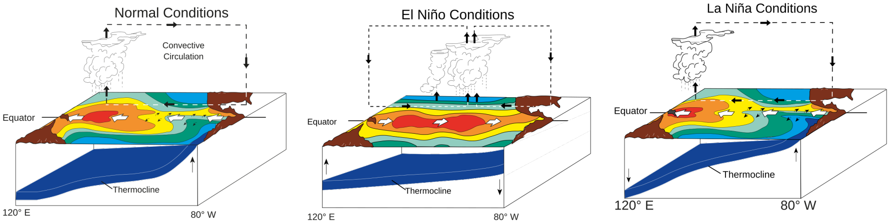
- Global climate phenomenon with significant ecological and societal impacts
- Cycles between
  - **El Niño**: Warm sea-surface temperature (SST) in eastern Pacific Ocean
  - **La Niña**: Cool SST in eastern Pacific Ocean
- Mainly impacts tropics (e.g. Southeast Asia, Australia, South America)
- Changing climate conditions can impact food security, air and water quality, ecosystems, human health, and disease transmission

El Niño–Southern Oscillation



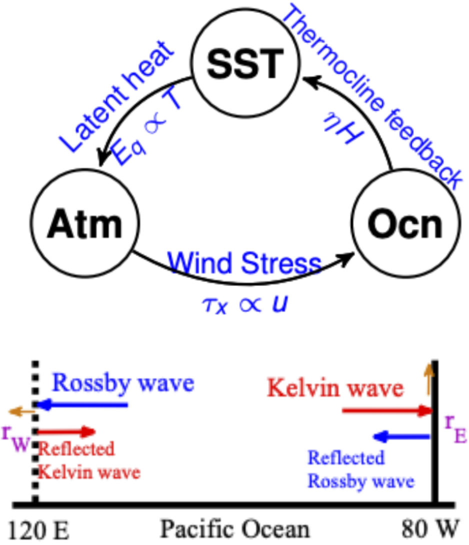
El Niño–Southern Oscillation

- Modelled as interaction between SST, ocean, and atmosphere
  - SST forces atmosphere through evaporation
  - Atmosphere forces ocean by wind stress
  - Ocean forces SST through thermocline feedback



Adapted from NOAA/PMEL/TAO diagrams. Public domain via Wikimedia Commons.

ENSO Model



ENSO coupling (top) and reflection of the waves at the boundaries (bottom)

ENSO Model

Atmosphere model :

$$\partial_t K^A(x,t) + \partial_x K^A(x,t) = \chi_A \alpha_q (2 - 2\overline{Q})^{-1} T(x,t)$$

$$\partial_t R^A(x,t) - \partial_x R^A(x,t)/3 = \chi_A \alpha_q (3 - 3\overline{Q})^{-1} T(x,t)$$

where $K^A(x,t)$ and $R^A(x,t)$ are **atmospheric Kelvin and Rossby waves**,respectively. Boundary conditions :

$$K^A(0,t) = r_w R^A(0,t) \quad \text{and} \quad R^A(L_O,t) = r_E K^A(L_O,t)$$

Ocean model:

$$\partial_t K^O(x,t) + c\partial_x K^O(x,t) = \chi_O \kappa c/2(K^A(x,t) - R^A(x,t))$$

$$\partial_t R^O(x,t) - c/3\partial_x R^O(x,t) = -\chi_O \kappa c/3(K^A(x,t) - R^A(x,t)))$$

where $K^O(x,t)$ and $R^O(x,t)$ are **oceanic Kelvin and Rossby waves**. Boundary conditions :

$$K^O(0,t) = r_W R^O(0,t) \quad \text{and} \quad R^O(L_O,t) = r_E K^O(L_O,t)$$

Sea Surface Temperature (SST) model :

$$\partial_t T(x,t) = -c\zeta\alpha_q T + c\eta(K^O(x,t) + R^O(x,t))$$

where $T(x,t)$ is the **SST**.

Based on the paper **Thual, S., Majda, A. J., Chen, N., & Stechmann, S. N. (2016). Simple stochastic model for El Niño with westerly wind bursts.** We modified atmosphere model and boundary conditions for simplicity [Chen et al., 2018].

In order to do numerical simulation, we need to discretize the equations in space and time. We use the first-order forward/backward difference.

Example: Discretization of Kelvin atmosphere equation

First-order forward time–backward space difference:

$$\frac{\partial K^A(t,x)}{\partial t} + \frac{\partial K^A(t,x)}{\partial x} = CT(t,x) \qquad (C = \chi_A \alpha_q (2 - 2\overline{Q})^{-1})$$

$$\Longrightarrow \frac{K^A(t+\Delta t, x) - K^A(t,x)}{\Delta t} + \frac{K^A(t,x) - K^A(t, x-\Delta x)}{\Delta x} = CT(t,x)$$

$$\Longrightarrow K^A(t+\Delta t, x) = K^A(t,x) - \frac{\Delta t}{\Delta x}(K^A(t,x) - K^A(t, x-\Delta x)) + C\Delta t T(t,x)$$

$$\Longrightarrow K^{A,j}_{i+1} = K^{A,j}_i - \frac{\Delta t}{\Delta x}(K^{A,j}_i - K^{A,j-1}_i) + C\Delta t T^j_i \qquad (i = t, j = x)$$

Initial conditions ($j = 0$): $K^{A,0}_{i+1} = K^{A,0}_i - \frac{\Delta t}{\Delta x}(K^{A,0}_i - K^{A,-1}_i) + C\Delta t T^0_i$

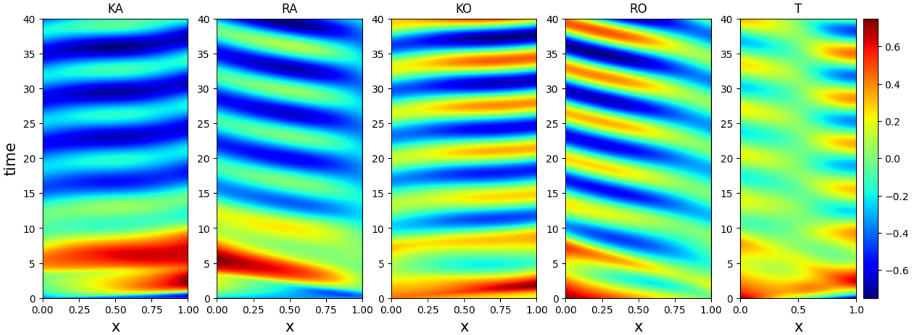
$$= K^{A,0}_i - \frac{\Delta t}{\Delta x}(K^{A,0}_i - r_w R^{A,0}_i) + C\Delta t T^0_i$$

Modelling & Simulation Parameters

Parameter	Value	Description
c	0.5	Ocean phase speed
χ_A	0.3	Meridional projection coefficient atm.
χ_O	1	Meridional projection coefficient ocean
ξ	1	Latent heating exchange coefficient
α_q	3	Latent heating factor
γ	6.5	Wind stress coefficient
η	$1.5 + 0.5\tanh(7.5(x - L/2))$	Profile of thermocline feedback
$\hat{Q}$	0.01	Mean vertical moisture gradient
M	8000	Number of time steps
N	200	Number of space steps
L	1	Space length
T_f	40	Time length

Parameter values used in the model

Simulation Results



K^A, R^A, K^O, R^O, and T over space (x-axis) and time (y-axis)

Contents

What:
Dimensionality reduction is the process of transforming high-dimensional data into a lower-dimensional representation while preserving its key features.
Why:
Improve accuracy:
High-dimensional data often contains redundant or irrelevant features.
Avoid the Curse of Dimensionality:
Dimensionality reduction removes these, improving model performance.
As the number of dimensions increases data becomes sparse, making it harder to analyze and visualize.

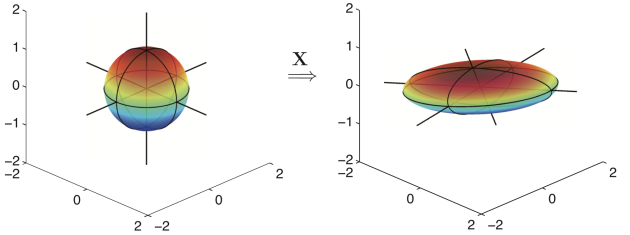


Figure: Geometric illustration of the dimensionality reduction as a mapping from a sphere in R^n to an ellipsoid in R^m [Brunton and Kutz, 2022]

