COLLEGE OF SCIENCE
ACADEMY OF DATA SCIENCE
VIRGINIA TECH

COMPUTATIONAL MODELING
AND DATA ANALYTICS

Sandia National Laboratories

Data-driven Reduced Modeling in the Time and Frequency Domains

Ionut-Gabriel Farcas, Virginia Tech
Shane A. McQuarrie, Sandia National Laboratories
Steffen Werner, Virginia Tech

March 03–07, 2025
SIAM Conference on Computational Science & Engineering, Fort Worth, TX

SAND2025-02461C

Introduction: Learning Dynamical Systems from Data

The Problem: High-dimensional Models / Data

Problem: We have a expensive computational model of a physical process, called the full-order model (FOM):

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) &= \mathbf{f}(t, \mathbf{q}, \mathbf{u}), \quad \mathbf{q}(t_0) = \mathbf{q}_0, \\ \mathbf{y}(t) &= \mathbf{g}(t, \mathbf{q}, \mathbf{u}) \end{aligned} \tag{FOM}$$

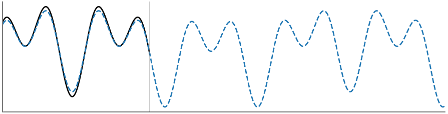
with high-dimensional state $\mathbf{q}(t) \in \mathbb{R}^n$ and input $\mathbf{u}(t) \in \mathbb{R}^u$ and output $\mathbf{y}(t) \in \mathbb{R}^\ell$.
The FOM is so large that we can only afford to solve it for a short time window or for a limited number of initial conditions.

Goal: Given a training data set, construct a computationally efficient surrogate—a reduced-order model (ROM)—to solve in place of the FOM.

Goals of Data-driven Model Reduction

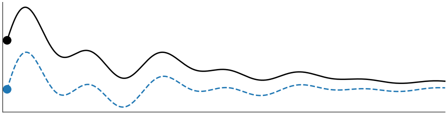
Prediction in time

$[t_0, t_f] \to [t_0, t_f']$



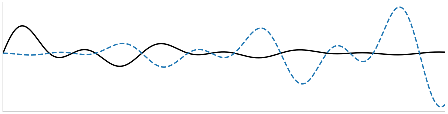
Prediction to new initial conditions

$\mathbf{q}_0 \to \mathbf{q}_0'$



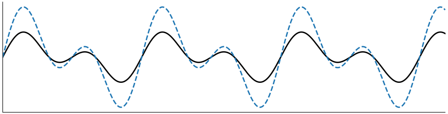
Prediction to new inputs

$\mathbf{u}(t) \to \mathbf{u}'(t)$



Prediction w.r.t. system parameters

$\mathbf{f}(t, \mathbf{q}, \mathbf{u}; \boldsymbol{\mu}) \to \mathbf{f}(t, \mathbf{q}, \mathbf{u}; \boldsymbol{\mu}')$



Classical Model Reduction Uses Intrusive Information

Galerkin projection is a classical approach to constructing a reduced-order model.

1. Construct an orthonormal basis matrix $\mathbf{V}_r \in \mathbb{R}^{n\times r}$.
2. Approximate $\mathbf{q}(t) \approx \mathbf{V}_r\hat{\mathbf{q}}(t)$, where $\hat{\mathbf{q}}(t) \in \mathbb{R}^r$ are the latent variables with $r \ll n$.
3. Substitute the approximation into the FOM and use orthogonality:

$$\mathbf{V}_r^{\mathsf{T}}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{V}_r\hat{\mathbf{q}}(t) = \mathbf{f}(t, \mathbf{V}_r\hat{\mathbf{q}}, \mathbf{u})\right) \quad \longrightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}(t, \hat{\mathbf{q}}, \mathbf{u}) := \mathbf{V}_r^{\mathsf{T}}\mathbf{f}(t, \mathbf{V}_r\hat{\mathbf{q}}, \mathbf{u}) \quad \text{(ROM)}$$

4. Solve the ROM defined by $\hat{\mathbf{f}}$ with initial condition $\hat{\mathbf{q}}(t_0) = \mathbf{V}_r^{\mathsf{T}}\mathbf{q}_0$
5. The ROM solution approximates the true solution, $\mathbf{q}(t) \approx \mathbf{q}_{\mathrm{rom}}(t) := \mathbf{V}_r\hat{\mathbf{q}}(t)$.

The problem: The ROM operator $\hat{\mathbf{f}}$ depends explicitly on the FOM operator $\mathbf{f}$.
Can we construct $\hat{\mathbf{f}}$ without direct access to $\mathbf{f}$ if we do have access to data?

Classical Model Reduction Uses Intrusive Information

$$\mathbf{V}_r^\mathsf{T}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{V}_r\hat{\mathbf{q}}(t) = \mathbf{f}(t, \mathbf{V}_r\hat{\mathbf{q}}, \mathbf{u})\right) \quad \longrightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}(t, \hat{\mathbf{q}}, \mathbf{u}) := \mathbf{V}_r^\mathsf{T}\mathbf{f}(t, \mathbf{V}_r\hat{\mathbf{q}}, \mathbf{u})$$

Example: In linear-time invariant (LTI) systems,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}(t, \mathbf{q}, \mathbf{u}) = \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t)$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$. The Galerkin ROM is

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \mathbf{V}_r^\mathsf{T}\mathbf{f}(t, \mathbf{V}_r\hat{\mathbf{q}}, \mathbf{u}) = \underbrace{\mathbf{V}_r^\mathsf{T}\mathbf{A}\mathbf{V}_r}_{\hat{\mathbf{A}}}\hat{\mathbf{q}}(t) + \underbrace{\mathbf{V}_r^\mathsf{T}\mathbf{B}}_{\hat{\mathbf{B}}}\mathbf{u}(t)$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{B}} \in \mathbb{R}^{r\times m}$.

