The University of Texas at Austin
Oden Institute for Computational Engineering and Sciences

# Predictive data science
## for physical systems

### From model reduction to scientific machine learning

**Professor Karen E. Willcox**

New York Scientific Data Summit

June 12, 2019

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

# Contributors

**Dr. Boris Kramer**
MIT → UCSD

**Elizabeth Qian**
MIT

**Renee Swischuk**
MIT

**Prof. Benjamin Peherstorfer**
Courant Institute

Funding sources: Air Force Office of Sponsored Research MURI on Multi-Physics Multi-Source Information (F. Fahroo); Air Force Center of Excellence on Rocket Combustion (M. Birkan, F. Fahroo, D. Talley)

# Outline

1 **Predictive Data Science**

What & why

2 **A concrete example: Lift and Learn**

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

3 **Application example**

Rocket engine combustion

4 **Conclusions & Outlook**

1 Predictive Data Science

2 Concrete Example

3 Application Example

3 Conclusions & Outlook

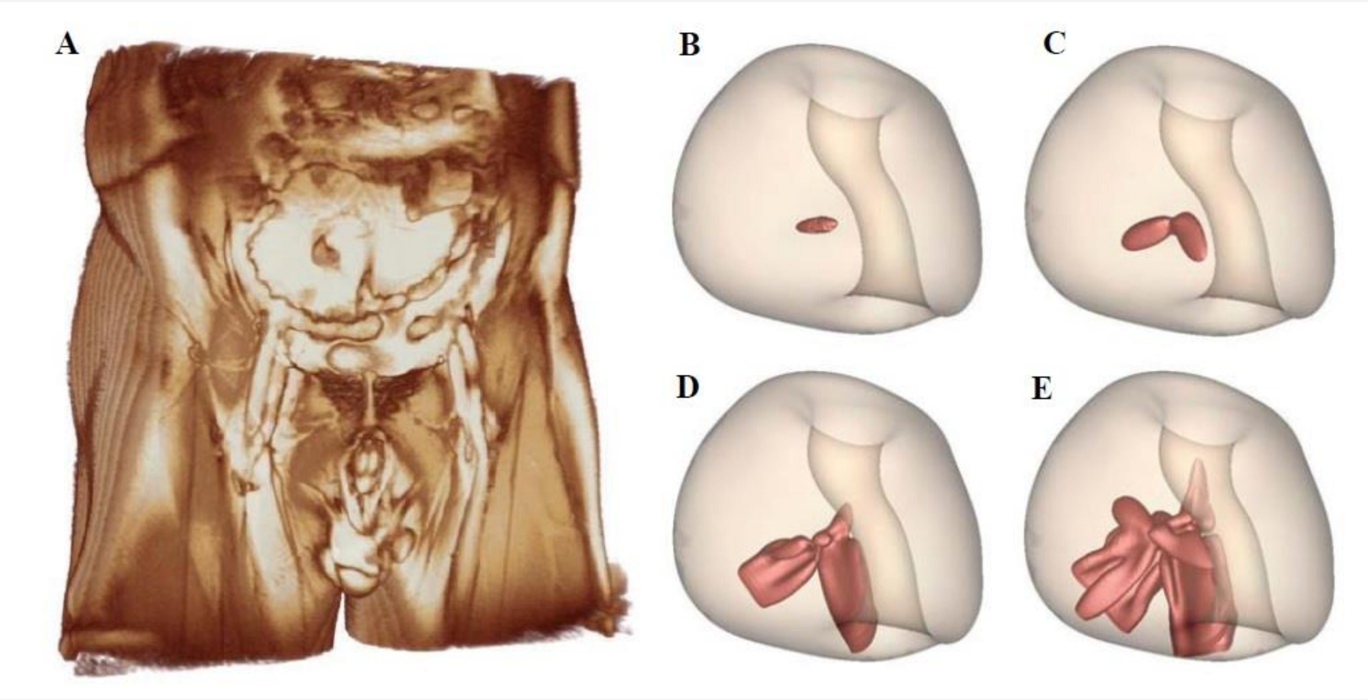
# Predictive Data Science

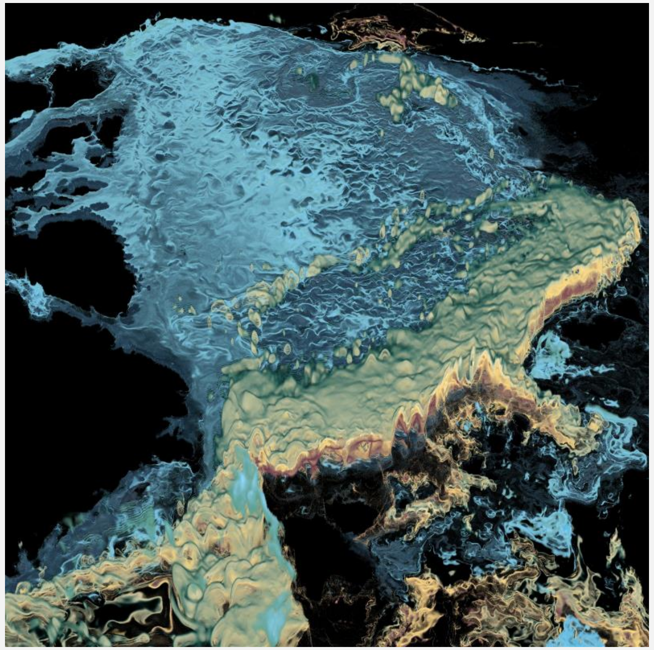
What and Why

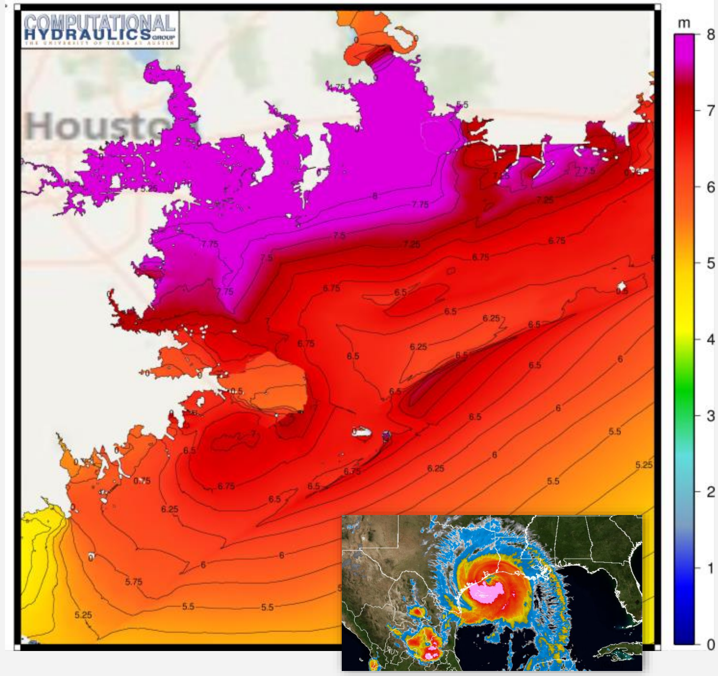
## Challenges in data science for physical systems

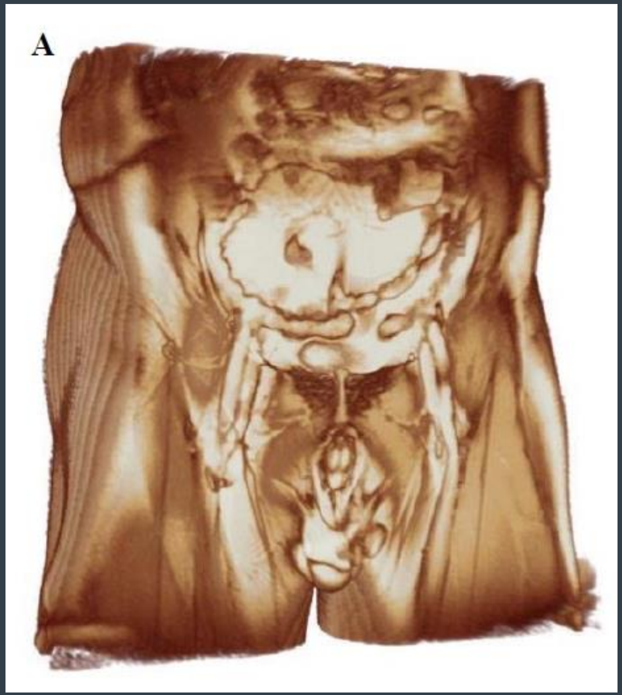
How do we harness the explosion of data to extract knowledge, insight and decisions?