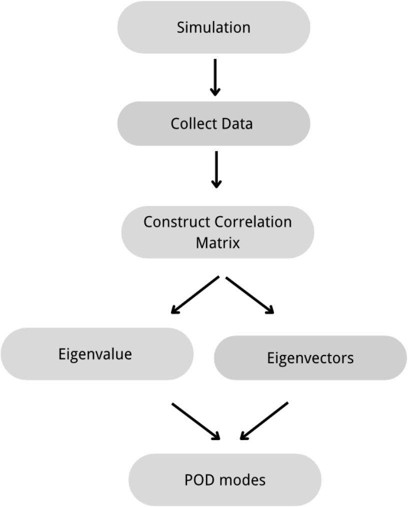
Function expansion

A common way is through eigenfunction expansions:

$$\boxed{u(t,x) \approx \sum_{k=1}^{r} a_k(t)\psi_k(x)}$$

- $\psi_k(x)$: spatial basis functions
- $a_k(t)$: corresponding time-dependent coefficients
- r: the number of basis elements

Algorithm



Simulation
Collect Data
Construct Correlation Matrix
Eigenvalue
Eigenvectors
POD modes

Computing Auto-Correlation Matrices

To construct the correlation matrix, we compute the auto-correlation matrices by multiplying each matrix with its transpose:

$$\mathbf{C}_{\mathbf{KA}(8000\times 8000)} = \mathbf{KA}_{8000\times 200} \cdot \mathbf{KA}^{T}_{200\times 8000}$$

$$\mathbf{C}_{\mathbf{RA}} = \mathbf{RA} \cdot \mathbf{RA}^{T}$$

$$\mathbf{C}_{\mathbf{KO}} = \mathbf{KO} \cdot \mathbf{KO}^{T}$$

$$\mathbf{C}_{\mathbf{RO}} = \mathbf{RO} \cdot \mathbf{RO}^{T}$$

$$\mathbf{C}_{\mathbf{T}} = \mathbf{T} \cdot \mathbf{T}^{T}$$

$$\boxed{\mathbf{C}_{8000\times 8000} = (\mathbf{C}_{KA} + \mathbf{C}_{RA} + \mathbf{C}_{KO} + \mathbf{C}_{RO} + \mathbf{C}_{T})/M}$$

Eigenvalue Decomposition

- Calculate the eigenvalues and eigenvectors of the correlation matrix **C**

$$\mathbf{C}v = \lambda v$$

- Construct the POD modes for each component
  - $$\phi_{KA} = \sum v \times KA/\sqrt{M \times \lambda}$$
  - $$\phi_{RA} = \sum v \times RA/\sqrt{M \times \lambda}$$
  - $$\phi_{KO} = \sum v \times KO/\sqrt{M \times \lambda}$$
  - $$\phi_{RO} = \sum v \times RO/\sqrt{M \times \lambda}$$
  - $$\phi_{T} = \sum v \times T/\sqrt{M \times \lambda}$$
- Combine them

$$\psi = (\phi_{KA}, \phi_{RA}, \phi_{KO}, \phi_{RO}, \phi_T)$$

Eigenvalues of Correlation Matrix

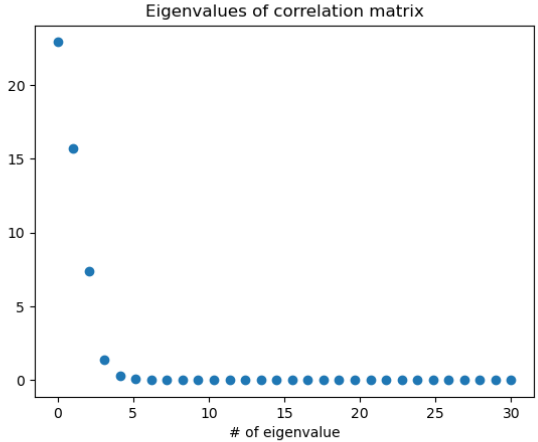


Figure: From the graph, we can clearly see that the first four points stand out the most. After the fourth point, the gaps between them become smaller, indicating that the remaining components contribute less significant information and can be disregarded.

Root Mean Square Error (RMSE) Using Different Number of Modes

The sharp initial drop indicates the first few principle components are needed.