Learning Reduced-order Models from Data

Our goal: Construct a suitable ROM from data, *without* direct access to a FOM.

$$\text{(FOM)} \qquad \cancel{\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) &= \mathbf{f}(t, \mathbf{q}, \mathbf{u}), \\ \mathbf{y}(t) &= \mathbf{g}(t, \mathbf{q}, \mathbf{u}) \end{aligned}} \qquad\qquad \begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) &= \hat{\mathbf{f}}(t, \hat{\mathbf{q}}, \mathbf{u}) \\ \mathbf{y}(t) &= \hat{\mathbf{g}}(t, \hat{\mathbf{q}}, \mathbf{u}) \end{aligned} \qquad \text{(ROM)}$$

Perspective 1: we want a time domain representation of the ROM.
Data are states or outputs, at various time instances, corresponding to known inputs.

$$\mathbf{u}(t) \quad \rightarrow \quad \left[\begin{array}{c|c|c|c} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{n_t-1} \end{array}\right] \in \mathbb{R}^{n \times n_t} \quad \text{or} \quad \left[\begin{array}{c|c|c|c} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{n_t-1} \end{array}\right] \in \mathbb{R}^{\ell \times n_t}$$

Perspective 2: we want a frequency domain representation of the ROM (next time).

Part 1: Learning from Time-domain State Space Data

Time-domain Methods for Data-driven Model Reduction

- **Dynamic mode decomposition** (DMD)
  [Schmid, 2010, Kutz et al., 2016, Schmid, 2022]
- **Operator inference** (OpInf)
  [Peherstorfer and Willcox, 2016, Ghattas and Willcox, 2021, Kramer et al., 2024]
- **Sparse Identification of Nonlinear Dynamics** (SINDy)
  [Brunton et al., 2016, Schaeffer, 2017, Brunton and Kutz, 2022]
- **Eigenvalue realization algorithm** (ERA, a.k.a. subspace identification)
  [Kramer and Gugercin, 2016, Kung, 1978]
- **Physics-informed neural networks** (PINNs)
  [Raissi et al., 2019, Karniadakis et al., 2021]
- Neural operators: **DeepONet**, **Fourier Neural Operator** (FNO), etc.
  [Lu et al., 2021, Kovachki et al., 2023]

Dynamic Mode Decomposition (DMD)

Based on Koopman theory, standard DMD assumes a discrete, linear relationship

$$\mathbf{q}_{j+1} \approx \mathbf{A}\mathbf{q}_j, \qquad \mathbf{A} = \mathrm{argmin}_{\bar{\mathbf{A}}} \sum_j \left\| \mathbf{q}_{j+1} - \bar{\mathbf{A}}\mathbf{q}_j \right\|_2^2,$$

where $\mathbf{q}_j = \mathbf{q}(t_j)$. A DMD ROM is defined as

$$\hat{\mathbf{q}}_{j+1} \approx \hat{\mathbf{A}}\hat{\mathbf{q}}_j, \qquad \hat{\mathbf{A}} = \mathbf{V}_r^{\mathsf{T}}\mathbf{A}\mathbf{V}_r = \mathbf{V}_r^{\mathsf{T}}\mathbf{Q}'\mathbf{W}_r\boldsymbol{\Sigma}_r \tag{ROM}$$

where

$$\mathbf{Q} = \left[\begin{array}{c|c|c|c} \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{n_t-2} \end{array}\right], \qquad \mathbf{Q}' = \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_{n_t-1} \end{array}\right].$$

and $\mathbf{Q} \approx \mathbf{V}_r\boldsymbol{\Sigma}_r\mathbf{W}_r^{\mathsf{T}}$ is the truncated SVD of $\mathbf{Q}$ with r retained singular modes.

Dynamic Mode Decomposition (DMD)

Advantages:

- Small, linear system—easy to analyze and work with
- Provides a spatiotemporal modal decomposition (coherent structures)
- Straightforward extensions to ODEs, noisy data, and some nonlinear dynamics

Disadvantages:

- Linear dynamics cannot always approximate nonlinear phenomena well
- Tends to perform poorly for transient dynamics
- For nonlinear dynamics, insight is typically required and the resulting linear system can be very large

Package: PyDMD

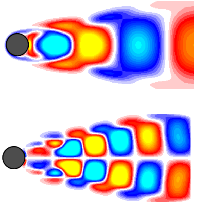


Figure: [Kutz et al., 2016]

Operator Inference (OpInf)

Galerkin projection preserves polynomial nonlinear forms. For a polynomial FOM, OpInf poses a ROM with the same polynomial form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{c} + \mathbf{A}\mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)] + \cdots + \mathbf{B}\mathbf{u}(t), \qquad \text{(FOM)}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] + \cdots + \hat{\mathbf{B}}\mathbf{u}(t). \qquad \text{(ROM)}$$

Reduced "operators" $\hat{\mathbf{c}}, \hat{\mathbf{A}}, \hat{\mathbf{H}}, \ldots, \hat{\mathbf{B}}$ are learned by minimizing the ROM equation residual with respect to observed states:

$$\hat{\mathbf{c}}, \hat{\mathbf{A}}, \hat{\mathbf{H}}, \ldots, \hat{\mathbf{B}} = \underset{\bar{\mathbf{c}}, \bar{\mathbf{A}}, \bar{\mathbf{H}}, \ldots, \bar{\mathbf{B}}}{\operatorname{argmin}} \sum_j \left\| \dot{\hat{\mathbf{q}}}_j - (\bar{\mathbf{c}} + \bar{\mathbf{A}}\hat{\mathbf{q}}_j + \bar{\mathbf{H}}[\hat{\mathbf{q}}_j \otimes \hat{\mathbf{q}}_j] + \cdots + \bar{\mathbf{B}}\mathbf{u}_j) \right\|_2^2,$$

where $\hat{\mathbf{q}}_j = \mathbf{V}_r^{\mathsf{T}}\mathbf{q}_j$ and $\dot{\hat{\mathbf{q}}}_j$ is an estimate of its time derivative.

Operator Inference (OpInf)

Advantages:

- Inference problem is linear least squares
- Covers a wide range of nonlinear phenomena
- Has extensions to discrete systems, noisy data, parametric problems, etc.

