# Efficient nonlinear manifold reduced order model

Youngkyu Kim, Youngsoo Choi, David Widemann, Tarek Zohdi

Lawrence Livermore National Laboratory

Berkeley UNIVERSITY OF CALIFORNIA

NEURAL INFORMATION PROCESSING SYSTEMS

## Why surrogate models?

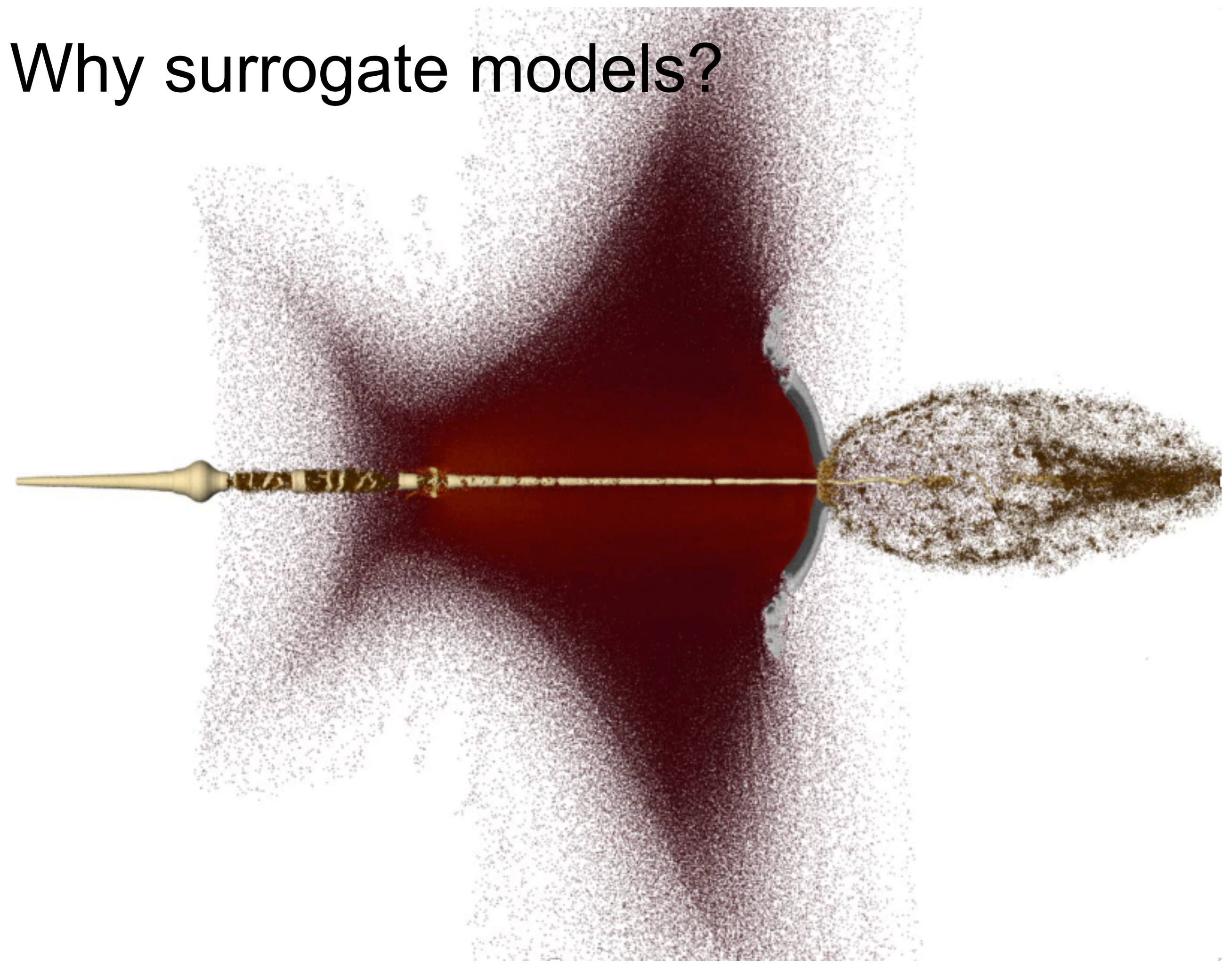
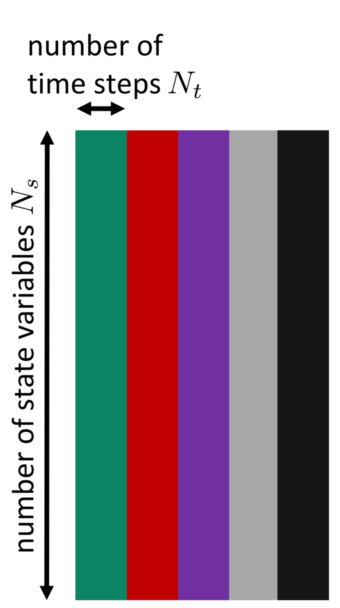


