Learning nonlinear dynamical systems from data using scientific machine learning

Romit Maulik
Assistant Computational Scientist, Argonne National Laboratory

August 12, 2022

Motivation

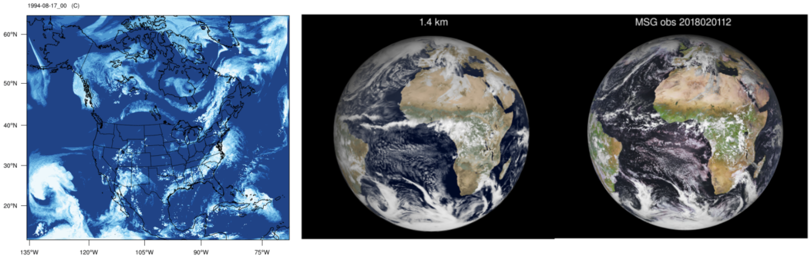


Figure: Cloud resolved weather and climate simulations are becoming a reality. 4km simulations of E3SM run over 100 forecast years require 120 million core hours (Theta-ANL) and 12 PB of storage data (250 GB/forecast day).

Image source: Jung et al., Simulations of E3SM on ANL-Theta, 2022 (top), ECMWF Simulations on ORNL-Summit, DOE E3SM All-hands meeting 2021.

Emulating dynamical systems from data

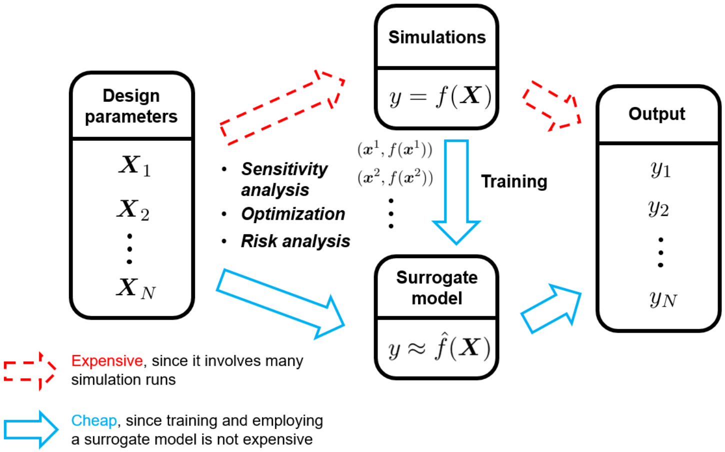


Figure: Source - “An introduction to surrogate modeling” - Shuai Guo.

Emulating dynamical systems from data

There may also be a requirement to construct 'non-intrusive' surrogate models - for example when dynamics are only partially understood/known - i.e., **No closed form governing laws available**.

This project is joint work with

- Prasanna Balaprakash (Argonne).
- Qi Tang, Joshua Burby (Los Alamos).
- Alec Linot, Mike Graham (Wisconsin).
- Varun Shankar, Vedant Puri, Venkat Vishwanathan (CMU).

Background: Neural ordinary differential equations

$$\frac{d\mathbf{a}}{dt} = f(\mathbf{a}, \theta), \quad (\theta) \in \Theta, \tag{1}$$

where $\Theta \subset \mathbb{R}^{N_w}$ is the space of trainable parameters of an arbitrary neural network. The NODE [3, 4, 5] approximates the latent-space evolution as a set of ordinary differential equations that can be trained through adjoint-based (i.e., continuous) backpropagation [3, 5], i.e.,

$$L(\tilde{\mathbf{a}}^T) = L(\mathbf{a}^0 + \int_{t=0}^{t=T} f(\mathbf{a}(\mathbf{t}), \theta) dt) \tag{2}$$

$$\frac{d\mathbf{z}}{dt} = -\mathbf{z}^T \frac{\partial f(\mathbf{a}, t, \theta)}{\partial \mathbf{a}}, \quad \mathbf{z}(t) = \frac{\partial L}{\partial \mathbf{a}(t)} \tag{3}$$

$$\frac{dL}{d\theta} = -\int_{t=T}^{t=0} \mathbf{z}(t)^T \frac{\partial f(\mathbf{a}(\mathbf{t}), \theta)}{\partial \theta} dt. \tag{4}$$

Chaotic dynamics: The Kuramoto-Sivashinsky equation

We want to address the surrogate modeling of chaotic systems. Traditionally, most data-driven time-series modeling techniques suffer with deterministic chaos.