Disadvantages:

- Tends to perform poorly for transient dynamics
- Sensitive to the accuracy of the time derivative estimates $\dot{\hat{\mathbf{q}}}_j \approx \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t)|_{t=t_j}$

Package: `opinf`

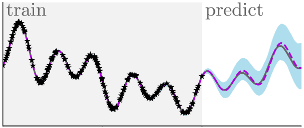


Figure: [McQuarrie et al., 2025]

Sparse Identification of Nonlinear Dynamics (SINDy)

For dynamics with unknown structure, SINDy uses sparsity-promoting (L^1) regression of (reduced) states to select coefficients for a library of nonlinear terms:

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) \approx \boldsymbol{\Xi}^{\mathsf{T}}\boldsymbol{\theta}(\hat{\mathbf{q}}(t)), \qquad \text{(ROM)}$$

$$\boldsymbol{\Xi} = [\,\boldsymbol{\xi}_1 \quad \cdots \quad \boldsymbol{\xi}_r\,], \qquad \boldsymbol{\xi}_i = \underset{\bar{\boldsymbol{\xi}}}{\operatorname{argmin}} \|\dot{\hat{\mathbf{q}}}^{(i)} - \boldsymbol{\Theta}(\hat{\mathbf{Q}})\bar{\boldsymbol{\xi}}\|_2 + \lambda\|\bar{\boldsymbol{\xi}}\|_1,$$

where

$$\begin{aligned}
\dot{\hat{\mathbf{q}}}^{(i)} &= [\,\dot{\hat{q}}_i(t_0) \quad \cdots \quad \dot{\hat{q}}_i(t_{n_t-1})\,]^{\mathsf{T}}, \\
\boldsymbol{\Theta}(\hat{\mathbf{Q}}) &= [\,\boldsymbol{\theta}(\hat{\mathbf{q}}_0) \quad \cdots \quad \boldsymbol{\theta}(\hat{\mathbf{q}}_{n_t-1})\,]^{\mathsf{T}}, \\
\boldsymbol{\theta}(\hat{\mathbf{q}}) &= [\,1 \quad \hat{\mathbf{q}}^{\mathsf{T}} \quad (\hat{\mathbf{q}} \otimes \hat{\mathbf{q}})^{\mathsf{T}} \quad \cdots \quad \sin(\hat{\mathbf{q}})^{\mathsf{T}} \quad \cos(\hat{\mathbf{q}})^{\mathsf{T}} \quad \cdots\,]^{\mathsf{T}}.
\end{aligned}$$

Sparse Identification of Nonlinear Dynamics (SINDy)

Advantages:

- Can learn a fully nonlinear model
- Resulting models are parsimonious (few terms), interpretable
- Has extensions to control systems, noisy data, parametric problems, etc.

Disadvantages:

- Library of candidate terms must be specified by hand
- Regression problem, though sparse, can be very large
- Sensitive to the accuracy of the time derivative estimates

Package: PySINDy

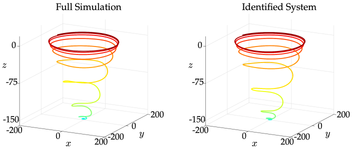


Figure: [Brunton et al., 2016]

Eigenvalue Realization (ERA) / Subspace Identification

These methods identify/learn linear input-output systems starting at rest,

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) &= \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{q}(t_0) = \mathbf{0}, \\
\mathbf{y}(t) &= \mathbf{C}\mathbf{q}(t) + \mathbf{D}\mathbf{u}(t).
\end{aligned}
$$

Markov parameters $\mathbf{M}_1, \mathbf{M}_2, \ldots$ can be estimated from long enough time series by solving a least-squares problem based on the solution formula

$$
\mathbf{y}(k) = \sum_{i=1}^{k} \mathbf{C}\mathbf{A}^{i-1}\mathbf{B}\mathbf{u}(i) + \mathbf{D}\mathbf{u}(k) = \sum_{i=1}^{k} \mathbf{M}_i \mathbf{u}(i) + \mathbf{D}u(k)
$$

The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are estimated from the Hankel matrix:

$$
\mathcal{H} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \mathbf{M}_3 & \cdots \\ \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 & \cdots \\ \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}
$$

Eigenvalue Realization (ERA) / Subspace Identification

Advantages

- Allows full system construction only from input-output pairs; no state data involved
- Methods yield error bounds on given data
- Preserves stability of the underlying system
- Has been extended to general noise models, etc.

Disadvantages

- Hankel matrices can be ill-conditioned
- System order depends on rank decision over estimated Markov parameters
- Assumes the data corresponds to a linear system

Machine Learning Methods

Physics Informed Neural Networks (PINNs) aim to solve

$$u_t + \mathcal{N}(u) = 0, \quad x \in \Omega, \quad t \in [0, T]$$

by approximating $\mathcal{N}$ with a neural network (NN). The standard NN loss function is augmented with the PDE residual and residuals for initial and boundary conditions.

Neural Operators, such as **DeepONet** and **Fourier Neural Operator**, establish a NN architecture designed to learn maps between infinite dimensional function spaces. Given a set of coefficients/boundary conditions, the aim is to learn the solution function of a (parametric) PDE.

Practical Considerations for Data-driven Modeling

- **Data availability and quality**
  How dense are the data in time? How many data samples are required?
  Are the data trustworthy, or is there observational noise?
- **Data transformations**
  What variables should the data be expressed in? Do all entries have the same units?
  How are multiple state variables combined?
- **Dimension reduction**
  How should the states be approximated with only a few degrees of freedom?
  How many degrees of freedom are needed? How do we avoid bias?
- **Model structure**
  What kind of model should be learned? What structure or properties should it have?
- **Computational scalability**
  What happens if the state dimension is very large ($n \sim 10^8$)?

Demonstration: Compressible Euler Flow of an Ideal Gas (1D)

`TimeDomain/CompressibleEuler1D/demo.ipynb`

Euler: Governing Equations

The following Euler equations model the compressible flow of an ideal gas with periodic boundary conditions in the 1D spatial domain $\Omega = [0, L]$.

$$\frac{\partial \vec{q}_{\text{c}}}{\partial t} = \frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho v \\ \rho e \end{bmatrix} = -\frac{\partial}{\partial x}\begin{bmatrix} \rho v \\ \rho v^2 + p \\ (\rho e + p) v \end{bmatrix}$$

$$\vec{q}_{\text{c}}(0, t) = \vec{q}_{\text{c}}(L, t) \qquad \vec{q}_{\text{c}}(x, t_0) = \vec{q}_{\text{c},0}(x)$$

	name	units
ρ	density	kg/m^3
v	velocity	m/s
e	internal energy	m^2/s^2
p	pressure	kg / m·s^2

The state variable are related via the ideal gas law

$$\rho e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2, \qquad \gamma = 1.4 \text{ (heat capacity ratio).}$$

Note that the dynamics are non-polynomially nonlinear with respect to ρ, ρv, and ρe.

