ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

Challenges and progress in learning physics-based reduced models for combustion processes

Karen E. Willcox

NSF Workshop on Exuberance of Machine Learning in Transport Phenomena
Dallas, TX
February 10-11, 2020

TEXAS
The University of Texas at Austin

The Team

Dr. Cheng Huang

U. Michigan

Prof. Boris Kramer

UCSD

Elizabeth Qian

MIT

Prof. Benjamin Peherstorfer

Courant Institute

Renee Swischuk

MIT → Caliper

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**

Outline

1 **Motivation**

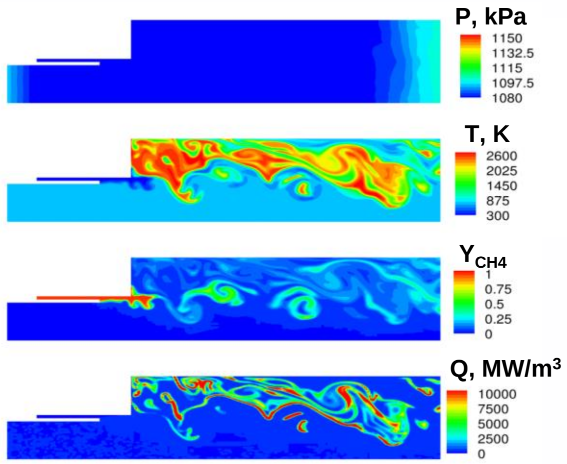
Machine learning vs. model reduction
to create efficient surrogate models

2 **Lift & Learn**

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

3 **Conclusions & Outlook**

Motivating example



Modeling combustion in a rocket engine:
Conservation of mass (ρ), momentum ($\rho\vec{v}$), energy (E),
species concentration ($Y_{\mathrm{CH_4}}$, $Y_{\mathrm{O_2}}$, $Y_{\mathrm{CO_2}}$, $Y_{\mathrm{H_2O}}$)

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

Reduced/surrogate models enable **rapid prediction**, **inversion**, **design**, and **uncertainty quantification** of large-scale scientific and engineering systems

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]

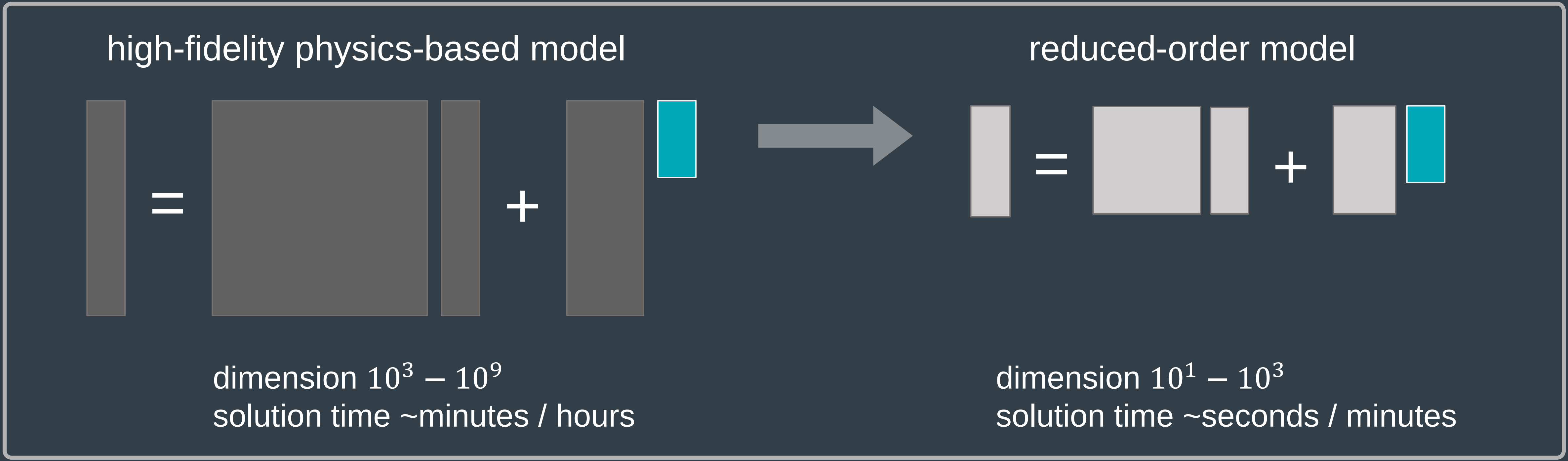
Reduced-order modeling

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

Reduced-order modeling & **machine learning**:
Two different paths to efficient surrogate models

Model reduction methods have grown from CSE, with a focus on ***reducing* high-dimensional models** that arise from physics-based modeling, whereas machine learning has grown from CS, with a focus on ***creating* low-dimensional models** from black-box data streams. [Swischuk et al., *Computers & Fluids*, 2018]

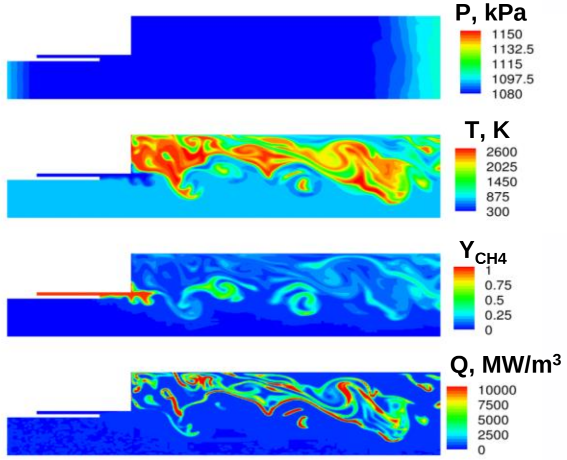
Can we get the best of both worlds?



