Predictive Data Science for physical systems

From model reduction to scientific machine learning

Professor Karen E. Willcox
Mathematics of Reduced Order Models | ICERM | 2-20-202
The Team

Funding sources:

- US Air Force **Computational Math Program** (F. Fahroo)
- US Air Force **Center of Excellence on Rocket Combustion** (M. Birkan, F. Fahroo, R. Munipalli, D. Talley)
- US Department of Energy **AEOLUS MMICC** (S. Lee, W. Spotz)
- SUTD-MIT **International Design Centre**

Michael Kapteyn  
**MIT**

Dr. Parisa Khodabakhshi  
**Oden Institute**

Prof. Boris Kramer  
**UCSD**

Prof. Benjamin Peherstorfer  
**Courant Institute**

Elizabeth Qian  
**MIT**

Renee Swischuk  
**Caliper**
1 Scientific Machine Learning
   What, Why & How?

2 Lift & Learn
   Projection-based model reduction as a lens through which to learn predictive models

3 Conclusions & Outlook
Scientific Machine Learning

“Scientific machine learning (SciML) is a core component of artificial intelligence (AI) and a computational technology that can be trained, with scientific data, to augment or automate human skills.

Across the Department of Energy (DOE), SciML has the potential to transform science and energy research. Breakthroughs and major progress will be enabled by harnessing DOE investments in massive data from scientific user facilities, software for predictive models and algorithms, high-performance computing platforms, and the national workforce.”
Scientific Machine Learning

What role for model reduction?

1 reduce the cost of training 2 foundational shift in ML perspectives

- Embed domain knowledge
- Integrate heterogeneous, noisy & incomplete data
- Respect physical constraints
- Bring interpretability to results
- Get predictions with quantified uncertainties
Predictive Digital Twin
via component-based ROMs and interpretable machine learning
ROMs embed predictive modeling and reduce the cost of training

**Offline:**
Construct library of ROMs representing different asset states
Use model library to train a classifier that predicts asset state based on sensor data

**Online:**
sensor data
current Digital Twin
updated Digital Twin
Analysis Prediction Optimization

[Kapteyn, Knezevic, W. AIAA Scitech 2020]
Machine learning
“The scientific study of algorithms & statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns & inference instead.” [Wikipedia]

Reduced-order modeling
“Model order reduction (MOR) is a technique for reducing the computational complexity of mathematical models in numerical simulations.” [Wikipedia]

What is the connection between reduced-order modeling and machine learning?

Model reduction methods have grown from Computational Science & Engineering, with focus on reducing high-dimensional models that arise from physics-based modeling, whereas machine learning has grown from Computer Science, with a focus on creating low-dimensional models from black-box data streams. [Swischuk et al., Computers & Fluids, 2019]
Machine learning
“The scientific study of algorithms & statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns & inference instead.” [Wikipedia]

Reduced-order modeling
“Model order reduction (MOR) is a technique for reducing the computational complexity of mathematical models in numerical simulations.” [Wikipedia]

Reduced-order modeling & machine learning: Can we get the best of both worlds?

- Discover hidden structure
- Non-intrusive implementation
- Black-box & flexible
- Accessible & available

- Embed governing equations
- Structure-preserving
- Predictive (error estimators)
- Stability-preserving
Lift & Learn

Projection-based model reduction as a lens through which to learn low-dimensional predictive models
Lift & Learn: Ingredients

1. A **physics-based model**
   Typically described by a set of PDEs or ODEs

2. **Lens of projection** to define a structure-preserving low-dimensional model

3. **Non-intrusive learning** of the reduced model

4. **Variable transformations** that expose polynomial structure in the model → can be exploited with non-intrusive learning
Start with a physics-based model