When these questions relate to
**high-consequence decisions**
in engineering, science and medicine,
we need more than just the data —
we need to build in domain knowledge.





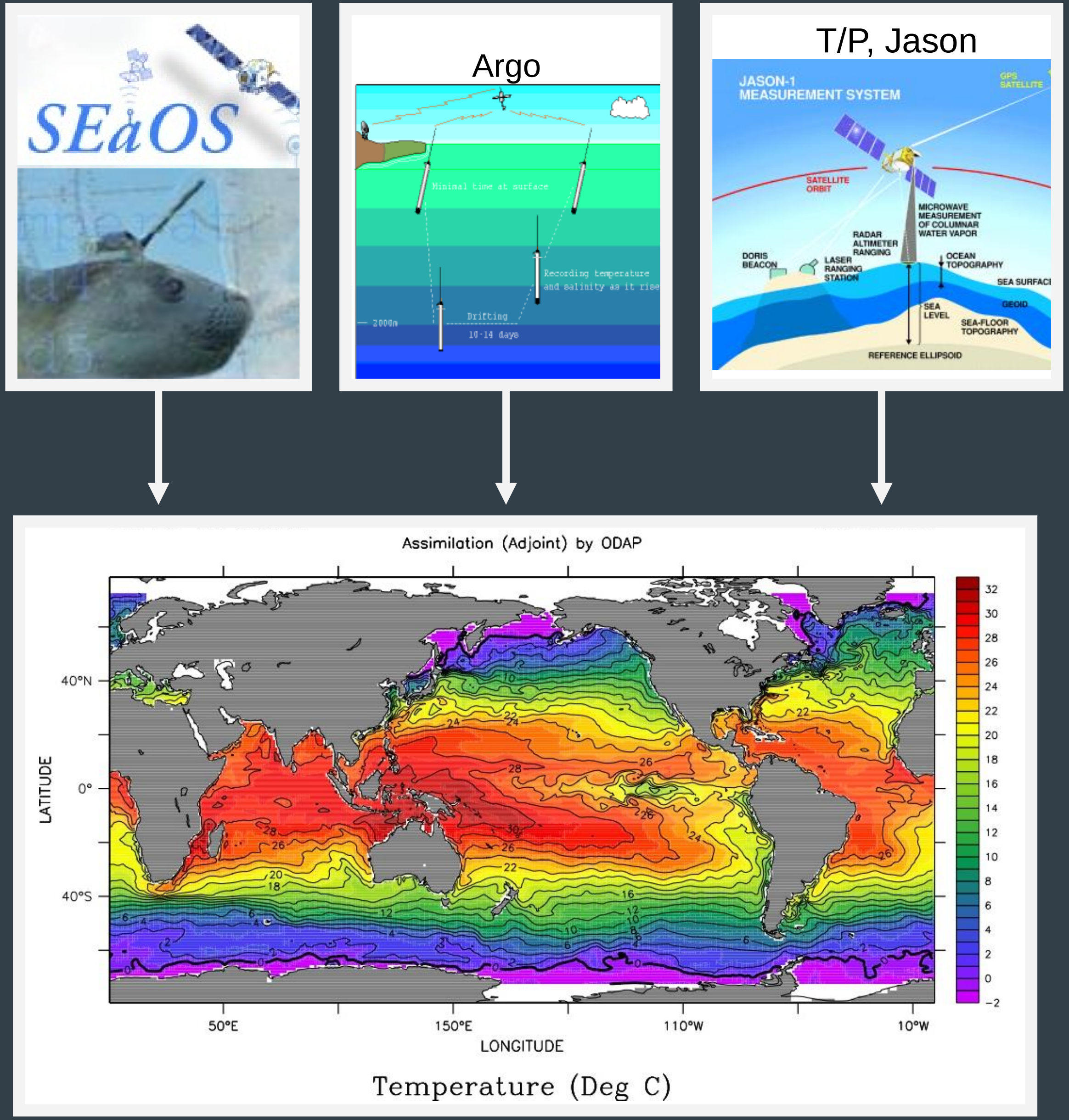



Example 1

Integrating MRI & ultrasound data with phase-field models to create the first **patient-specific prostate cancer model**

Physics-based **predictions** that capture the interplay between prostate geometry & tumor progression


**Computational Mechanics Group**
**T. Hughes**




Example 2

Integrating multiple heterogeneous data to infer ocean state & parameters characterizing **large-scale climate model**

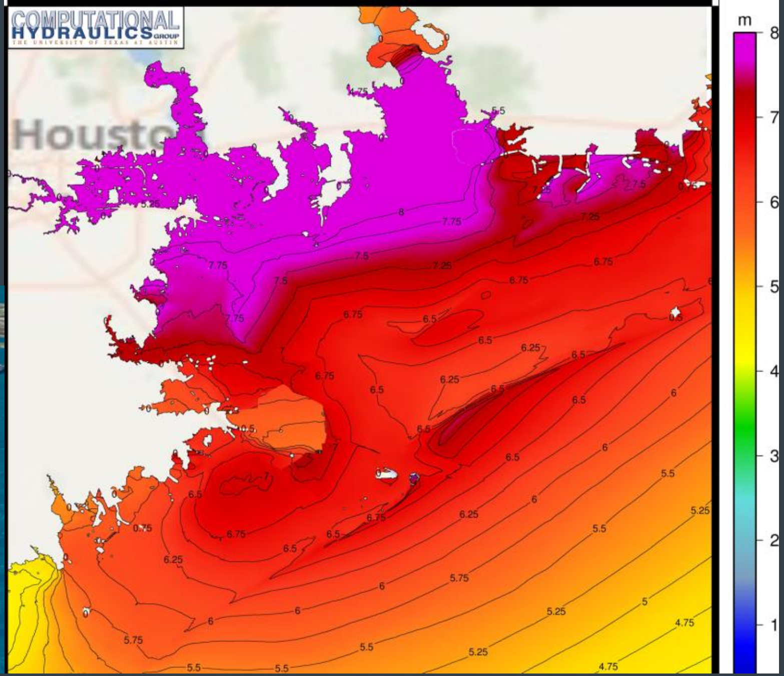
complex interactions • sparse & expensive sensors • multiphysics multiscale dynamics • data cannot by themselves reveal key climate indices needed to issue **predictions**