Projection-based model reduction

1 **Train**: Solve PDEs to generate training data (<u>snapshots</u>)
2 **Identify structure**: Compute a <u>low-dimensional basis</u>
3 **Reduce**: <u>Project</u> PDE model onto the low-dimensional subspace

Start with a physics-based model



Example: modeling combustion in a rocket engine
Conservation of mass (ρ), momentum ($\rho\vec{v}$), energy (E), species ($Y_{\mathrm{CH_4}}$, $Y_{\mathrm{O_2}}$, $Y_{\mathrm{CO_2}}$, $Y_{\mathrm{H_2O}}$)

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

Discretize:
Spatially discretized computational fluid dynamic (CFD) model

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

discretized **state $\mathbf{x}$** contains mass, momentum, energy, species concentrations at N_z spatial grid points

$N_z \sim O(10^4 - 10^6)$

$$\mathbf{x} = \begin{bmatrix}\rho_1\\ (\rho v_x)_1\\ (\rho v_y)_1\\ (\rho E)_1\\ (\rho Y)_1\\ \vdots\\ \rho_{N_z}\\ \vdots\\ \vdots\\ (\rho Y_{n_{sp}})_{N_z}\end{bmatrix}$$

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

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

$$\downarrow$$

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

$$\downarrow$$

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

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?**

Non-intrusive implementation
Discover hidden structure
Black-box
Flexible

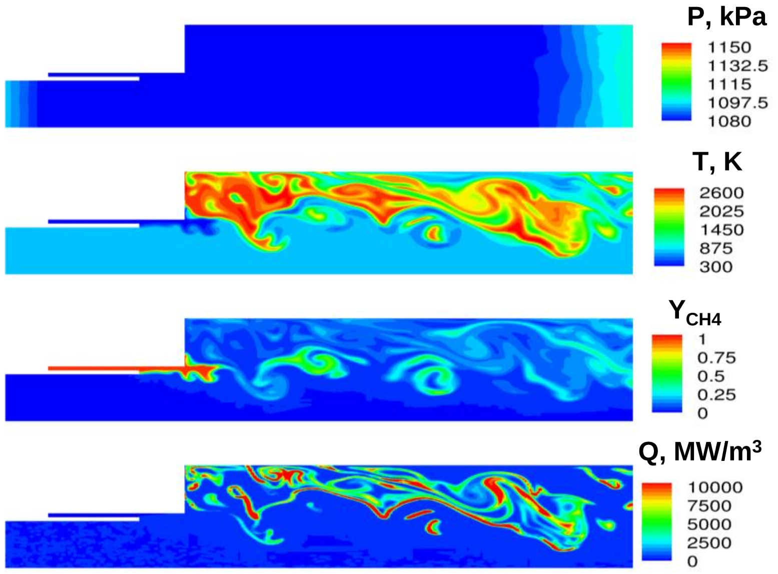
Embed governing equations
Structure-preserving
Predictive (error estimators)
Stability

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

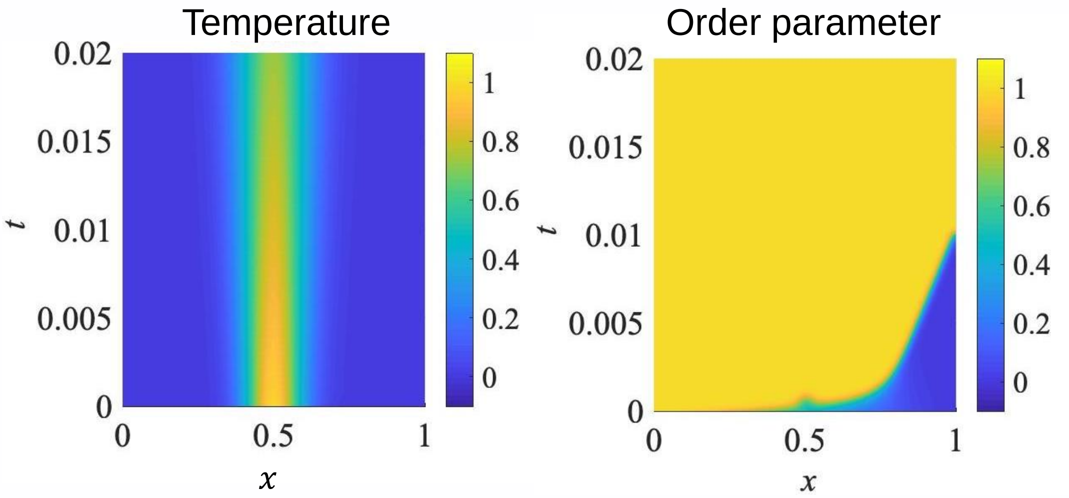


Rocket combustion

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



Solidification process in additive manufacturing

Operator inference

Non-intrusive learning of reduced models from simulation snapshot data

Given state data, learn the system

In principle could learn a large, sparse system e.g., Schaeffer, Tran & Ward, 2017

$$\dot{\mathbf{x}} = \underbrace{\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}}_{\text{linear}} + \underbrace{\mathbf{H}(\mathbf{x} \otimes \mathbf{x})}_{\text{quadratic}}$$

Given state data ($\mathbf{X}$) and velocity data ($\dot{\mathbf{X}}$):

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

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

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

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 velocity 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 appropriately generated, 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]