Figure 1. Smooth Particle Hydrodynamic (SPH) modeling for shape charge penetration [courtesy: Karen Wang]

High-fidelity, forward physical simulations are computationally expensive. For example, the shape charge penetration simulation in Figure 1 uses the smooth particle hydrodynamic method, which takes 7.4 days with 1,440 processors and 10.1 million particles. It is not ideal in multi-query applications, such as design optimization and uncertainty quantification. Thus, developing a **surrogate model** that accelerates a computationally expensive simulation without losing much accuracy is a great interest for many applications.

## Approach: Data-driven physical simulation

- ***Collect data***: solve full order model (FOM) for a sample, $\mu \in \mathcal{D}_{\text{sample}}$
- ***Machine learning***: build a surrogate model
- ***Accelerate*** physical simulation, $\mu \in \mathcal{D}_{\text{query}}$




***ML techniques:***

- Proper orthogonal decomposition
- Tucker decomposition
- Canonical Polyadic decomposition
- Autoencoder
- Generative adversarial network

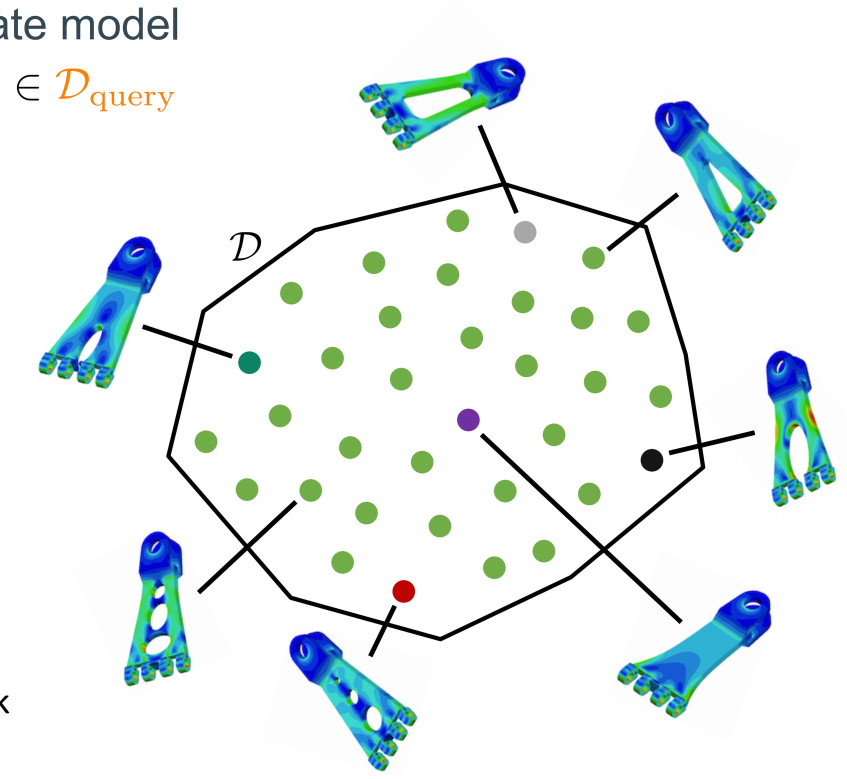
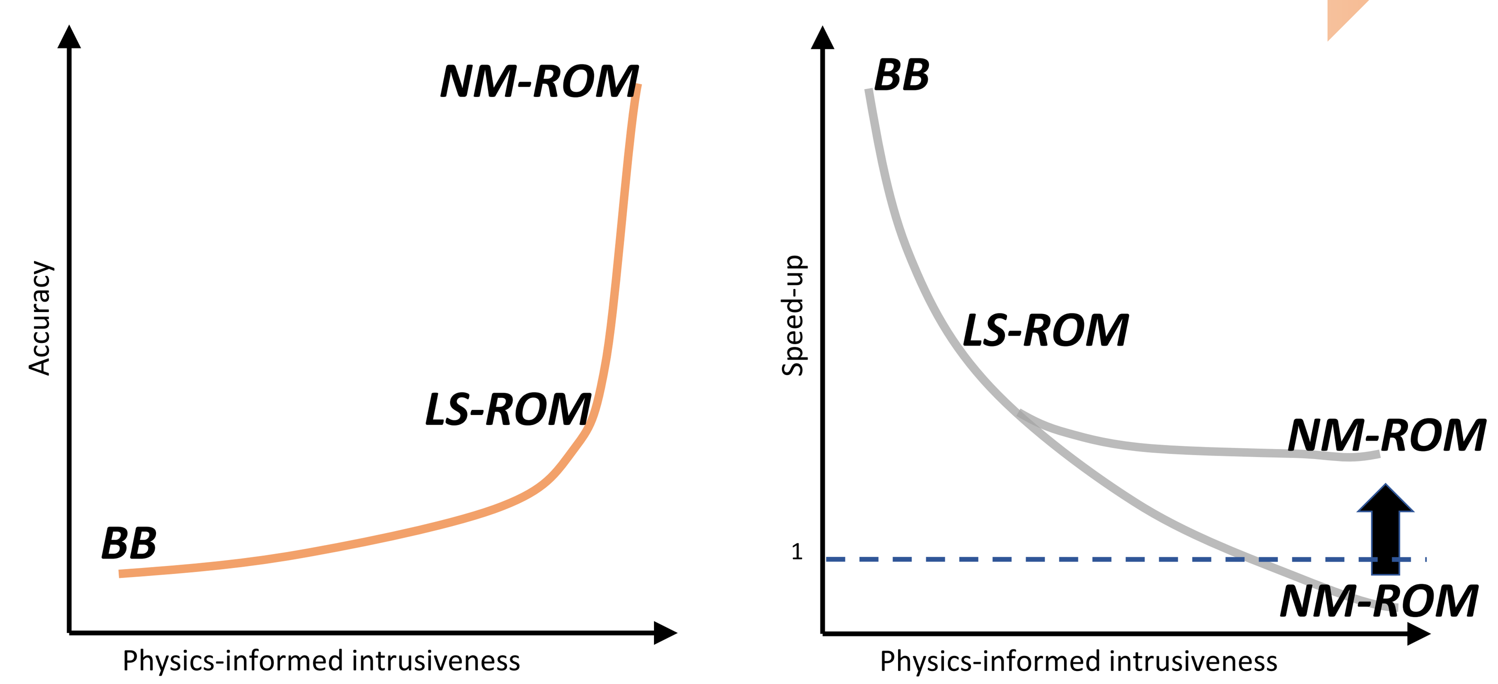



Figure 2. sampled solutions in parameter domain [1]

## Category of surrogate models via level of intrusiveness

***BB***: Black-box approach | ***LS-ROM:*** Linear subspace ROM | ***NM-ROM:*** Nonlinear manifold ROM




Roughly speaking, more intrusiveness means more accuracy. Therefore, reduced order models (ROMs) are more accurate than the black-box approach. On the other hand, more intrusiveness means less speed-up. Thus, the BB method is faster than ROMs in general. We are particularly interested in the most accurate surrogate model, that is, NM-ROM. NM-ROM used to be slower than the corresponding FOM. However, we made it faster so that it is much faster than FOM.

