Predictive Data Science for physical systems

From model reduction to scientific machine learning

Professor Karen E. Willcox
Mathematics of Reduced Order Models | ICERM | 2-20-20[2]

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

TEXAS
The University of Texas at Austin

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

Outline

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

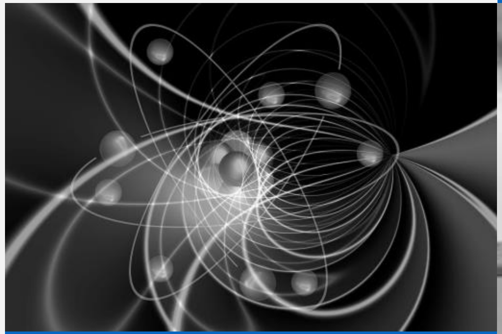
“Scientific machine learning (SciML) is a core component of artificial intelligence (AI) and a computational technology that can be trained, with scientific data, to augment or automate human skills.

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

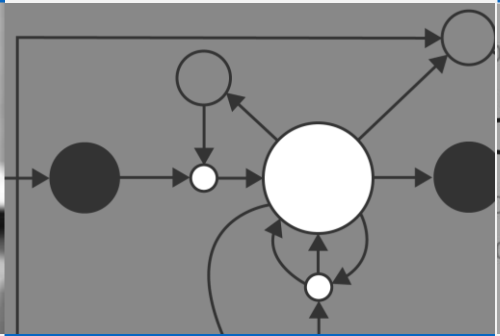
What role for model reduction?

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



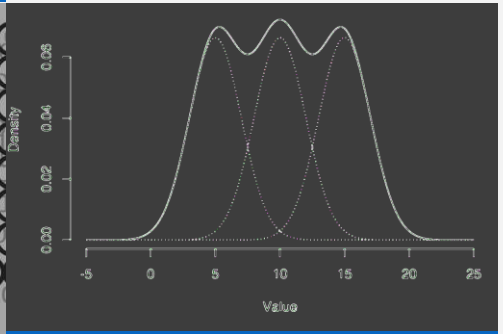
Respect physical constraints

Embed domain knowledge



Bring interpretability to results

Integrate heterogeneous, noisy & incomplete data



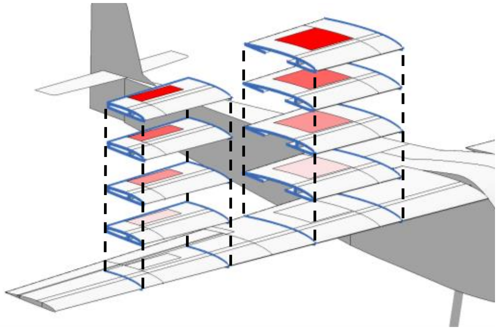
Get predictions with quantified uncertainties

Predictive Digital Twin

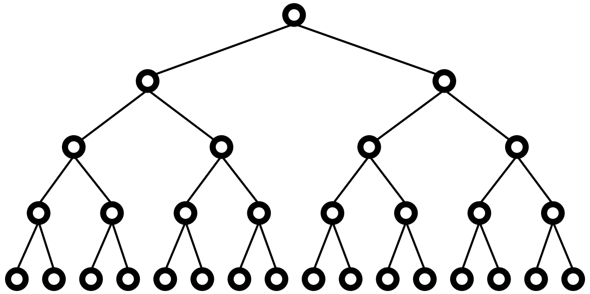
via component-based ROMs and interpretable machine learning

ROMs embed predictive modeling and reduce the cost of training

Offline:



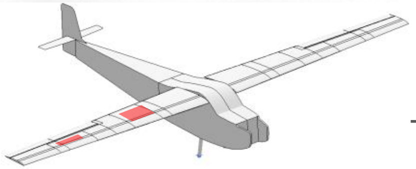
Construct library of ROMs representing different asset states



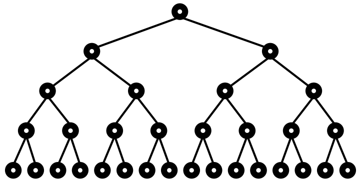
Use model library to train a classifier that predicts asset state based on sensor data

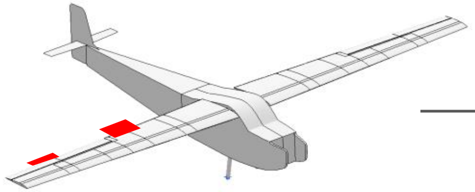
Online:

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	Embed governing equations
Non-intrusive implementation	Structure-preserving
Black-box & flexible	Predictive (error estimators)
Accessible & available	Stability-preserving

Lift & Learn

Projection-based model reduction as a lens through which to learn low-dimensional predictive models

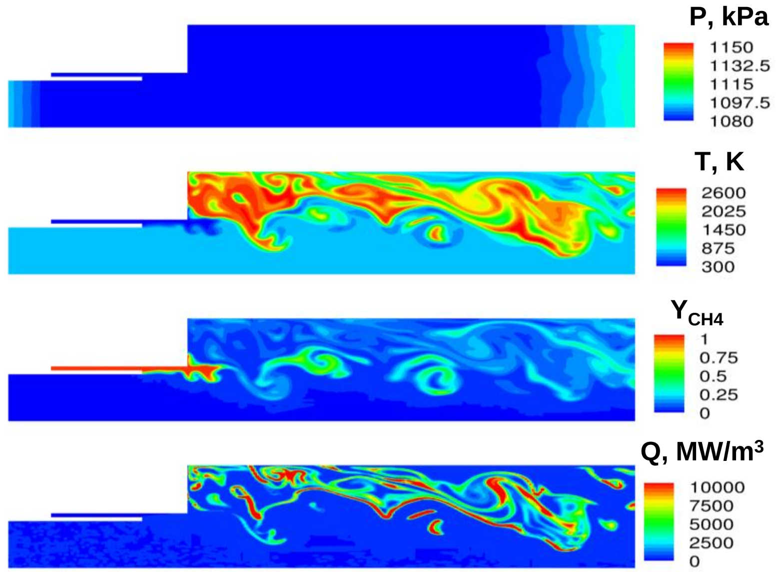
Lift & Learn: Ingredients

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

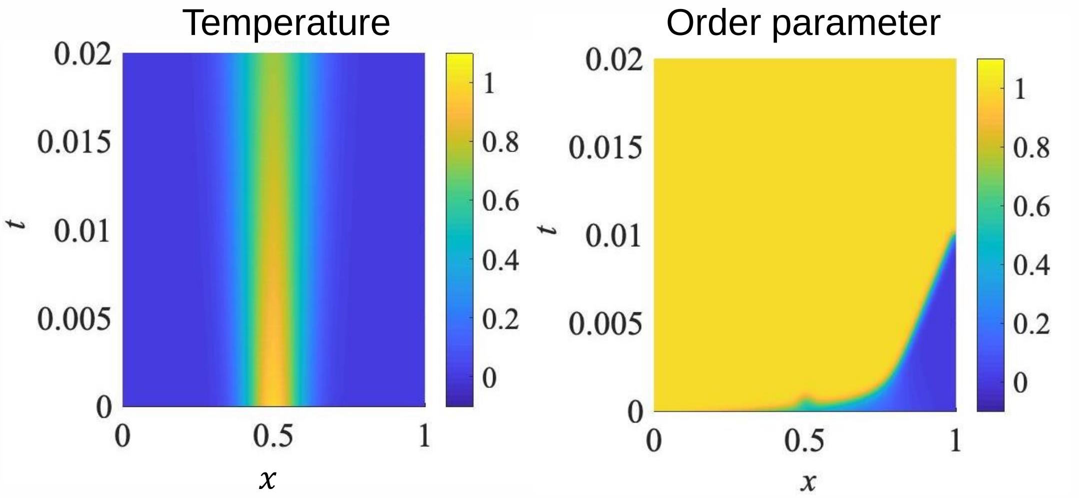
2. Lens of **projection** to define a structure-preserving low-dimensional model

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



Rocket combustion

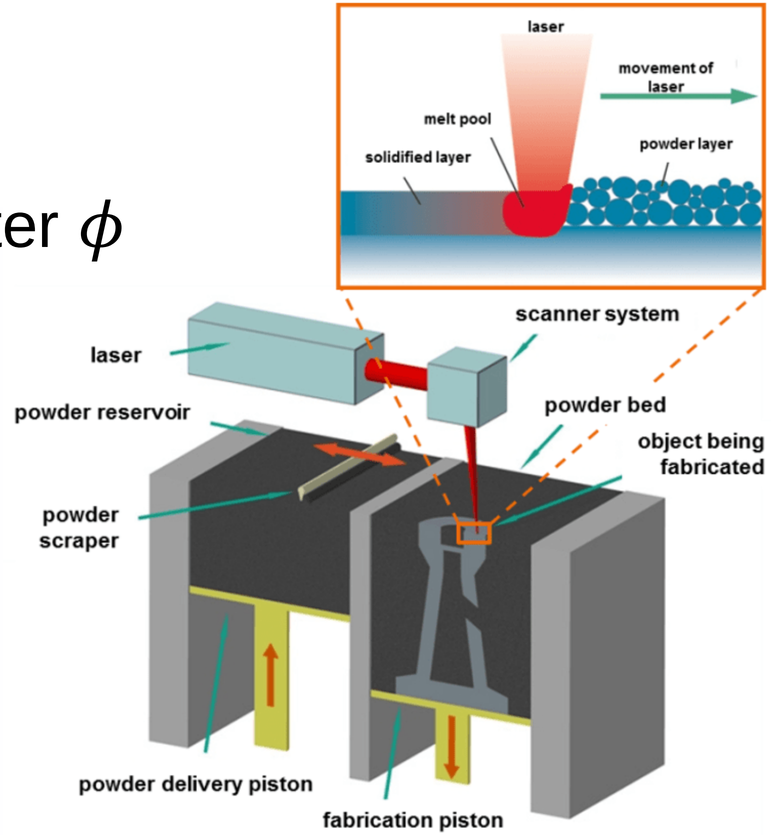


Solidification process in additive manufacturing

Start with a physics-based model

Example: modeling solidification in additive manufacturing

Space/time evolution of temperature T and phase parameter ϕ



$$\dot{T} + L\dot{\phi} = \nabla.\left(K(\phi)\nabla T\right)$$

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

with

$$K(\phi) = K_0\left(1 - h(\phi)\right) + K_1 h(\phi)$$

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

$$p(\phi) = \frac{1}{4}\phi^2(1-\phi)^2 \qquad q(T,\phi) = \frac{\beta}{2}\phi(\phi-1)\tanh\left[\gamma\left(T_{\text{melt}} - T\right)\right]$$