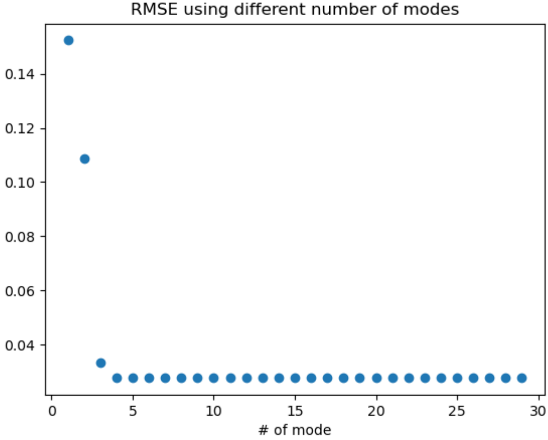


Figure: The reconstruction RMSE of SST using different number of modes

POD modes and time series

$$\boxed{u(t,x) \approx \sum_{k=1}^{4} a_k(t)\psi_k(x)}$$

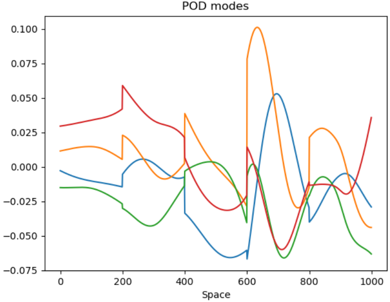


Figure: Spatial modes $\Psi_k(x)$

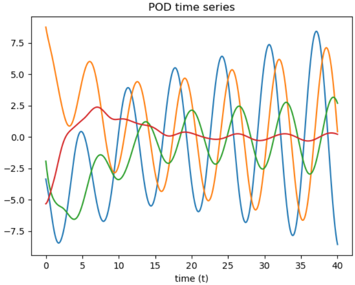


Figure: Time-dependent coefficients $a_k(t)$

Contents

SINDy Equation

- Find ODE model from data for dynamic system
- Assume dynamics controlled by only a few variables
- Balances complexity of model with accuracy

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{array}{cc} \text{state} \longrightarrow & \\ \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} & \text{time} \downarrow \end{array}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}.$$

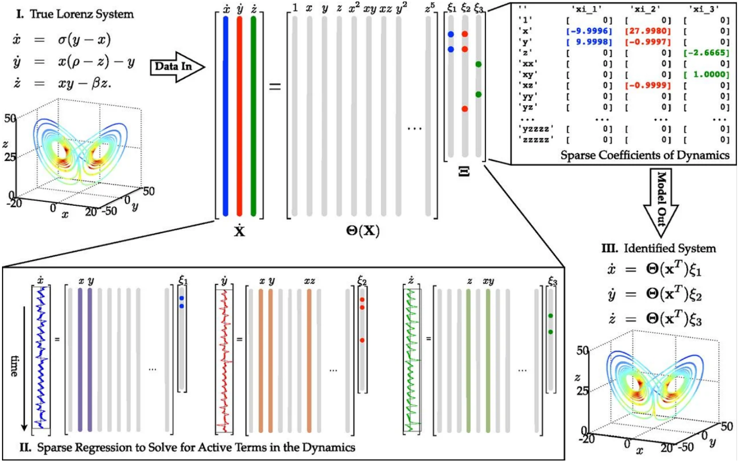
$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & & | & | & \\ 1 & \mathbf{X} & \mathbf{X}^{P_2} & \mathbf{X}^{P_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & \end{bmatrix}$$

Solve for these coefficients

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi.$$

- Use sequential threshold techniques for optimization
  - Candidate library and threshold can be adjusted as needed
- Use strategies for model selection which penalize large number of terms [Brunton et al., 2016]