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:

$$w_3 = xw_1 \qquad \dot{w}_3 = \dot{x}w_1 + x\dot{w}_1$$

$$= w_1w_2 + w_1u + 2w_1w_2 + 2w_1u$$

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

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

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

$$\dot{\psi} = \frac{1}{Pe}\psi_{ss} - \psi_s - \mathcal{D}\psi e^{\gamma - \frac{\gamma}{\theta}}$$
$$\dot{\theta} = \frac{1}{Pe}\theta_{ss} - \theta_s - \beta(\theta - \theta_{\text{ref}}) + \mathcal{B}\mathcal{D}\psi e^{\gamma - \frac{\gamma}{\theta}}$$

original equations

$$\dot{\psi} = \underbrace{\frac{1}{Pe}\psi_{ss} - \psi_s - \mathcal{D}w_4}_{\text{linear}}$$
$$\dot{\theta} = \underbrace{\frac{1}{Pe}\theta_{ss} - \theta_s - \beta(\theta - \theta_{\text{ref}}) + \mathcal{B}\mathcal{D}w_4}_{\text{linear}}$$
$$\dot{w}_1 = \gamma \; w_6 \left[\frac{1}{Pe}\psi_{ss} - \psi_s\right] + \gamma\mathcal{B}\mathcal{D} \; w_4 w_6$$
$$\dot{w}_2 = -2 \; w_5 \odot \left[\frac{1}{Pe}\psi_{ss} - \psi_s\right] - 2\mathcal{B}\mathcal{D} \; w_4 w_5$$
$$\dot{w}_3 = -w_2 \odot \left[\frac{1}{Pe}\psi_{ss} - \psi_s\right] - \mathcal{B}\mathcal{D} \; w_2 w_4$$
$$0 = w_4 - w_1\psi$$
$$0 = w_5 - w_2 w_3$$
$$0 = w_6 - w_1 w_2$$

quadratic-bilinear lifted equations

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., *Physica D*, 2020]

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., *Physica D*, 2020]

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)

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

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., *Physica D*, 2020]

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., *Physica D*, 2020]

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., *Physica D*, 2020]

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., *Physica D*, 2020]

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

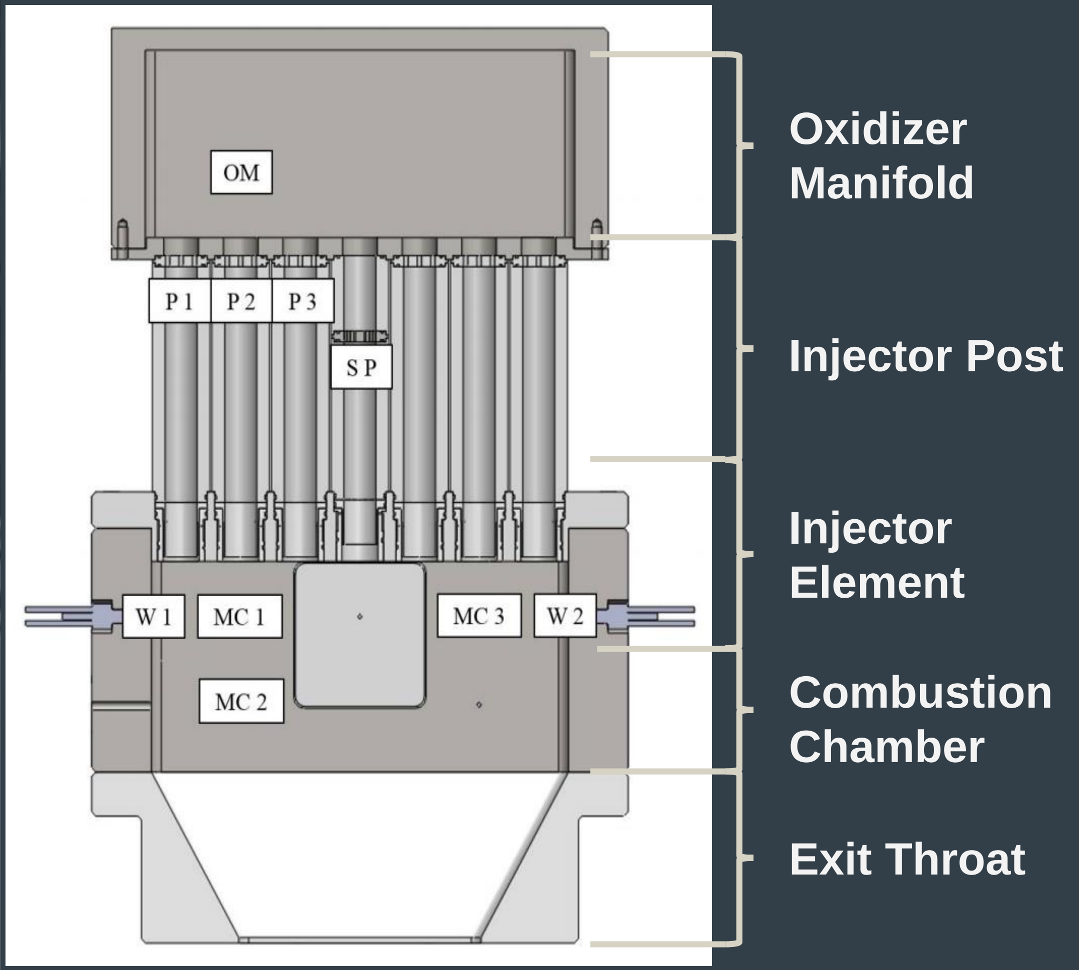
→ **convenience** of black-box learning +
rigor of projection-based reduction +
structure imposed by physics