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

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

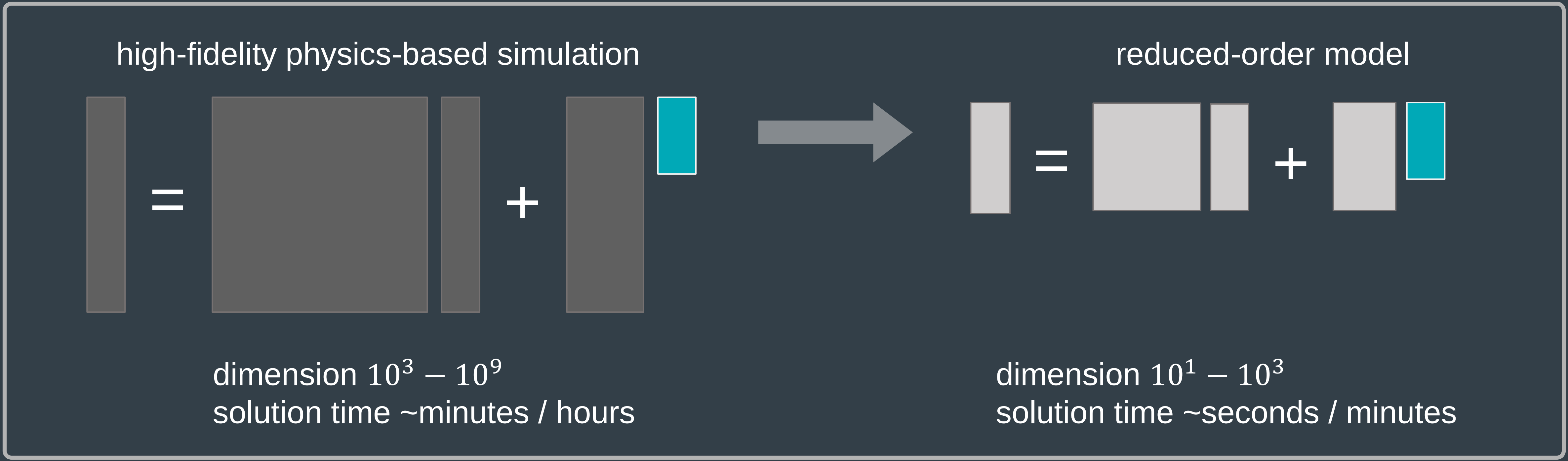
Discretize:
Spatially discretized finite element model

$$\boxed{\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{f}(\mathbf{x},\mathbf{u})}$$

discretized **state $\mathbf{x}$** contains temperature and phase field order parameter at N_z spatial grid points

$N_z \sim O(10^3 - 10^9)$

$$\mathbf{x} = \begin{bmatrix} T_1 \\ \vdots \\ T_{N_z} \\ \phi_1 \\ \vdots \\ \phi_{N_z} \end{bmatrix}$$



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

Projecting a linear system

Full-order model (FOM)
state $\mathbf{x} \in \mathbb{R}^N$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

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

Residual: N eqs $\gg$ r dof

$$\mathbf{r} = \mathbf{V}\dot{\mathbf{x}}_r - \mathbf{A}\mathbf{V}\mathbf{x}_r - \mathbf{B}\mathbf{u}$$

Project
$\mathbf{W}^\top\mathbf{r} = 0$
(Galerkin: $\mathbf{W} = \mathbf{V}$)

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

$$\dot{\mathbf{x}}_r = \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{u}$$

$$\boxed{\begin{aligned}\mathbf{A}_r &= \mathbf{V}^\top\mathbf{A}\mathbf{V}\\ \mathbf{B}_r &= \mathbf{V}^\top\mathbf{B}\end{aligned}}$$

Linear Model

FOM: $$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

ROM: $$\dot{\mathbf{x}}_r = \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{u}$$

Precompute the ROM matrices:

$$\mathbf{A}_r = \mathbf{V}^\top\mathbf{A}\mathbf{V}, \ \ \mathbf{B}_r = \mathbf{V}^\top\mathbf{B}$$

Quadratic Model

FOM: $$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{H}(\mathbf{x} \otimes \mathbf{x}) + \mathbf{B}\mathbf{u}$$

ROM: $$\dot{\mathbf{x}}_r = \mathbf{A}_r\mathbf{x}_r + \mathbf{H}_r(\mathbf{x}_r \otimes \mathbf{x}_r) + \mathbf{B}_r\mathbf{u}$$

Precompute the ROM matrices and tensor:

$$\mathbf{H}_r = \mathbf{V}^\top\mathbf{H}(\mathbf{V} \otimes \mathbf{V})$$

projection preserves structure $\longleftrightarrow$ 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

Peherstorfer & W.
Data-driven operator inference for nonintrusive projection-based model reduction, *Computer Methods in Applied Mechanics and Engineering*, 2016

$$\boxed{\dot{\hat{\mathbf{x}}} = \hat{\mathbf{A}}\hat{\mathbf{x}} + \hat{\mathbf{B}}\mathbf{u} + \hat{\mathbf{H}}(\hat{\mathbf{x}} \otimes \hat{\mathbf{x}})}$$

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

$$\hat{\mathbf{X}} = \begin{bmatrix} | & & | \\ \hat{\mathbf{x}}(t_1) & \dots & \hat{\mathbf{x}}(t_K) \\ | & & | \end{bmatrix} \qquad \dot{\hat{\mathbf{X}}} = \begin{bmatrix} | & & | \\ \dot{\hat{\mathbf{x}}}(t_1) & \dots & \dot{\hat{\mathbf{x}}}(t_K) \\ | & & | \end{bmatrix}$$

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

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{H}}} \left\| \hat{\mathbf{X}}^\top \hat{\mathbf{A}}^\top + \left(\hat{\mathbf{X}} \otimes \hat{\mathbf{X}}\right)^\top \hat{\mathbf{H}}^\top + \mathbf{U}^\top \hat{\mathbf{B}}^\top - \dot{\hat{\mathbf{X}}}^\top \right\|$$

- Generate $\hat{\mathbf{X}}$ data by projection of $\mathbf{X}$ snapshot data onto POD basis
- If data are Markovian, Operator Inference recovers the intrusive POD reduced model [Peherstorfer, 2019]

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ (E+p)u\end{pmatrix}=0$$

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho u^2$$

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ u\\ p\end{pmatrix}+\begin{pmatrix}\rho\frac{\partial u}{\partial x}+u\frac{\partial \rho}{\partial x}\\ u\frac{\partial u}{\partial x}+\frac{1}{\rho}\frac{\partial p}{\partial x}\\ \gamma p\frac{\partial u}{\partial x}+u\frac{\partial p}{\partial x}\end{pmatrix}=0$$

$$\frac{\partial}{\partial t}\begin{pmatrix}u\\ p\\ q\end{pmatrix}+\begin{pmatrix}u\frac{\partial u}{\partial x}+q\frac{\partial p}{\partial x}\\ \gamma p\frac{\partial u}{\partial x}+u\frac{\partial p}{\partial x}\\ q\frac{\partial u}{\partial x}+u\frac{\partial q}{\partial x}\end{pmatrix}=0$$

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

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho w\\ E\end{pmatrix}+\frac{\partial}{\partial z}\begin{pmatrix}\rho w\\ \rho w^2+p\\ (E+p)w\end{pmatrix}=0$$

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho w^2$$

conservative variables
mass, momentum, energy

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ w\\ p\end{pmatrix}+\begin{pmatrix}\rho\frac{\partial w}{\partial z}+w\frac{\partial \rho}{\partial z}\\ w\frac{\partial w}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial z}\\ \gamma p\frac{\partial w}{\partial z}+w\frac{\partial p}{\partial z}\end{pmatrix}=0$$

primitive variables
mass, velocity, pressure

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

$$\frac{\partial}{\partial t}\begin{pmatrix}w\\ p\\ q\end{pmatrix}+\begin{pmatrix}w\frac{\partial w}{\partial z}+q\frac{\partial p}{\partial z}\\ \gamma p\frac{\partial w}{\partial z}+w\frac{\partial p}{\partial z}\\ q\frac{\partial w}{\partial z}+w\frac{\partial q}{\partial z}\end{pmatrix}=0$$

specific volume variables

$$\dot{\mathbf{x}}=\mathbf{H}(\mathbf{x}\otimes\mathbf{x})+\mathbf{B}\mathbf{u}$$

transformed system
has quadratic structure

$$\dot{\mathbf{x}}_r=\mathbf{H}_r(\mathbf{x}_r\otimes\mathbf{x}_r)+\mathbf{B}_r\mathbf{u}$$

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,*

$$\mathcal{T} : \mathcal{S} \to \mathcal{W} \subset \mathbb{R}^{d_w}, \quad d_w \geq d_s, \tag{14}$$

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

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

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 $\boxed{\dot{x} = x^4 + u.}$

Introduce auxiliary variables: $w_1 = x^2 \quad w_2 = w_1^2$

Chain rule:
$$\dot{w}_1 = 2x[w_1^2 + u] = 2x[w_2 + u]$$
$$\dot{w}_2 = 2w_1\dot{w}_1 = 4xw_1[w_2 + u]$$

Need additional variable to make auxiliary dynamics quadratic:

$$w_3 = xw_1 \qquad \begin{aligned}\dot{w}_3 &= \dot{x}w_1 + x\dot{w}_1 \\ &= w_1w_2 + w_1u + 2w_1w_2 + 2w_1u\end{aligned}$$

QB-ODE

$$\boxed{\begin{aligned}\dot{x} &= w_2 + u \\ \dot{w}_1 &= 2xw_2 + 2xu \\ \dot{w}_2 &= 4w_2w_3 + 4w_3u \\ \dot{w}_3 &= 3w_1w_2 + 3w_1u\end{aligned}}$$

QB-DAE

$$\boxed{\begin{aligned}\dot{x} &= w_1^2 + u \\ 0 &= w_1 - x^2\end{aligned}}$$

$$\dot{T} + L\dot{\phi} = \nabla.\left(K\left(\phi\right)\nabla T\right)$$
$$\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right)$$

original equations

Many different forms of nonlinear PDEs can be lifted to polynomial form

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

cubic lifted equations

[Khodabakhshi, W. In preparation.]