Example: modeling solidification in additive manufacturing

Space/time evolution of temperature $T$ and phase parameter $\phi$

\[
\dot{T} + L \phi = \nabla \cdot (K(\phi) \nabla T)
\]

\[
\alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p' (\phi) - q (T, \phi)
\]

with

\[
K(\phi) = K_0 (1 - h(\phi)) + K_1 h(\phi)
\]

\[
h(\phi) = 6\phi^5 - 15\phi^4 + 10\phi^3
\]

\[
p(\phi) = \frac{1}{4} \phi^2 (1 - \phi)^2
\]

\[
q(T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh [\gamma (T_{melt} - T)]
\]

Figure from: https://www.bintoa.com/powder-bed-fusion/

Model based on Kobayashi, 1993; collaboration with Bao & Biros

Discretize:
Spatially discretized finite element model

\[
\dot{x} = Ax + Bu + f(x, u)
\]

Discretized state $x$ contains temperature and phase field order parameter at $N_z$ spatial grid points

\[
N_z \sim O(10^3 - 10^9)
\]
Projection-based model reduction

1 **Train**: Solve PDEs to generate training data (snapshots)
2 **Identify structure**: Compute a low-dimensional basis
3 **Reduce**: Project PDE model onto the low-dimensional subspace
Full-order model (FOM) state $x \in \mathbb{R}^N$

Reduced-order model (ROM) state $x_r \in \mathbb{R}^r$

$\dot{x} = Ax + Bu$

Approximate $x \approx Vx_r$  
$V \in \mathbb{R}^{N \times r}$

Residual: $N$ eqs $\gg r$ dof

$r = V\dot{x}_r - AVx_r - Bu$

Project

$W^T r = 0$  
(Galerkin: $W = V$)

$\dot{x}_r = A_r x_r + B_r u$

$A_r = V^T AV$  
$B_r = V^T B$
Linear Model

FOM: \( \dot{x} = Ax + Bu \)

ROM: \( \dot{x}_r = A_r x_r + B_r u \)

Precompute the ROM matrices:

\[
A_r = V^T AV, \quad B_r = V^T B
\]

Quadratic Model

FOM: \( \dot{x} = Ax + H(x \otimes x) + Bu \)

ROM: \( \dot{x}_r = A_r x_r + H_r (x_r \otimes x_r) + B_r u \)

Precompute the ROM matrices and tensor:

\[
H_r = V^T H(V \otimes V)
\]

projection preserves structure \( \leftrightarrow \) structure embeds physical constraints
Operator inference

Non-intrusive learning of reduced models from simulation snapshot data
Given reduced state data, learn the reduced model

Operator Inference using proper orthogonal decomposition (POD) aka PCA


\[
\hat{x} = \hat{A}x + \hat{B}u + \hat{H}(\hat{x} \otimes \hat{x})
\]

Given reduced state data (\(\hat{X}\)) and derivative data (\(\dot{\hat{X}}\)):

\[
\hat{X} = \begin{bmatrix}
\hat{x}(t_1) & \ldots & \hat{x}(t_K)
\end{bmatrix}, \quad \dot{\hat{X}} = \begin{bmatrix}
\dot{\hat{x}}(t_1) & \ldots & \dot{\hat{x}}(t_K)
\end{bmatrix}
\]

Find the operators \(\hat{A}, \hat{B}, \hat{H}\) by solving the least squares problem:

\[
\min_{\hat{A}, \hat{B}, \hat{H}} \left\| \hat{X}^T \hat{A}^T + (\hat{X} \otimes \hat{X})^T \hat{H}^T + U^T \hat{B}^T - \dot{\hat{X}}^T \right\|
\]

- Generate \(\hat{X}\) data by projection of \(X\) snapshot data onto POD basis
- If data are Markovian, Operator Inference recovers the intrusive POD reduced model [Peherstorfer, 2019]
Variable Transformations & Lifting

The physical governing equations reveal variable transformations and manipulations that expose polynomial structure.
There are multiple ways to write the Euler equations

Different choices of variables leads to different structure in the discretized system

- Define specific volume: \( q = \frac{1}{\rho} \)
- Take derivative: 
  \[
  \frac{\partial q}{\partial t} = -\frac{1}{\rho^2} \frac{\partial \rho}{\partial t} = \frac{1}{\rho^2} \left( -\rho \frac{\partial u}{\partial z} - u \frac{\partial \rho}{\partial z} \right) = q \frac{\partial u}{\partial z} - u \frac{\partial q}{\partial z}
  \]

\[
  \frac{\partial}{\partial t} \left( \begin{array}{c} \rho \wv \\ \rho w^2 + p \\ E \end{array} \right) + \frac{\partial}{\partial z} \left( \begin{array}{c} \rho \wv \\ \rho w^2 + p \\ E \end{array} + \begin{array}{c} 0 \\ (E + p)w \end{array} \right) = 0
  
  E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho w^2
\]

Conservative variables
mass, momentum, energy

Primitve variables
mass, velocity, pressure

\[
  \frac{\partial}{\partial t} \left( \begin{array}{c} \rho \wv \\ \rho \frac{w}{E} \frac{w}{z} + w \frac{\partial \rho}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} \\ \gamma p \frac{w}{z} + w \frac{\partial p}{\partial z} \end{array} \right) = 0
\]

\[
  \frac{\partial}{\partial t} \left( \begin{array}{c} w \frac{\partial w}{\partial z} + q \frac{\partial p}{\partial z} \\ \frac{\partial w}{\partial z} + \frac{\partial p}{\partial z} \\ \frac{q}{\partial z} + w \frac{\partial q}{\partial z} \end{array} \right) = 0
\]

Specific volume variables

\[
  \dot{x} = H(x \otimes x) + Bu
\]

Transformed system has quadratic structure

\[
  \dot{x}_r = H_r (x_r \otimes x_r) + B_r u
\]

ROM has quadratic structure
Introducing auxiliary variables can expose structure → **lifting**

[McCormick 1976; Gu 2011]

- original state \(s(x, t)\) dimension \(d_s\)
- lifted state \(w(x, t)\) dimension \(d_w\)
- lifted PDE has quadratic form

Definition 1. Define the lifting map,

\[
\mathcal{T} : \mathcal{S} \rightarrow \mathcal{W} \subset \mathbb{R}^{d_w}, \quad d_w \geq d_s,
\]

and let \(w(x, t) = \mathcal{T}(s(x, t))\). \(\mathcal{T}\) is a quadratic lifting of eq. (1) if the following conditions are met:

1. the map \(\mathcal{T}\) is differentiable with respect to \(s\) with bounded derivative, i.e., if \(\mathcal{J}(s)\) is the Jacobian of \(\mathcal{T}\) with respect to \(s\), then

\[
\sup_{s \in \mathcal{S}} \|\mathcal{J}(s)\| \leq c, \tag{15}
\]

for some \(c > 0\), and

2. the lifted state \(w\) satisfies

\[
\frac{\partial w}{\partial t} = a(w) + h(w), \tag{16}
\]

where

\[
a(w) = \begin{pmatrix} a_1(w) \\ \vdots \\ a_{d_w}(w) \end{pmatrix}, \quad h(w) = \begin{pmatrix} h_1(w) \\ \vdots \\ h_{d_w}(w) \end{pmatrix}, \tag{17}
\]

for some linear functions \(a_j\) and quadratic functions \(h_j, j = 1, 2, \ldots, d_w\).

[Qian, Kramer, Peherstorfer, W. Physica D, 2020]
Introducing auxiliary variables can expose structure → **lifting**

[McCormick 1976; Gu 2011]

Example: Lifting a quartic ODE to quadratic-bilinear form

Can either lift to a system of ODEs or to a system of DAEs

Consider the quartic system

\[
\dot{x} = x^4 + u.
\]

Introduce auxiliary variables:

\[
w_1 = x^2 \quad w_2 = w_1^2
\]

Chain rule:

\[
\begin{align*}
\dot{w}_1 &= 2x[w_1^2 + u] = 2x[w_2 + u] \\
\dot{w}_2 &= 2w_1\dot{w}_1 = 4xw_1[w_2 + u]
\end{align*}
\]

Need additional variable to make auxiliary dynamics quadratic:

\[
w_3 = xw_1 \quad \dot{w}_3 = \dot{x}w_1 + x\dot{w}_1 = w_1w_2 + w_1u + 2w_1w_2 + 2w_1u
\]

**QB-ODE**

\[
\begin{align*}
\dot{x} &= w_2 + u \\
\dot{w}_1 &= 2xw_2 + 2xu \\
\dot{w}_2 &= 4w_2w_3 + 4w_3u \\
\dot{w}_3 &= 3w_1w_2 + 3w_1u
\end{align*}
\]

**QB-DAE**

\[
\begin{align*}
\dot{x} &= w_1^2 + u \\
0 &= w_1 - x^2
\end{align*}
\]
Many different forms of nonlinear PDEs can be lifted to polynomial form.

\[
\begin{align*}
\dot{T} + L\dot{\phi} &= \nabla . (K(\phi) \nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - p'(\phi) - q(T, \phi)
\end{align*}
\]

\[
\begin{align*}
\dot{T} + L\dot{\phi} &= \nabla . (K \nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0 \\
\dot{K} &= \frac{120(K_1 - K_0)}{\alpha \xi^2} p \left[\xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0\right] \\
\dot{p} &= \frac{1}{\alpha \xi^2} p' \left[\xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0\right] \\
\dot{p}' &= \frac{1}{\alpha \xi^2} p'' \left[\xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0\right] \\
\dot{p}'' &= \frac{3}{\alpha \xi^2} (2\phi - 1) \left[\xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0\right] \\
m_0 &= -\gamma y \left\{ \nabla . (K \nabla T) - \frac{L}{\alpha \xi^2} \left[\xi^2 \Delta \phi - \frac{\beta}{6} \left(p'' - \frac{1}{2}\right) m_0\right] \right\} \\
y &= 1 - m_0^2
\end{align*}
\]
Solidification of a Pure Material

Nonlinear system for 1D solidification

\[
\begin{align*}
\dot{T} + \dot{\phi} &= \nabla \cdot (K(\phi) \nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - p'(\phi) - q(T, \phi)
\end{align*}
\]

with
\[
K(\phi) = K_0 (1 - h(\phi)) + K_1 h(\phi)
\]
\[h(\phi) = 6\phi^5 - 15\phi^4 + 10\phi^3\]
\[p(\phi) = \frac{1}{4} \phi^2 (1 - \phi)^2\]
\[q(T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh [\gamma (T_{\text{melt}} - T)]\]

DebRoy et al. *Progress in Materials Science, 2018*
Nonlinear system for 1D solidification

\[
\begin{align*}
\dot{T} + L\dot{\phi} &= \nabla \cdot (K(\phi) \nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - p'(\phi) - q(T, \phi)
\end{align*}
\]

with

\[
K(\phi) = K_0 (1 - h(\phi)) + K_1 h(\phi)
\]

\[
h(\phi) = 6\phi^5 - 15\phi^4 + 10\phi^3
\]

\[
p(\phi) = \frac{1}{4} \phi^2 (1 - \phi)^2
\]

\[
q(T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh[\gamma (T_{\text{melt}} - T)]
\]

Chain rule:

\[
K = K_0 [1 - h(\phi)] + K_1 h(\phi)
\]

\[
\dot{K} = (K_1 - K_0) h'(\phi) \dot{\phi}
\]

\[
= \frac{120 (K_1 - K_0)}{\alpha \xi^2} p \left[ \xi^2 \Delta \phi - p'(\phi) - q(T, \phi) \right]
\]
Nonlinear system for 1D solidification

\[ T + L\dot{\phi} = \nabla \cdot (K \nabla T) \]

\[ \alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p'(\phi) - q(T, \phi) \]

\[ \dot{K} = \frac{120}{\alpha \xi^2} \left( K_1 - K_0 \right) p \left[ \xi^2 \Delta \phi - p'(\phi) - q(T, \phi) \right] \]

with \( p(\phi) = \frac{1}{4} \phi^2 (1 - \phi)^2 \)

\[ q(T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh [\gamma (T_{\text{melt}} - T)] \]

Chain rule:

\[ p = \frac{1}{4} \phi^2 (1 - \phi)^2 \]

\[ \dot{p} = p'(\phi) \dot{\phi} = \frac{1}{\alpha \xi^2} p' \left[ \xi^2 \Delta \phi - p'(\phi) - q(T, \phi) \right] \]
Solidification of a Pure Material

Nonlinear system for 1D solidification

\[
\begin{align*}
\dot{T} + L\phi &= \nabla \cdot (K \nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - p' (\phi) - q (T, \phi) \\
\dot{K} &= \frac{120 (K_1 - K_0)}{\alpha \xi^2} p \left[ \xi^2 \Delta \phi - p' (\phi) - q (T, \phi) \right] \\
\dot{p} &= \frac{1}{\alpha \xi^2} p' \left[ \xi^2 \Delta \phi - p' (\phi) - q (T, \phi) \right]
\end{align*}
\]

with \( q (T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh [\gamma (T_{\text{melt}} - T)] \)

Chain rule:

\[
\begin{align*}
p' &= \frac{1}{2} \phi (1 - \phi) (1 - 2\phi) \\
\dot{p}' &= p'' (\phi) \dot{\phi} = \frac{1}{\alpha \xi^2} p'' \left[ \xi^2 \Delta \phi - p' (\phi) - q (T, \phi) \right]
\end{align*}
\]
Solidification of a Pure Material

Nonlinear system for 1D solidification

\[
\begin{align*}
\dot{T} + L\phi &= \nabla \cdot (K\nabla T) \\
\alpha \xi^2 \dot{\phi} &= \xi^2 \Delta \phi - p' - q(T, \phi) \\
\dot{K} &= \frac{120}{\alpha \xi^2} \left( K_1 - K_0 \right) p \left[ \xi^2 \Delta \phi - p' - q(T, \phi) \right] \\
\dot{p} &= \frac{1}{\alpha \xi^2} p' \left[ \xi^2 \Delta \phi - p' - q(T, \phi) \right] \\
\dot{p}' &= \frac{1}{\alpha \xi^2} p'' \left[ \xi^2 \Delta \phi - p' - q(T, \phi) \right]
\end{align*}
\]

with \( q(T, \phi) = \frac{\beta}{2} \phi (\phi - 1) \tanh [\gamma (T_{\text{melt}} - T)] \)

Chain rule:

\[
\begin{align*}
p'' &= 3 (\phi^2 - \phi) + \frac{1}{2} \\
p'' &= 3 (2\phi - 1) \phi = \frac{3}{\alpha \xi^2} (2\phi - 1) \left[ \xi^2 \Delta \phi - p' - q(T, \phi) \right]
\end{align*}
\]
Nonlinear system for 1D solidification

\[
\dot{T} + L\dot{\phi} = \nabla \cdot (K\nabla T)
\]

\[
\alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0(T)
\]

\[
\dot{K} = \frac{120 (K_1 - K_0)}{\alpha \xi^2} p \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0(T) \right]
\]

\[
\dot{p} = \frac{1}{\alpha \xi^2} p' \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0(T) \right]
\]

\[
\dot{p}' = \frac{1}{\alpha \xi^2} p'' \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0(T) \right]
\]

\[
\dot{p}'' = \frac{3}{\alpha \xi^2} (2\phi - 1) \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0(T) \right]
\]

Chain rule:

\[
m_0 = \tanh \left[ \gamma (T_{\text{melt}} - T) \right]
\]

\[
\dot{m}_0 = -\gamma \left(1 - m_0^2\right) \dot{T}
\]

\[
= -\gamma \left(1 - m_0^2\right) \left\{ \nabla \cdot (K \nabla T) - \frac{L}{\alpha \xi^2} \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \right\}
\]
Solidification of a Pure Material

Nonlinear system for 1D solidification

\[ \dot{T} + L \dot{\phi} = \nabla \cdot (K \nabla T) \]
\[ \alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \]
\[ \dot{K} = \frac{120 (K_1 - K_0)}{\alpha \xi^2} p \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \]
\[ \dot{p} = \frac{1}{\alpha \xi^2} p' \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \]
\[ \dot{p}' = \frac{1}{\alpha \xi^2} p'' \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \]
\[ \dot{p}'' = \frac{3}{\alpha \xi^2} (2\phi - 1) \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \]
\[ \dot{m}_0 = -\gamma y \left\{ \nabla \cdot (K \nabla T) - \frac{L}{\alpha \xi^2} \left[ \xi^2 \Delta \phi - p' - \frac{\beta}{6} \left( p'' - \frac{1}{2} \right) m_0 \right] \right\} \]
\[ y = 1 - m_0^2 \]

with original variables \( T, \phi \)

with lifted variables \( T, \phi, K, p, p', p'', m_0, y \)
Lift & Learn

Variable transformations to expose structure
+
non-intrusive learning that frees us to choose our variables
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

**Lift & Learn** [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)

\[
X_{\text{orig}} = \begin{bmatrix}
    x(t_1) & \ldots & x(t_K)
\end{bmatrix}, \quad \dot{X}_{\text{orig}} = \begin{bmatrix}
    \dot{x}(t_1) & \ldots & \dot{x}(t_K)
\end{bmatrix}
\]
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

Lift & Learn [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)
2. Transform snapshot data to get lifted snapshots (analyze the PDEs to expose system polynomial structure)

\[ \mathbf{X}_{\text{orig}} \rightarrow \mathbf{X} \quad \dot{\mathbf{X}}_{\text{orig}} \rightarrow \dot{\mathbf{X}} \]
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

Lift & Learn [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)
2. Transform snapshot data to get lifted snapshots
3. Compute POD basis from lifted trajectories

\[ X = V \Sigma W^T \]
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

**Lift & Learn** [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)
2. Transform snapshot data to get lifted snapshots
3. Compute POD basis from lifted trajectories
4. Project lifted trajectories onto POD basis, to obtain trajectories in low-dimensional POD coordinate space

\[ \hat{X} = V^T X \]
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

Lift & Learn [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)
2. Transform snapshot data to get lifted snapshots
3. Compute POD basis from lifted trajectories
4. Project lifted trajectories onto POD basis, to obtain trajectories in low-dimensional POD coordinate space
5. Solve least squares minimization problem to infer the low-dimensional model

\[
\min_{\hat{A}, \hat{B}, \hat{H}} \left\| \hat{X}^T \hat{A}^T + (\hat{X} \otimes \hat{X})^T \hat{H}^T + U^T \hat{B}^T - \dot{\hat{X}}^T \right\|
\]
Learning a low-dimensional model

Using only snapshot data from the original high-fidelity model (non-intrusive) but using variable transformations to expose and exploit structure

Lift & Learn [Qian, Kramer, Peherstorfer & W., 2019]

1. Generate full state trajectories (snapshots) (from high-fidelity simulation)
2. Transform snapshot data to get lifted snapshots
3. Compute POD basis from lifted trajectories
4. Project lifted trajectories onto POD basis, to obtain trajectories in low-dimensional POD coordinate space
5. Solve least squares minimization problem to infer the low-dimensional model

Under certain conditions, recovers the intrusive POD reduced model

→ convenience of black-box learning + rigor of projection-based reduction + structure imposed by physics
Additive Manufacturing

Lift & Learn reduced models for a highly nonlinear solidification process
Modeling solidification in additive manufacturing

\[
\dot{T} + L \dot{\phi} = \nabla \cdot (K(\phi) \nabla T)
\]

\[
\alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p'(\phi) - q(T, \phi)
\]

- Spatial domain discretized into 1,000 cells
- Initial conditions

\[
T(x, 0) = 0.4 \\
\phi(x, 0) = 0.5 \cos(\pi x) + 0.5
\]