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



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

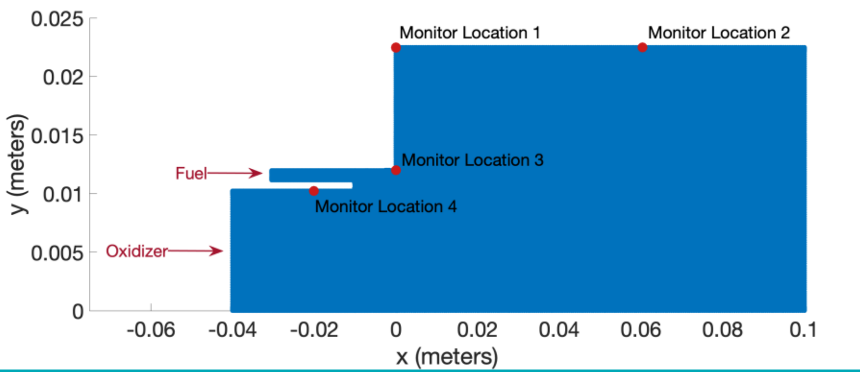
Modeling a single injector of a rocket engine combustor

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

  $$\mathbf{x} = [\mathbf{p} \quad \mathbf{u} \quad \mathbf{v} \quad \mathbf{1}/\boldsymbol{\rho} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{CH_4}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{O_2}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{CO_2}} \quad \boldsymbol{\rho}\mathbf{Y}_{\mathbf{H_2O}}]$$

  makes many (but not all) terms in governing equations quadratic
- Snapshot matrix $\mathbf{X} \in \mathbb{R}^{308,184 \times 10,000}$



Test data

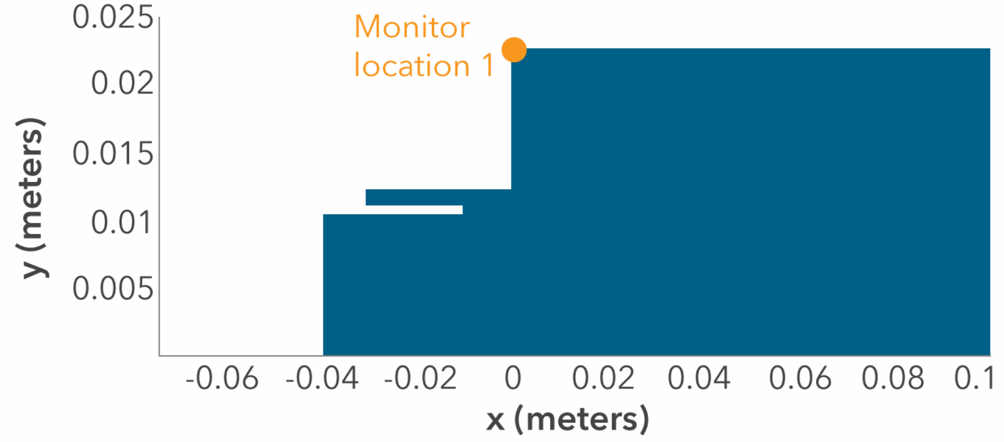
Additional 2 ms of data at four monitor locations (20,000 timesteps)