## LS-ROM vs NM-ROM[2]

Governing equation: $\dfrac{d\boldsymbol{w}}{dt} = \boldsymbol{f}(\boldsymbol{w}, t; \boldsymbol{\mu}), \quad \boldsymbol{w}, \boldsymbol{f} \in \mathbb{R}^{N_s}$

### LS-ROM

- **Solution approximation**:

$$\boldsymbol{w} \approx \tilde{\boldsymbol{w}} = \boldsymbol{w}_{\text{ref}} + \boldsymbol{\Phi}\hat{\boldsymbol{w}}, \; \boldsymbol{\Phi} \in \mathbb{R}^{N_s \times n_s}$$

- **Substitute & Discretize** (backward Euler) :

$$\tilde{\boldsymbol{r}}^n_{\text{LS}}(\hat{\boldsymbol{w}}) = \boldsymbol{\Phi}(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_{n-1}) - \boldsymbol{f}_n(\boldsymbol{w}_{\text{ref}} + \boldsymbol{\Phi}\hat{\boldsymbol{w}})$$

Scales with FOM size: $\mathbb{R}^{N_s}$

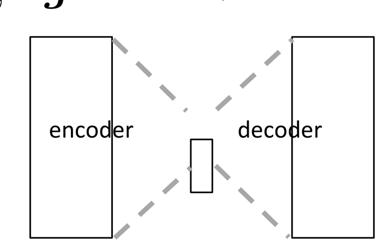
- **Project**: $\underset{\hat{\boldsymbol{w}} \in \mathbb{R}^{n_s}}{\arg\min} \; \frac{1}{2} \|\tilde{\boldsymbol{r}}^n_{\text{LS}}(\hat{\boldsymbol{w}})\|_2^2$

Hyper-reduction!

### NM-ROM

- **Solution approximation**:

$$\boldsymbol{w} \approx \tilde{\boldsymbol{w}} = \boldsymbol{w}_{\text{ref}} + \boldsymbol{g}(\hat{\boldsymbol{w}}), \quad \boldsymbol{g} : \mathbb{R}^{n_s} \rightarrow \mathbb{R}^{N_s}$$




- **Substitute & Discretize** (backward Euler):

$$\tilde{\boldsymbol{r}}^n_{\text{NM}}(\hat{\boldsymbol{w}}) = \boldsymbol{g}(\hat{\boldsymbol{w}}) - \boldsymbol{g}(\hat{\boldsymbol{w}}_{n-1}) - \Delta t \boldsymbol{f}_n(\boldsymbol{w}_{ref} + \boldsymbol{g}(\hat{\boldsymbol{w}}))$$

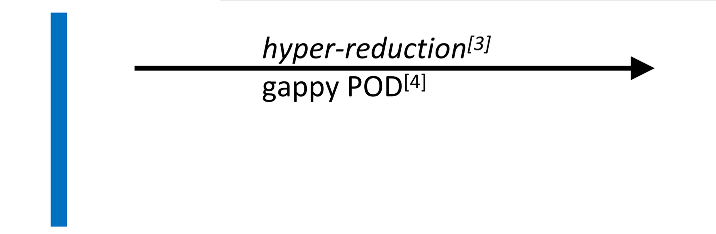
Scales with FOM size: $\mathbb{R}^{N_s}$

- **Project**: $\underset{\hat{\boldsymbol{w}} \in \mathbb{R}^{n_s}}{\arg\min} \; \frac{1}{2} \|\tilde{\boldsymbol{r}}^n_{\text{NM}}(\hat{\boldsymbol{w}})\|_2^2$

## **Hyper-reduction** enables a speed-up

- ***Hyper-reduction*** selects a subset of nonlinear terms

$$\boldsymbol{f}_n \in \mathbb{R}^{N_s} \rightarrow \boldsymbol{Z}^T\boldsymbol{f}_n \approx \boldsymbol{Z}^T\boldsymbol{\Phi}_f\hat{\boldsymbol{f}}_n \in \mathbb{R}^{n_f}$$




- Computation efficiency is achieved by a shallow sparse decoder

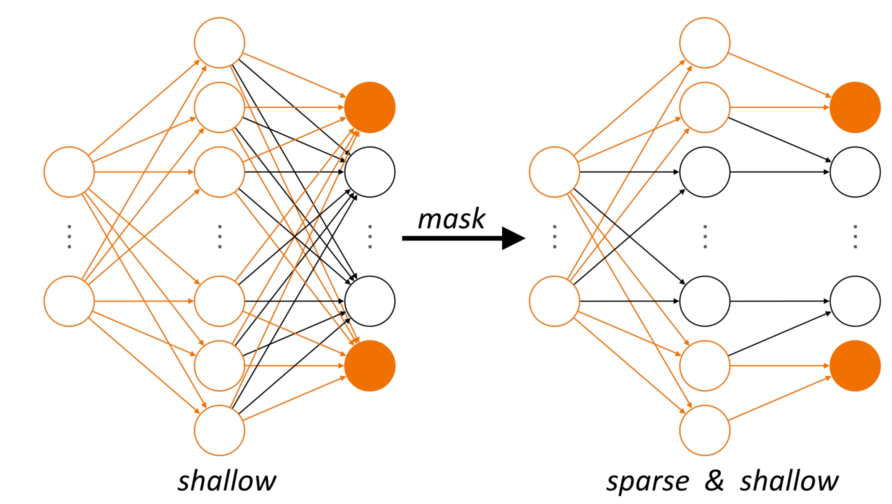



Figure 3. The shallower network involves the less number of hidden nodes for the computation of the selected outputs than a deep network. We build a sparser network network by applying a mask to further increase the efficiency.

## **Numerical result**: 2D viscous Burgers equation (advection-dominated)

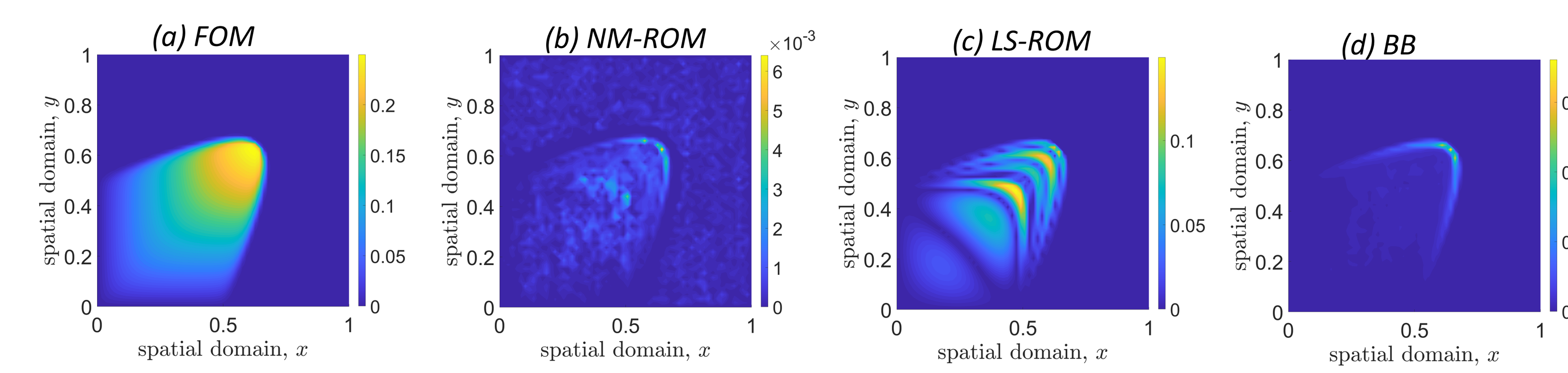



Figure 4. (a) FOM solution at the last time step and absolute differences of (b) NM-ROM, (c) LS-ROM, and (d) BB with respect to FOM solution at the last time step

| method | *NM-ROM* | *LS-ROM* | *BB*[5] |
|---|---|---|---|
| max. rel. error (%) | 0.93 | 34.4 | 38.6 |
| speed-up | 11.6 | 26.8 | 119 |

## References

[1] Xinyao Li and Liangli He (2020). “Shape optimization design for a centrifuge structure with multi topology configurations based on the B-spline FCM and GCMMA.” In: *Applied Sciences*, 10(2), p.620.

[2] Youngkyu Kim et al. (2020). “Efficient nonlinear manifold reduced order model.” In: *Advances in Neural Information Processing Systems.*

[3] Youngsoo Choi et al. (2020). “SNS: a solution-based nonlinear subspace method for time-dependent model order reduction.” In: *SIAM Journal on Scientific Computing*, 42(2):A1116-A1146.

[4] Richard Everson and Lawrence Sirovich (1995). “Karhunen-Loeve procedure for gappy data.” In: *JOSA*, 12(8):1657-1664.

[5] Byungsoo Kim, et al. (2019). “Deep fluids: a generative network for parameterized fluid simulations.” In: *Computer Graphics Forum*. 38(2).

LLNL-PRES-816543
This work was performed under the auspices of the U.S. Departement of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC