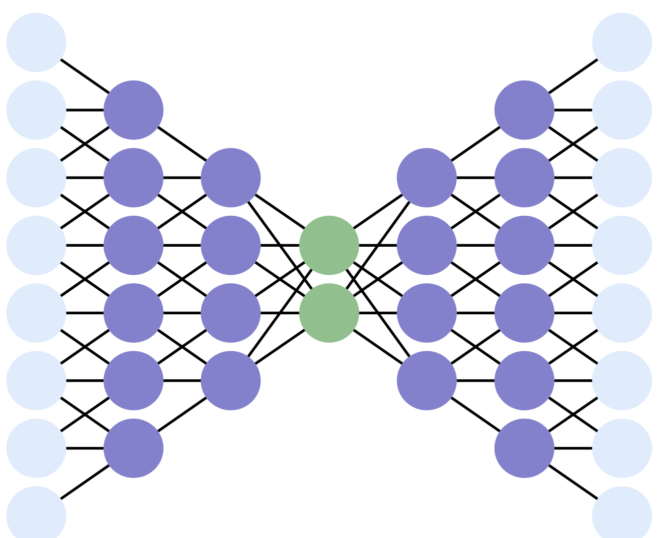
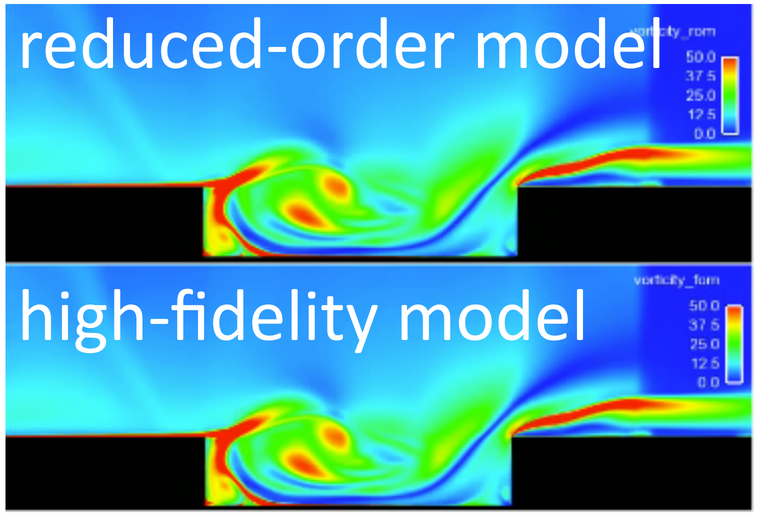
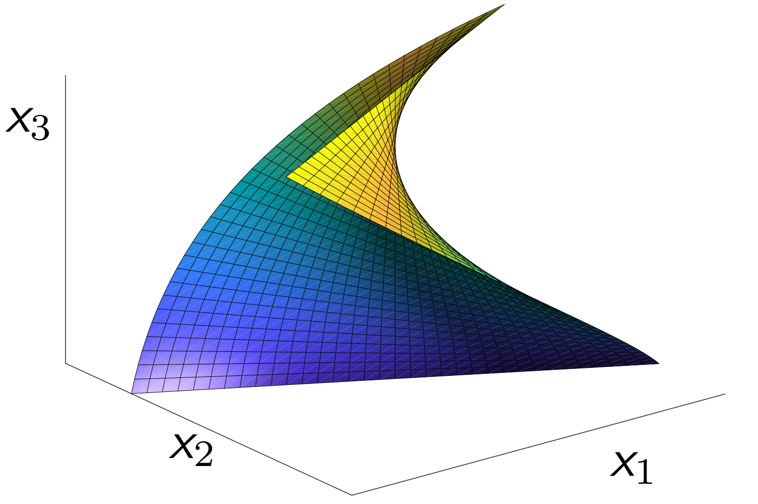
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Nonlinear model reduction

Using machine learning to enable rapid simulation of extreme-scale physics models







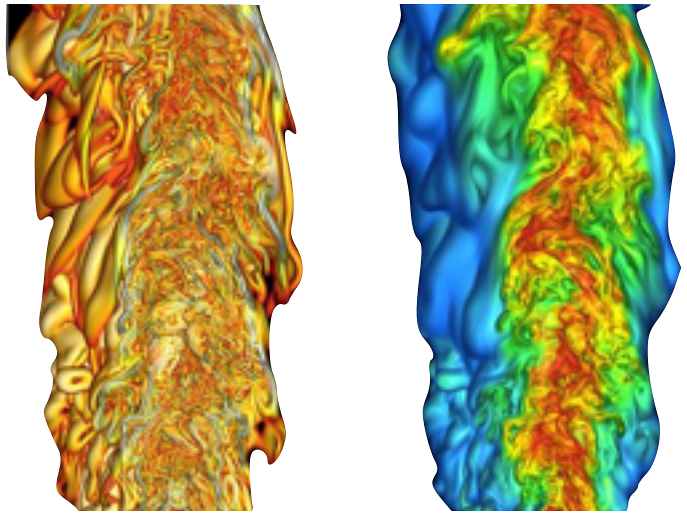
Kookjin Lee and <u>Kevin Carlberg</u>

Sandia National Laboratories

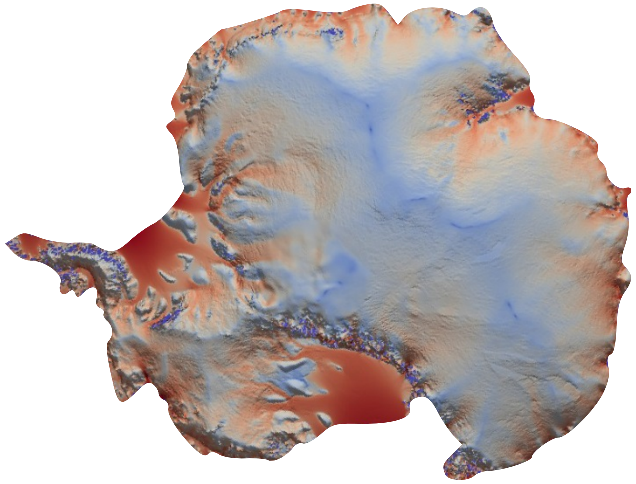
Stanford ICME Xpo
May 17, 2019

High-fidelity simulation

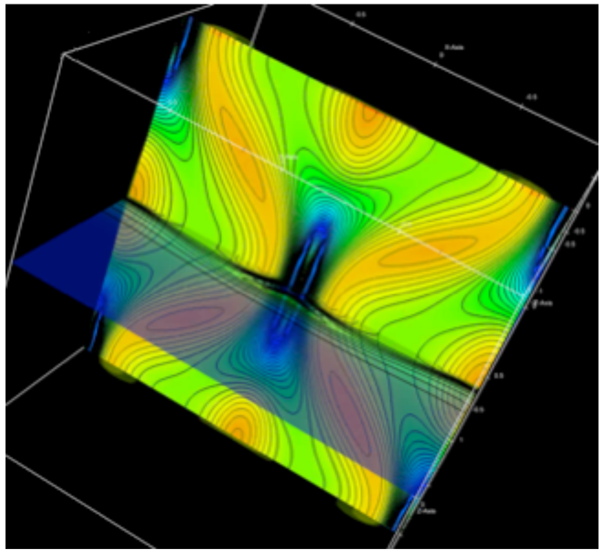
+ Indispensable across science, engineering, and entertainment
- *High fidelity*: extreme-scale computational models



Turbulent reacting flows
courtesy J. Chen, Sandia



Antarctic ice sheet modeling
courtesy R. Tuminaro, Sandia



Magnetohydrodynamics
courtesy J. Shadid, Sandia

computational barrier

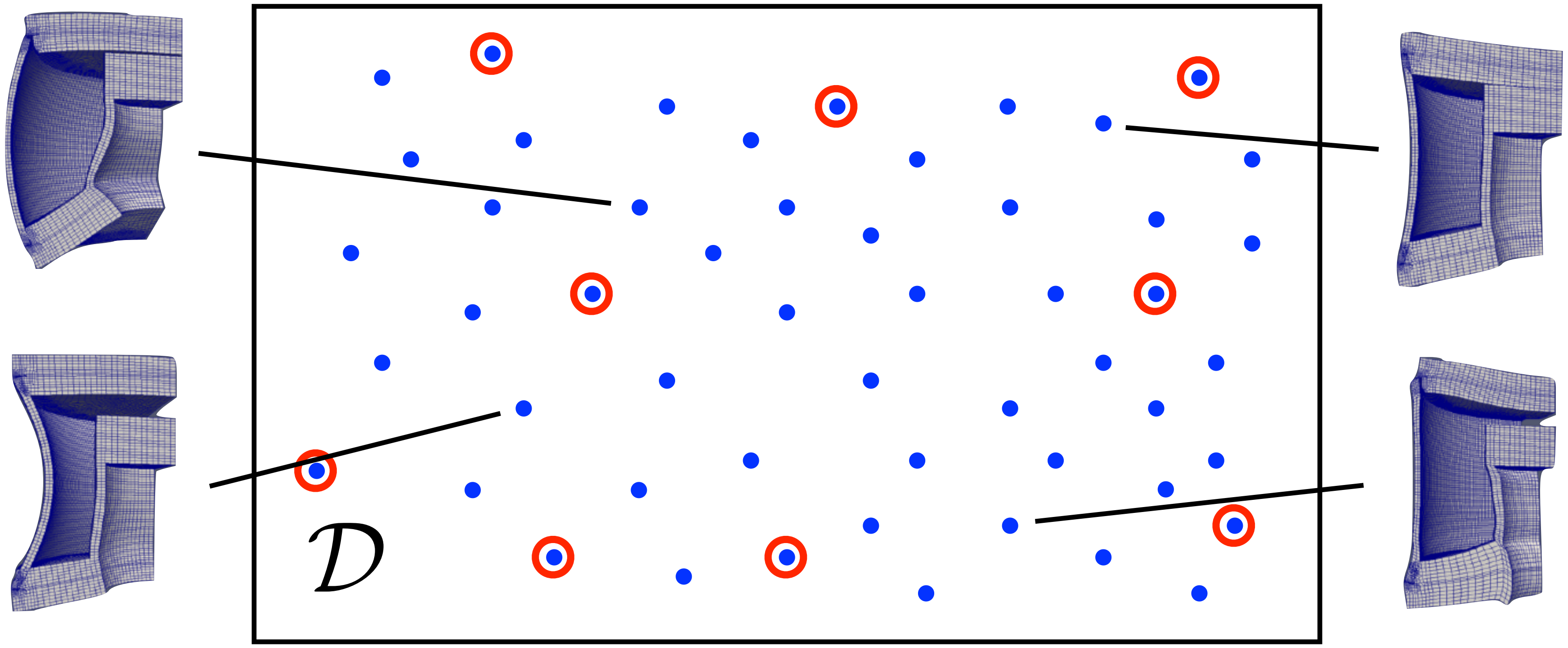
Time-critical problems

- model predictive control
- health monitoring
- interactive virtual environment
- design optimization

Approach: exploit simulation data

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu}), \quad \mathbf{x}(0, \boldsymbol{\mu}) = \mathbf{x}_0(\boldsymbol{\mu}), \quad t \in [0, T_{\text{final}}], \quad \boldsymbol{\mu} \in \mathcal{D}$$

***Time-critical problem**: rapidly solve ODE for* $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}}$



***Idea**: exploit simulation data collected at a few points*

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify structure in data
3. *Reduction:* Reduce cost of ODE solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Model reduction criteria

1. ***Accuracy:*** achieves <1% error

2. ***Low cost:*** achieves >100x computational savings

Model reduction criteria

1. ***Accuracy:*** achieves <1% error
   - autoencoders for accurate nonlinear manifolds [Lee, C., 2018]
   - optimal projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
2. ***Low cost:*** achieves >100x computational savings
   - sample mesh [C., Farhat, Cortial, Amsallem, 2013]
   - space–time projection [Choi, C., 2019]
3. ***Structure preservation:*** preserves important physical properties
   - enforce conservation laws [C., Choi, Sargsyan, 2018]
   - preserve Lagrangian structure and stability [C. Boggs, Tuminaro, 2015; Peng, C. 2017]
4. ***Generalization:*** always works, even in difficult cases
   - *h*-adaptivity [C., 2015]
   - vector-space sieving [Etter, C., 2019]
5. ***Certification:*** accurately quantifies the reduction error
   - machine-learning error models [Drohmann, C., 2015; Trehan, C., Durlofsky, 2017; Freno, C., 2019]
   - machine-learning closure models [Pagani, Manzoni,, C., 2019]

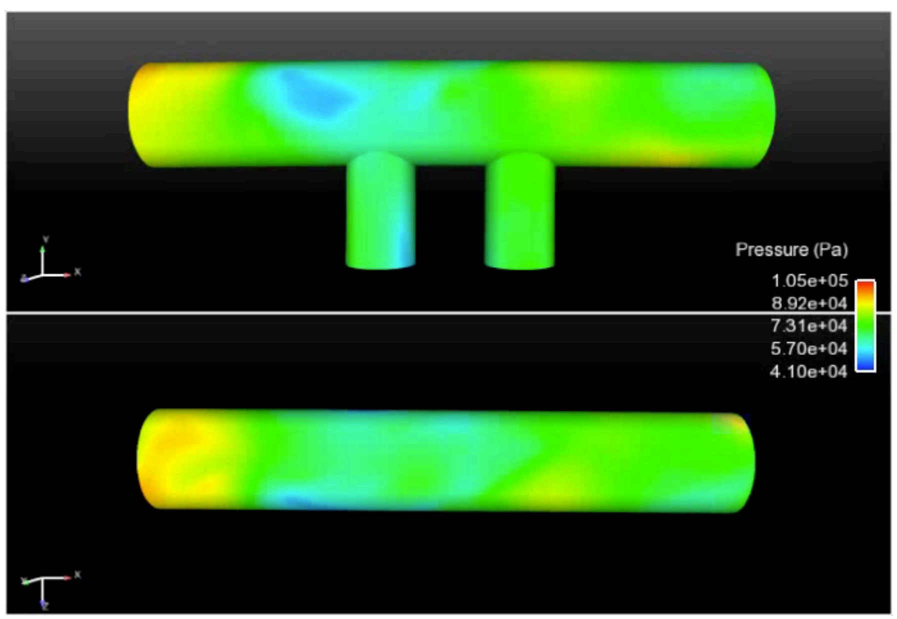
Model reduction criteria

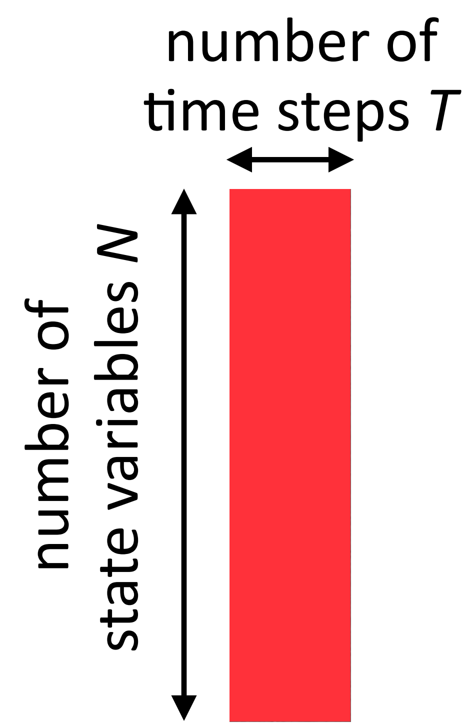
1. ***Accuracy:*** achieves <1% error
   - autoencoders for accurate nonlinear manifolds [Lee, C., 2018]
   - optimal projection [C., Bou-Mosleh, Farhat, 2011; C., Barone, Antil, 2017]
2. ***Low cost:*** achieves >100x computational savings
   - sample mesh [C., Farhat, Cortial, Amsallem, 2013]
   - space–time projection [Choi, C., 2019]
3. ***Structure preservation:*** preserves important physical properties
   - enforce conservation laws [C., Choi, Sargsyan, 2018]
   - preserve Lagrangian structure and stability [C., Boggs, Tuminaro, 2015; Peng, C. 2017]
4. ***Generalization:*** always works, even in difficult cases
   - *h*-adaptivity [C., 2015]
   - vector-space sieving [Etter, C., 2019]
5. ***Certification:*** accurately quantifies the reduction error
   - machine-learning error models [Drohmann, C., 2015; Trehan, C., Durlofsky, 2017; Freno, C., 2019]
   - machine-learning closure models [Pagani, Manzoni, C., 2019]

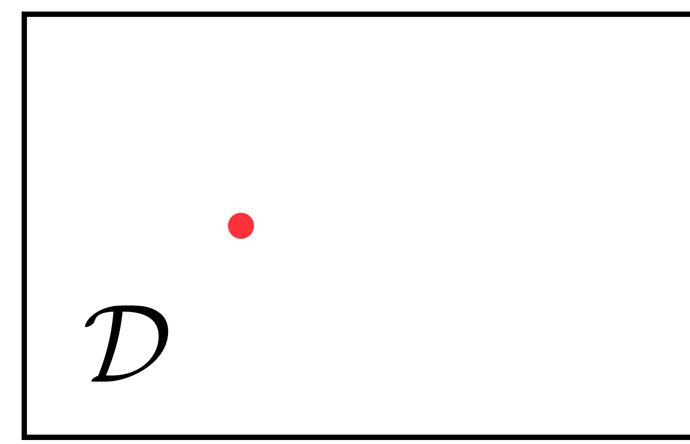
Training

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



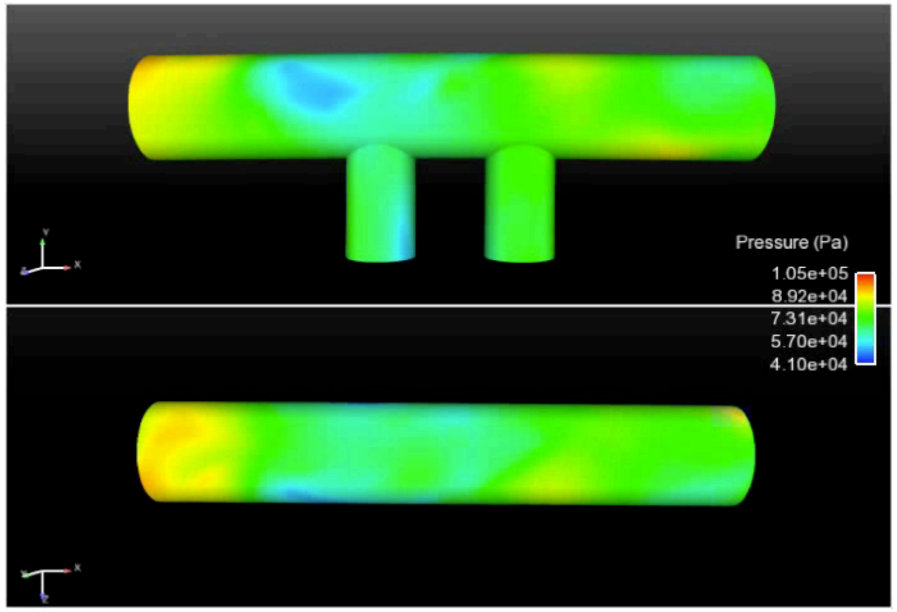




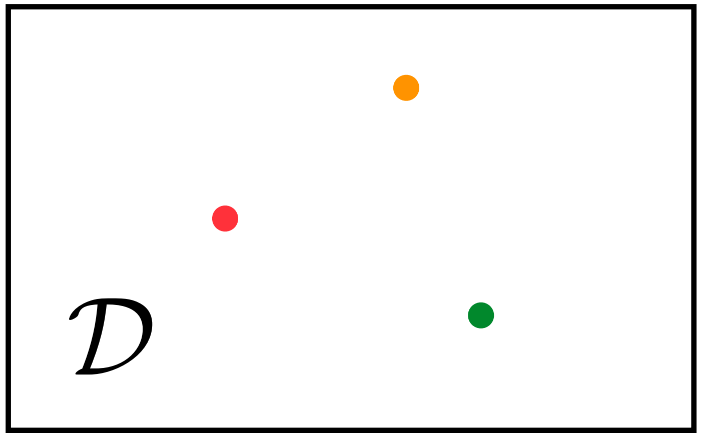
Training

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



$\mathbf{X} =$ 

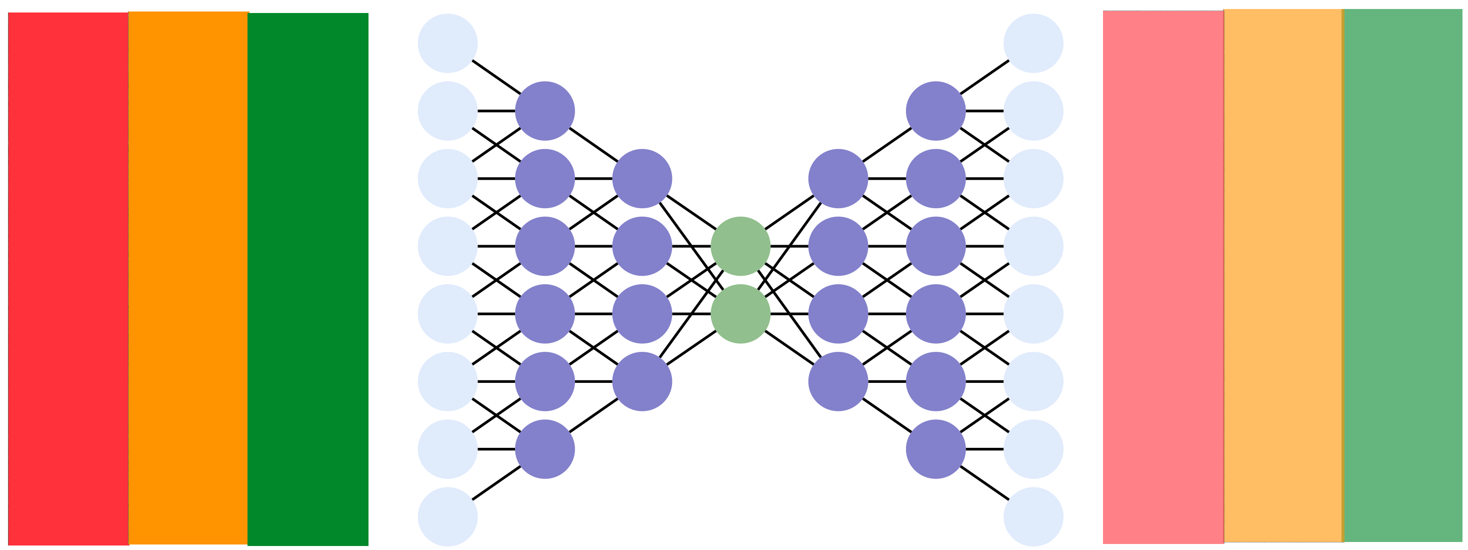


Machine learning

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

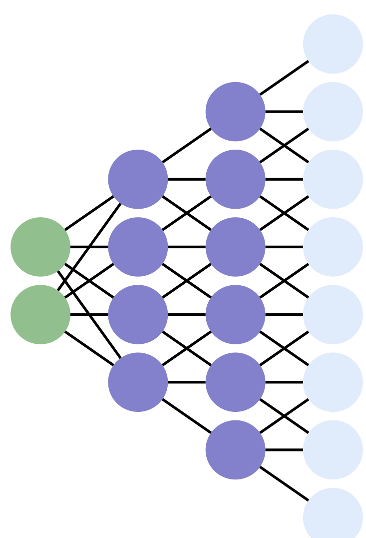
1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