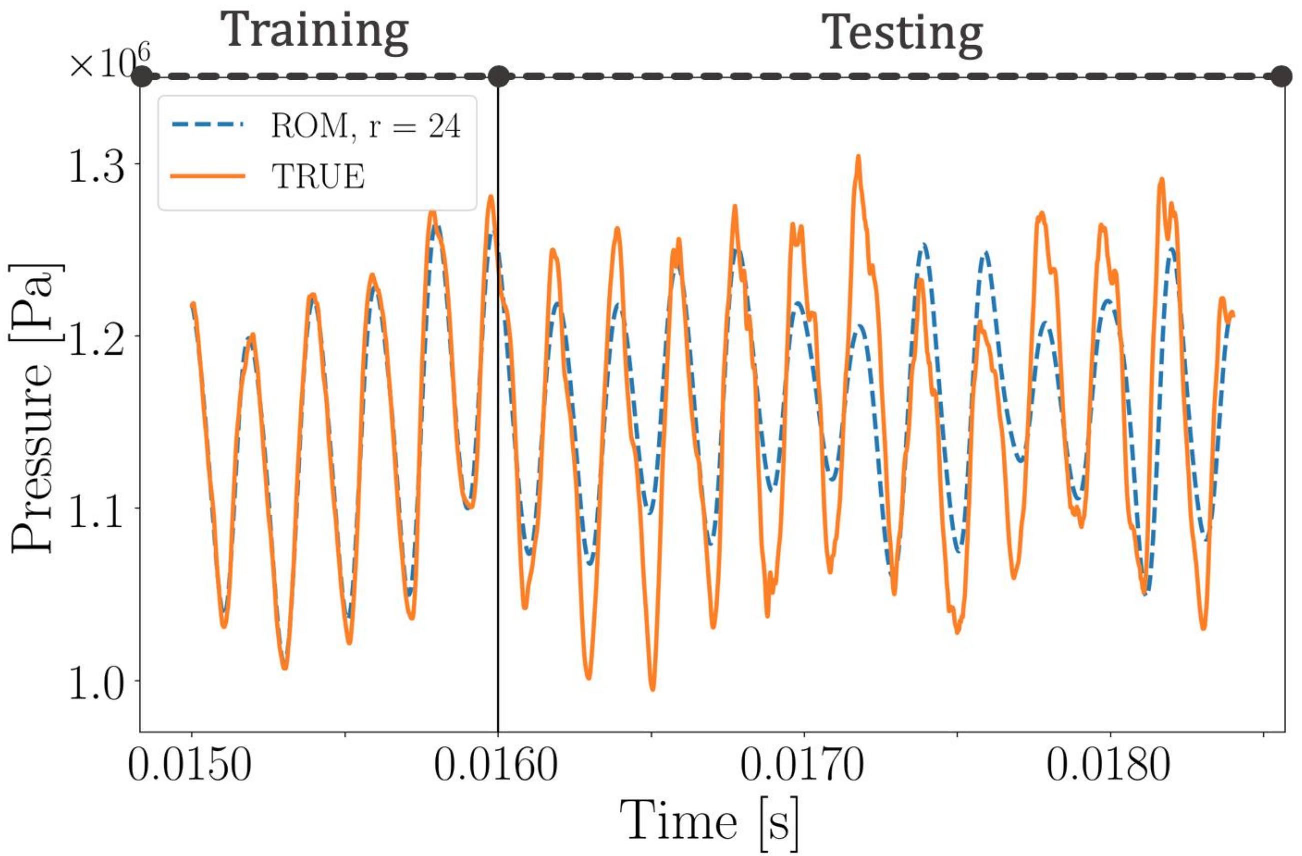
Performance of learned quadratic ROM

Pressure time traces at monitor location 1

Original CFD model 308,184 unknowns

POD basis size $r = 24$



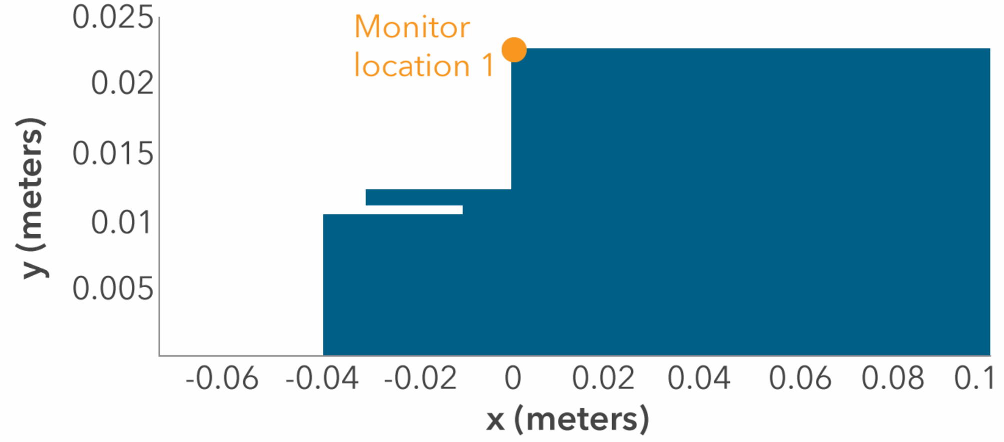


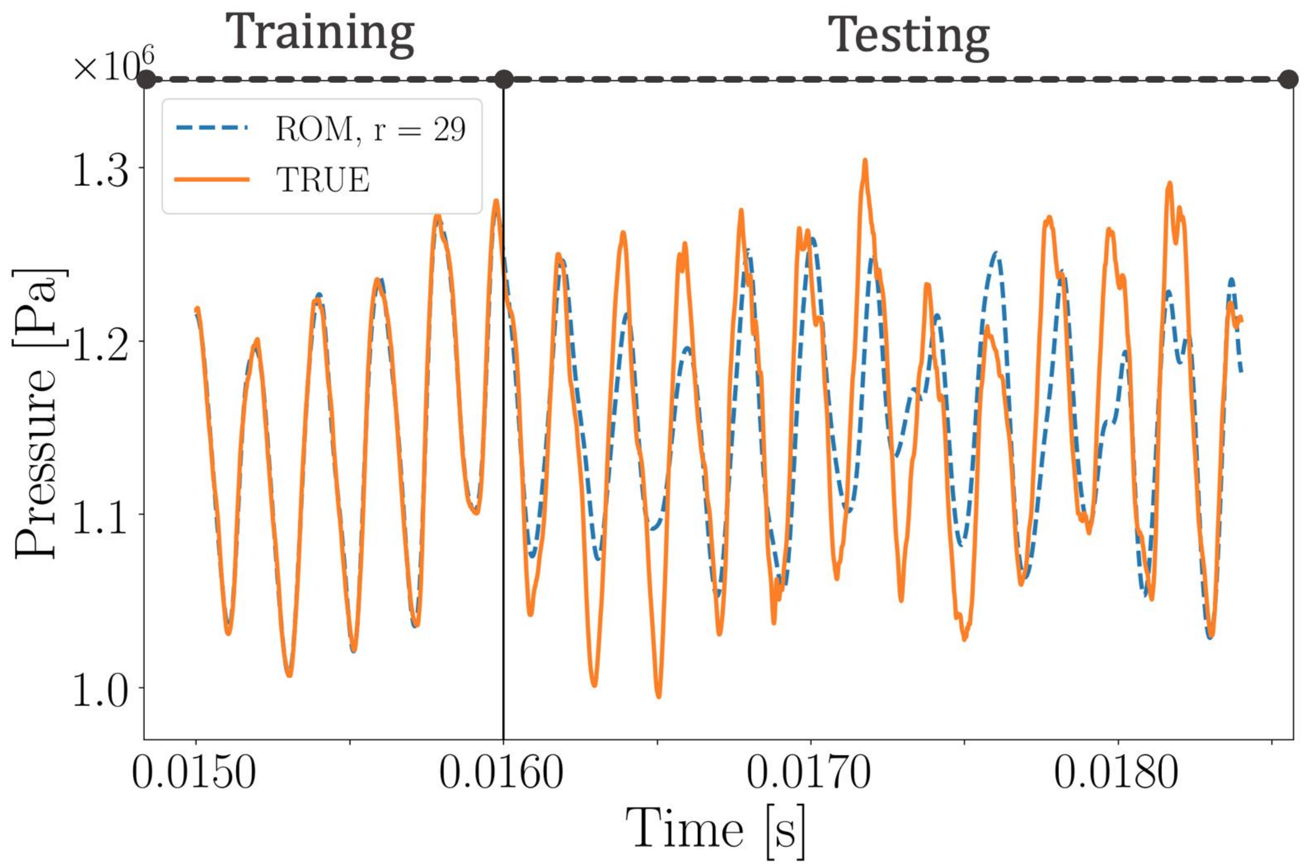
Performance of learned quadratic ROM

Pressure time traces at monitor location 1

Original CFD model 308,184 unknowns

POD basis size $r = 29$

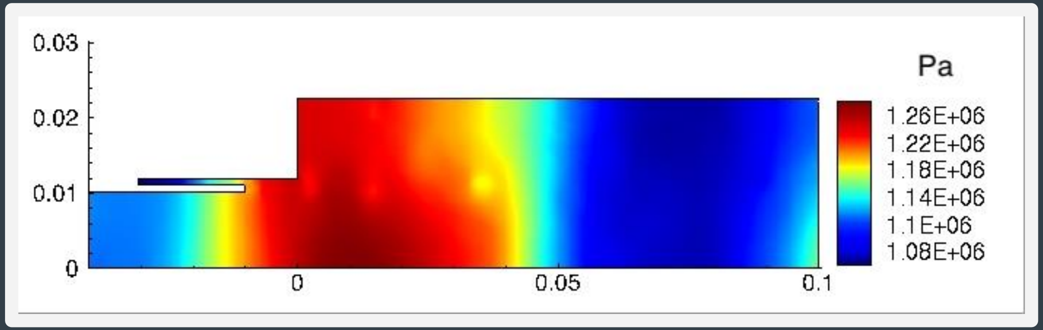


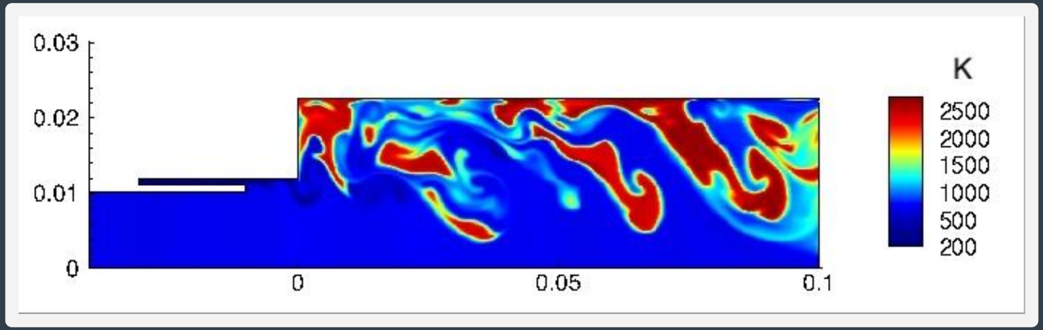


True

Pressure