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \tag{5}$$

$$u \in \mathbb{R}^{64}; x \in [-\pi, \pi] \subset \mathbb{R}^1 \tag{6}$$

A prototypical system to study chaotic dynamics, possesses a dissipative nature (i.e., an attractor in the long-term limit), challenging for state-of-the-art black-box forecasting methods.

A novel neural ODE for capturing chaotic attractors

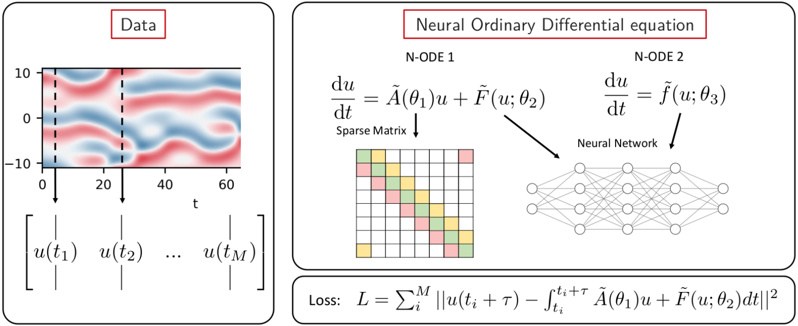


Figure: A novel neural-ODE for learning chaotic dynamics.

Linot, Burby, Tang, Balaprakash, Graham, RM. arXiv:2203.15706.

Example: The Kuramoto-Sivashinsky equation

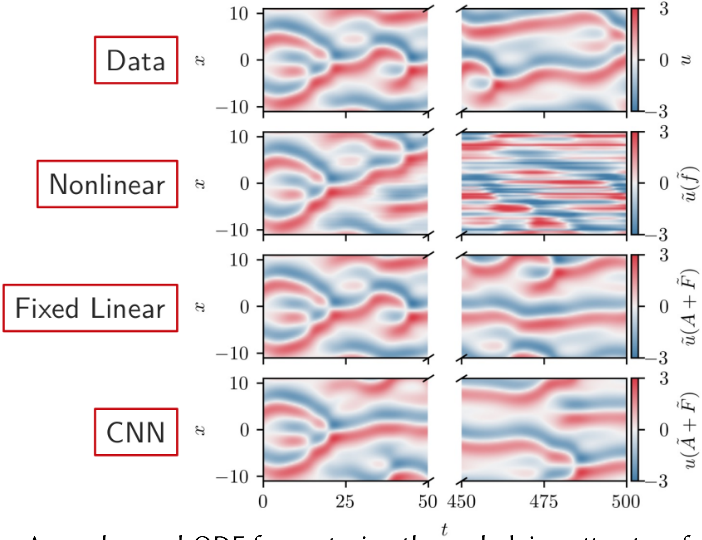
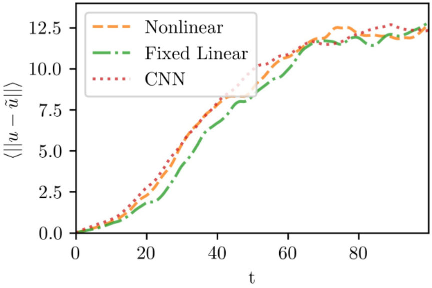


Figure: A novel neural-ODE for capturing the underlying attractors for the KS equations: Long-term stability.

Example: The Kuramoto-Sivashinsky equation

Tracking Error



Joint PDFs

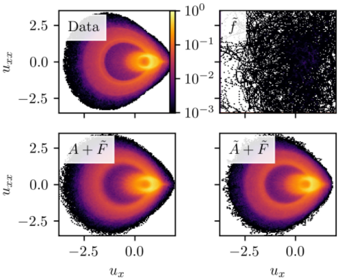


Figure: A novel neural-ODE for capturing the underlying attractors for the KS equations: Attractor captured successfully!

Approximate inertial manifold theory can then be used since we have a linear term.

Flipping the script: A reduced-order model of a surrogate

We can also reduce the order of this neural ODE a-posteriori by using the theory of approximate inertial manifolds [9]:

$$\frac{dp}{dt} = Ap + PF(p+q), \tag{7}$$

$$\frac{dq}{dt} = Aq + QF(p+q), \tag{8}$$

$$q = A^{-1}Q(p+q) \approx A^{-1}Q(p). \tag{9}$$

$$Q = I - P. \tag{10}$$

If we construct P using selected eigenvectors of the learned linear term A

$$AV = V\Lambda,$$

$$P = \tilde{V}\tilde{V}^{T} \tag{11}$$

where $\tilde{V}$ are a truncated subset of eigenvectors that promotes $\frac{dq}{dt} = 0$.

Ignoring the computation of q gives us the nonlinear Galerkin ROM, computing q with $A^{-1}Q(p)$ during the simulation gives us the AIM ROM and after the simulation gives us postprocessing ROMs.

Example: The Kuramoto-Sivashinsky equation

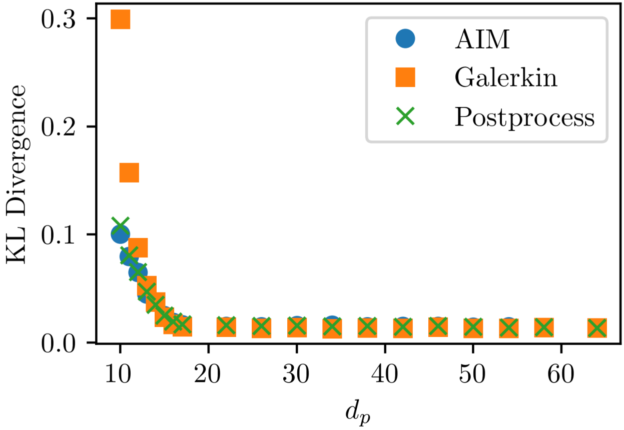


Figure: A reduced-order model from the proposed full-order neural ODE. KL-divergence of attractor statistics - model reduced to 25% of original size.

Example: Learning the viscous Burgers equations

The viscous Burgers equations are given by the following system

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} \tag{12}$$

in a domain with length $L = 1$ and with periodic boundary conditions. Our viscosity, $\nu = 8 \times 10^{-4}$. Initial conditions sampled from superpositions of frequencies in Fourier space for the same viscosity. Solution discretized on 512 grid points.

Example: Learning the viscous Burgers equations

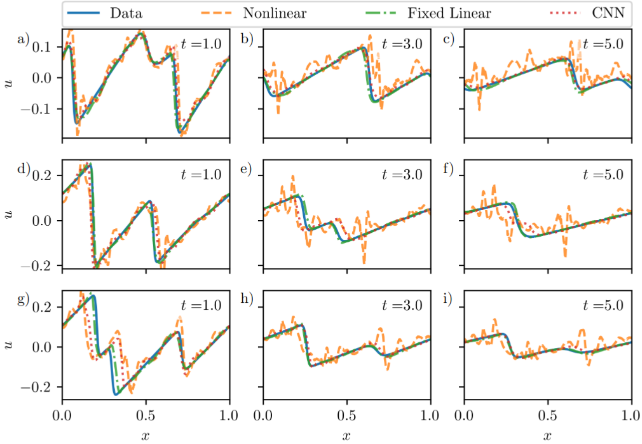


Figure: The stabilized neural ODE outperforms the standard neural ODE for learning the viscous Burgers equations.

Example: Learning the viscous Burgers equations

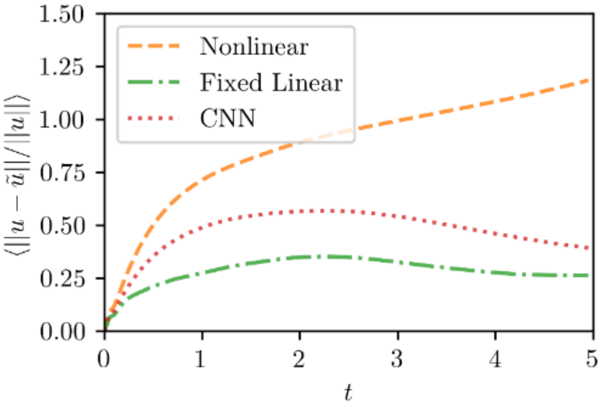


Figure: The stabilized neural ODE outperforms the standard neural ODE for learning the viscous Burgers equations - confirmed for an ensemble of test predictions.

Example: Learning the viscous Burgers equations

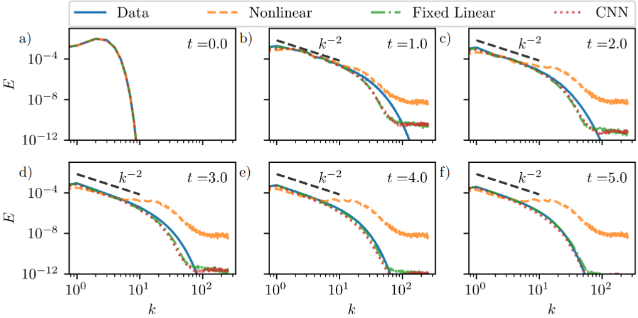


Figure: The stabilized neural ODE outperforms the standard neural ODE for learning the viscous Burgers equations - confirmed for an ensemble of test predictions.

Example: Learning the viscous Burgers equations

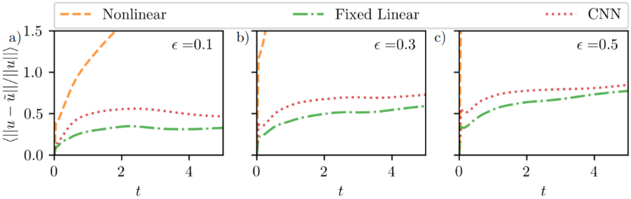


Figure: When adding noise to the initial conditions - the stabilized neural ODE performs more robustly

Example: Learning the viscous Burgers equations

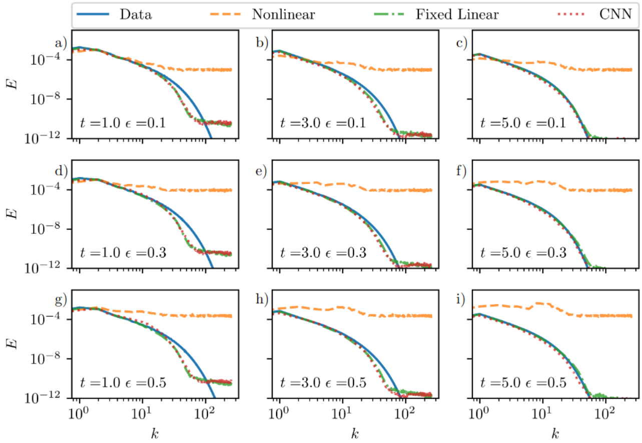


Figure: When adding noise to the initial conditions - the stabilized neural ODE performs more robustly

Under-resolved snapshot data? A preview.

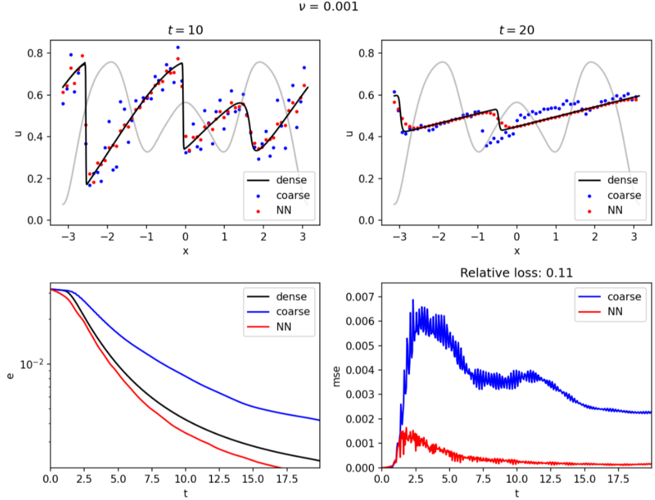


Figure: The stabilized NODE framework is also able to learn a stabilized coarse-grained evolution (i.e., if snapshot resolution is inadequate). Fine-grid 4096 DOF, coarse-grid 64 DOF.

Under-resolved snapshot data? A preview.

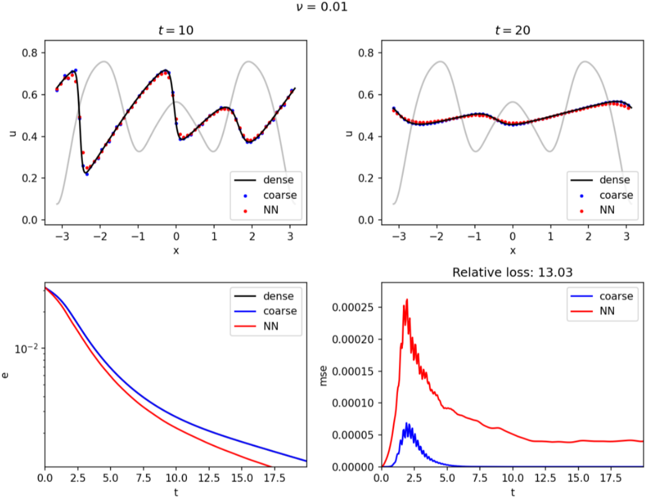


Figure: The stabilized NODE framework is also able to learn a stabilized coarse-grained evolution (i.e., if snapshot resolution is inadequate). Fine-grid 4096 DOF, coarse-grid 64 DOF.

Under-resolved snapshot data? A preview.

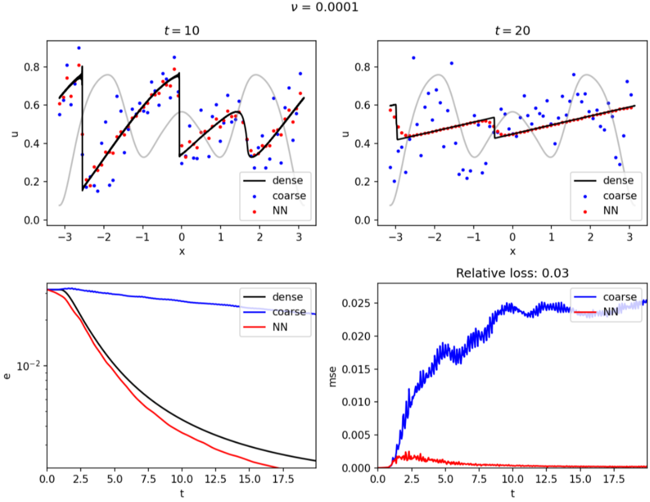


Figure: The stabilized NODE framework is also able to learn a stabilized coarse-grained evolution (i.e., if snapshot resolution is inadequate). Fine-grid 4096 DOF, coarse-grid 64 DOF.

Example: Learning the invariant manifold of the sea-surface temperature

NOAA OI SST V2 High Resolution Dataset

Data on and after 2016 is now v2.1

Brief Description:

- NOAA High-resolution Blended Analysis of Daily SST and Ice. Data is from Sep 1981 and is on a 1/4 deg global grid. *More Details...*

Temporal Coverage:

- Daily values from 1981/09 to present
- **Sea Ice Concentration data is missing for Dec 6th 1987- Jan 10th 1988.**

Spatial Coverage:

- 0.25 degree latitude x 0.25 degree longitude global grid (1440x720).
- 89.875S - 89.875N,0.125E to 359.875E.

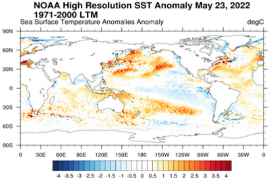


Figure: A sea-surface temperature dataset obtained from satellite and ship observations.

Example: Learning the invariant manifold of the sea-surface temperature

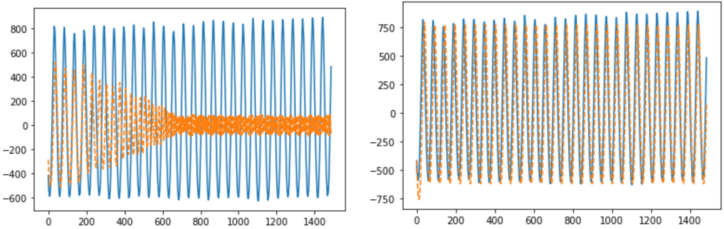


Figure: Test results for learning the POD coefficients of this dataset using regular (left) and stabilized (right) neural ODEs.

Example: Learning the invariant manifold of the sea-surface temperature

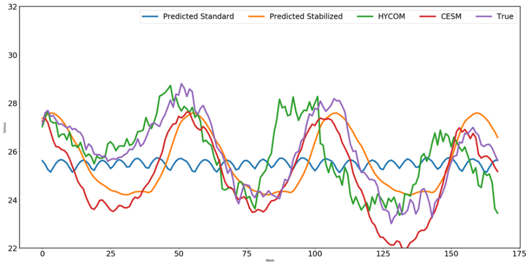


Figure: Preliminary results indicate that predictive dynamics do not decay to fixed point. Probe for solution at 95 degrees latitude and 250 degrees longitude.