SINDy Example

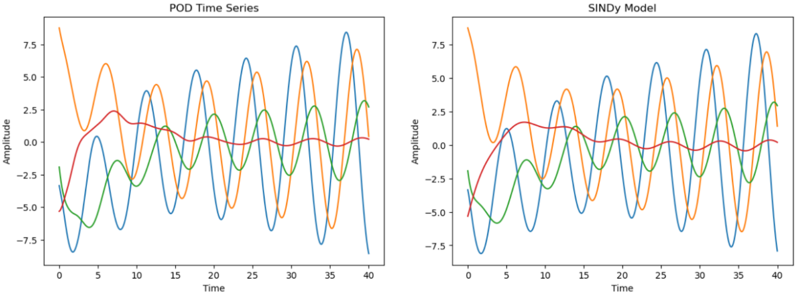


Coefficient Matrix for Model Equations

	q1'	q2'	q3'	q4'
1	0	0	0	0
q1	0.0268	0.6805	0.2003	0
q2	-0.9408	-0.0611	0.2855	0.0715
q3	-0.8154	-0.4754	-0.0325	-0.1290
q4	-0.2203	0.0329	0.2664	-0.2531
q1^2	0	0	0	0
q1 q2	0	0	0	0
q1 q3	0	0	0	0
q1 q4	0	0	0.0250	0
q2^2	0	0	0	0
q2 q3	0	0	0	0
q2 q4	0	0	0	0
q3^2	0	0	0	0
q3 q4	0	0	0.0382	0
q4^2	0.2691	0	0	0
q1^3	0	0	0	0
q1^2 q2	0	0	0	0
q1^2 q3	0	0	0	0
q1^2 q4	0	0	0	0
q1 q2^2	0	0	0	0
q1 q2 q3	0	0	0	0
q1 q2 q4	0	0	0	0
q1 q3^2	0	0	0	0
q1 q3 q4	0	0	0	0
q1 q4^2	0	0	0	0
q2^3	0	0	0	0
q2^2 q3	0	0	0	0
q2^2 q4	0	0	0	0
q2 q3^2	0	0	0	0
q2 q3 q4	0	0	0	0
q2 q4^2	0	0	0	0
q3^3	0	0	0	0
q3^2 q4	0	0	0	0
q3 q4^2	0.0864	0	0	0
q4^3	0	0	0.0410	0

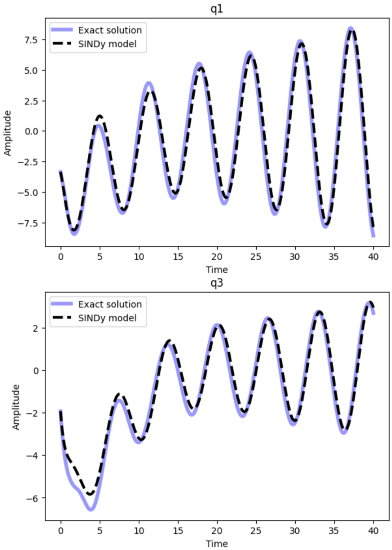
Figure: From top to bottom polynomials up to third degree

SINDy Results



- Used sequentially thresholded least squares optimization with threshold 0.02
- Candidate library is polynomials up to degree 3
- Used finite difference for differentiation

Exact Dynamics compared to SINDy Model



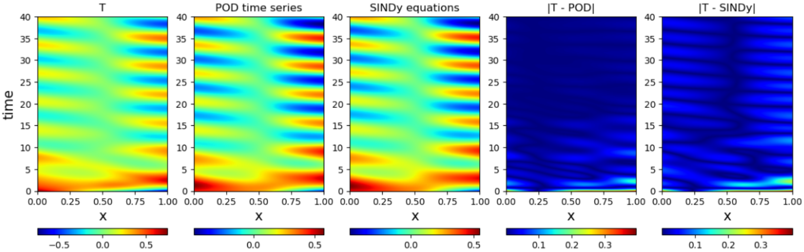


Figure: Temperature, POD reconstruction, SINDy reconstruction, L1 error for POD, L1 error for SINDy

Conclusion

- ENSO can be mathematically modeled using coupled PDEs describing interactions between sea surface temperature, the atmosphere, and the ocean.
- The PDEs are discretized using finite differences for numerical simulation, enabling the computation of the observed temperature T over space and time.
- Dimensionality reduction is applied to reduce the system's dimension from 200 to 4.
- The Root Mean Square Error (RMSE) is calculated for temperature, POD modes, and time series.
- The SINDy algorithm is used to identify an ODE model for the dynamical system.
- The original data is compared to the reconstructed data obtained from the SINDy equations.

Brunton, S. L. and Kutz, J. N. (2022).
Data-driven science and engineering: Machine learning, dynamical systems, and control.
Cambridge University Press.

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.

Chen, N., Majda, A. J., and Thual, S. (2018).
Observations and mechanisms of a simple stochastic dynamical model capturing el niño diversity.
Journal of Climate, 31(1):449–471.