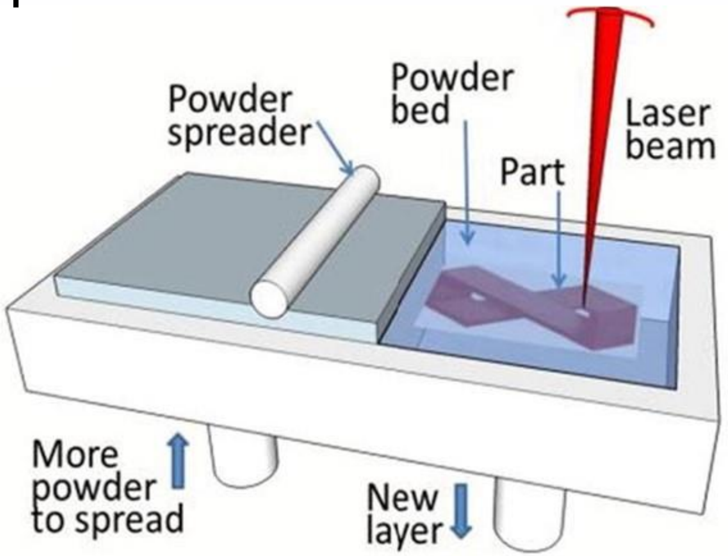
Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \end{bmatrix}$$

Nonlinear system for 1D solidification

$$\boxed{\begin{aligned} &\dot{T} + L\dot{\phi} = \nabla . \left(K\left(\phi\right)\nabla T\right) \\ &\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right) \end{aligned}}$$

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



DebRoy et al. *Progress in Materials Science*, 2018

Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \end{bmatrix}$$

Nonlinear system for 1D solidification

$$\boxed{\begin{aligned} &\dot{T} + L\dot{\phi} = \nabla.\left(K\left(\phi\right)\nabla T\right) \\ &\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right) \end{aligned}}$$

with
$$\begin{aligned} K\left(\phi\right) &= K_0\left(1-h\left(\phi\right)\right) + K_1 h\left(\phi\right) \\ h\left(\phi\right) &= 6\phi^5 - 15\phi^4 + 10\phi^3 \\ p\left(\phi\right) &= \frac{1}{4}\phi^2\left(1-\phi\right)^2 \\ q\left(T,\phi\right) &= \frac{\beta}{2}\phi\left(\phi-1\right)\tanh\left[\gamma\left(T_{\mathrm{melt}}-T\right)\right] \end{aligned}$$

Chain rule:

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

Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \\ K \end{bmatrix}$$

Nonlinear system for 1D solidification

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

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

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

Chain rule:

$$p = \frac{1}{4}\phi^2\left(1-\phi\right)^2$$

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

Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \\ K \\ p \end{bmatrix}$$

Nonlinear system for 1D solidification

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

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

Chain rule:

$$p' = \frac{1}{2}\phi\left(1-\phi\right)\left(1-2\phi\right)$$

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

Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \\ K \\ p \\ p' \end{bmatrix}$$

Nonlinear system for 1D solidification

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

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

Chain rule:

$$p'' = 3\left(\phi^2 - \phi\right) + \frac{1}{2}$$

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

Solidification of a Pure Material

$$\begin{bmatrix} T \\ \phi \\ K \\ p \\ p' \\ p'' \end{bmatrix}$$

Nonlinear system for 1D solidification

$$\boxed{\begin{aligned}
&\dot{T} + L\dot{\phi} = \nabla.\left(K\nabla T\right) \\
&\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\left(T\right) \\
&\dot{K} = \frac{120\left(K_1 - K_0\right)}{\alpha\xi^2}p\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\left(T\right)\right] \\
&\dot{p} = \frac{1}{\alpha\xi^2}p'\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\left(T\right)\right] \\
&\dot{p'} = \frac{1}{\alpha\xi^2}p''\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\left(T\right)\right] \\
&\dot{p''} = \frac{3}{\alpha\xi^2}\left(2\phi - 1\right)\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\left(T\right)\right]
\end{aligned}}$$

Chain rule:

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

$$\begin{aligned}
\dot{m}_0 &= -\gamma\left(1 - m_0^2\right)\dot{T} \\
&= -\gamma\left(1 - m_0^2\right)\left\{\nabla.\left(K\nabla T\right) - \frac{L}{\alpha\xi^2}\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]\right\}
\end{aligned}$$

Solidification of a Pure Material

Original system:

$$\dot{T} + L\dot{\phi} = \nabla.\left(K\left(\phi\right)\nabla T\right)$$
$$\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right)$$

with original variables T, ϕ

Nonlinear system for 1D solidification

$$\dot{T} + L\dot{\phi} = \nabla.\left(K\nabla T\right)$$ **Quadratic**

$$\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0$$ **Quadratic**

$$\dot{K} = \frac{120\left(K_1 - K_0\right)}{\alpha\xi^2}p\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]$$ **Cubic**

$$\dot{p} = \frac{1}{\alpha\xi^2}p'\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]$$ **Cubic**

$$\dot{p'} = \frac{1}{\alpha\xi^2}p''\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]$$ **Cubic**

$$\dot{p''} = \frac{3}{\alpha\xi^2}\left(2\phi - 1\right)\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]$$ **Cubic**

$$\dot{m}_0 = -\gamma y\left\{\nabla.\left(K\nabla T\right) - \frac{L}{\alpha\xi^2}\left[\xi^2\Delta\phi - p' - \frac{\beta}{6}\left(p'' - \frac{1}{2}\right)m_0\right]\right\}$$ **Cubic**

$$y = 1 - m_0^2$$ **Quadratic**

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)

$$\mathbf{X}_{\mathbf{orig}} = \begin{bmatrix} | & & | \\ \mathbf{x}(t_1) & \dots & \mathbf{x}(t_K) \\ | & & | \end{bmatrix} \quad \dot{\mathbf{X}}_{\mathbf{orig}} = \begin{bmatrix} | & & | \\ \dot{\mathbf{x}}(t_1) & \dots & \dot{\mathbf{x}}(t_K) \\ | & & | \end{bmatrix}$$

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)

$$\mathbf{X_{orig}} \longrightarrow \mathbf{X} \qquad \dot{\mathbf{X}}_{\mathbf{orig}} \longrightarrow \dot{\mathbf{X}}$$

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

$$\mathbf{X} = \mathbf{V}\,\boldsymbol{\Sigma}\,\mathbf{W}^\top$$

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

$$\hat{\mathbf{X}} = \mathbf{V}^{\top}\mathbf{X}$$

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
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

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{H}}} \left\| \hat{\mathbf{X}}^\top \hat{\mathbf{A}}^\top + \left(\hat{\mathbf{X}} \otimes \hat{\mathbf{X}}\right)^\top \hat{\mathbf{H}}^\top + \mathbf{U}^\top \hat{\mathbf{B}}^\top - \dot{\hat{\mathbf{X}}}^\top \right\|$$