Example: Learning the invariant manifold of the sea-surface temperature

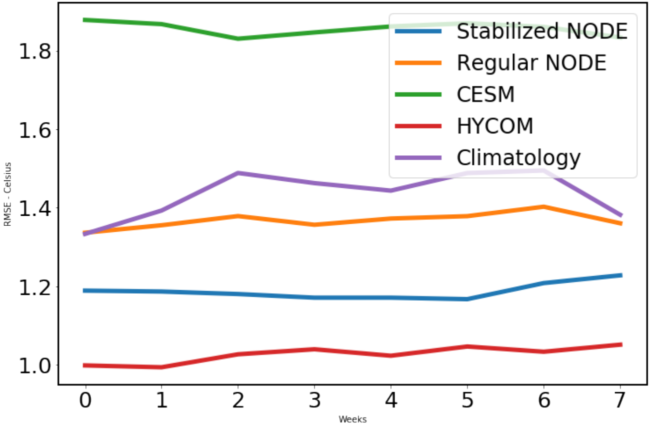


Figure: Comparisons on test data across different methods.

Emulating the sea-surface temperature: Worth it?

Cost to construct our NODE-ROM: 2 node hours of CPU-only laptop, cost to evaluate - negligible.

Cost to evaluate HYCOM: 44800 core hours per forecast day of Cray XC40 system.

Cost to evaluate CESM: 510 million core-hours on Yellowstone, NCAR's high-performance computing resource.

Extensions: Interfacing SST-ROMs as a 'boundary condition' to E3SM atmosphere.

Acknowledgements

U.S. Department of Energy, Advanced Scientific Computing Research
(DOE-FOA2493: Data intensive scientific machine learning, PI-Maulik)

U.S. Department of Energy, Advanced Scientific Computing Research
(SCIDAC-RAPIDS Institute, PI-Ross)

Argonne Leadership Computing Facility
For compute resources and Margaret Butler Fellowship
(DE-AC02-06CH11357)

Thanks for listening!

`romit-maulik.github.io`

References I

Lawrence Sirovich.
Turbulence and the dynamics of coherent structures. i. coherent structures.
Quarterly of applied mathematics, 45(3):561–571, 1987.

AE Deane, IG Kevrekidis, G Em Karniadakis, and SA0746 Orszag.
Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders.
Physics of Fluids A: Fluid Dynamics, 3(10):2337–2354, 1991.

R Rico-Martinez, K Krischer, IG Kevrekidis, MC Kube, and JL Hudson.
Discrete-vs. continuous-time nonlinear signal processing of cu electrodissolution data.
Chemical Engineering Communications, 118(1):25–48, 1992.

Ramiro Rico-Martinez and Ioannis G Kevrekidis.
Continuous time modeling of nonlinear systems: A neural network-based approach.
In *IEEE International Conference on Neural Networks*, pages 1522–1525. IEEE, 1993.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud.
Neural ordinary differential equations.
arXiv preprint arXiv:1806.07366, 2018.

Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.

Alex Sherstinsky.
Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network.
Physica D: Nonlinear Phenomena, 404:132306, 2020.

References II

Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, and Daniel Livescu.
Time-series learning of latent-space dynamics for reduced-order model closure.
Physica D: Nonlinear Phenomena, 405:132368, 2020.

Michael S Jolly, IG Kevrekidis, and Edriss S Titi.
Approximate inertial manifolds for the kuramoto-sivashinsky equation: analysis and computations.
Physica D: Nonlinear Phenomena, 44(1-2):38–60, 1990.

Sebastian Mika, Bernhard Schölkopf, Alexander J Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch.
Kernel pca and de-noising in feature spaces.
In *NIPS*, volume 11, pages 536–542, 1998.

Ronald R Coifman and Stéphane Lafon.
Diffusion maps.
Applied and computational harmonic analysis, 21(1):5–30, 2006.

Mark A Kramer.
Nonlinear principal component analysis using autoassociative neural networks.
AIChE journal, 37(2):233–243, 1991.

Romit Maulik, Romain Egele, Bethany Lusch, and Prasanna Balaprakash.
Recurrent neural network architecture search for geophysical emulation.
In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14. IEEE, 2020.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski.
Deepar: Probabilistic forecasting with autoregressive recurrent networks.
International Journal of Forecasting, 36(3):1181–1191, 2020.