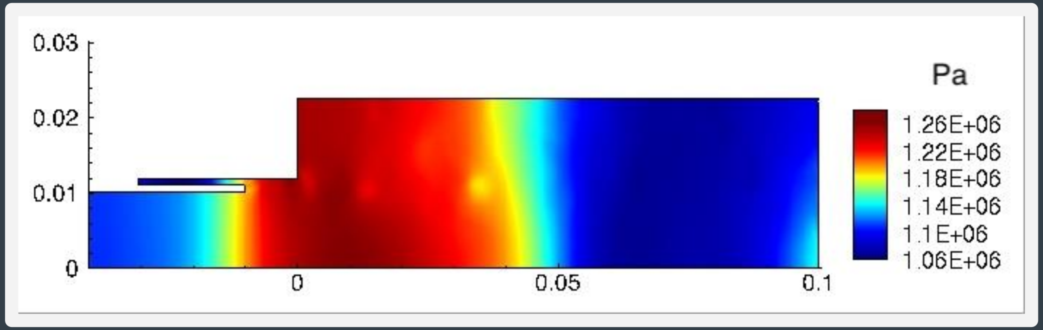
Temperature

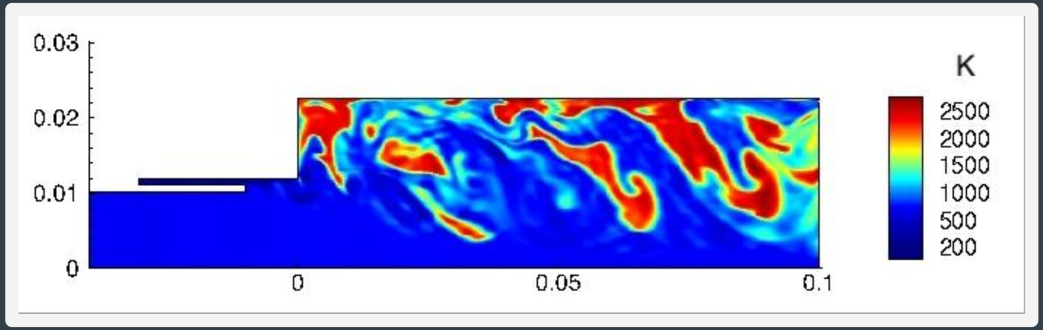




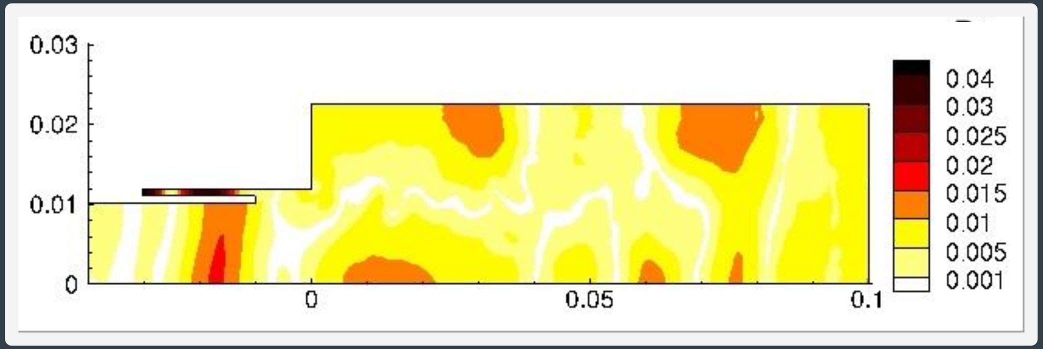
Predicted

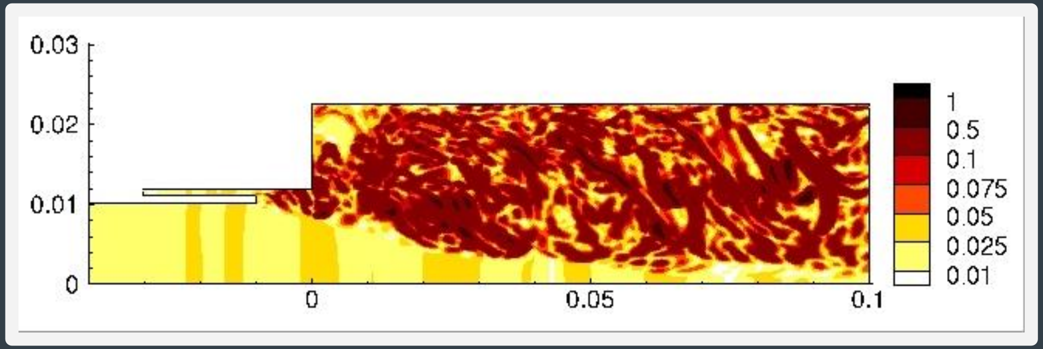
$r = 29$ POD modes





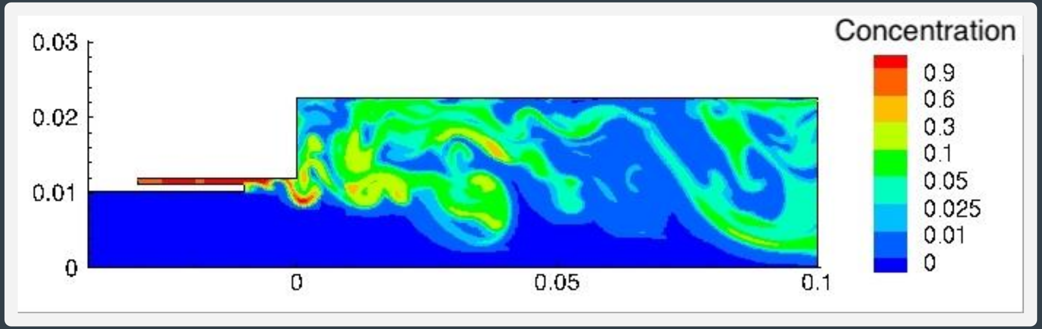
Relative error

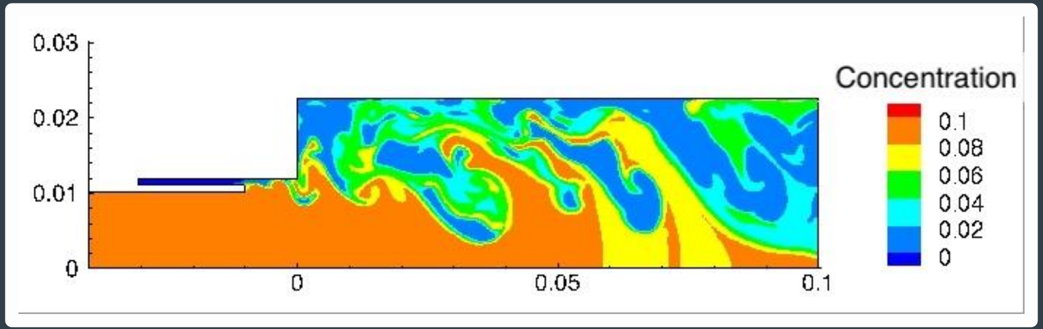




True

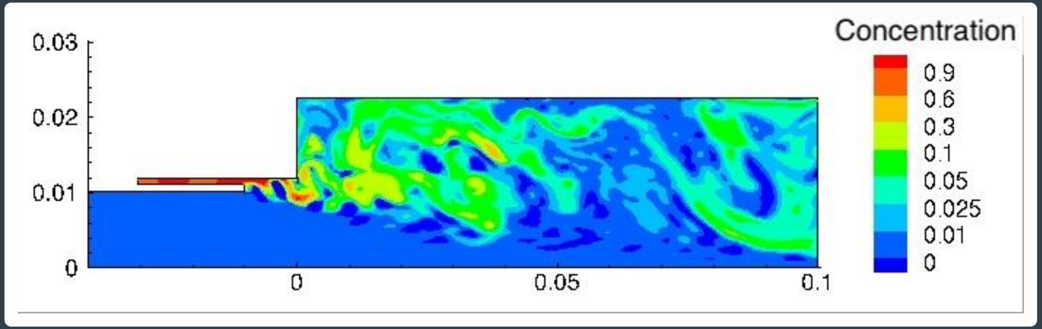
CH_4 **O_2**

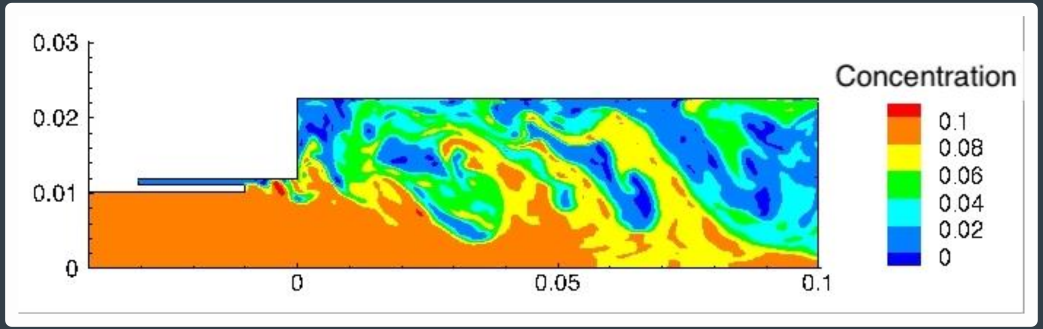




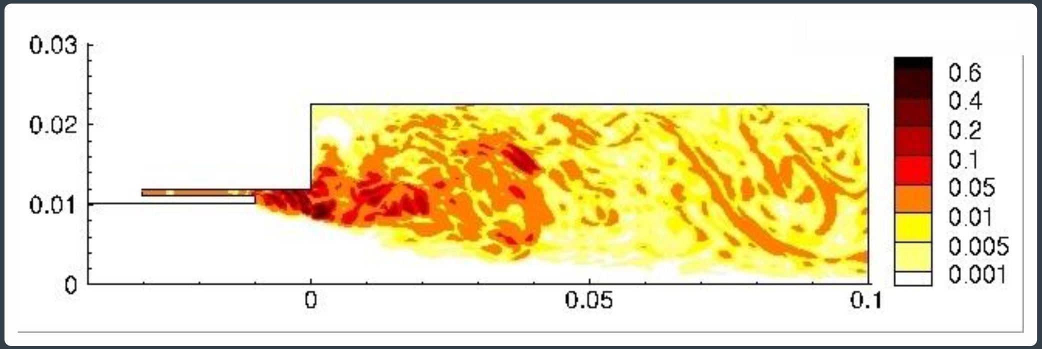
Predicted

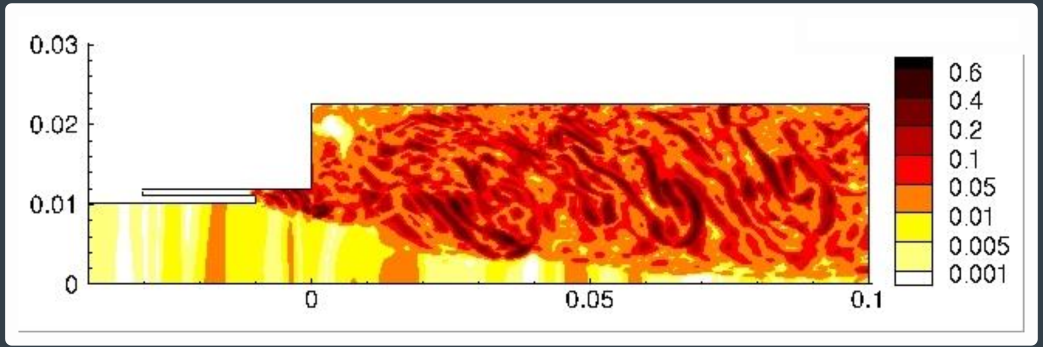
$r = 29$ POD modes





Normalized absolute error



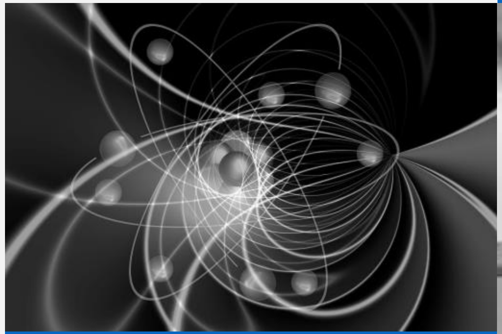


Conclusions & Outlook

Challenges & opportunities for machine learning in complex scientific & engineering applications

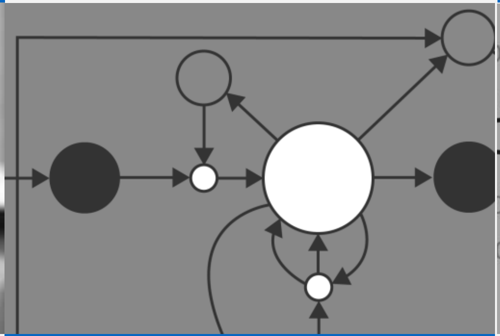
Predictive Data Science

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



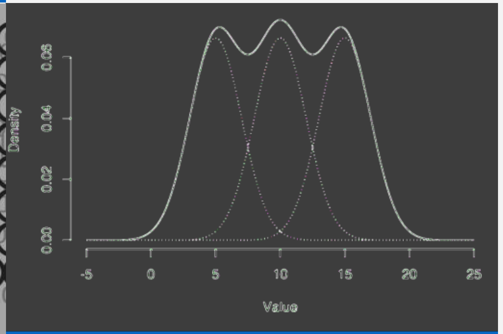
Respect physical constraints

Embed domain knowledge



Bring interpretability to results

Integrate heterogeneous, noisy & incomplete data



Get predictions with quantified uncertainties

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES
ODEN

Forward simulations
Advancing scientific discovery & engineering innovation

Optimization & inverse problems
Advancing estimation, design & control

Uncertainty quantification
Towards Predictive Science

Scientific machine learning
Towards Predictive Data Science

6 decades of Computational Science & Engineering

How do we harness the explosion of data to extract knowledge, insight and decisions?

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