Euler: Full-order Model and Training Data

The FOM discretizes the spatial derivatives with finite differences over an equidistant grid $0 = x_0 < x_1 < \cdots < x_{n_x} = L$.

$$\mathbf{q}_{\mathsf{c}}(t) = \begin{bmatrix} \boldsymbol{\rho}(t) \\ \boldsymbol{\rho v}(t) \\ \boldsymbol{\rho e}(t) \end{bmatrix} = \begin{bmatrix} \rho(x_0, t) \\ \vdots \\ \rho(x_{n_x - 1}, t) \\ (\rho v)(x_0, t) \\ \vdots \\ (\rho v)(x_{n_x - 1}, t) \\ (\rho e)(x_0, t) \\ \vdots \\ (\rho e)(x_{n_x - 1}, t) \end{bmatrix}$$

Training trajectories are generated for 16 initial conditions over a limited time domain $t \in [0, t_{\mathsf{obs}}]$ with 200 time steps after the initial condition.

$$\mathbf{Q}_{\mathsf{c}}^{(1)} = \begin{bmatrix} \mathbf{q}_{\mathsf{c},0}^{(1)} & \cdots & \mathbf{q}_{\mathsf{c},200}^{(1)} \end{bmatrix}$$

$$\vdots$$

$$\mathbf{Q}_{\mathsf{c}}^{(16)} = \begin{bmatrix} \mathbf{q}_{\mathsf{c},0}^{(16)} & \cdots & \mathbf{q}_{\mathsf{c},200}^{(16)} \end{bmatrix}$$

Review: Operator Inference

For a polynomial FOM, OpInf poses a ROM with the same polynomial form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)] \qquad \text{(FOM)}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] \qquad \text{(ROM)}$$

The reduced "operator" $\hat{\mathbf{H}}$ is learned by minimizing the ROM equation residual with respect to observed states:

$$\hat{\mathbf{H}} = \underset{\bar{\mathbf{H}}}{\operatorname{argmin}} \sum_{j} \left\| \dot{\hat{\mathbf{q}}}_j - \bar{\mathbf{H}}[\hat{\mathbf{q}}_j \otimes \hat{\mathbf{q}}_j] \right\|^2, \qquad \hat{\mathbf{q}}_j = \mathbf{V}_r^{\mathsf{T}}\mathbf{q}_j$$

This problem has no inputs ($\mathbf{u}(t) \equiv 0$) and—after some work—only quadratic state terms.

Euler: Lift to a Polynomial Form

Problem: OpInf constructs ROMs with polynomial structure, but the dynamics of this system are non-polynomially nonlinear with respect to the state $\vec{q}_{\text{c}} = (\rho, \rho v, \rho e)$.

Solution: Let $\vec{q} = (v, p, \zeta)$, where $\zeta = 1/\rho$ is the specific volume [m^3/kg]. Then

$$\frac{\partial \vec{q}_{\text{c}}}{\partial t} = -\frac{\partial}{\partial x}\begin{bmatrix} \rho v \\ \rho v^2 + p \\ (\rho e + p) v \end{bmatrix} \qquad \longrightarrow \qquad \frac{\partial \vec{q}}{\partial t} = \frac{\partial}{\partial t}\begin{bmatrix} v \\ p \\ \zeta \end{bmatrix} = \begin{bmatrix} -v\frac{\partial v}{\partial x} - \zeta\frac{\partial p}{\partial x} \\ -\gamma p\frac{\partial v}{\partial x} - v\frac{\partial p}{\partial x} \\ -v\frac{\partial \zeta}{\partial x} + \zeta\frac{\partial v}{\partial x} \end{bmatrix}.$$

The transformed system is quadratic with respect to $\vec{q} = (v, p, \zeta)$.

Euler: Lift to a Polynomial Form

Problem: OpInf constructs ROMs with polynomial structure, but the dynamics of this system are non-polynomially nonlinear with respect to the state $\vec{q}_{\text{c}} = (\rho, \rho v, \rho e)$.

Solution: Transform the training data from $\vec{q}_{\text{c}} = (\rho, \rho v, \rho e)$ to $\vec{q} = (v, p, \zeta)$:

$$\mathbf{Q}_{\text{c}}^{(i)} = \left[\begin{array}{c} \mathbf{Q}_{\boldsymbol{\rho}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{\rho v}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{\rho e}}^{(i)} \end{array}\right] \mapsto \left[\begin{array}{c} \mathbf{Q}_{\boldsymbol{v}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{p}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{\zeta}}^{(i)} \end{array}\right] = \mathbf{Q}^{(i)}, \qquad i = 1, \ldots, 16.$$

This motivates learning a ROM with quadratic structure.

Euler: Center and/or Scale Data

Problem: The values of v, p, and ζ have very different units and scales.

	name	units	minimum	maximum
v	velocity	m/s	9.38×10^{1}	1.06×10^{2}
p	pressure	kg/ms^2	9.07×10^{4}	1.10×10^{5}
ζ	specific volume	m^3/kg	3.81×10^{-2}	5.46×10^{-2}

Solution: Non-dimensionalize (scale) the training data by normalizing each variable by an appropriate characteristic scale.

$$\mathbf{Q}^{(i)} = \begin{bmatrix} \mathbf{Q}_{\boldsymbol{v}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{p}}^{(i)} \\ \mathbf{Q}_{\boldsymbol{\zeta}}^{(i)} \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{Q}_{\boldsymbol{v}}^{(i)} \;/\; (10^2\ \text{m/s}) \\ \mathbf{Q}_{\boldsymbol{p}}^{(i)} \;/\; (10^5\ \text{kg/ms}^2) \\ \mathbf{Q}_{\boldsymbol{\zeta}}^{(i)} \;/\; (10^{-1}\ \text{m}^3\text{/kg}) \end{bmatrix}, \quad i = 1, \ldots, 16.$$

Euler: Reduce Data Dimensionality

Problem: The transformed, scaled state variable $\mathbf{q}(t) \in \mathbb{R}^n$ is high-dimensional.[1]

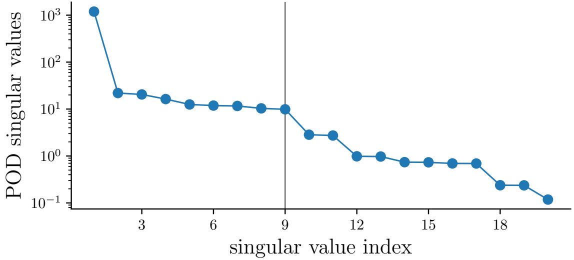
Solution: Approximate with a low-dimensional state representation,

$$\mathbf{q}(t) \approx \mathbf{V}_r \hat{\mathbf{q}}(t), \qquad \mathbf{V}_r \in \mathbb{R}^{n \times r}, \;\; \hat{\mathbf{q}}(t) \in \mathbb{R}^r, \;\; r \ll n.$$

The usual choice for the basis matrix $\mathbf{V}_r$ is proper orthogonal decomposition (POD):

$$\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^\mathsf{T} = \mathrm{svd}\left(\left[\, \mathbf{Q}^{(1)} \;\; \cdots \;\; \mathbf{Q}^{(16)} \,\right]\right),$$

$$\mathbf{V}_r := \boldsymbol{\Phi}_{:,:r}.$$



The singular values $\mathrm{diag}(\boldsymbol{\Sigma})$ can inform the choice for r.

[1] Well, $n = 3n_x = 600$ is not large, but in larger applications n can be on the order of 10^6–10^9.

Euler: Reduce Data Dimensionality

Problem: The transformed, scaled state variable $\mathbf{q}(t) \in \mathbb{R}^n$ is high-dimensional.

Solution: With $\mathbf{V}_r \in \mathbb{R}^{n\times r}$ fixed, compress the data by minimizing approximation error:

$$\mathbf{q}(t) \approx \mathbf{V}_r\hat{\mathbf{q}}(t) \qquad \longrightarrow \qquad \hat{\mathbf{Q}}^{(i)} = \text{argmin}_{\hat{\mathbf{Q}}} \left\|\mathbf{Q}^{(i)} - \mathbf{V}_r\hat{\mathbf{Q}}\right\|_F = \mathbf{V}_r^{\mathsf{T}}\mathbf{Q}^{(i)}.$$

Sanity Check: Compute the reconstruction error,

$$\left\|\mathbf{Q}^{(i)} - \mathbf{V}_r\hat{\mathbf{Q}}^{(i)}\right\|_F = \left\|\mathbf{Q}^{(i)} - \mathbf{V}_r\mathbf{V}_r^{\mathsf{T}}\mathbf{Q}^{(i)}\right\|_F.$$

This should be small for the data as a whole **and** for individual state variables.

Euler: Learn Reduced Operators from Data

The OpInf ROM is defined by a matrix $\hat{\mathbf{H}}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)], \qquad \hat{\mathbf{H}} = \underset{\bar{\mathbf{H}}}{\operatorname{argmin}} \sum_{j=1}^{K} \left\| \bar{\mathbf{H}}[\hat{\mathbf{q}}_j \otimes \hat{\mathbf{q}}_j] - \dot{\hat{\mathbf{q}}}_j \right\|_2^2 + \lambda^2 \left\| \bar{\mathbf{H}} \right\|_F^2 .$$

We need time derivative estimates $\dot{\hat{\mathbf{q}}}_j \approx \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t)|_{t=t_j}$, e.g., via finite differences:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t)\right|_{t=t_j} \approx \frac{-25\hat{\mathbf{q}}(t_j) + 48\hat{\mathbf{q}}(t_j+\delta t) - 36\hat{\mathbf{q}}(t_j+2\delta t) + 16\hat{\mathbf{q}}(t_j+3\delta t) - 3\hat{\mathbf{q}}(t_j+4\delta t)}{12\delta t}.$$

This fourth-order forward difference does not give estimates for the last 4 time steps.

$$\mathrm{ddt}(\hat{\mathbf{Q}}^{(i)}) \mapsto \hat{\mathbf{Q}}^{(i)}, \dot{\hat{\mathbf{Q}}}^{(i)}, \quad i = 1, \ldots, 16$$

$$\hat{\mathbf{Q}} = \left[\ \hat{\mathbf{Q}}^{(1)} \ \cdots \ \hat{\mathbf{Q}}^{(16)}\ \right]$$
$$\dot{\hat{\mathbf{Q}}} = \left[\ \dot{\hat{\mathbf{Q}}}^{(1)} \ \cdots \ \dot{\hat{\mathbf{Q}}}^{(16)}\ \right]$$

Euler: Learn Reduced Operators from Data

The OpInf ROM is defined by a matrix $\hat{\mathbf{H}}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)], \qquad \begin{aligned}\hat{\mathbf{H}} &= \operatorname*{argmin}_{\bar{\mathbf{H}}} \sum_j \left\| \bar{\mathbf{H}}[\hat{\mathbf{q}}_j \otimes \hat{\mathbf{q}}_j] - \dot{\hat{\mathbf{q}}}_j \right\|_2^2 + \lambda^2 \left\| \bar{\mathbf{H}} \right\|_F^2 . \\ &= \operatorname*{argmin}_{\bar{\mathbf{H}}} \left\| (\hat{\mathbf{Q}} \odot \hat{\mathbf{Q}})^\mathsf{T} \bar{\mathbf{H}}^\mathsf{T} - \dot{\hat{\mathbf{Q}}}^\mathsf{T} \right\|_2^2 + \lambda^2 \left\| \bar{\mathbf{H}} \right\|_F^2 ,\end{aligned}$$

where $\odot$ applies $\otimes$ columnwise.

Problem: Representation of $\hat{\mathbf{H}}$ is not unique.
Solution: Use a "compressed" Kronecker product.