$\mathbf{X} =$



$= \tilde{\mathbf{X}}(\boldsymbol{\theta})$

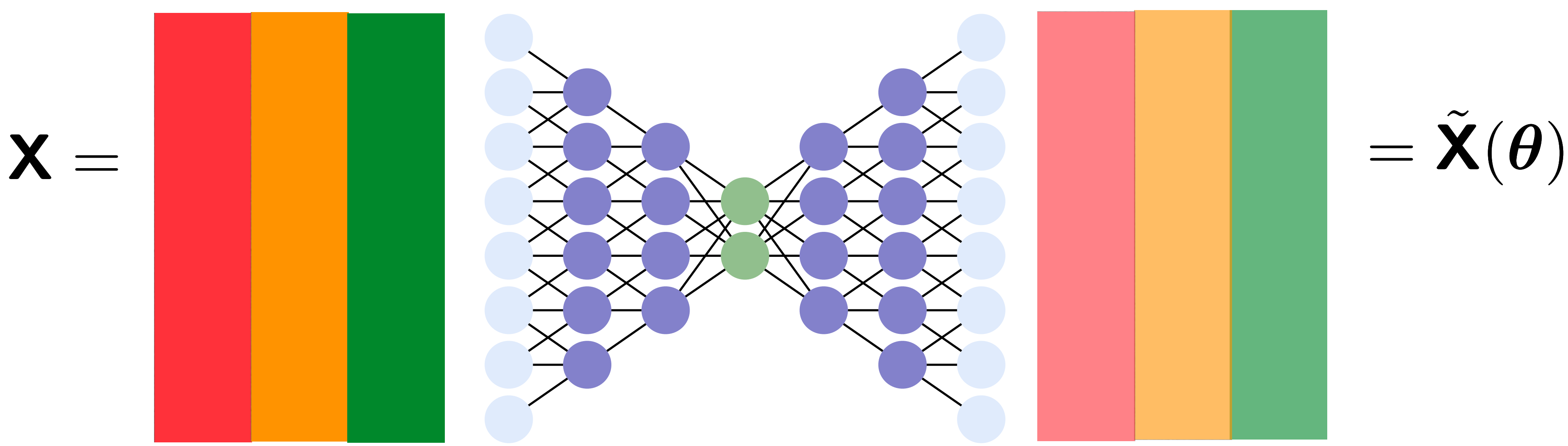
- Define low-dim manifold from decoder:



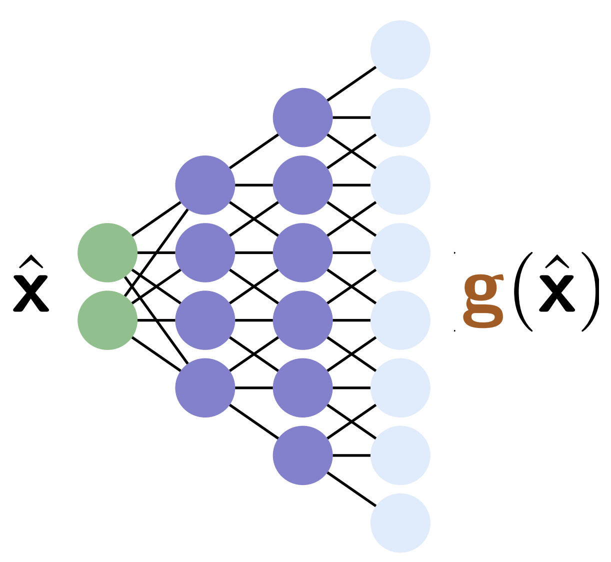
Machine learning

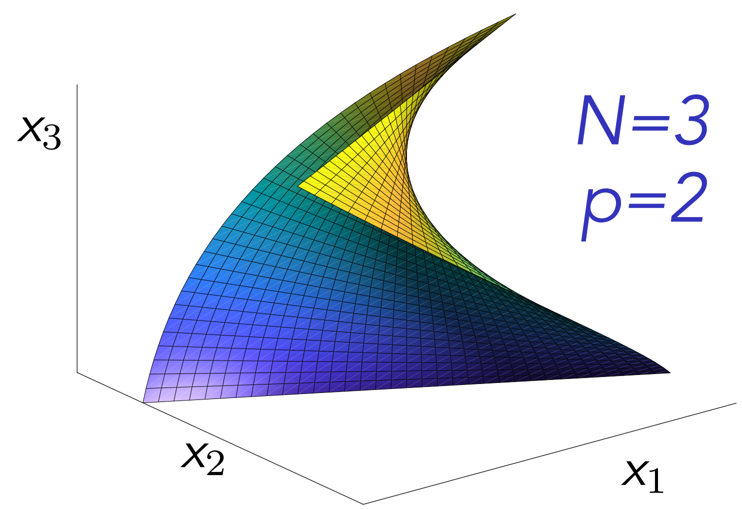
$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$



- Define low-dim manifold from decoder: $\mathcal{S} := \{\mathbf{g}(\hat{\mathbf{x}}) \,|\, \hat{\mathbf{x}} \in \mathbb{R}^p\} \subseteq \mathbb{R}^N$





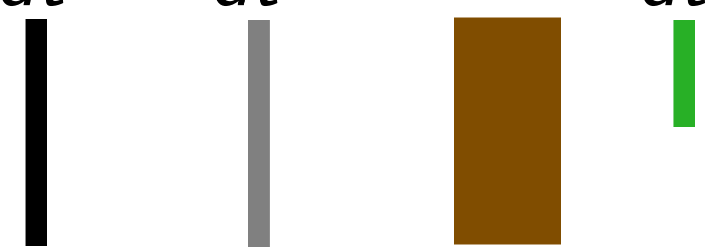
Reduction

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S} \qquad \frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla \mathbf{g}(\hat{\mathbf{x}}) \frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$$



Perform optimal projection

$$\frac{d\tilde{\mathbf{x}}}{dt}(\hat{\mathbf{x}}) \quad \text{satisfies} \quad \underset{\mathbf{v} \in T_{\hat{\mathbf{x}}}\mathcal{S}}{\text{minimize}} \left\| \mathbf{v} - \mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); \mathrm{t}, \boldsymbol{\mu}) \right\|_2$$

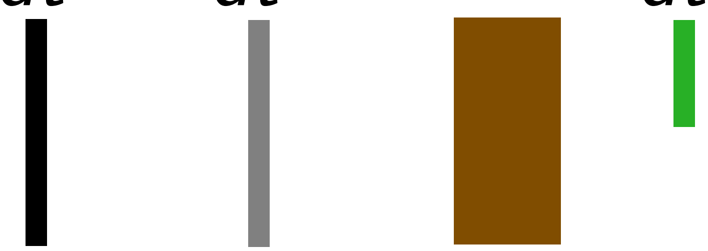
Reduction

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S} \qquad \frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla \mathbf{g}(\hat{\mathbf{x}}) \frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$$



Perform optimal projection with physics constraints

$$\frac{d\tilde{\mathbf{x}}}{dt}(\hat{\mathbf{x}}) \quad \text{satisfies} \quad \underset{\mathbf{v} \in T_{\hat{\mathbf{x}}}\mathcal{S}}{\text{minimize}} \; \|\mathbf{v} - \mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); \mathrm{t}, \boldsymbol{\mu})\|_2$$

$$\text{subject to } \mathbf{c}(\mathbf{v}, \mathbf{g}(\hat{\mathbf{x}}); t, \boldsymbol{\mu}) = \mathbf{0}$$

Reduction

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S} \qquad \frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla \mathbf{g}(\hat{\mathbf{x}}) \frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$$

Perform optimal projection with physics constraints

$$\frac{d\tilde{\mathbf{x}}}{dt}(\hat{\mathbf{x}}) \quad \text{satisfies} \quad \underset{\mathrm{v} \in T_{\hat{\mathbf{x}}}\mathcal{S}}{\text{minimize}} \; \|\mathrm{v} - \mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); \mathrm{t}, \boldsymbol{\mu})\|_2$$

$$\text{subject to} \quad \mathbf{c}(\mathrm{v}, \mathbf{g}(\hat{\mathbf{x}}); t, \boldsymbol{\mu}) = \mathbf{0}$$

+ Model integrates computational physics with deep learning

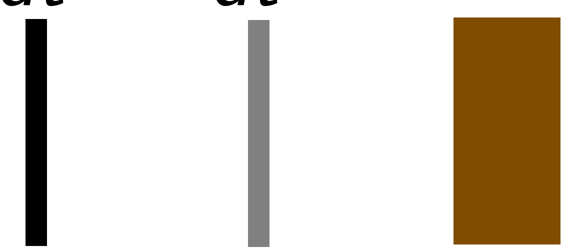
Reduction

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu})$$

1. *Training:* Solve ODE for $\boldsymbol{\mu} \in \mathcal{D}_{\text{training}}$ and collect simulation data
2. *Machine learning:* Identify low-dimensional manifold
3. *Reduction:* Project ODE onto manifold and solve for $\boldsymbol{\mu} \in \mathcal{D}_{\text{query}} \setminus \mathcal{D}_{\text{training}}$

Reduce the number of unknowns

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) = \mathbf{g}(\hat{\mathbf{x}}(t)) \in \mathcal{S} \qquad \frac{d\mathbf{x}}{dt} \approx \frac{d\tilde{\mathbf{x}}}{dt} = \nabla \mathbf{g}(\hat{\mathbf{x}}) \frac{d\hat{\mathbf{x}}}{dt} \in T_{\hat{\mathbf{x}}}\mathcal{S}$$



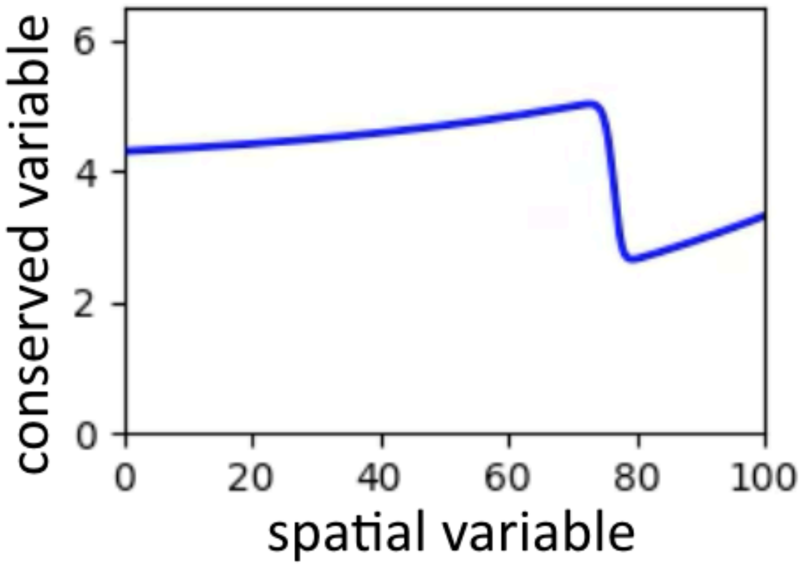
Perform optimal projection with physics constraints

$$\frac{d\tilde{\mathbf{x}}}{dt}(\hat{\mathbf{x}}) \quad \text{satisfies} \quad \underset{\mathbf{v} \in T_{\hat{\mathbf{x}}}\mathcal{S}}{\text{minimize}} \; \|\mathbf{v} - \mathbf{f}(\mathbf{g}(\hat{\mathbf{x}}); t, \boldsymbol{\mu})\|_2$$

$$\text{subject to } \mathbf{c}(\mathbf{v}, \mathbf{g}(\hat{\mathbf{x}}); t, \boldsymbol{\mu}) = \mathbf{0}$$

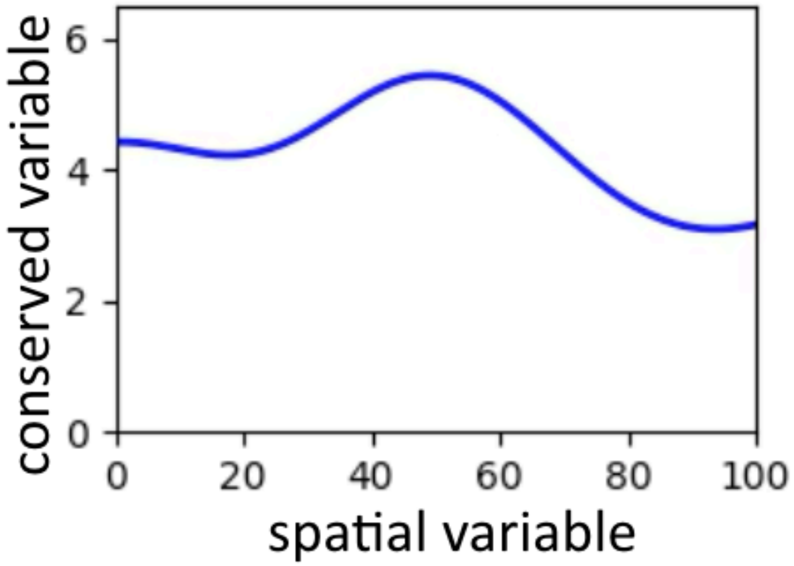
+ Model integrates computational physics with deep learning
+ Physics constraints exactly satisfied

High-fidelity model



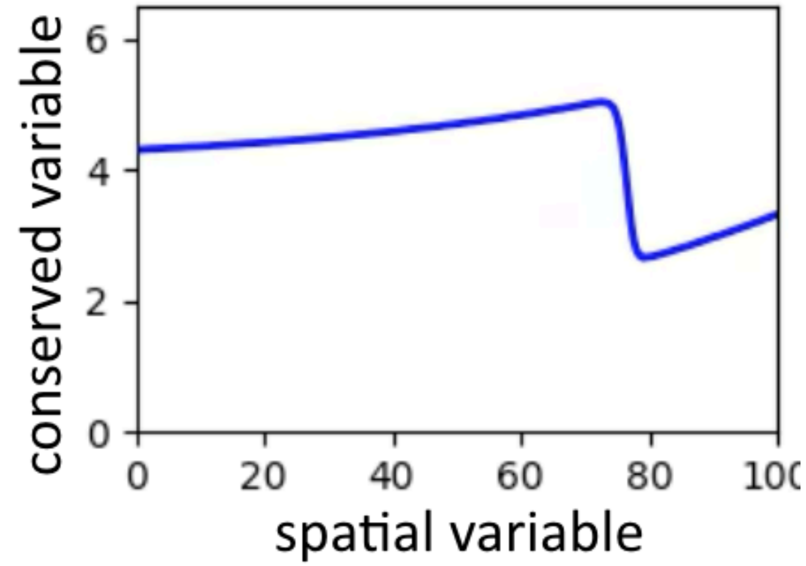
Reduced-order models

PCA subspace



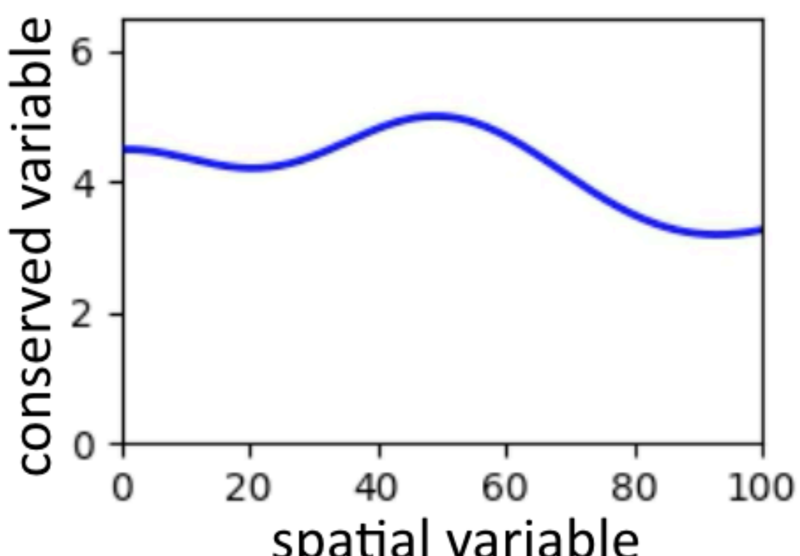
Solution error: 13%
Conservation violation: 16%

Autoencoder manifold



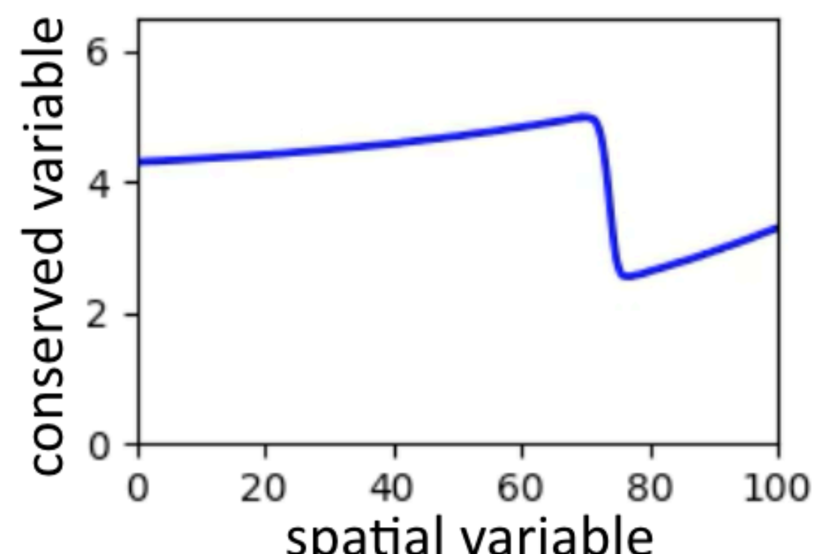
Solution error: 0.5%
Conservation violation: 1%

PCA subspace with conservation constraints



Solution error: 12%
Conservation violation: <0.001%

Autoencoder manifold with conservation constraints



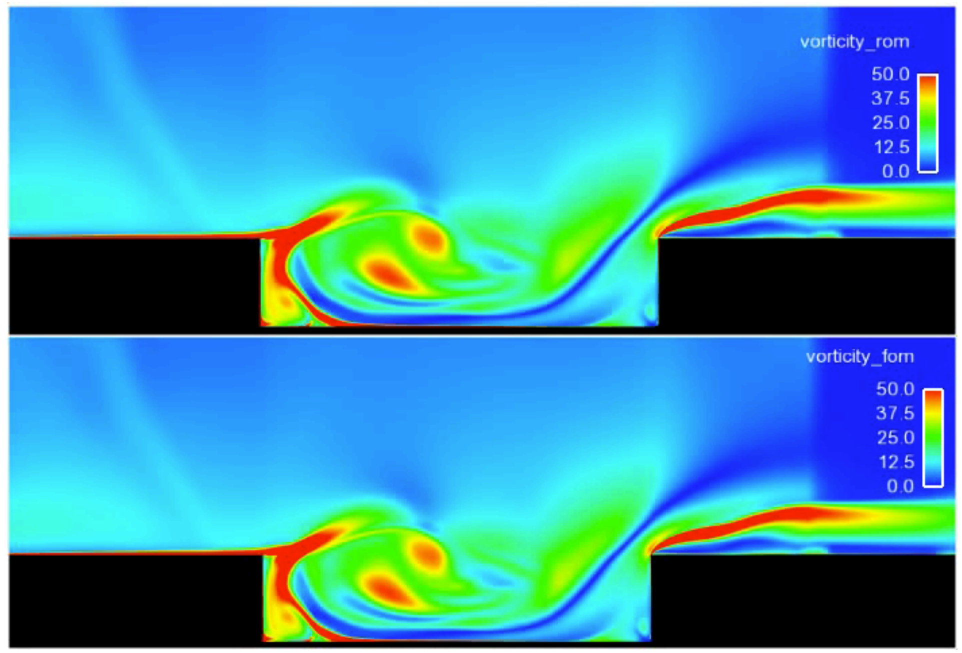
Solution error: 0.2%
Conservation violation: <0.001%

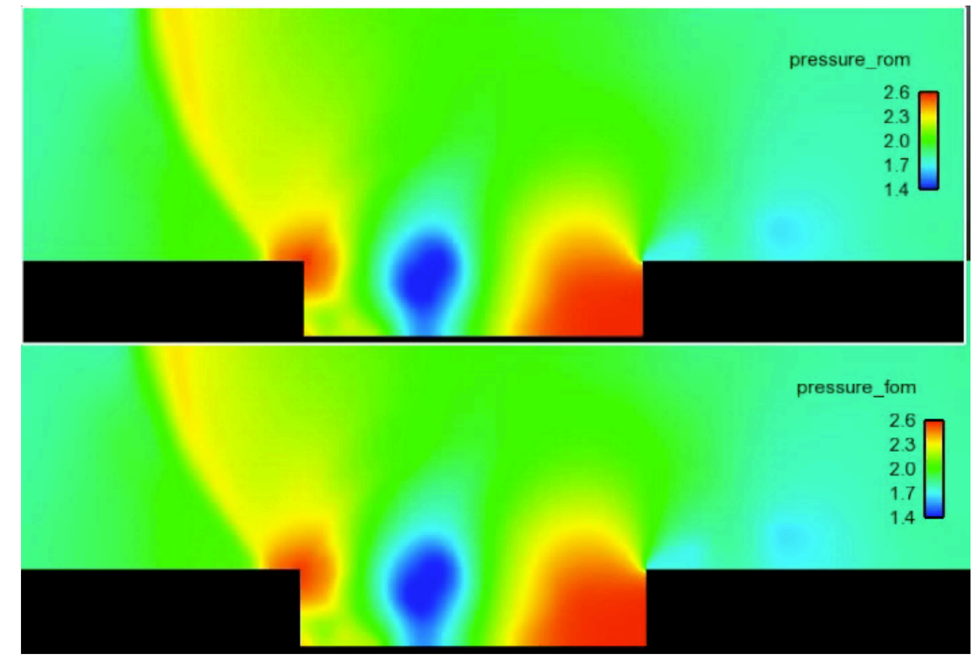
Currently implementing in large-scale code

vorticity field *pressure field*

Reduced-order model
PCA subspace
32 min, 2 cores

high-fidelity model
5 hours, 48 cores





Text

+ *229x savings in core–hours*
+ *< 1% error in time-averaged drag*

References:

- K. Lee and K. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. arXiv e-print, (1812.08373), 2018.
- K. Carlberg, Y. Choi, and S. Sargsyan. Conservative model reduction for finite-volume models. Journal of Computational Physics, 371:280–314, 2018.
- K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. Journal of Computational Physics, 330:693–734, 2017.