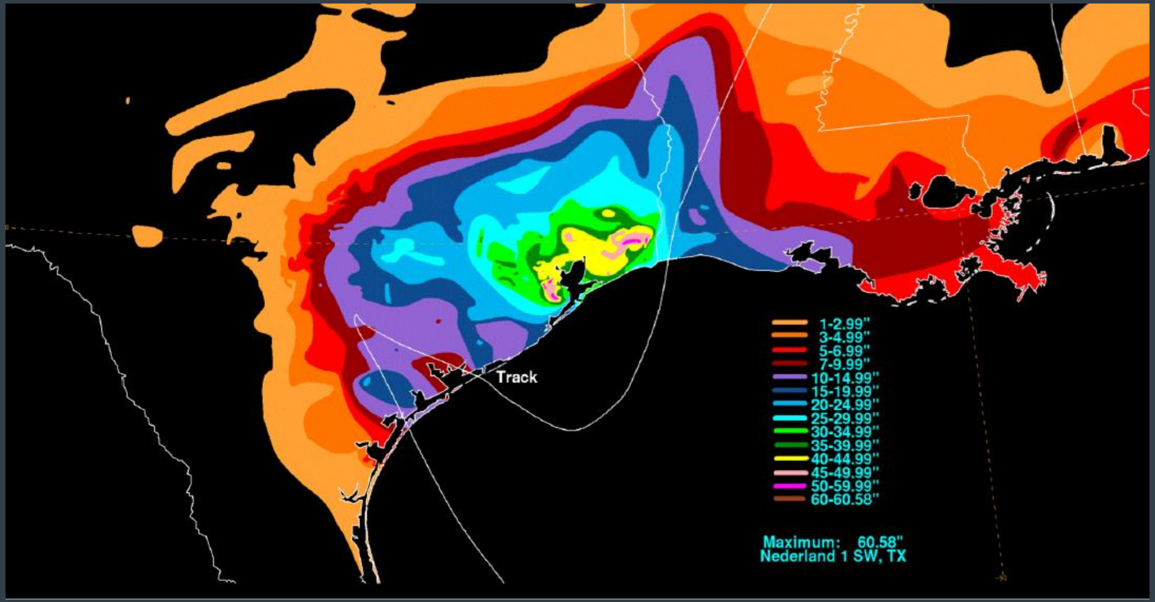
**Computational Research in Ice and Oceans Group**
**P. Heimbach**

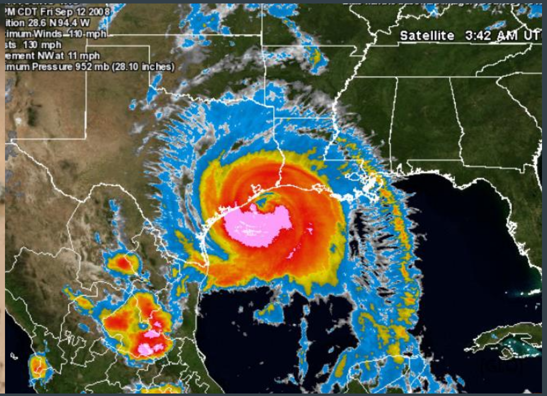
Example 3

Integrating multiple heterogeneous data with circulation & transport modeling for **disaster response & preparedness**

Billion-dollar **decisions** on where & how to build coastal protection


Computational Hydraulics Group
C. Dawson

# Predictive Data Science

a **convergence** of **Data Science** and **Computational Science & Engineering**

**Computational Science & Engineering (CSE): an interdisciplinary field that uses mathematical modeling combined with advanced computing capabilities to understand and solve complex problems

At its core CSE involves the development of models and simulations to understand natural systems

# Predictive Data Science

a **convergence** of
**Data Science** and **CSE**

## Challenges

1 **high-consequence** applications are characterized by **complex multiscale multiphysics** dynamics

2 **high** (and even infinite) **dimensional parameters**

3 **data** are relatively **sparse** and **expensive to acquire**

4 **uncertainty quantification** in model inference and **certified predictions** in regimes **beyond training data**

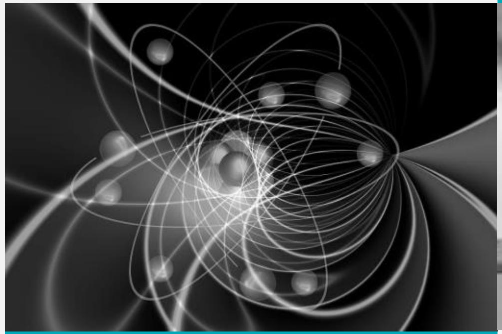
# Predictive Data Science

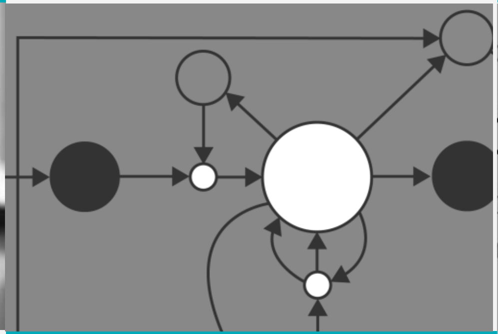
**Learning from data through the lens of models** is a way to exploit structure in an otherwise intractable problem.

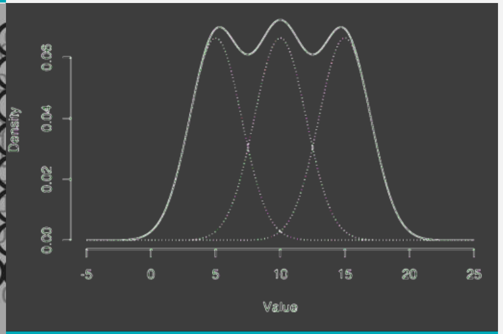
...

**Embed domain knowledge**

**Integrate heterogeneous, noisy & incomplete data**






**Respect physical constraints**

**Bring interpretability to results**

**Get predictions with quantified uncertainties**

# Learning from data through the lens of models…

dynamical systems

uncertainty quantification

meshing methods

Bayesian inference

inverse problems

finite volumes

large scale optimization

convergence analysis

randomized methods

high performance computing

regularization

finite elements

image processing

physics-based models

projection-based model reduction

boundary values

partial differential equations

data assimilation

approximation theory

big data analysis

adjoints

importance sampling

error analysis

low rank approximation

# Learning from data through the lens of models…

dynamical systems
uncertainty quantification
meshing methods
Bayesian inference
inverse problems
finite volumes
large scale optimization
convergence analysis
randomized methods
high performance computing
regularization
image processing
finite elements
physics-based models
projection-based model reduction
boundary values
partial differential equations
data assimilation
approximation theory
big data analysis
adjoints
importance sampling
error analysis
low rank approximation

1 Predictive Data Science

**2 Concrete Example**

3 Application Example

4 Conclusions & Outlook

# Lift & Learn

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

# What is a physics-based model?

**PDEs**:
1D Euler equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho w \\ E\end{pmatrix} + \frac{\partial}{\partial z}\begin{pmatrix}\rho u \\ \rho w^2 + p \\ (E+p)w\end{pmatrix} = 0 \qquad E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho w^2$$

+ boundary conditions & initial conditions

conservation of mass ($\rho$), momentum ($\rho w$), and energy ($E$) for compressible flow

**Discretize**:
Spatially discretized computational fluid dynamic (CFD) model

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

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

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

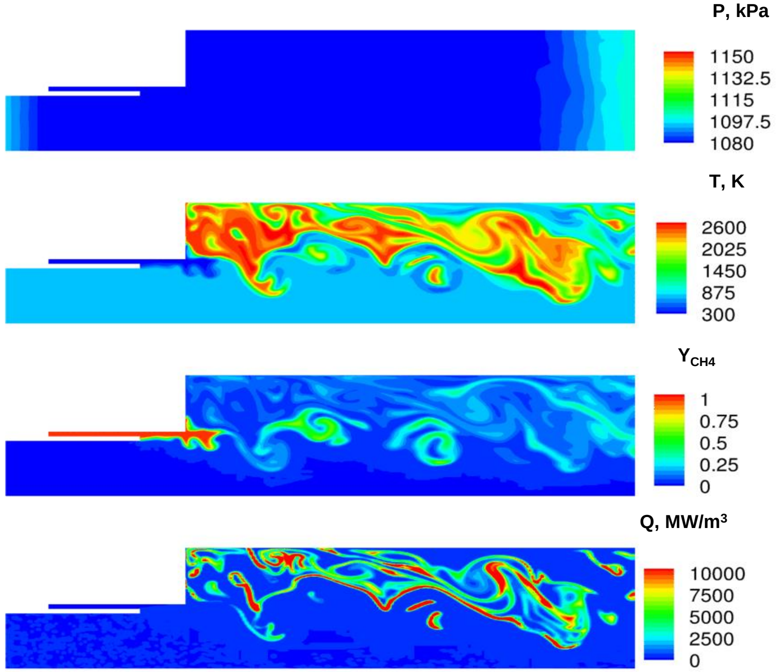
$$\mathbf{x} = \begin{bmatrix} \rho_1 \\ \rho w_1 \\ E_1 \\ \rho_2 \\ \rho w_2 \\ \vdots \\ \rho_{N_z} \\ \rho w_{N_z} \\ E_{N_z} \end{bmatrix}$$

**Solve**: given initial state $\mathbf{x}(0)$ and input $\mathbf{u}(t)$, compute state trajectory $\mathbf{x}(t)$

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

$\mathbf{x}(t_i)$: $i$th snapshot
$\mathbf{X}$: snapshot matrix

# What is a physics-based model?




Example: modeling combustion in a rocket engine

Conservation of mass ($\rho$), momentum ($\rho\vec{w}$), energy ($E$), species ($Y_{CH_4}$, $Y_{O_2}$, $Y_{CO_2}$, $Y_{H_2O}$)

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

- discretized state $\mathbf{x}(t)$ contains reacting flow unknowns
  $\rho, \rho\vec{w}, E, Y_{CH_4}, Y_{O_2}, Y_{CO_2}, Y_{H_2O}$ discretized over computational domain
- $\mathbf{u}(t)$: forcing inputs
  oscillation of inlet mass flow rate, stagnation temperature, back pressure, …
- $\mathbf{p}$: other parameters of interest:
  fuel-to-oxidizer ratio, combustion zone length, fuel temperature, oxidizer temperature, …

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

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

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

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

# 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
→ ***lifting***

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

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

conservative variables
mass, momentum, energy

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

primitive variables
mass, velocity, pressure

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

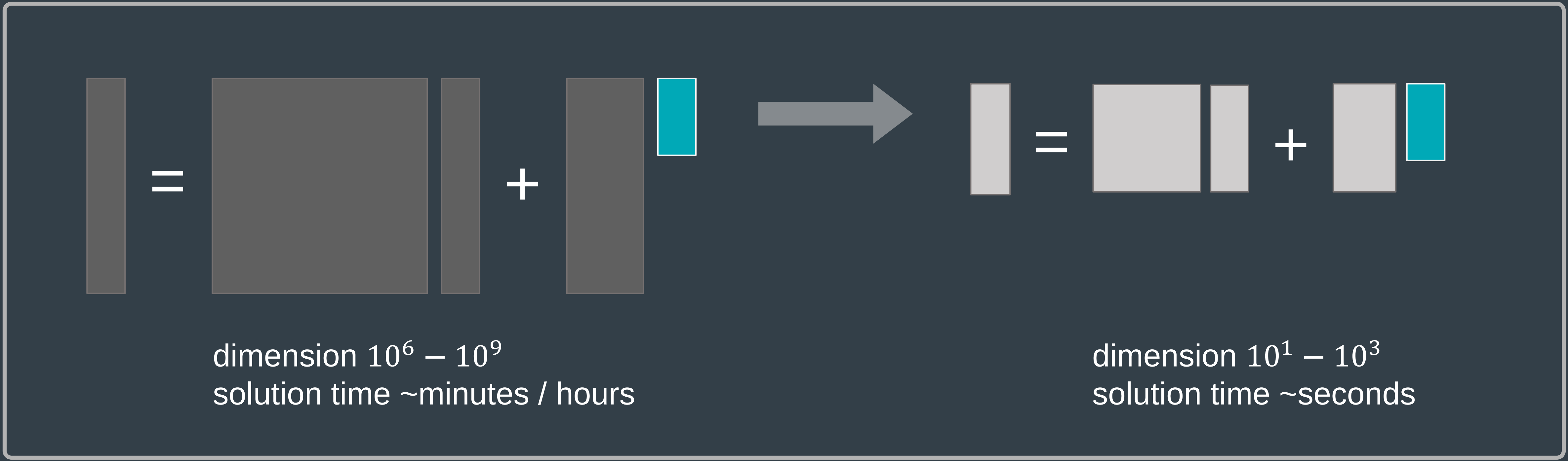
$$\frac{\partial}{\partial t}\begin{pmatrix}u\\ p\\ q\end{pmatrix} + \begin{pmatrix}u\frac{\partial u}{\partial x} + q\frac{\partial p}{\partial x}\\ \gamma p\frac{\partial u}{\partial x} + u\frac{\partial p}{\partial x}\\ q\frac{\partial u}{\partial x} + u\frac{\partial q}{\partial x}\end{pmatrix} =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

specific volume variables

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

transformed system
has linear-quadratic structure

cf. $\mathbf{E}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})$




# 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

# Reduced models

1 Train

2 Identify structure

3 Reduce

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

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

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

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

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

$$\mathbf{r} = \mathbf{E}\mathbf{V}\dot{\mathbf{x}}_r - \mathbf{A}\mathbf{V}\mathbf{x}_r - \mathbf{H}(\mathbf{V}\mathbf{x}_r \otimes \mathbf{V}\mathbf{x}_r) - \mathbf{B}\mathbf{u}$$

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

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

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

## Linear Model

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

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

Precompute the ROM matrices:

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

## Quadratic Model

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

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

Precompute the ROM matrices and tensor:

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

**projection preserves structure $\longleftrightarrow$ structure embeds physical constraints**

# (Some) Large-Scale Model Reduction Methods

Different mathematical foundations lead to different ways to compute the basis and the reduced model

Overview in Benner, Gugercin & Willcox, *SIAM Review*, 2015

- **Proper orthogonal decomposition** (POD)
  *[Lumley, 1967; Sirovich, 1981; Berkooz, 1991; Deane et al. 1991; Holmes et al. 1996]*
  - aka PCA, EOF, KLE, etc.
- **Krylov-subspace** methods
  *[Gallivan, Grimme, & van Dooren, 1994; Feldmann & Freund, 1995; Grimme, 1997, Gugercin et al., 2008]*
- **Balanced truncation**
  *[Moore, 1981; Sorensen & Antoulas, 2002; Li & White, 2002]*
- **Reduced basis** methods
  *[Noor & Peters, 1980; Patera & Rozza, 2007]*
- **Eigensystem realization algorithm** (ERA) *[Juang & Pappa, 1985]*, **Dynamic mode decomposition** (DMD) *[Schmid, 2010]*, **Loewner** model reduction *[Mayo & Antoulas, 2007]*

## Machine learning

“Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed.” [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 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. Yet recent years have seen an increased blending of the two perspectives and a recognition of the associated opportunities.
[Swischuk et al., *Computers & Fluids*, 2018]

# Lift & Learn

Variable transformations to expose structure

+ learning structured ROMs from simulation snapshot data

# Given state data, learn the system

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

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

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

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

# Given *reduced* state data, learn the *reduced* model

## Operator Inference

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

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

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

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

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

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

# Learning a low-dimensional model

Using only snapshot data from the high-fidelity model (non-intrusive) but learning the POD reduced model

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

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

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

# Learning a low-dimensional model

Using only snapshot data from the high-fidelity model (non-intrusive) but learning the POD reduced model

**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 (expose system polynomial structure)

# Learning a low-dimensional model

Using only snapshot data from the high-fidelity model (non-intrusive) but learning the POD reduced model

**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

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

# Learning a low-dimensional model

Using only snapshot data from the high-fidelity model (non-intrusive) but learning the POD reduced model

**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 high-fidelity model (non-intrusive) but learning the POD reduced model

**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

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

# Learning a low-dimensional model

Using only snapshot data from the high-fidelity model (non-intrusive) but learning the POD reduced model

**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

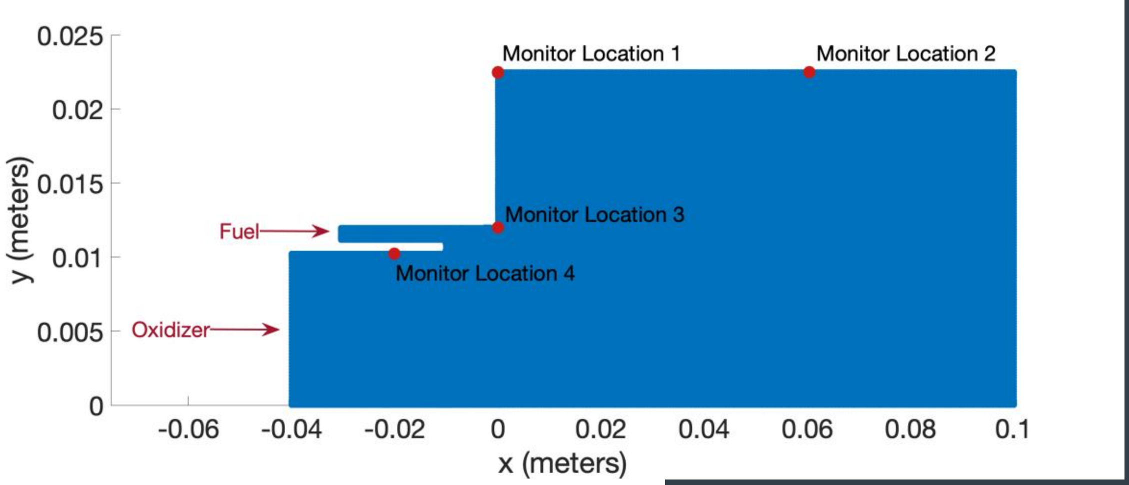
→ **convenience** of black-box learning +
**rigor** of projection-based reduction +
**structure** imposed by physics

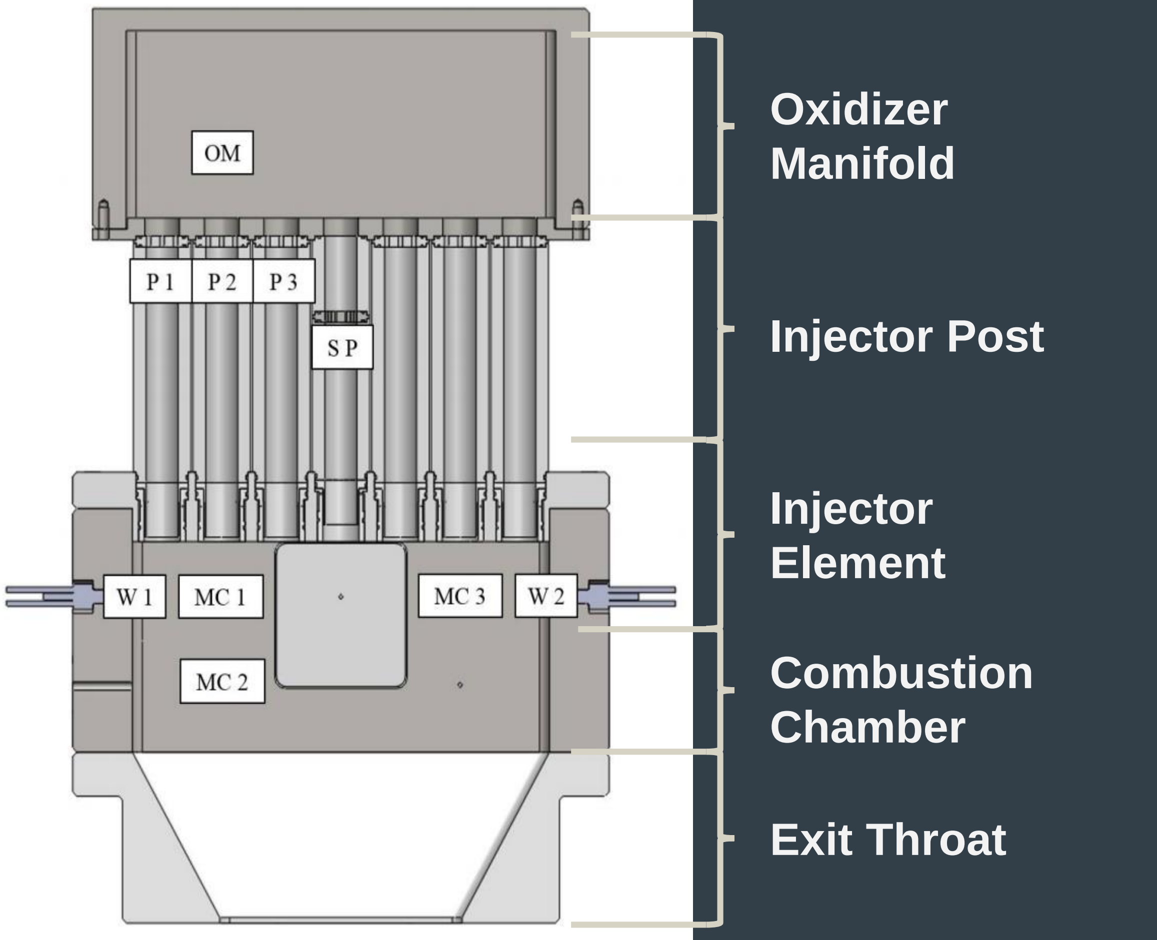
# Rocket Engine Combustion

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

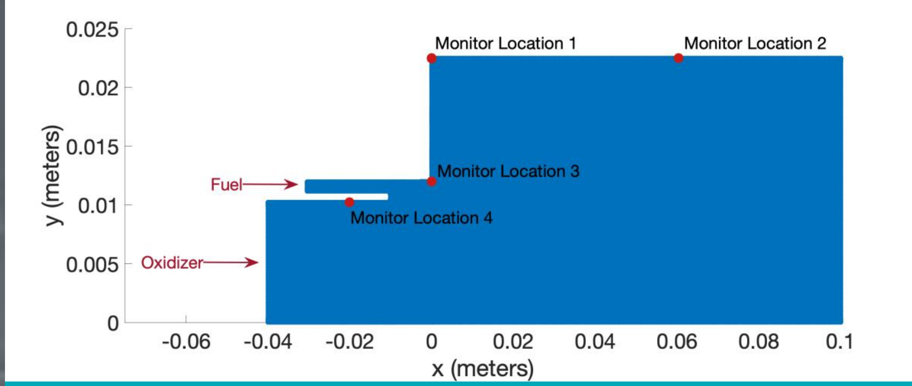
# Modeling a single injector of a rocket engine combustor

- Spatial domain discretized into 38,523 cells
- Pressure monitored at 4 locations
- Oxidizer input: 0.37 $\frac{kg}{s}$ of 42% $O_2$ / 58% $H_2O$
- Fuel input: 5.0 $\frac{kg}{s}$ of $CH_4$
- Governing equations: conservation of mass, momentum, energy, species
- Forced by a back pressure boundary condition at exit throat



# Modeling a single injector of a rocket engine combustor




**Training data**

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

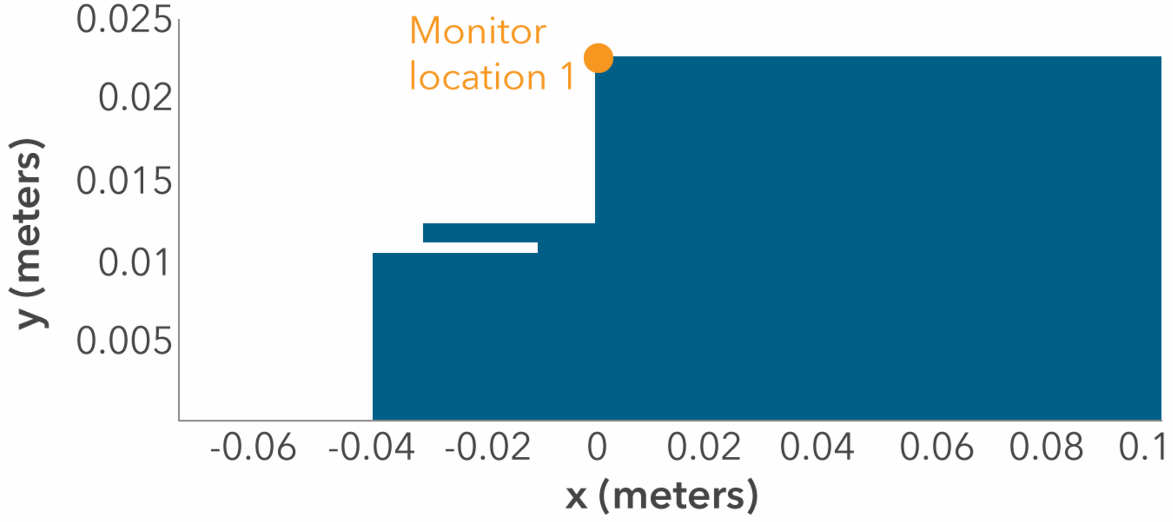
**Test data**

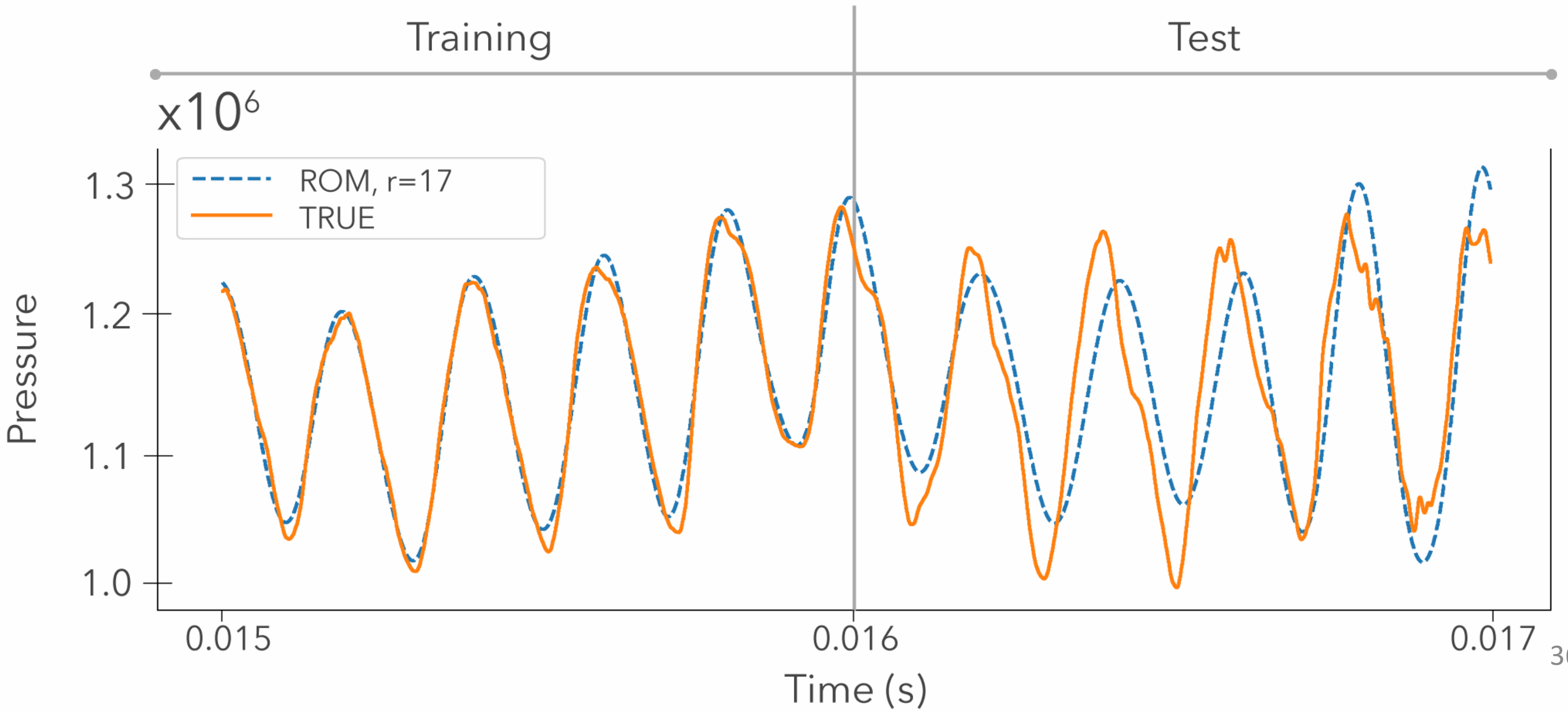
Additional 1 ms of data at monitor locations (10,000 timesteps)

# Performance of learned quadratic ROM

Pressure time traces at monitor location 1

**Basis size $r = 17$**

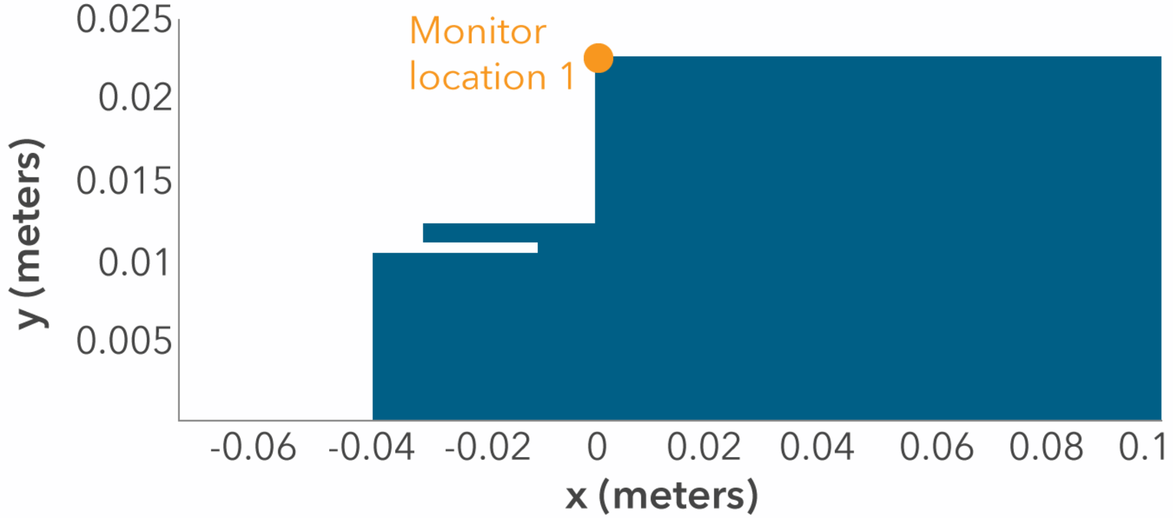


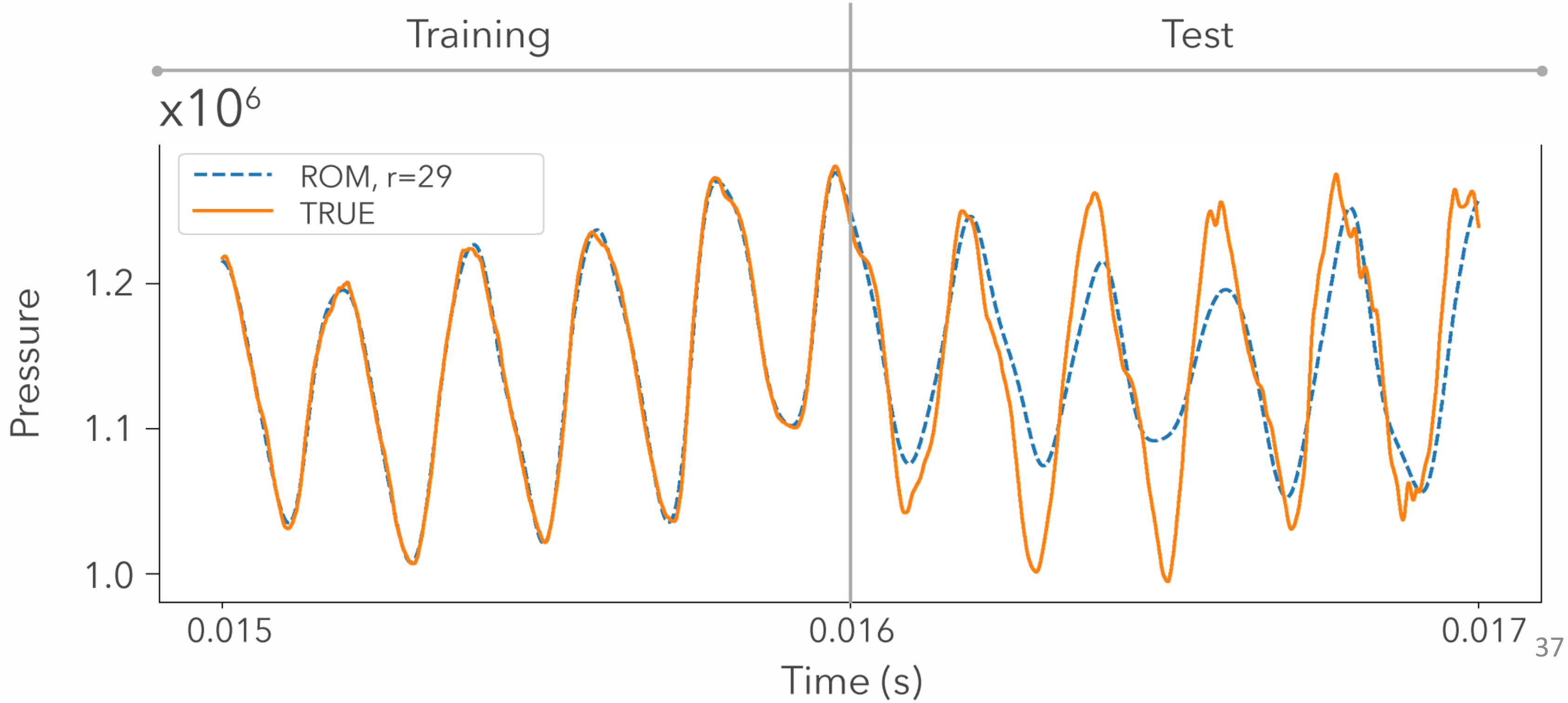



# Performance of learned quadratic ROM

Pressure time traces at monitor location 1

**Basis size** $r = 29$

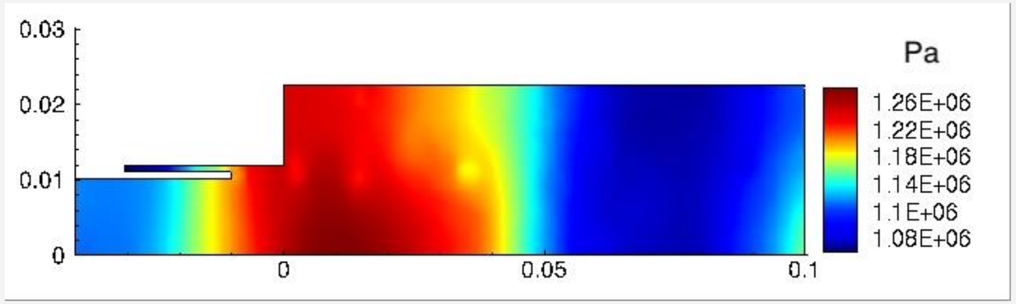


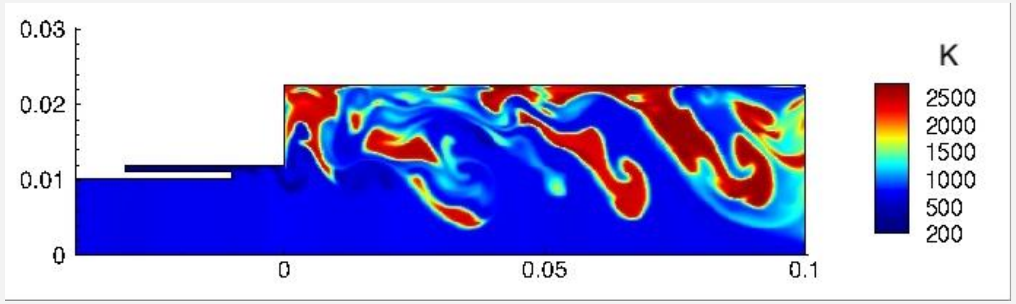



# True

Pressure

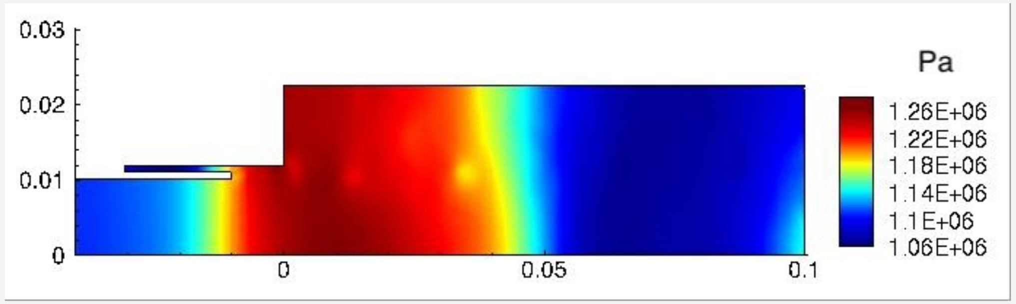
Temperature

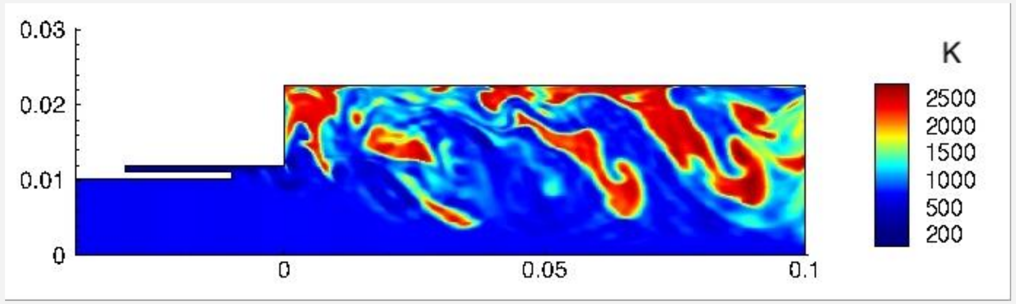


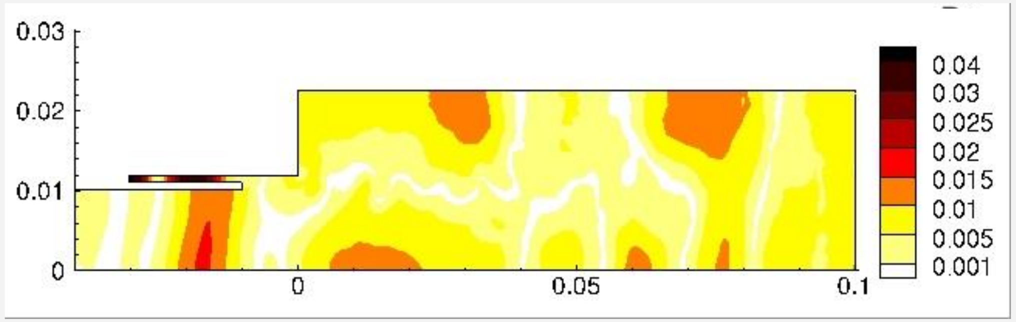

# Predicted

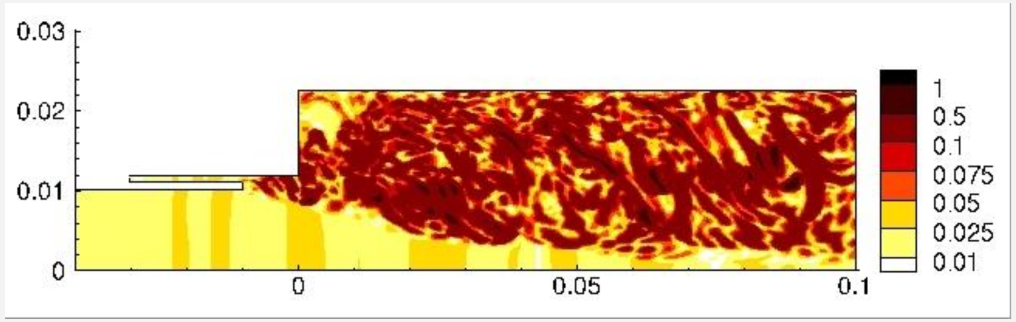
$r = 29$ POD modes



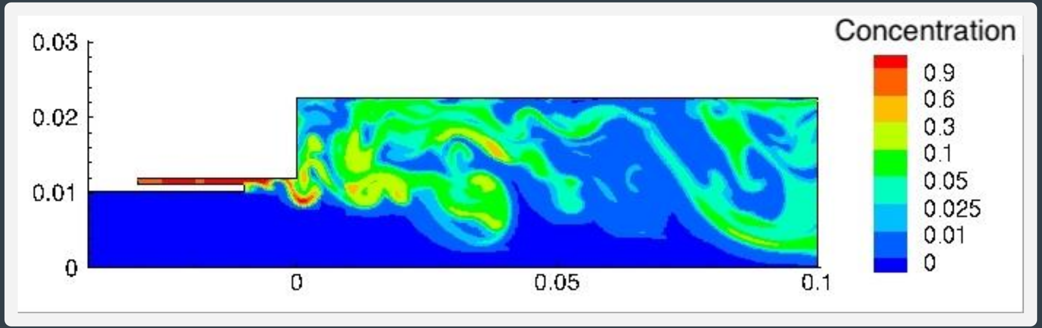

# Relative error

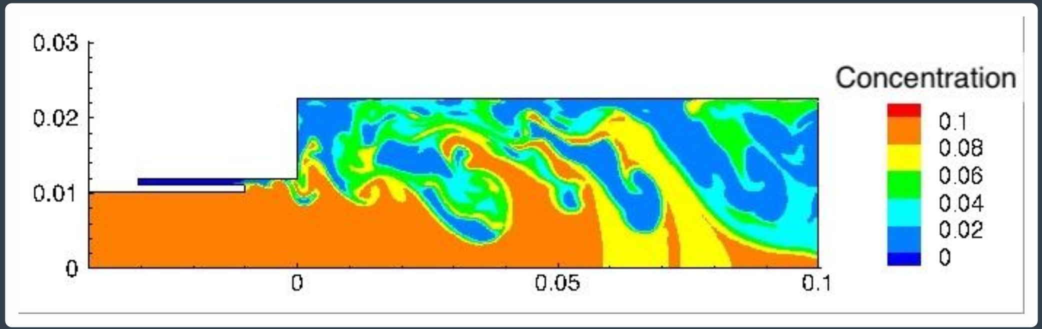


# True

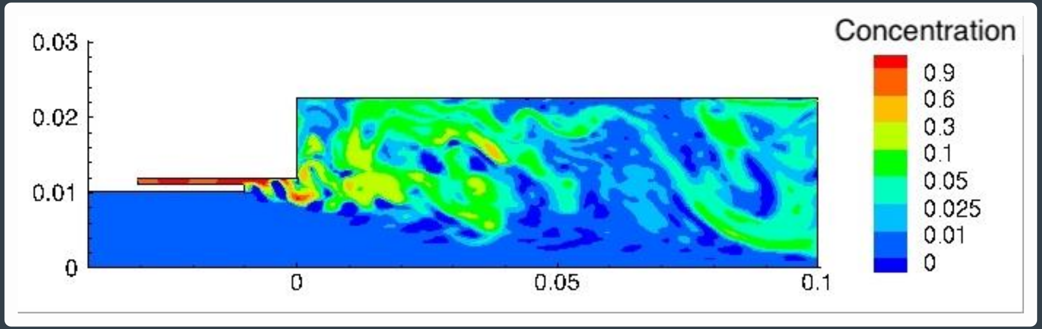
$CH_4$ $O_2$

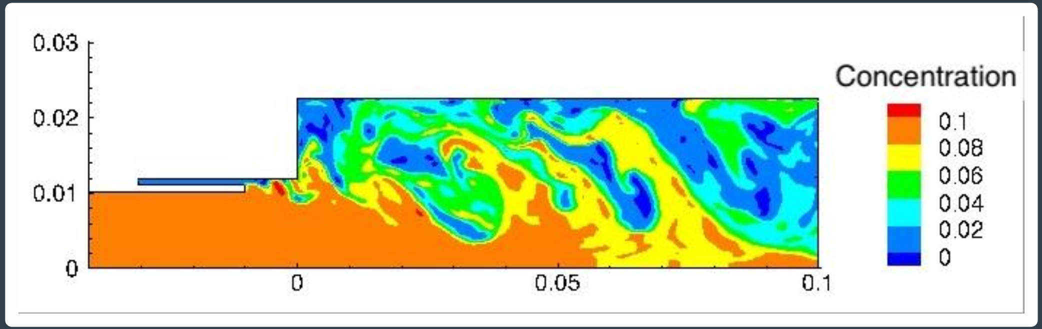


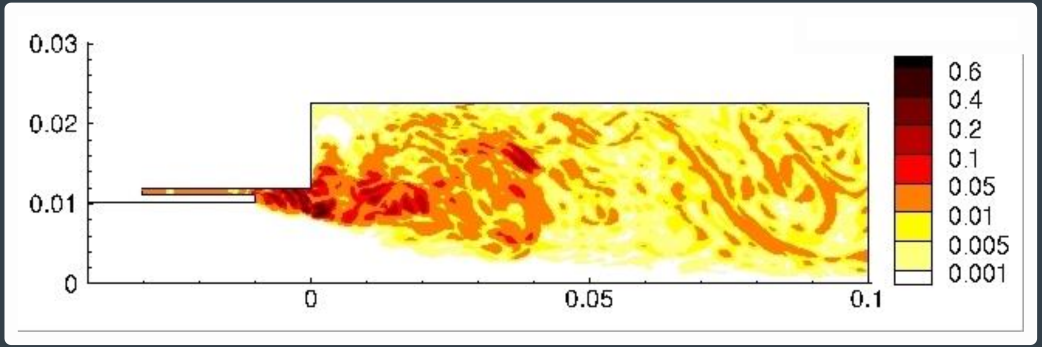

# Predicted

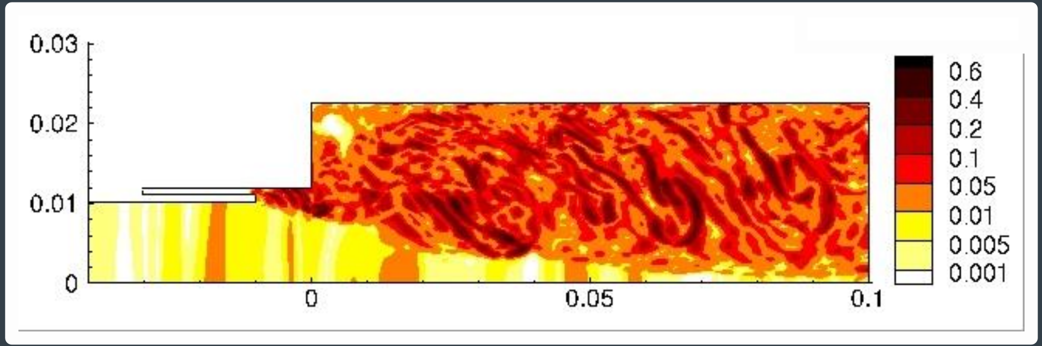
$r = 29$ POD modes




# Normalized absolute error



# Conclusions & Outlook

The future of Predictive Data Science

**Data Science**

Computational Science & Engineering

# Predictive Data Science

Revolutionizing decision-making for **high-consequence applications** in science, engineering & medicine

# Predictive Data Science

Needs **interdisciplinary research** & **education** at the interfaces

1. Embedding domain knowledge
2. Learning from data through the lens of models
3. Principled approximations that exploit low-dimensional structure
4. Explicit modeling & treatment of uncertainty

# 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