- Boundary conditions

\[
T(0, t) = T(\ell, t) 0.4 \\
\left. \frac{\partial \phi}{\partial x} \right|_{x=0} = \left. \frac{\partial \phi}{\partial x} \right|_{x=\ell} = 0
\]

https://www.bintoa.com/powder-bed-fusion
Modeling solidification in additive manufacturing

Training data

- 800 snapshots collected over time $t = [0, 0.02]$
- Parameters: $\ell = 1, \alpha = 3, \xi = 0.1, \beta = 0.9$, $T_{\text{melt}} = 1.0, L = 0.5, \gamma = 2.0, K_0 = 1, K_1 = 0.1$
- Variables used for learning cubic ROMs
  $$\mathbf{x} = [T, \phi, K, p, p', p'', m_0, \gamma]$$

\[\dot{T} + L \dot{\phi} = \nabla \cdot (K(\phi) \nabla T)\]
\[\alpha \xi^2 \dot{\phi} = \xi^2 \Delta \phi - p'(\phi) - q(T, \phi)\]
Lift & Learn reduced model performance

- $r = 23$ POD basis functions
- 16 modes for differential eqs + 7 modes for algebraic eqs
Lift & Learn reduced model performance

- $r = 32$ POD basis functions
- 22 modes for differential eqs + 10 modes for algebraic eqs
Rocket Engine Combustion

Lift & Learn reduced models for a complex Air Force combustion problem
Modeling a single injector of a rocket engine combustor

- Spatial domain (2D) discretized into 38,523 cells
- Oxidizer input: $0.37 \frac{\text{kg}}{\text{s}}$ of 42% $\text{O}_2$ / 58% $\text{H}_2\text{O}$
- Fuel input: $5.0 \frac{\text{kg}}{\text{s}}$ of $\text{CH}_4$
- Forced by a back pressure boundary condition at exit throat
Modeling a single injector of a rocket engine combustor

Training data

- 1 ms of full state solutions generated using Air Force GEMS code (~200 hours CPU time)
- Timestep $\Delta t = 10^{-7}$ s; 10,000 total snapshots
- Variables used for learning ROMs
  
  $$x = [p \ u \ v \ 1/\rho \ \rho Y_{CH4} \ \rho Y_{O2} \ \rho Y_{CO2} \ \rho Y_{H2O}]$$
  
  makes many (but not all) terms in governing equations quadratic
- Snapshot matrix $X \in \mathbb{R}^{308,184 \times 10,000}$

Test data

Additional 2 ms of data at four monitor locations (20,000 timesteps)
Performance of learned quadratic ROM

Pressure time traces at monitor location 1

Basis size $r = 24$
Performance of learned quadratic ROM

Pressure time traces at monitor location 1

Basis size $r = 29$
True

Predicted
$r = 29$ POD modes

Relative error
True

Predicted
\( r = 29 \) POD modes

Normalized absolute error
Conclusions & Outlook

What future for model reduction?
Scientific Machine Learning

What role for model reduction?
reduce the cost of training | foundational shift in ML perspectives

- Embed domain knowledge
- Integrate heterogeneous, noisy & incomplete data

- Respect physical constraints
- Bring interpretability to results
- Get predictions with quantified uncertainties
Scientific Machine Learning

Learning from data through the lens of models is a way to exploit structure in an otherwise intractable problem.

- Embed domain knowledge
- Integrate heterogeneous, noisy & incomplete data
- Respect physical constraints
- Bring interpretability to results
- Get predictions with quantified uncertainties
Scientific Machine Learning

What future for model reduction?

1. **Rigor**
   issuing predictions with certified uncertainty for high-consequence applications

2. **Relevance**
   towards real-world scientific and engineering applications

3. **Accessibility**
   accessible algorithms, community software, benchmark problems

4. **Impact & adoption**
   depend on all of the above
Data-driven decisions

building the mathematical foundations and computational methods to enable design of the next generation of engineered systems