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

$$\dot{T} + L\dot{\phi} = \nabla.\left(K\left(\phi\right)\nabla T\right)$$

$$\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right)$$

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

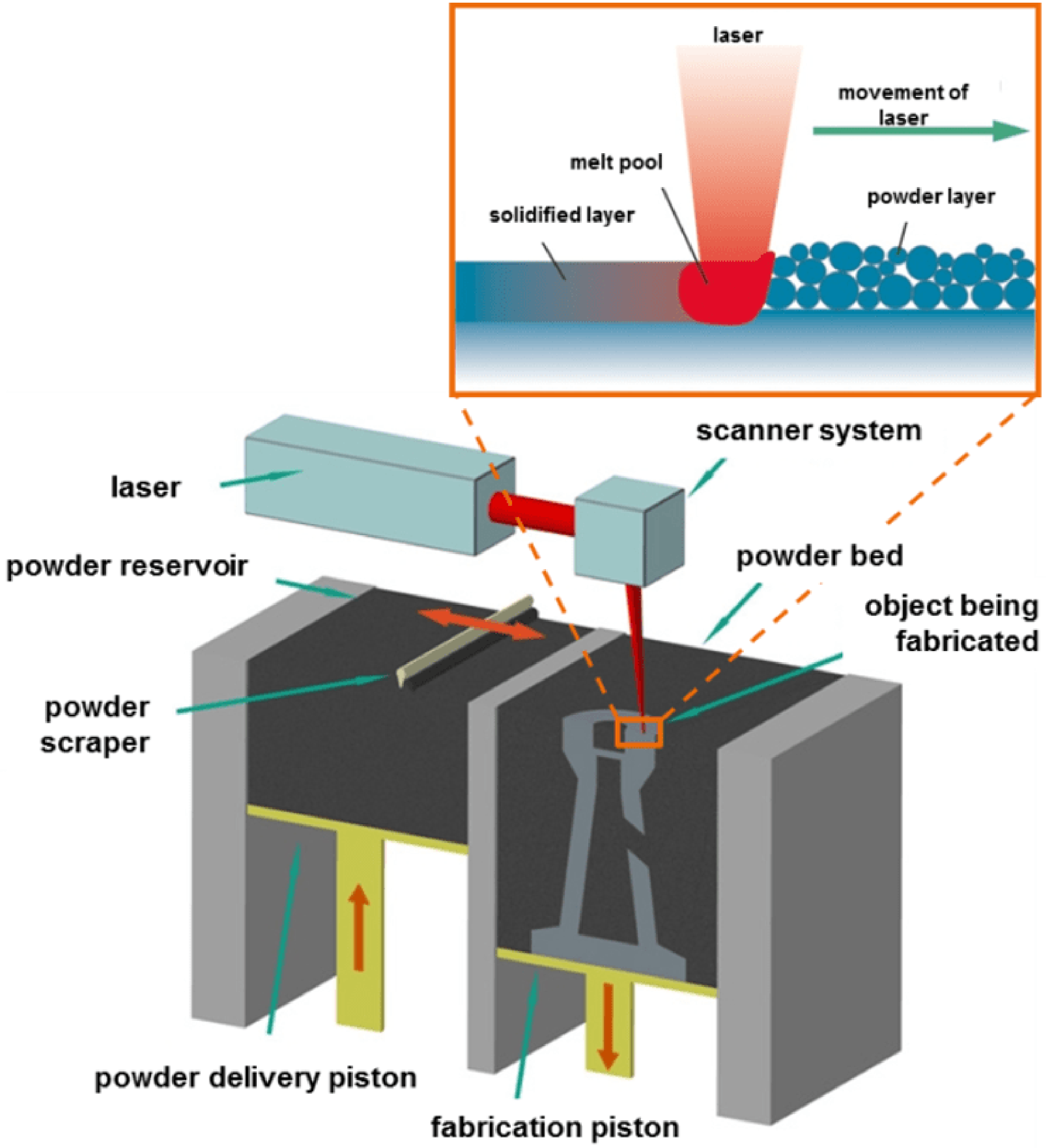
$$T\left(x,0\right) = 0.4$$

$$\phi\left(x,0\right) = 0.5\cos\left(\pi x\right) + 0.5$$

- Boundary conditions

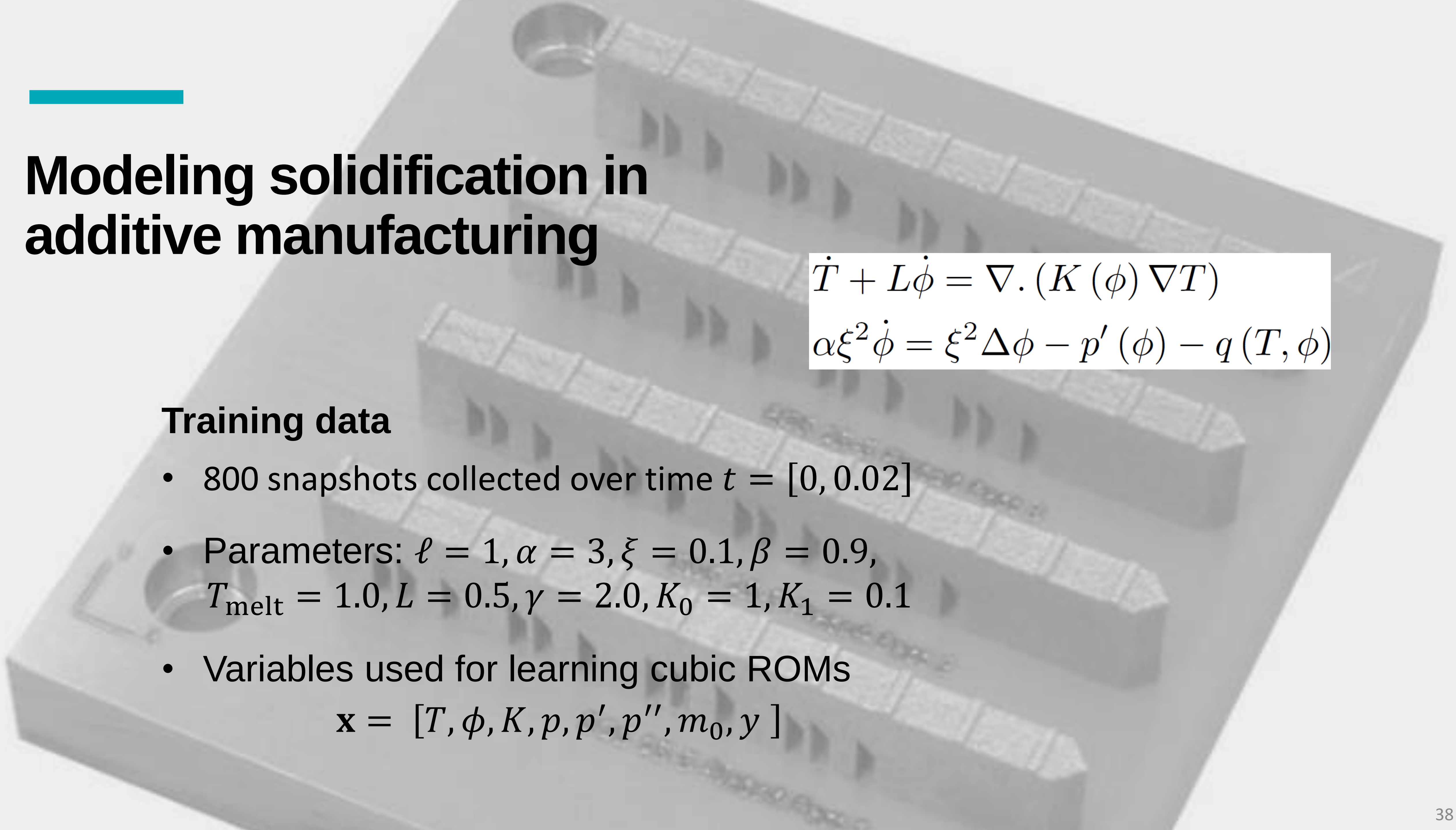
$$T\left(0,t\right) = T\left(\ell,t\right) 0.4$$

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=0} = \left.\frac{\partial\phi}{\partial x}\right|_{x=\ell} = 0$$



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

Modeling solidification in additive manufacturing



$$\dot{T} + L\dot{\phi} = \nabla.\left(K\left(\phi\right)\nabla T\right)$$

$$\alpha\xi^2\dot{\phi} = \xi^2\Delta\phi - p'\left(\phi\right) - q\left(T,\phi\right)$$

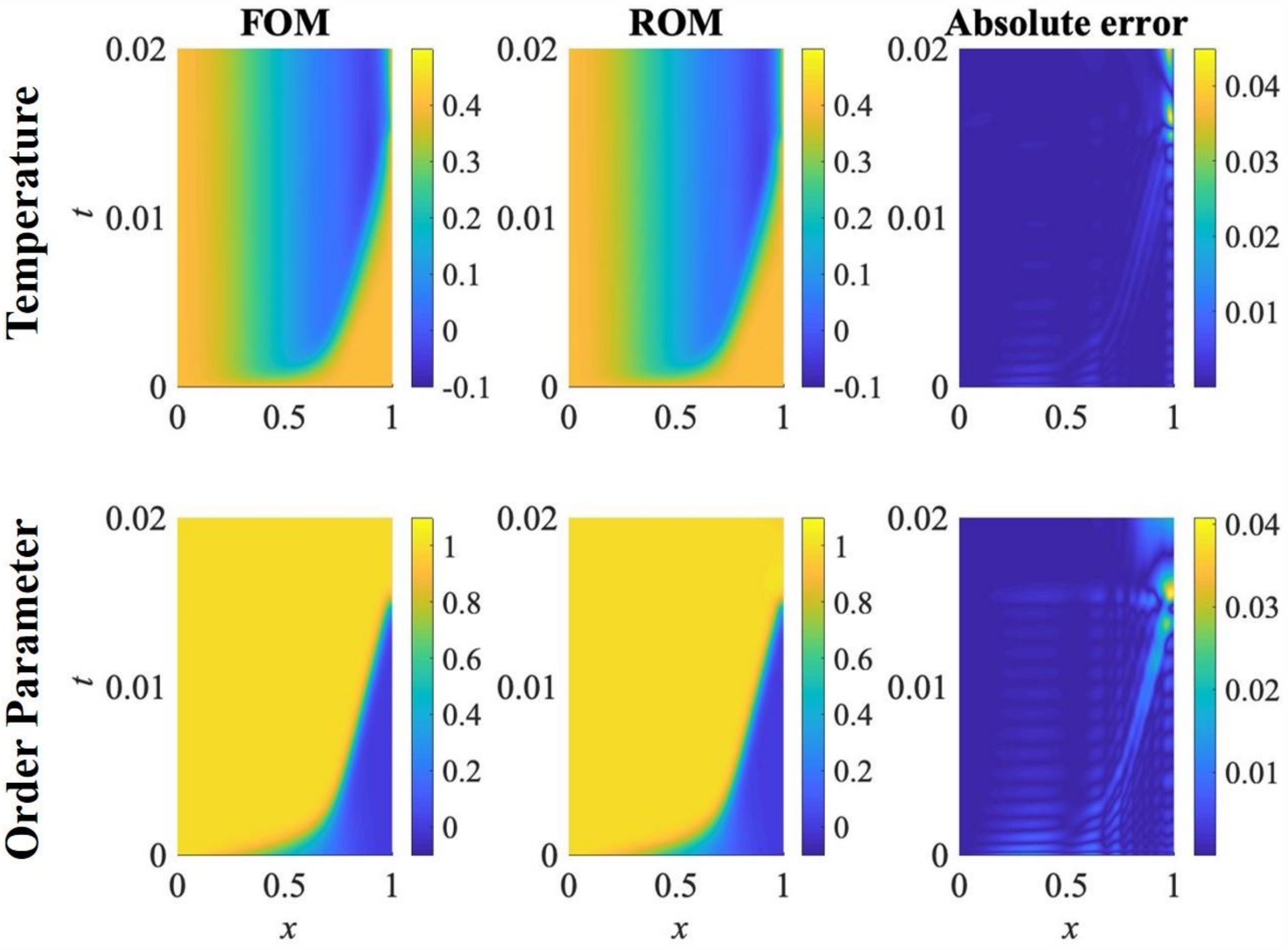
Training data

- 800 snapshots collected over time $t = [0, 0.02]$
- Parameters: $\ell = 1, \alpha = 3, \xi = 0.1, \beta = 0.9,$ $T_{\mathrm{melt}} = 1.0, L = 0.5, \gamma = 2.0, K_0 = 1, K_1 = 0.1$
- Variables used for learning cubic ROMs