Form $\mathbf{D} = (\hat{\mathbf{Q}} \odot \hat{\mathbf{Q}})^\mathsf{T}$ and solve

$$\left(\mathbf{D}^\mathsf{T}\mathbf{D} + \lambda^2 \mathbf{I}\right) \hat{\mathbf{H}}^\mathsf{T} = \mathbf{D}^\mathsf{T} \dot{\hat{\mathbf{Q}}}^\mathsf{T}$$

Euler: Summary

FOM data must be pre-processed:

- Lift/transform to a polynomial form
- Center and/or scale
- Reduce data dimensionality
- Learn reduced operators from data

ROM solutions must be post-processed:

- Decompress reduced states
- Unscale and/or uncenter
- Unlift/untransform to original form
- Sanity check: compare to training data

Demonstration: Vortex-shedding Flow Past a Cylinder (2D)

`TimeDomain/VortexShedding2D/demo.ipynb`

Addressing Scalability Bottlenecks

Problem: In large-scale problems, the snapshot dimension can be extremely large (It may not even be possible to load the full snapshots into memory!)

Solution 1: **Domain decomposition** [Farcas et al., 2024b]

- Split the domain into overlapping regions
- Localize the reduced basis in space
- Couple data-driven models for each subdomain

Solution 2: **Distributed memory computing** [Farcas et al., 2024a]

- Process data and learn models in parallel
- Key step: POD through the method of snapshots [Sirovich, 1987]

Additional tutorial: `github.com/ionutfarcas/distributed_Operator_Inference`

Summary

Data-driven ROMs are derived from data, not from FOM operators

This part: **Time domain perspective**

- Data are the states or outputs at various times
- Data should be pre-processed appropriately
- ROM structure should be designed with the true dynamical structure in mind

Next part: **Frequency domain perspective**

Part 2: Learning from Frequency Data

Transforming to the Frequency Domain with the Laplace Transform

For a given time domain signal $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, the Laplace transform of f is the function[2]

$$F(s) := \mathcal{L}\{f\} = \int_0^\infty f(t)e^{-st}\,\mathrm{d}t,$$

so that $F\colon \mathbb{C} \to \mathbb{C}^n$. The frequency domain lives in the complex numbers.

Time domain system

$$\mathbf{E}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{q}(0) = \mathbf{0}$$
$$\mathbf{y}(t) = \mathbf{C}\mathbf{q}(t)$$

Differential equations

$$\xrightarrow{\mathcal{L}}$$

Frequency domain system

$$s\mathbf{E}\mathbf{Q}(s) = \mathbf{A}\mathbf{Q}(s) + \mathbf{B}\mathbf{U}(s)$$
$$\mathbf{Y}(s) = \mathbf{C}\mathbf{Q}(s)$$

Algebraic equations

$$\mathbf{Q} = \mathcal{L}\{\mathbf{q}\}, \quad \mathbf{U} = \mathcal{L}\{\mathbf{u}\}, \quad \mathbf{Y} = \mathcal{L}\{\mathbf{y}\}$$

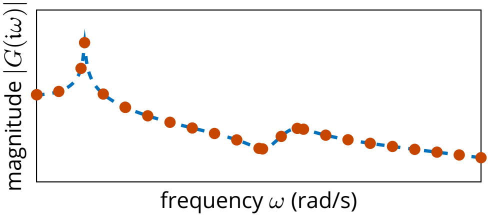
[2] The discrete-time equivalent is called the Z-transform and works analogously.

Transfer Function Data

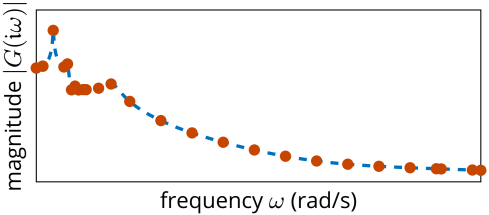
The transfer function $G\colon \mathbb{C} \to \mathbb{C}^{p\times m}$ describes the input-to-output behavior of linear systems in the frequency domain

$$\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s) = \left(\mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\right)\mathbf{U}(s).$$

Frequency-domain data are transfer function evaluations $\{(\omega_i, \mathbf{g}_i = \mathbf{G}(\mathrm{j}\omega_i))\}_{i=1}^{N}$.



(a) Data from an artificial fishtail.



(b) Data from a butterfly gyroscope.

Frequency vs. Time Domain

Equivalence of Representations

The time and frequency domain descriptions of systems are equivalent and high accuracy in either domain ensures overall high accuracy.

	Time domain	Frequency domain
Problem setting	• fit differential/difference equations	• fit (rational) complex functions
Data	• time series data	• function evaluations
Noise effects	• carries through time evolution	• isolated in frequency points
Dynamics types	• suited for linear and nonlinear dynamics	• suited for linear dynamics (nonlinear dynamics are currently researched)

Frequency-domain Methods for Data-driven MOR

- **Loewner Framework**
  [Antoulas and Anderson, 1986, Mayo and Antoulas, 2007]
- **Vector Fitting** (VF)
  [Gustavsen and Semlyen, 1999, Drmač et al., 2015]
- **Rational Krylov Fitting** (RKFIT)
  [Berljafa and Güttel, 2017, Elsworth and Güttel, 2019]
- **Adaptive Antoulas-Anderson** (AAA)
  [Nakatsukasa et al., 2018, Gosea and Güttel, 2021]
- **Quadrature-based reduced-order modeling**
  [Gosea et al., 2022]
- **Optimization-based reduced-order modeling**
  [Hund et al., 2022, Mlinarić and Gugercin, 2023, Schwerdtner and Voigt, 2023]