$$\mathbf{x} = \ [T, \phi, K, p, p', p'', m_0, y\ ]$$

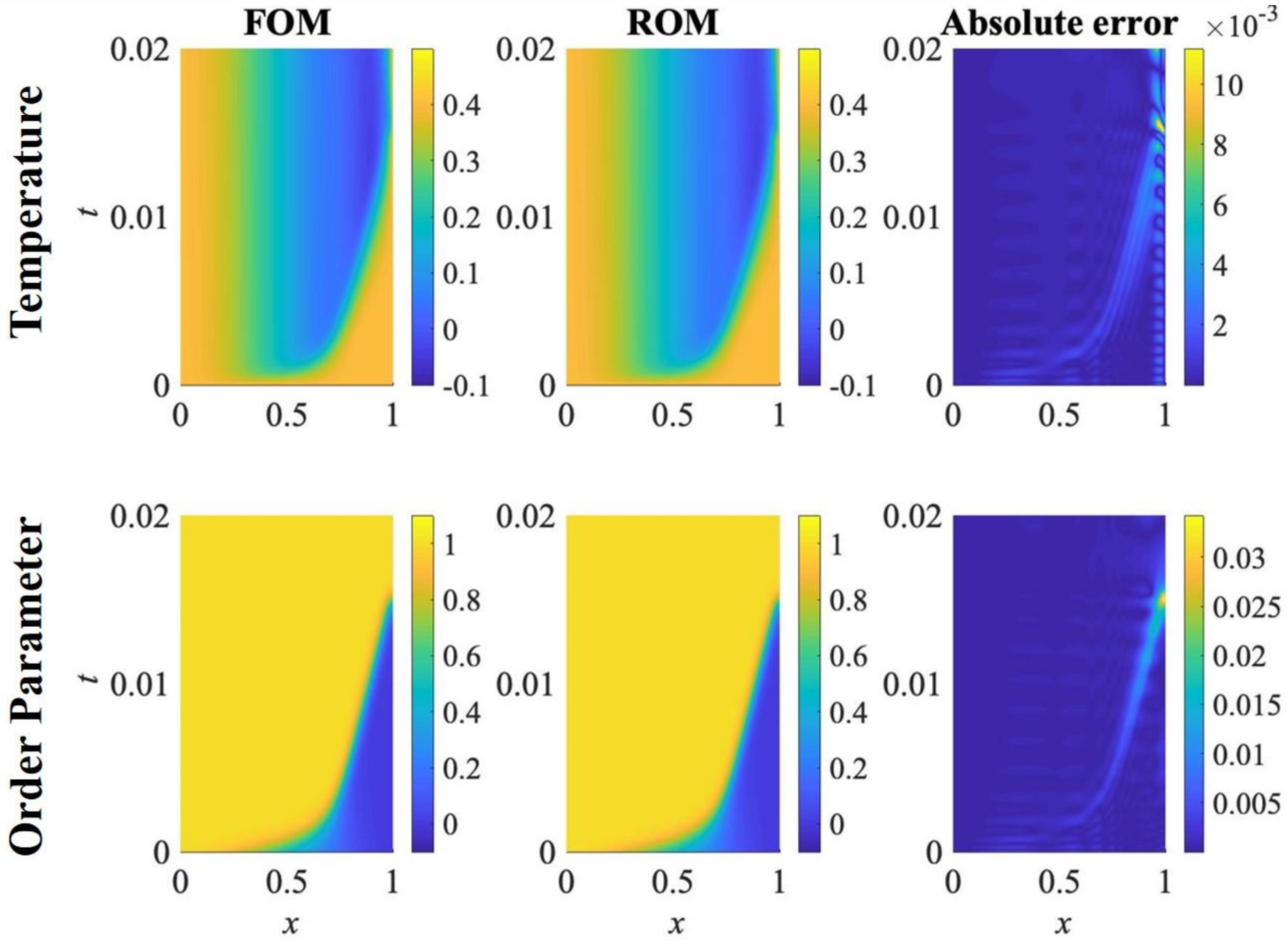
Lift & Learn reduced model performance

- 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

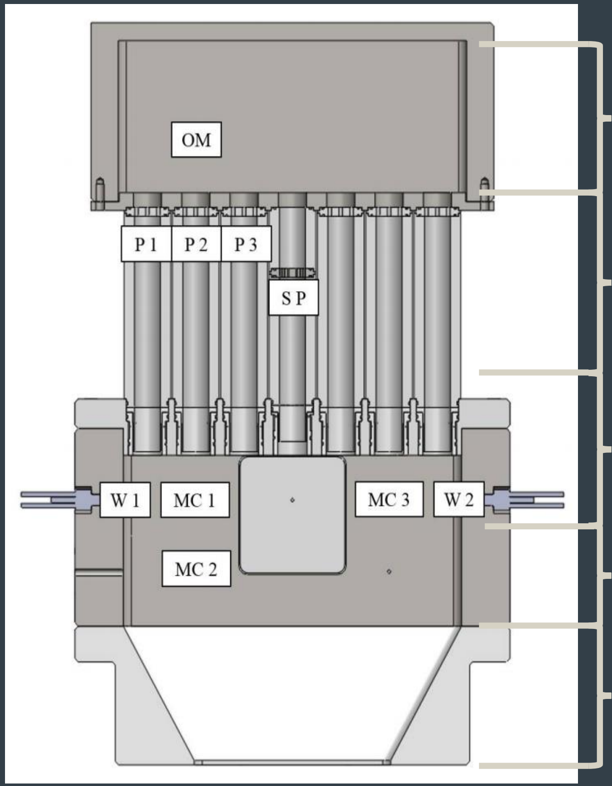


Rocket Engine Combustion

Lift & Learn reduced models for a complex Air Force combustion problem

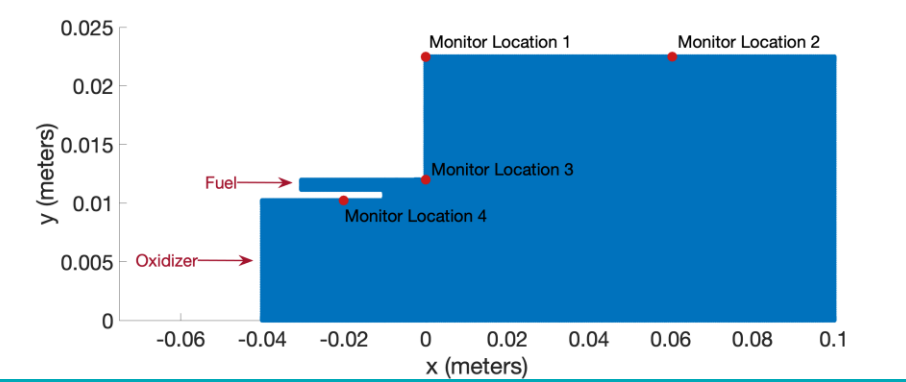
Modeling a single injector of a rocket engine combustor

- Spatial domain (2D) discretized into $38{,}523$ cells
- Oxidizer input: 0.37 $\frac{\mathrm{kg}}{\mathrm{s}}$ of 42% $\mathrm{O_2}$ / 58% $\mathrm{H_2O}$
- Fuel input: 5.0 $\frac{\mathrm{kg}}{\mathrm{s}}$ of $\mathrm{CH_4}$
- Forced by a back pressure boundary condition at exit throat



$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho v_x\\ \rho v_y\\ \rho E\\ \rho Y_1\\ \vdots\\ \rho Y_{n_{\mathrm{sp}}}\end{bmatrix} + \nabla\cdot\left(\begin{bmatrix}\rho v_x\\ \rho v_x^2 + p\\ \rho v_x v_y\\ \rho v_x E + p v_x\\ \rho v_x Y_1\\ \vdots\\ \rho v_x Y_{n_{\mathrm{sp}}}\end{bmatrix}\vec{i} + \begin{bmatrix}\rho v_y\\ \rho v_x v_y\\ \rho v_y^2 + p\\ \rho v_y E + p v_y\\ \rho v_y Y_1\\ \vdots\\ \rho v_y Y_{n_{\mathrm{sp}}}\end{bmatrix}\vec{j} - \begin{bmatrix}0\\ \tau_{xx}\\ \tau_{yx}\\ \tau_{xx} v_x + \tau_{yx} v_y - j_x^q\\ -j_{1,x}^m\\ \vdots\\ -j_{n_{\mathrm{sp}},x}^m\end{bmatrix}\vec{i} - \begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{xy} v_x + \tau_{yy} v_y - j_y^q\\ -j_{1,y}^m\\ \vdots\\ -j_{n_{\mathrm{sp}},y}^m\end{bmatrix}\vec{j}\right) = \begin{bmatrix}0\\ 0\\ 0\\ 0\\ \dot{\omega}_1\\ \vdots\\ \dot{\omega}_{n_{\mathrm{sp}}}\end{bmatrix}$$