Loewner framework

The Loewner framework constructs frequency domain models $\widehat{\mathbf{G}}$ via interpolation:
Given the data $\{(\mu_i, \mathbf{g}_i)\}_i$, find $\widehat{\mathbf{G}} = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}$ so that

$$\widehat{\mathbf{G}}(\mu_i) = \mathbf{g}_i \quad \text{for all } i.$$

The model is constructed directly from data via Loewner matrices

$$\mathbb{L} = \begin{bmatrix} \frac{g_{\ell,1} - g_{\mathrm{r},1}}{\mu_{\ell,1} - \mu_{\mathrm{r},1}} & \frac{g_{\ell,1} - g_{\mathrm{r},2}}{\mu_{\ell,1} - \mu_{\mathrm{r},2}} & \cdots \\ \frac{g_{\ell,2} - g_{\mathrm{r},1}}{\mu_{\ell,2} - \mu_{\mathrm{r},1}} & \frac{g_{\ell,2} - g_{\mathrm{r},2}}{\mu_{\ell,2} - \mu_{\mathrm{r},2}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \qquad \mathbb{L}_{\mathrm{s}} = \begin{bmatrix} \frac{\mu_{\ell,1} g_{\ell,1} - \mu_{\mathrm{r},1} g_{\mathrm{r},1}}{\mu_{\ell,1} - \mu_{\mathrm{r},1}} & \frac{\mu_{\ell,1} g_{\ell,1} - \mu_{\mathrm{r},2} g_{\mathrm{r},2}}{\mu_{\ell,1} - \mu_{\mathrm{r},2}} & \cdots \\ \frac{\mu_{\ell,2} g_{\ell,2} - \mu_{\mathrm{r},1} g_{\mathrm{r},1}}{\mu_{\ell,2} - \mu_{\mathrm{r},1}} & \frac{\mu_{\ell,2} g_{\ell,2} - \mu_{\mathrm{r},2} g_{\mathrm{r},2}}{\mu_{\ell,2} - \mu_{\mathrm{r},2}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix},$$

$$\mathbf{B}_{\mathbb{L}} = \begin{bmatrix} g_{\ell,1} \\ g_{\ell,2} \\ \vdots \end{bmatrix}, \qquad \mathbf{C}_{\mathbb{L}} = \begin{bmatrix} g_{\mathrm{r},1} & g_{\mathrm{r},2} & \cdots \end{bmatrix}.$$

Loewner framework

Advantages

- Can exactly identify unknown linear systems
- Easy and efficient construction of approximations
- Extensions for preserving system structures and properties (stability, passivity, ...)

Disadvantages

- Rank reductions break theoretic results
- Approximation error away from interpolation points can be large
- Interpolation is not suited for noisy data

Demonstration: Mass-Spring-Damper System

FrequencyDomain/MassSpringDamper/demo.ipynb

Demonstration: Thermal Diffusion System

`FrequencyDomain/ThermalDiffusion/demo.ipynb`

Quadrature-based methods

These methods are based on solving integrals appearing in model order reduction via quadrature rules, for example, in balanced truncation we need to compute

$$P = \int_{-\infty}^{\infty} (\mathrm{j}\omega \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}\mathbf{B}^{\mathsf{T}} (-\mathrm{j}\omega \mathbf{E} - \mathbf{A})^{-\mathsf{T}} \,\mathrm{d}\,\omega$$

$$\approx \sum_{k=1}^{N} \phi_k (\mathrm{j}\omega_k \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}\mathbf{B}^{\mathsf{T}} (-\mathrm{j}\omega_k \mathbf{E} - \mathbf{A})^{-\mathsf{T}}.$$

Reduced-order models $\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{C}}, \widehat{\mathbf{E}}$ can be directly written from data and quadrature weights using Loewner matrices.

Quadrature-based methods

Advantages

- Bounds on the approximation error exist
- Sophisticated reduced-order models can be constructed without intermediate medium-scale approximations
- Properties of intrusive model reduction methods can carry over (stability, passivity, ...)

Disadvantages

- High quadrature accuracy is needed and depends on unknown problem structure
- Loses most theoretical guarantees when applied to fixed data
- Intermediate Loewner matrices can become very large

Least-squares Methods

General problem: Least-squares methods aim to compute transfer functions (or related objects) to fit the data in a least-squares sense

$$\min \sum_{j=1}^{N} \|g_j - \mathbf{G}(\mu_j)\|_F^2.$$

- In vector fitting, the transfer function is reformulated in barycentric form

$$\widehat{\mathbf{G}}(s) = \left(\sum_{k=1}^{r} \frac{\alpha_{\mathbf{k}}}{s - \lambda_k} \right) \bigg/ \left(1 + \sum_{k=1}^{r} \frac{\beta_k}{s - \lambda_k} \right)$$

  and then fitted via dynamically weighted linear least squares.
- In RKFIT, RKFUNs are fitted to the data using rational Krylov subspaces iteratively. RKFUNs can then be transformed into transfer functions.

Least-squares methods

Advantages

- Works well with noisy data
- Fast run times due to intermediate linearizations

Disadvantages

- Non-robust convergence behavior
- Can get stuck in suboptimal local minima

Package: `RKFIT Toolbox`

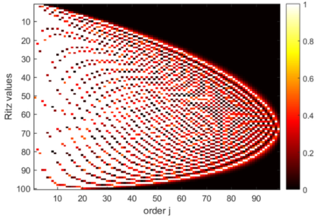


Figure: RKFIT Website

Optimization-based methods

Optimization methods generalize least-squares approaches to brute-force fit system parameters to given data.

Advantages

- Can provide current approximation error during execution
- Different system structures can be included

Disadvantages

- Expensive training costs (time and memory) for brute force optimization
- Easily stuck in suboptimal local minima

AAA Algorithm

The Adaptive Antoulas-Anderson (AAA) algorithm is based on the idea of combining the best of interpolation and least-squares:

- First, interpolate in data points that maximize the approximation error
- Second, fit the remaining data in a linear least-squares sense

The key component for the algorithm's performance is the interpolatory barycentric form

$$\widehat{\mathbf{G}} = \frac{\sum_{j=1}^{r} \frac{\mathbf{g}_j w_j}{s - \lambda_j}}{1 + \sum_{j=1}^{r} \frac{w_j}{s - \lambda_j}},$$

which interpolates selected points by construction but has leftover degrees of freedom for the least-squares fit.

AAA Algorithm

Advantages

- Incorporates the complete data set without overparametrization
- Provides accuracy in crucial points via interpolation
- Adaptive choice of reduced order

Disadvantages

- No guaranteed monotonic error decay
- Linearized least-squares error can be strongly different from true approximation error
- Assumes the existence of “good enough” data for interpolation

Demonstration: Porous Bone Vibrations

`FrequencyDomain/PorousBone/demo.ipynb`

Summary

Data-driven ROMs are derived from data, not from FOM operators

Time domain perspective

- Data are the states or outputs at various times
- Data should be pre-processed appropriately
- ROM structure should be designed with the true dynamical structure in mind

Frequency domain perspective

- Data are frequency / transfer function evaluation pairs
- Methods should be chosen with respect to amount and quality of data
- Usually targets linear time-invariant systems

References I

Antoulas, A. C. and Anderson, B. D. O. (1986).
On the scalar rational interpolation problem.
IMA Journal of Mathematical Control and Information, 3(2-3):61–8.

Berljafa, M. and Güttel, S. (2017).
The RKFIT algorithm for nonlinear rational approximation.
SIAM Journal on Scientific Computing, 39(5):A2049–a2071.

Brunton, S. L. and Kutz, J. N. (2022).
Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.
Cambridge University Press, 2nd edition.

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

- Drmač, Z., Gugercin, S., and Beattie, C. (2015).
  Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation.
  SIAM Journal on Scientific Computing, 37(2):A625–a652.
- Elsworth, S. and Güttel, S. (2019).
  Conversions between barycentric, RKFUN, and Newton representations of rational interpolants.
  Linear Algebra and its Applications, 576:246–257.
- Farcas, I.-G., Gundevia, R. P., Munipalli, R., and Willcox, K. E. (2024a).
  Distributed computing for physics-based data-driven reduced modeling at scale: Application to a rotating detonation rocket engine.
  https://arxiv.org/abs/2407.09994.

References III

Farcas, I.-G., Gundevia, R. P., Munipalli, R., and Willcox, K. E. (2024b).
Domain decomposition for data-driven reduced modeling of large-scale systems.
AIAA Journal, 62(11):4071–4086.

Ghattas, O. and Willcox, K. (2021).
Learning physics-based models from data: Perspectives from inverse problems and model reduction.
Acta Numerica, 30:445–554.

Gosea, I. V., Gugercin, S., and Beattie, C. (2022).
Data-driven balancing of linear dynamical systems.
SIAM Journal on Scientific Computing, 44(1):A554–a582.

Gosea, I. V. and Güttel, S. (2021).
Algorithms for the rational approximation of matrix-valued functions.
SIAM Journal on Scientific Computing, 43(5):A3033–a3054.

References IV

Gustavsen, B. and Semlyen, A. (1999).
Rational approximation of frequency domain responses by vector fitting.
IEEE Transactions on Power Delivery, 14(3):1052–1061.

Hund, M., Mitchell, T., Mlincarić, P., and Saak, J. (2022).
Optimization-based parametric model order reduction via $\mathcal{H}_2 \otimes \mathcal{L}_2$ first-order necessary conditions.
SIAM Journal on Scientific Computing, 44(3):A1554–a1578.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021).
Physics-informed machine learning.
Nature Reviews Physics, 3:422–440.

References V

- Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2023).
  Neural operator: Learning maps between function spaces with applications to PDEs.
  Journal of Machine Learning Research, 24(89):1–97.
- Kramer, B. and Gugercin, S. (2016).
  Tangential interpolation-based eigensystem realization algorithm for MIMO systems.
  Mathematical and Computer Modelling of Dynamical Systems, 22(4):282–306.
- Kramer, B., Peherstorfer, B., and Willcox, K. E. (2024).
  Learning nonlinear reduced models from data with operator inference.
  Annual Review of Fluid Mechanics, 56(1):521–548.

References VI

- Kung, S.-Y. (1978).
  A new identification and model reduction algorithm via singular value decomposition.
  In *Proceedings of the 12th Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA*, pages 705–714.
- Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. (2016).
  Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems.
  Siam.
- Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. (2021).
  Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators.
  Nature Machine Intelligence, 3:218–229.

References VII

Mayo, A. J. and Antoulas, A. C. (2007).
A framework for the solution of the generalized realization problem.
Linear Algebra and its Applications, 425(2-3):634–662.
Special issue in honor of P. A. Fuhrmann, Edited by A. C. Antoulas, U. Helmke, J. Rosenthal, V. Vinnikov, and E. Zerz.

McQuarrie, S. A., Chaudhuri, A., Willcox, K. E., and Guo, M. (2025).
Bayesian learning with Gaussian processes for low-dimensional representations of time-dependent nonlinear systems.
Physica D: Nonlinear Phenomena, 475:134572.

Mlinarić, P. and Gugercin, S. (2023).
A unifying framework for interpolatory $\mathcal{L}_2$-optimal reduced-order modeling.
SIAM Journal on Numerical Analysis, 61(5):2133–2156.

References VIII

- Nakatsukasa, Y., Sète, O., and Trefethen, L. N. (2018).
  The AAA algorithm for rational approximation.
  SIAM Journal on Scientific Computing, 40(3):A1494–a1522.
- Peherstorfer, B. and Willcox, K. (2016).
  Data-driven operator inference for nonintrusive projection-based model reduction.
  Computer Methods in Applied Mechanics and Engineering, 306:196–215.
- Raissi, M., Perdikaris, P., and Karniadakis, G. (2019).
  Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
  Journal of Computational Physics, 378:686–707.
- Schaeffer, H. (2017).
  Learning partial differential equations via data discovery and sparse optimization.
  Proceedings of the Royal Society A, 473:20160446.

References IX

Schmid, P. J. (2010).
Dynamic mode decomposition of numerical and experimental data.
Journal of Fluid Mechanics, 656:5–28.

Schmid, P. J. (2022).
Dynamic mode decomposition and its variants.
Annual Review of Fluid Mechanics, 54:225–254.

Schwerdtner, P. and Voigt, M. (2023).
SOBMOR: Structured optimization-based model order reduction.
SIAM Journal on Scientific Computing, 45(2):A502–a529.

Sirovich, L. (1987).
Turbulence and the dynamics of coherent structures part i: Coherent structures.
Quarterly of Applied Mathematics, 45(3):561–571.

Acknowledgments

S.A.M. was funded in part by the John von Neumann Postdoctoral Fellowship, a position at Sandia National Laboratories sponsored by the Applied Mathematics Program of the U.S. Department of Energy Office of Advanced Scientific Computing Research.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government.