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
  $\mathbf{x} = [\mathbf{p} \quad \mathbf{u} \quad \mathbf{v} \quad 1/\boldsymbol{\rho} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{CH_4}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{O_2}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{CO_2}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{H_2O}}]$
  makes many (but not all) terms in governing equations quadratic
- Snapshot matrix $\mathbf{X} \in \mathbb{R}^{308{,}184 \times 10{,}000}$

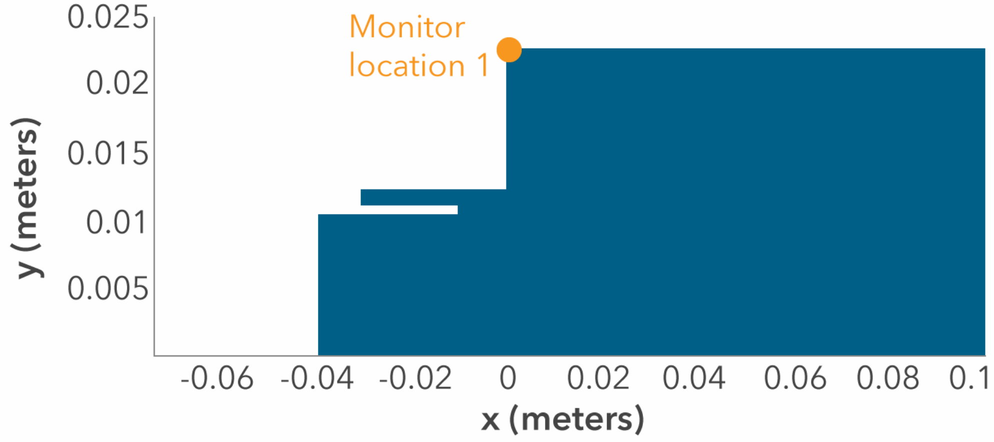
Test data

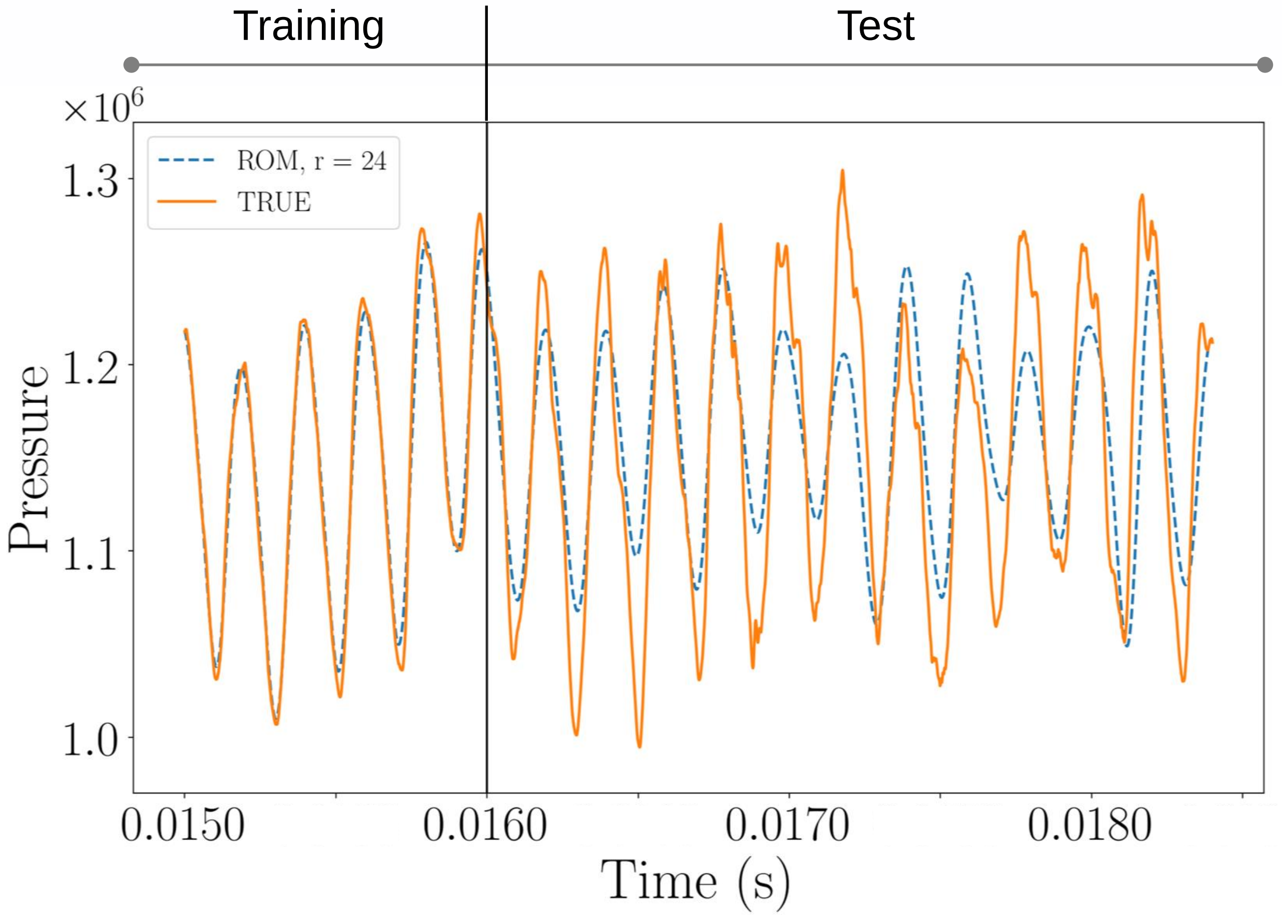
Additional 2 ms of data at four monitor locations (20,000 timesteps)

Performance of learned quadratic ROM

Pressure time traces at monitor location 1

Basis size $r = 24$

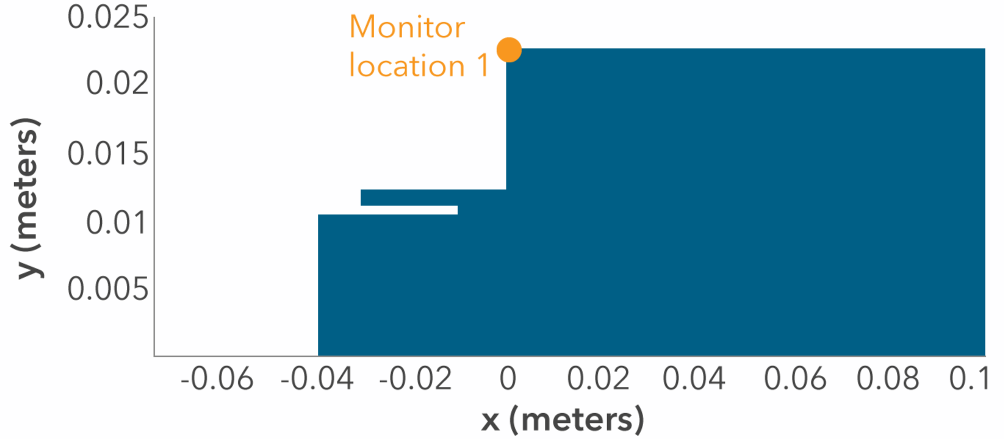


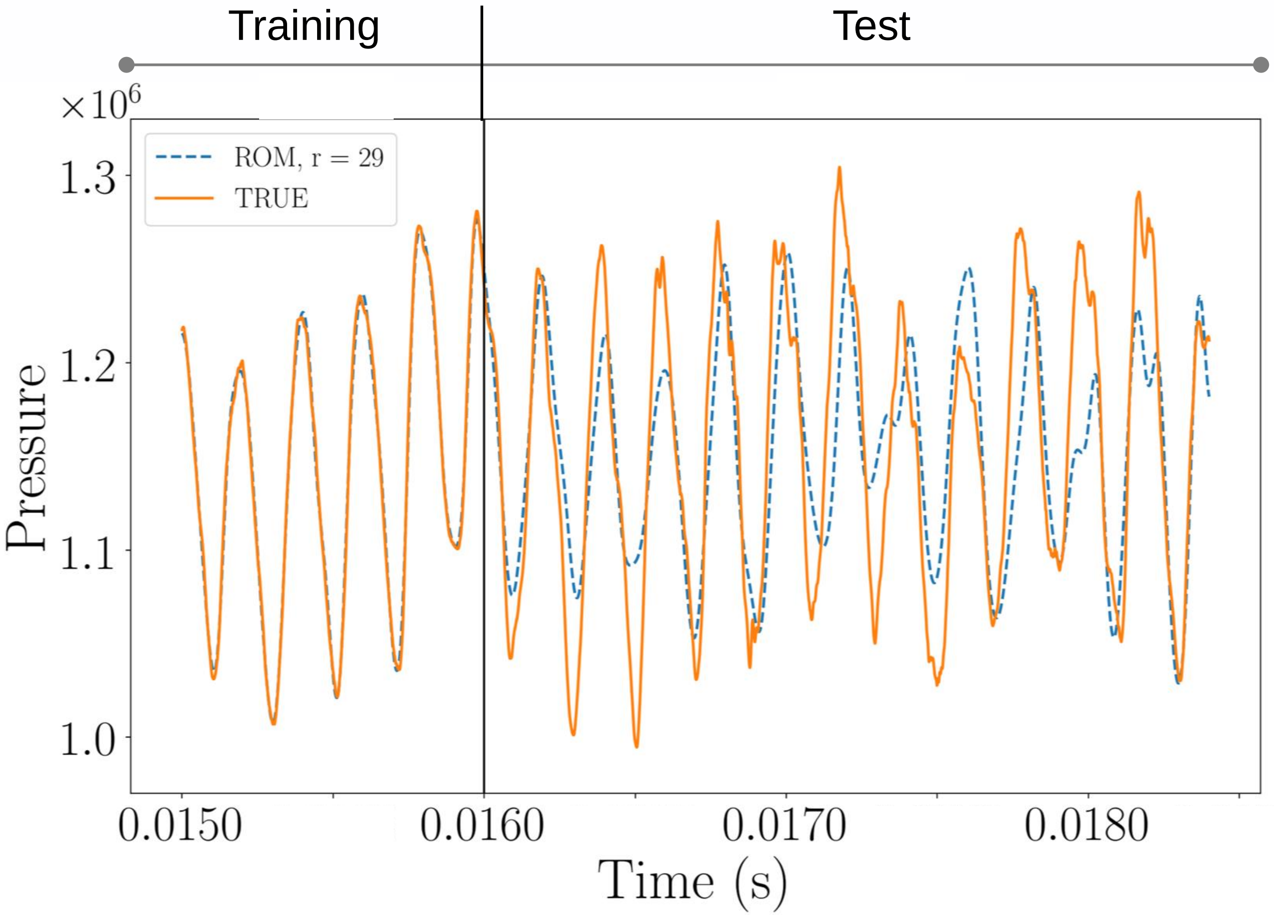


Performance of learned quadratic ROM

Pressure time traces at monitor location 1

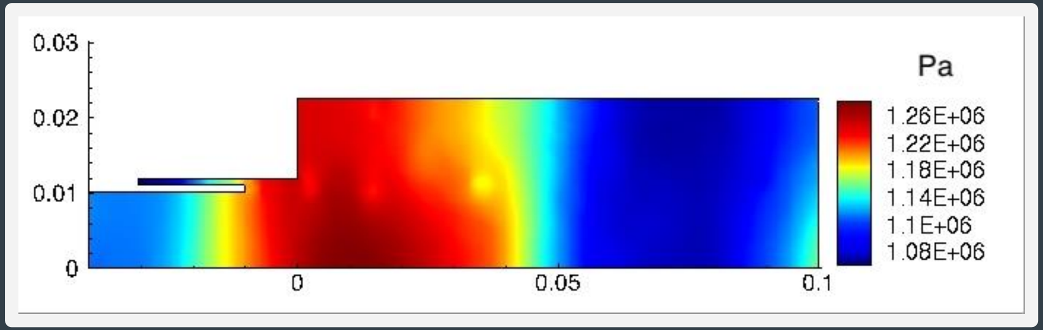
Basis size $r = 29$



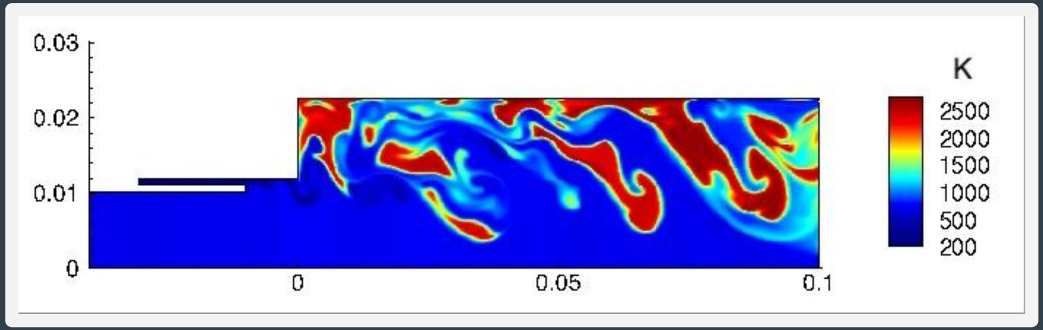


True

Pressure

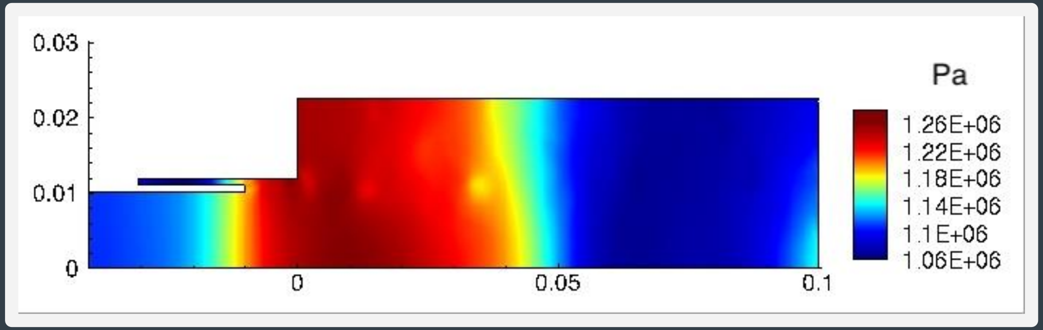


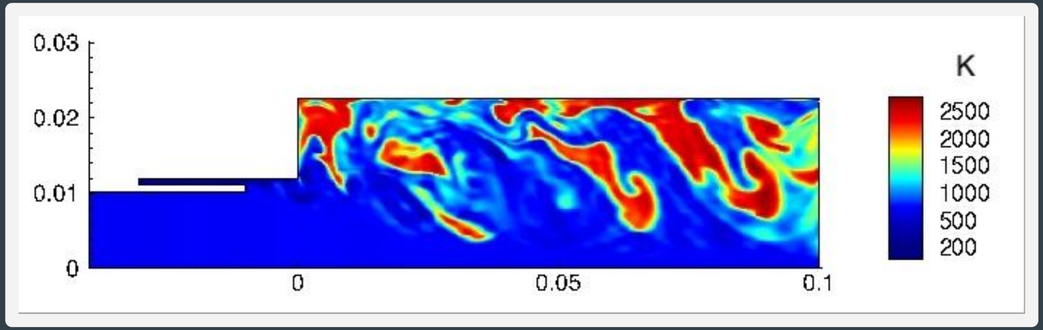
Temperature



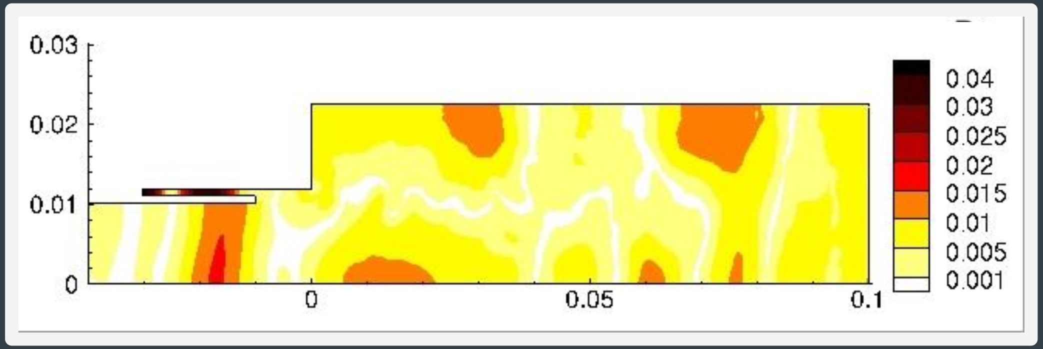
Predicted

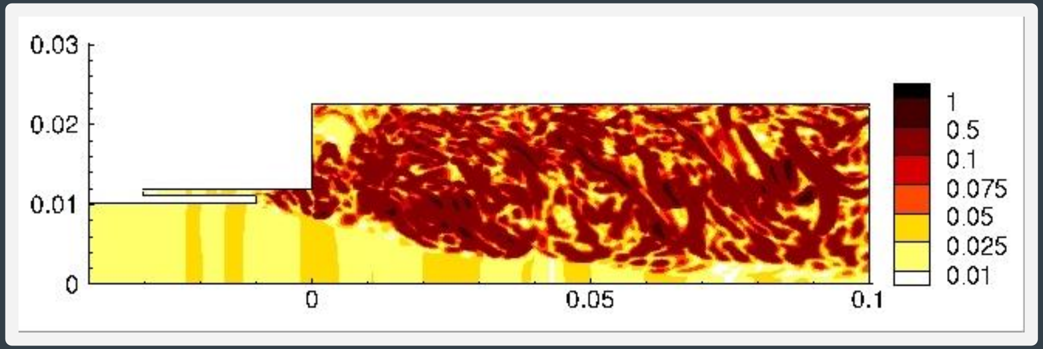
$r = 29$ POD modes





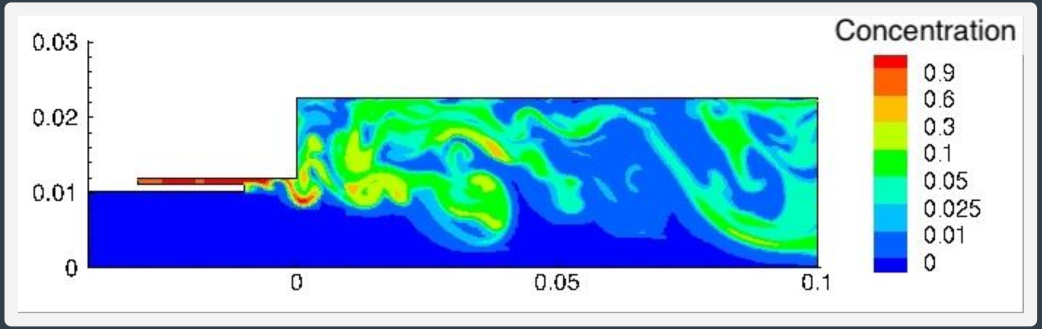
Relative error

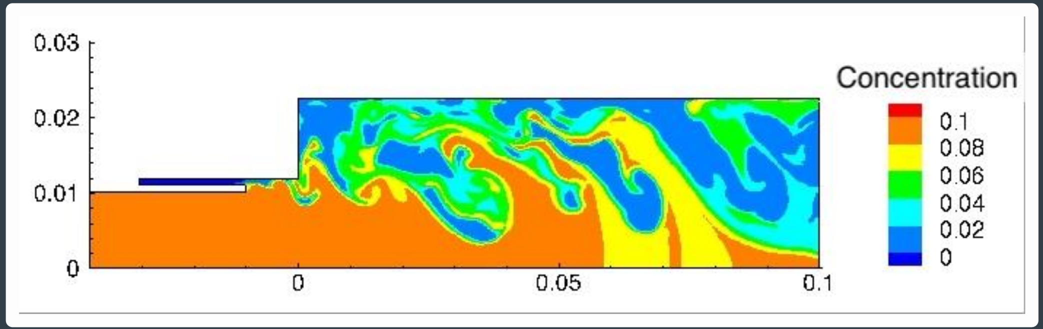




True

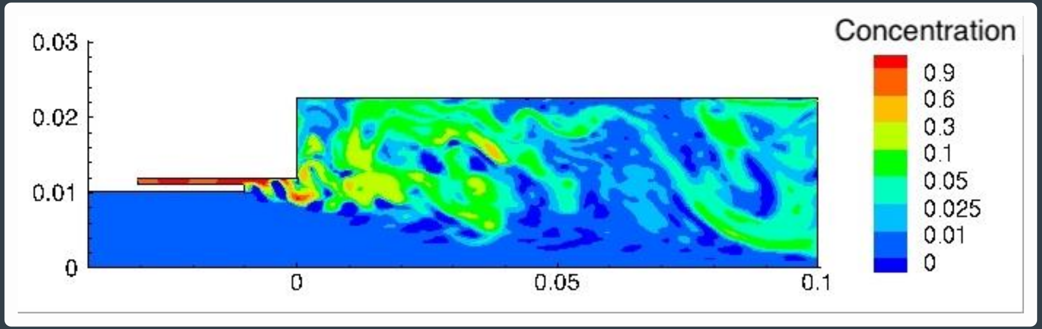
CH_4 O_2

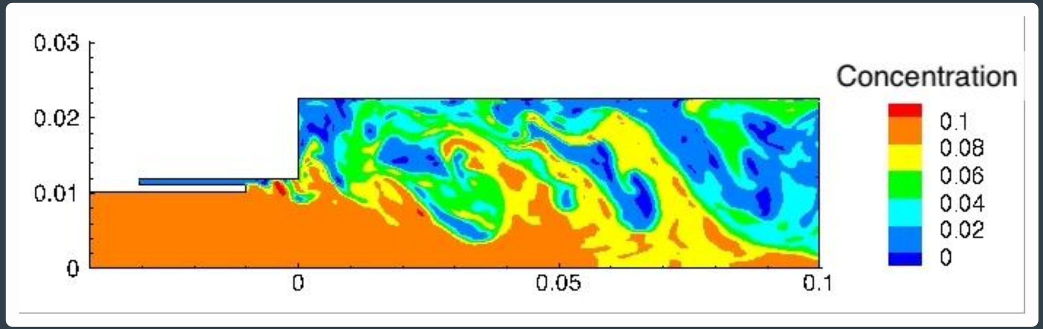




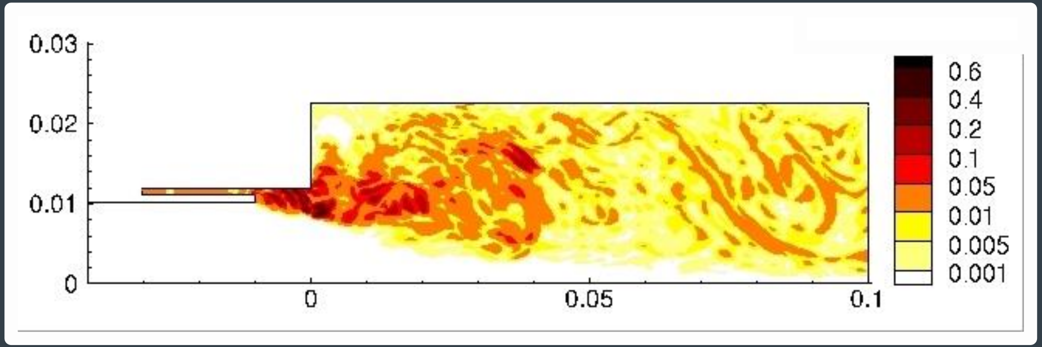
Predicted

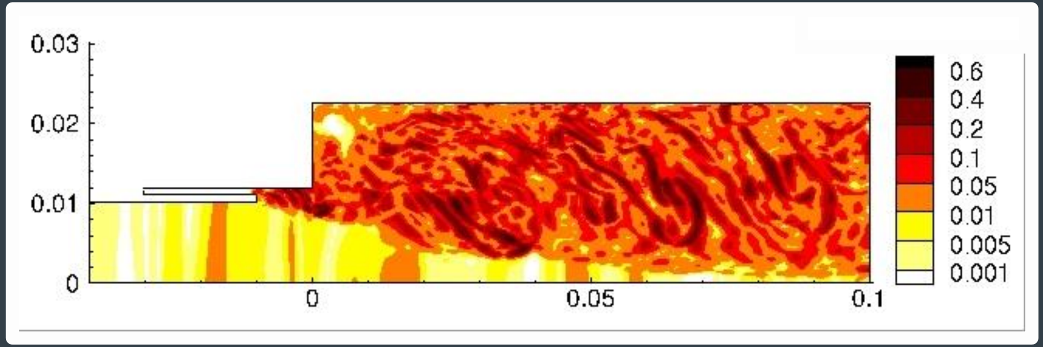
$r = 29$ POD modes





Normalized absolute error





Conclusions & Outlook

What future for model reduction?

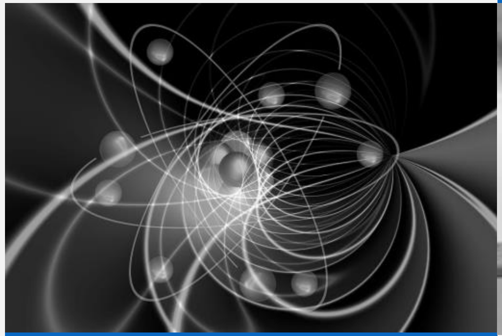
Scientific Machine Learning

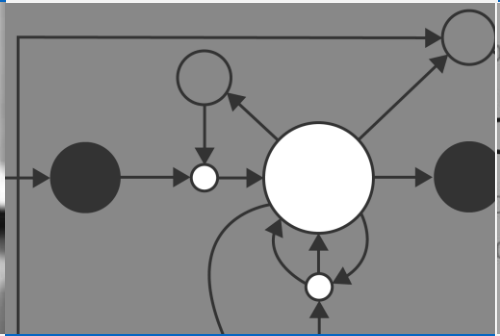
What role for model reduction?

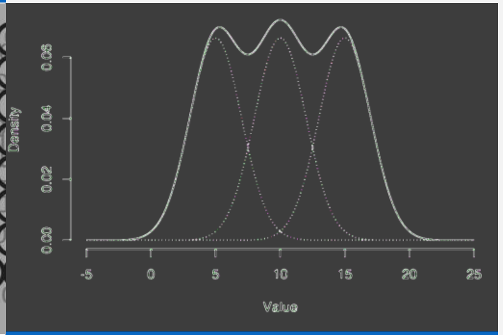
reduce the cost of training | foundational shift in ML perspectives

Embed domain knowledge

Integrate heterogeneous, noisy & incomplete data







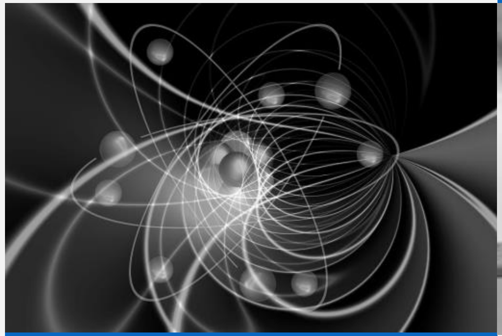
Respect physical constraints

Bring interpretability to results

Get predictions with quantified uncertainties

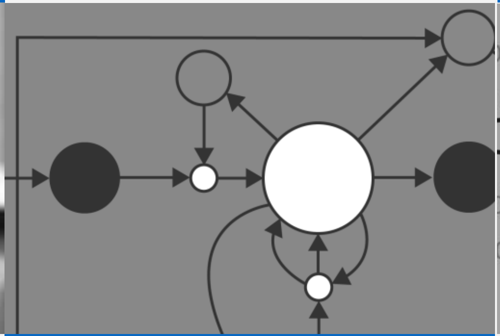
Scientific Machine Learning

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



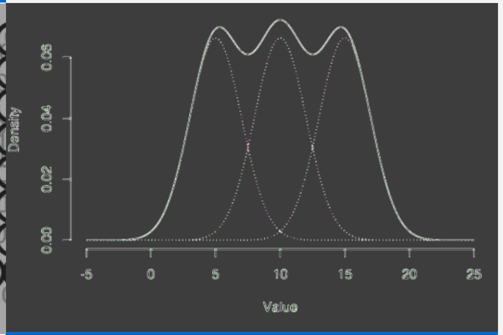
Respect physical constraints

Embed domain knowledge



Bring interpretability to results

Integrate heterogeneous, noisy & incomplete data



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

KIWI.ODEN.UTEXAS.EDU

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES