



Role of Hyper-Reduction in Enhancing the Efficacy of Digital Twins

Suparno Bhattacharyya, Jian Tao,
Eduardo Gildin, Jean C. Ragusa

TEXAS A&M
UNIVERSITY®

Team at Texas A&M



Jean Ragusa
Nuclear
Engineering



Eduardo
Gildin
Petroleum
Engineering



Jian Tao
Visualization
/TAMIDS



Suparno
Bhattacharyya
TAMIDS



Quincy Huhn
Nuclear
Engineering



Pravija Danda
Computer
Science

Reduced order models

- Reduced Order Models (ROMs) are instrumental in improving computational efficiency of large-scale system simulation.
- Complex models describing neutron diffusion, neutron transport phenomena in nuclear reactors, or flow and transport models in porous media, etc. are inherently high-dimensional and/or nonlinear, and consume vast computational resources during simulation.
- ROMs, in principle, can simplify such models, improving computational efficiency.
- Hyper-reduction techniques further improves ROMs: manage non-linearities effectively.

ROMs for Digital Twin

- Digital twin (DT) technology provides a virtual duplicate of a physical asset, allowing for real-time data exchange, analysis, and control.
- In the realm of nuclear reactors, DT has shown promise with the following:
 - Real-Time Monitoring & Control for enhanced safety.
 - Forecast and mitigate potential issues, such as heat fluctuations.
 - Reduce operational costs.
 - Enhanced reactor design optimization.
- DT requires fast and precise simulations; hyper-reduced ROMs are capable of bolstering both computational efficiency and accuracy.

System of interest: nonlinear heat conduction(ss)

- Steady-state nonlinear heat conduction model provides a simplified yet representative elliptic partial differential equation (PDE) for study.
- Models describing intricate thermo-fluid dynamics, and energetics of the nuclear reactor core include nonlinear elliptic operator.
- Other large-scale models of interest such as steady-state mass transfer in porous media are often characterized by elliptic PDEs.

Nonlinear heat conduction: governing equations

$$\rho c_v \frac{\partial T}{\partial t} = \nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T) \quad x \in \Omega$$

Parameter

$$0 = \nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T)$$

Steady-state

- **Fixed temperature at boundary** (Dirichlet): $T|_{\Gamma_d} = T_b$
- **Fixed heat flux at boundary** (Neuman): $-k(\mathbf{x}, T, \mu) \frac{\partial T}{\partial n} \Big|_{\Gamma_n} = q_n$
- **Robin (Mixed) Condition :** $-k(\mathbf{x}, T, \mu) \frac{\partial T}{\partial n} \Big|_{\Gamma_r} + h(T - T_{\text{ext}})|_{\Gamma_r} = 0$

Weak-form for FEM analysis

$$0 = \nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T)$$

$$0 = \int_{\Omega} [\nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T)] v(\mathbf{x}) d\Omega$$

$$\int_{\Omega} k(\mathbf{x}, T, \mu) \nabla T \cdot \nabla v d\Omega - \int_{\Gamma} k(\mathbf{x}, T, \mu) (\nabla T \cdot \mathbf{n}) v d\Gamma = - \int_{\Omega} q(\mathbf{x}, T) v d\Omega$$

Stiffness

Boundary term

Source term

Weak-form for FEM analysis

$$0 = \nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T)$$

$$0 = \int_{\Omega} [\nabla \cdot (k(\mathbf{x}, T, \mu) \nabla T) + q(\mathbf{x}, T)] v(\mathbf{x}) d\Omega$$

$$\int_{\Omega} k(\mathbf{x}, T, \mu) \nabla T \cdot \nabla v d\Omega - \int_{\Gamma} k(\mathbf{x}, T, \mu) (\nabla T \cdot \mathbf{n}) v d\Gamma = - \int_{\Omega} q(\mathbf{x}, T) v d\Omega$$

Stiffness Boundary term Source term

Dirichlet

Reflective

The diagram illustrates the decomposition of the boundary term in the weak form. A red arrow points from the boundary integral term $-\int_{\Gamma} k(\mathbf{x}, T, \mu) (\nabla T \cdot \mathbf{n}) v d\Gamma$ to the word "Dirichlet" above it. Another red arrow points from the same boundary integral term to the word "Reflective" below it.

Assembled FEM model

$$\mathbf{K}_{N_h \times N_h} \mathbf{T}_{N_h \times 1} = \mathbf{F}_{N_h \times 1}$$

$$K_{ij} = \sum_{e=1}^{n_e} \int_{\Omega_e} k(\mathbf{x}, T, \mu) \nabla N_i^e \cdot \nabla N_j^e d\Omega_e$$

$$F_i = \sum_{e=1}^{n_e} - \int_{\Omega_e} N_i^e q(\mathbf{x}, T) d\Omega_e$$

Nonlinear model solution requires iterative evaluation of T, K, F

Study with large set of parameters becomes computationally expensive for large N_h

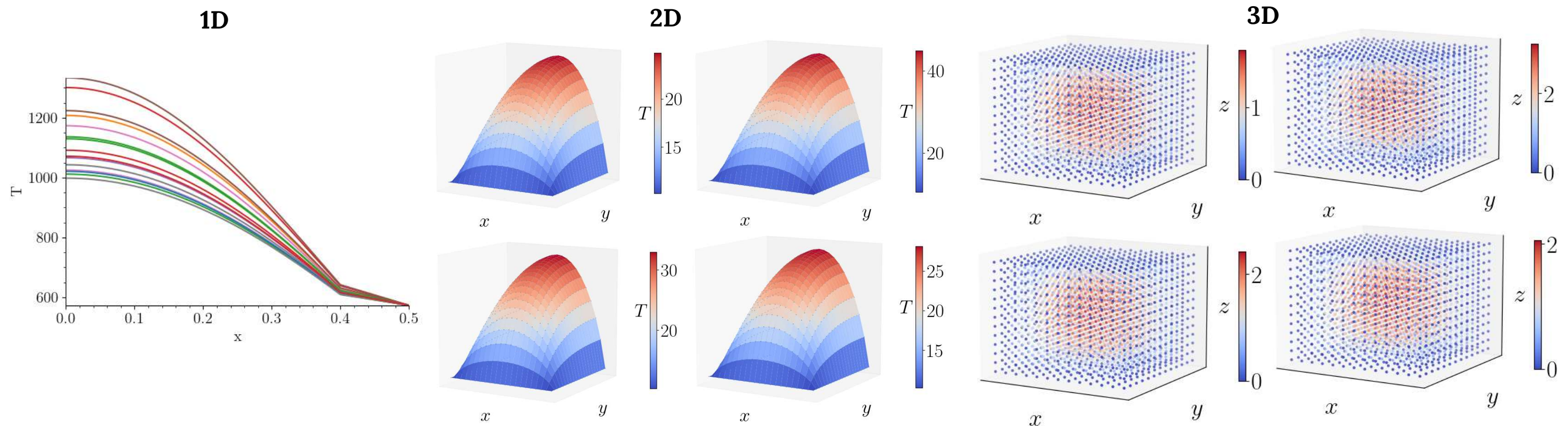
$$\mathbf{K}_{N_h \times N_h} \mathbf{T}_{N_h \times 1} = \mathbf{F}_{N_h \times 1}$$

$$K_{ij} = \sum_{e=1}^{n_e} \int_{\Omega_e} k(\mathbf{x}, T, \mu) \nabla N_i^e \cdot \nabla N_j^e d\Omega_e$$

$$F_i = \sum_{e=1}^{n_e} - \int_{\Omega_e} N_i^e q(\mathbf{x}, T) d\Omega_e$$

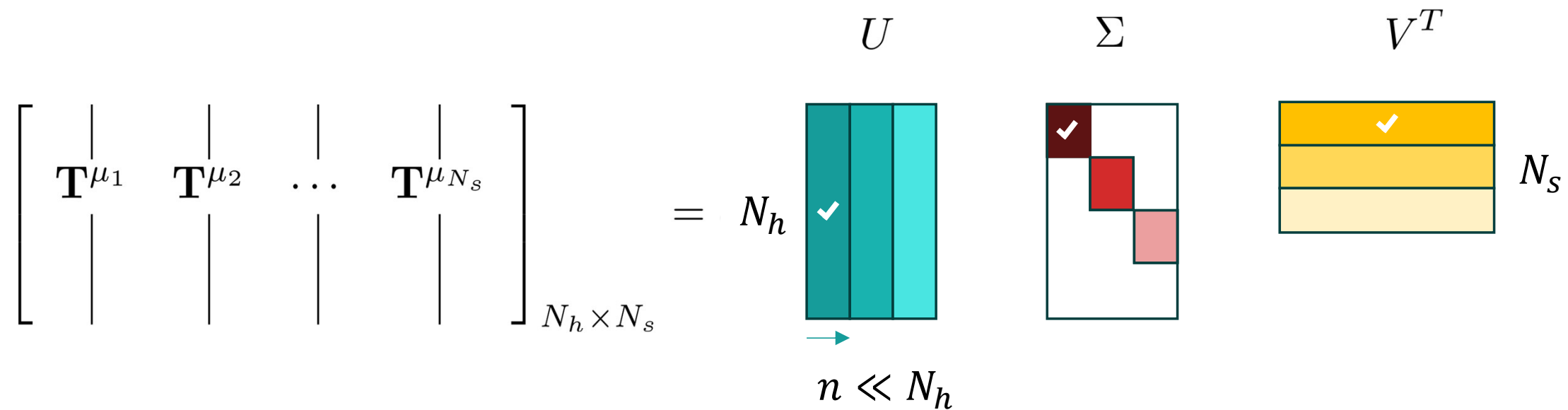
Model order reduction (MOR) for parametric nonlinear systems

- SVD on solution snapshots corresponding to different μ .



- Determine reduced subspace, spanned by the most dominant left singular vectors.

Derive nonlinear reduced order models

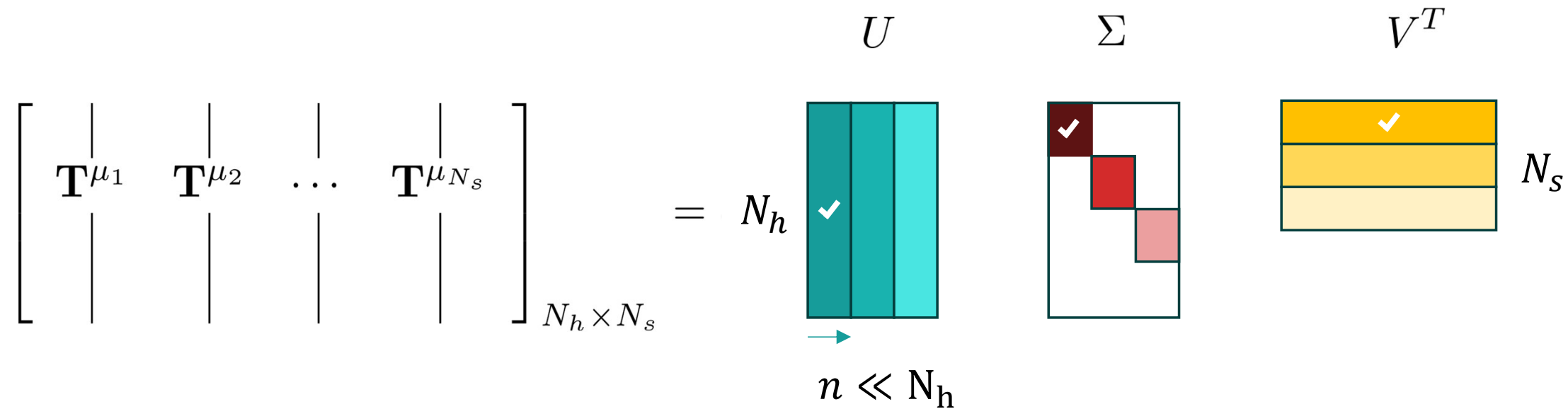


$$\tilde{\mathbf{U}} = \mathbf{U}[:, : n] \in \mathbb{R}^{N_h \times n}$$

$$\tilde{\mathbf{U}}^T \mathbf{K} \tilde{\mathbf{U}} \mathbf{T}_n = \tilde{\mathbf{U}}^T \mathbf{F}$$

$$\text{ROM: } \mathbf{K}_n \mathbf{T}_n = \mathbf{F}_n \quad \mathbf{T} = \tilde{\mathbf{U}} \mathbf{T}_n$$

Nonlinear ROM **still** requires iterative evaluation of $\mathbf{T}, \mathbf{K}, \mathbf{F}$



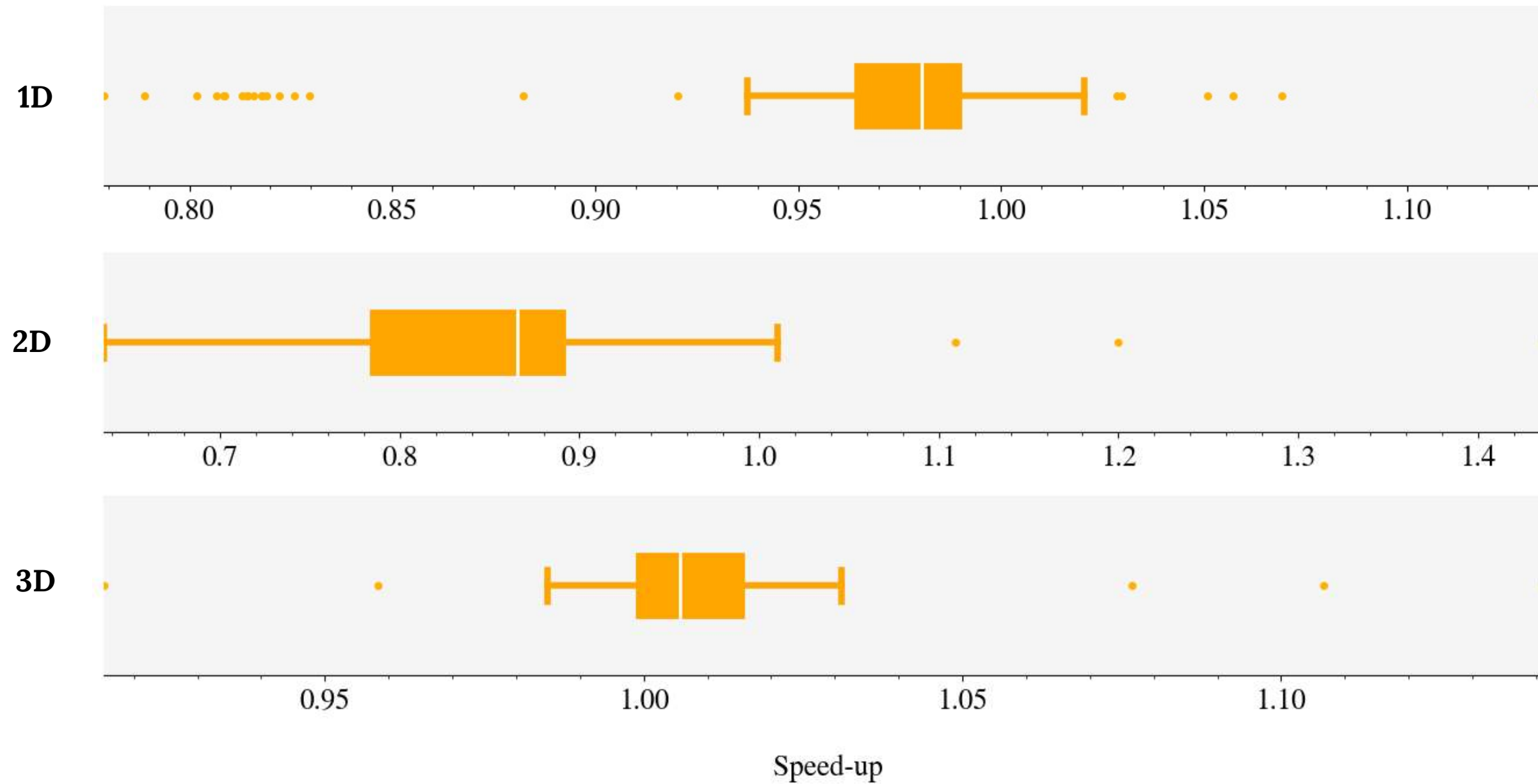
$$\tilde{\mathbf{U}} = \mathbf{U}[:, :n] \in \mathbb{R}^{N_h \times n}$$

$$\tilde{\mathbf{U}}^T \boxed{\mathbf{K}} \tilde{\mathbf{U}} \mathbf{T}_n = \tilde{\mathbf{U}}^T \boxed{\mathbf{F}}$$

$$\text{ROM: } \mathbf{K}_n \mathbf{T}_n = \mathbf{F}_n \quad \boxed{\mathbf{T}} = \tilde{\mathbf{U}} \mathbf{T}_n$$

Iteratively evaluate
 \mathbf{K}, \mathbf{F} using \mathbf{T} to
 calculate $\mathbf{K}_n, \mathbf{F}_n$

ROMs are hardly any faster!



Hyper-Reduction in parametric ROMs

- Minimization of computational complexity associated with evaluating nonlinear terms in ROMs.
- The Discrete Empirical Interpolation Method (DEIM).
- Energy conserving sampling and weighing (ECSW) method.

The Discrete Empirical Interpolation Method (DEIM)

- DEIM employs a specific kind of reduced basis, denoted as \mathbf{U}^f , to approximate the nonlinear term \mathbf{F} .

$$\mathbf{F}_{N_h \times 1} \approx \mathbf{F}_{N_h \times 1}^{\text{approx}} = \mathbf{U}_{N_h \times r}^f \hat{\mathbf{F}}_{r \times 1}$$

- Basis \mathbf{U}^f typically originates from snapshots of \mathbf{F} (e.g., by applying SVD).
- A sampling matrix \mathbf{P} is further calculated that selects specific components of \mathbf{F} , which is used to determine $\mathbf{F}^{\text{approx}}$ using

Evaluate the r rows of \mathbf{F}
indicated by \mathbf{P}

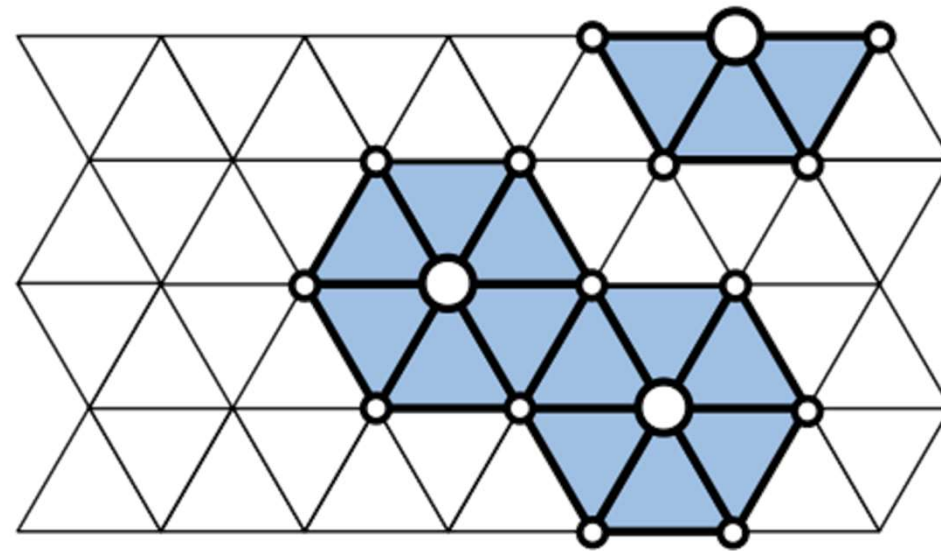
$$\hat{\mathbf{F}}_{r \times 1} = (\mathbf{P}_{r \times N_h}^T \mathbf{U}_{N_h \times r}^f)^\dagger \mathbf{P}_{r \times N_h}^T \mathbf{F}_{N_h \times 1}$$

The Discrete Empirical Interpolation Method (DEIM)

- The reduced force vector can then be calculated as

$$\mathbf{F}_n = \underbrace{\tilde{\mathbf{U}}^T \mathbf{F}^{\text{approx}}}_{\text{Approximate then project}} = \underbrace{\tilde{\mathbf{U}}^T \mathbf{U}^f (\mathbf{P}^T \mathbf{U}^f)^{\dagger}}_{\text{Pre-computed}} \mathbf{P}^T \mathbf{F}$$

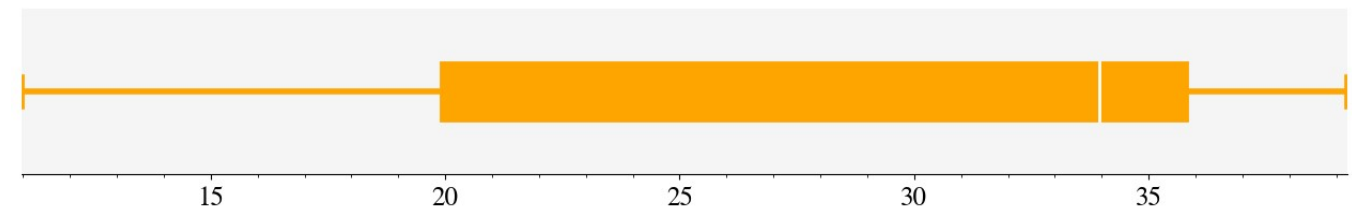
DEIM with FEM



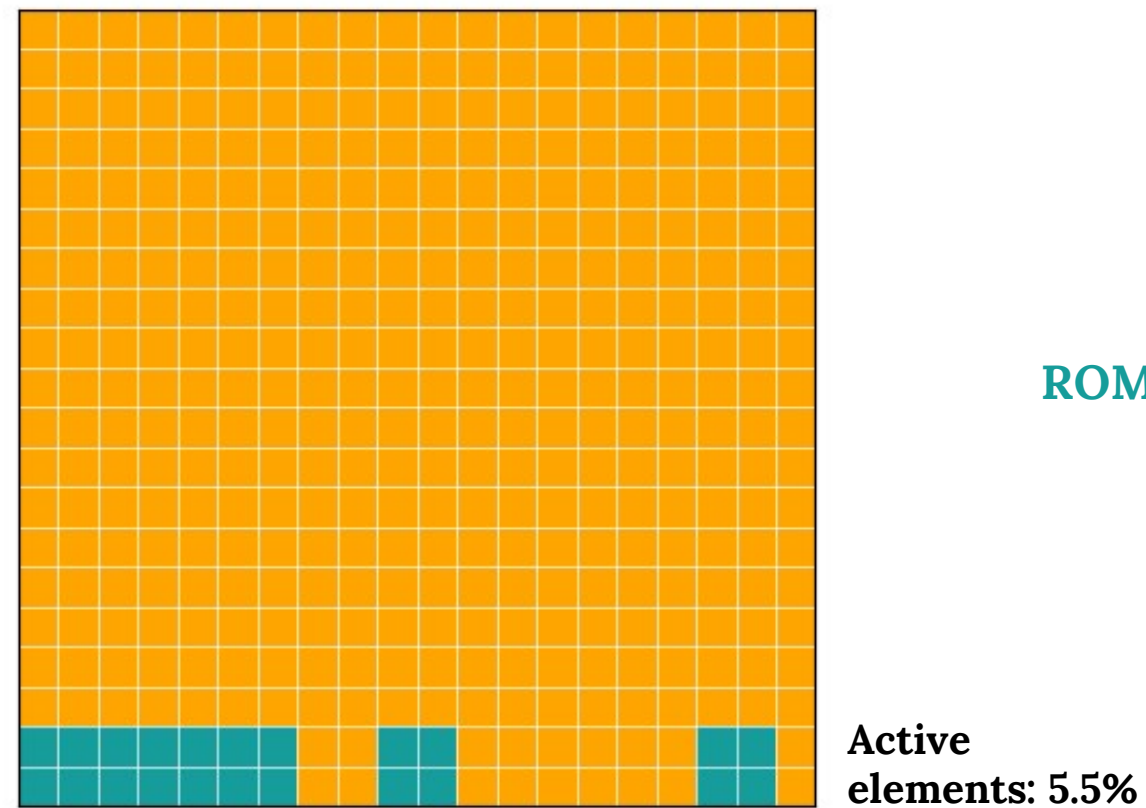
- Sampled rows of the nonlinear force vector, correspond to specific nodes.
- Calculate contribution only from elements associated with the nodes.
- Computational cost is a mere **fraction** of the total cost associated with iteratively evaluating and assembling **all** elements in the domain.

Results for Hyper-Reduced ROMs (DEIM)

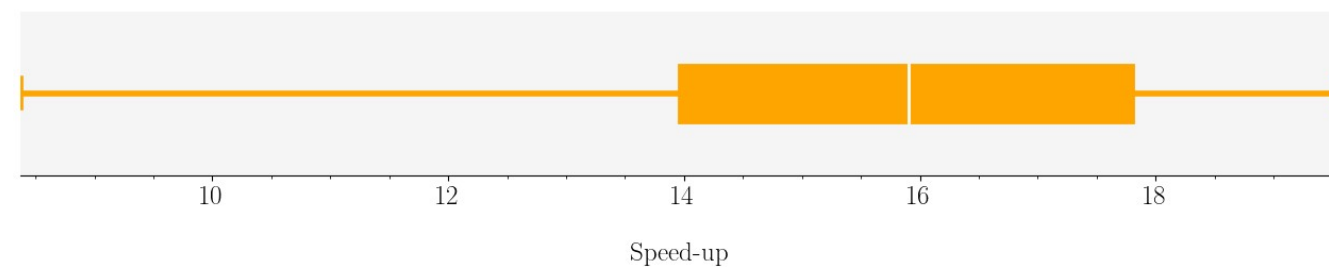
1D



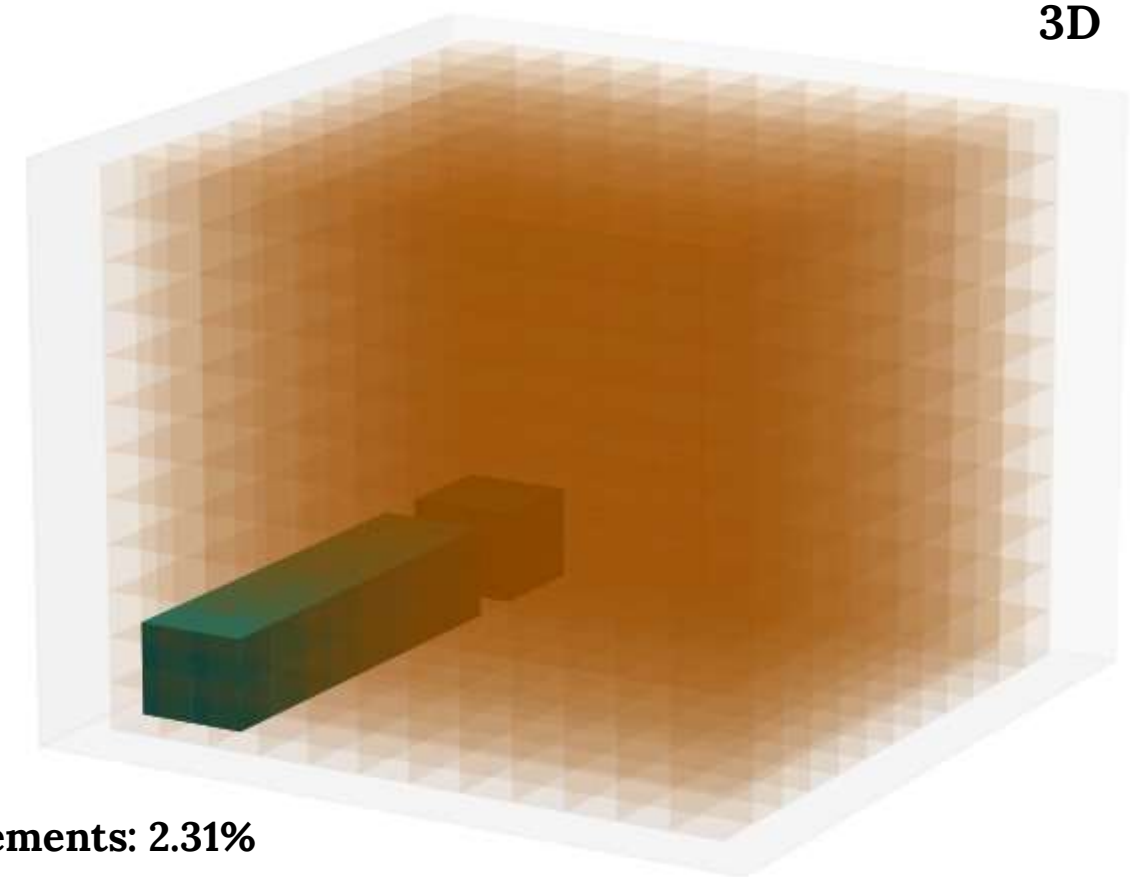
2D



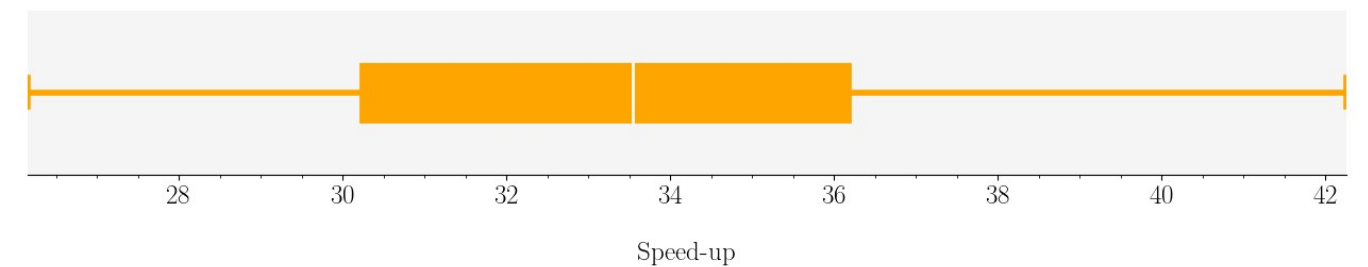
ROM error: $< 10^{-5}$



3D



Active elements: 2.31%



Energy conserving sampling and weighing (ECSW) method.

- Project-then-approximate hyper-reduction method.
- The basic idea is to match the “virtual-work” done by the internal “forces” onto the “displacements” induced by the reduction basis $\tilde{\mathbf{U}}$.

- The nonlinear projected term: $\mathbb{F} = \sum_{e=1}^{n_e} \tilde{\mathbf{U}}^e{}^T (\mathbf{K}^e \mathbf{T}^e - \mathbf{F}^e)$

Energy conserving sampling and weighing (ECSW) method.

- Weights ξ_e are determined by matching the “work” done by the nonlinear forces onto the reduction basis for a set of N_s sampled training forces.

$$\mathbb{F} = \sum_{e=1}^{n_e} \tilde{\mathbf{U}}^e{}^T (\mathbf{K}^e \mathbf{T}^e - \mathbf{F}^e)$$
$$\tilde{\mathbb{F}}^{\text{approx}} = \sum_{e \in E} \xi_e \tilde{\mathbf{U}}^e{}^T (\mathbf{K}^e \mathbf{T}^e - \mathbf{F}^e)$$

where ξ_e are positive weights and $|E| < n_e$.

Energy conserving sampling and weighing (ECSW) method.

- Underlying optimization problem:

$$\xi : \arg \min_{\tilde{\xi} \in \mathbb{R}^{n_e}, \tilde{\xi} \geq 0} \left\| \mathbf{G}\tilde{\xi} - \mathbf{b} \right\|_2$$

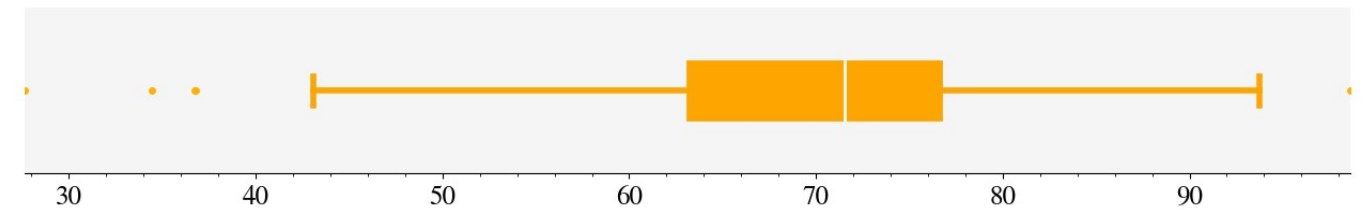
$$\mathbf{G}_{nN_s \times n_e} = \begin{bmatrix} \mathbf{g}^{1 \ 1} & \dots & \mathbf{g}^{1 \ n_e} \\ \vdots & \ddots & \vdots \\ \mathbf{g}^{N_s \ 1} & \dots & \mathbf{g}^{N_s \ n_e} \end{bmatrix} \quad \mathbf{g}_{n \times 1}^{ie} = \tilde{\mathbf{U}}^e{}^T \left(\mathbf{K}_i^e \tilde{\mathbf{U}}^e \tilde{\mathbf{U}}^e{}^T \mathbf{T}^{\mu_i} - \mathbf{F}_i^e \left(\tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{T}^{\mu_i} \right) \right)$$

$$\mathbf{b} = \mathbf{G}\mathbf{1}$$

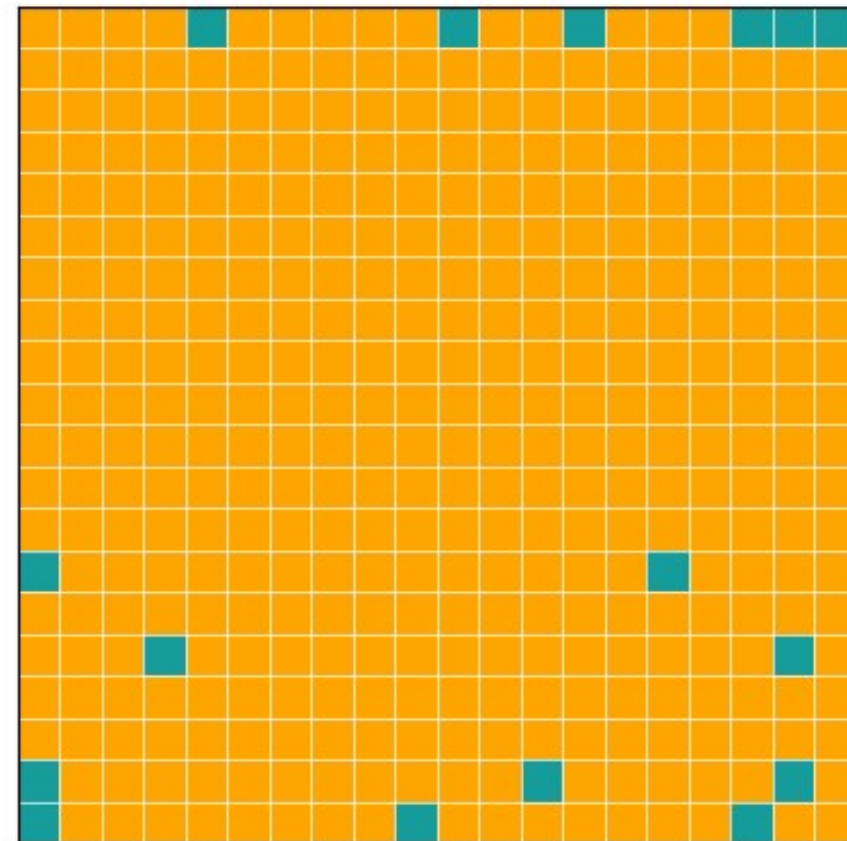
- Use non-negative least squares (NNLS) to solve the optimization problem and obtain sparse ξ

Results for Hyper-Reduced ROMs (ECSW)

1D

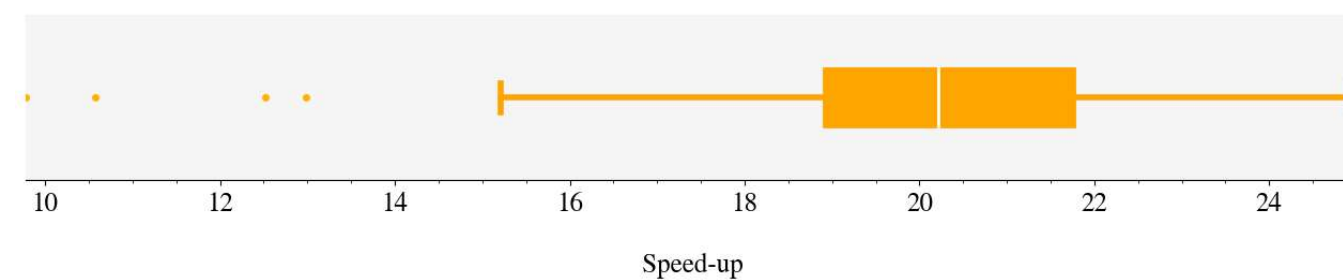


2D

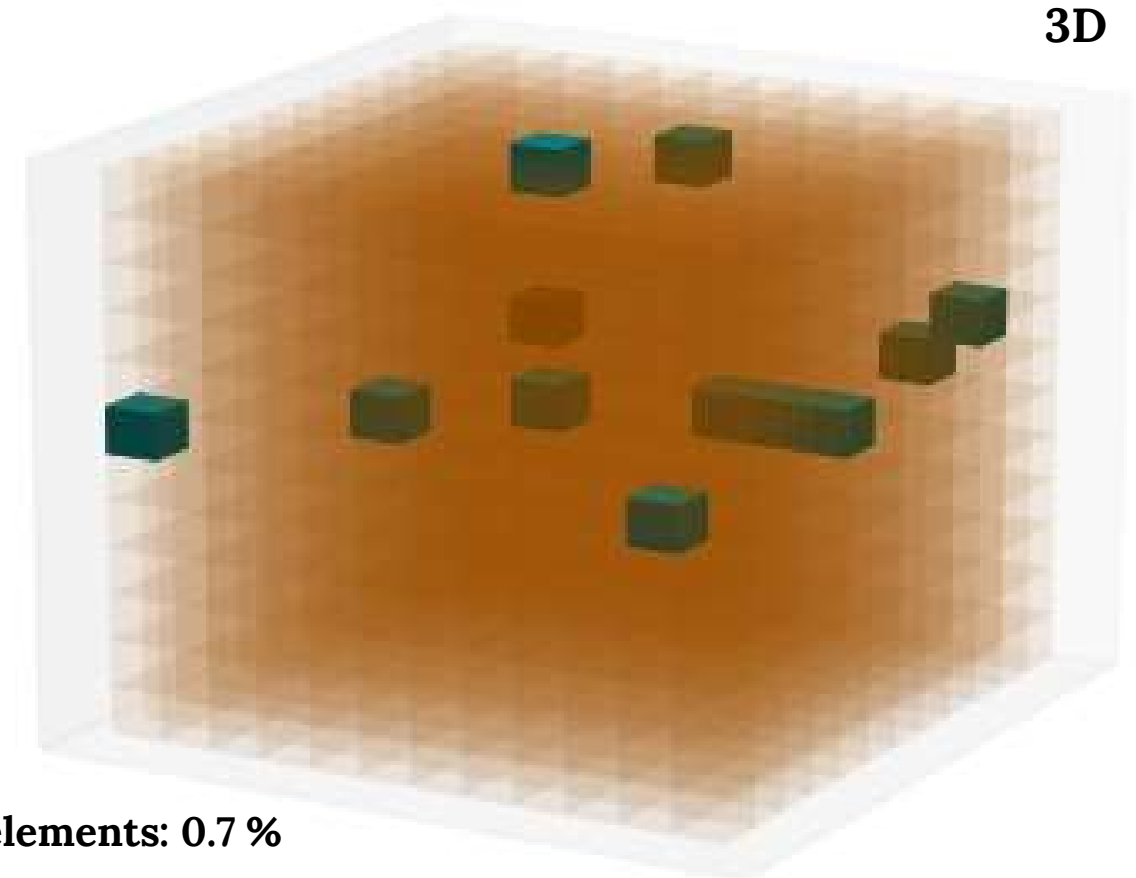


ROM error: $< 10^{-5}$

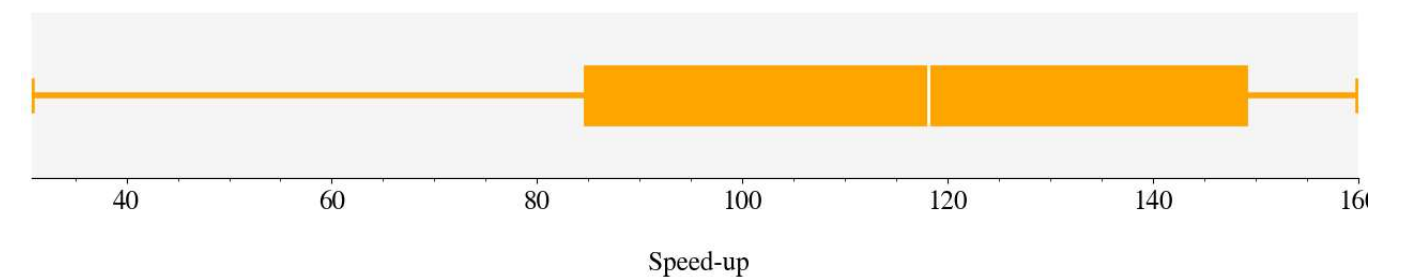
Active elements: 4.0%



3D

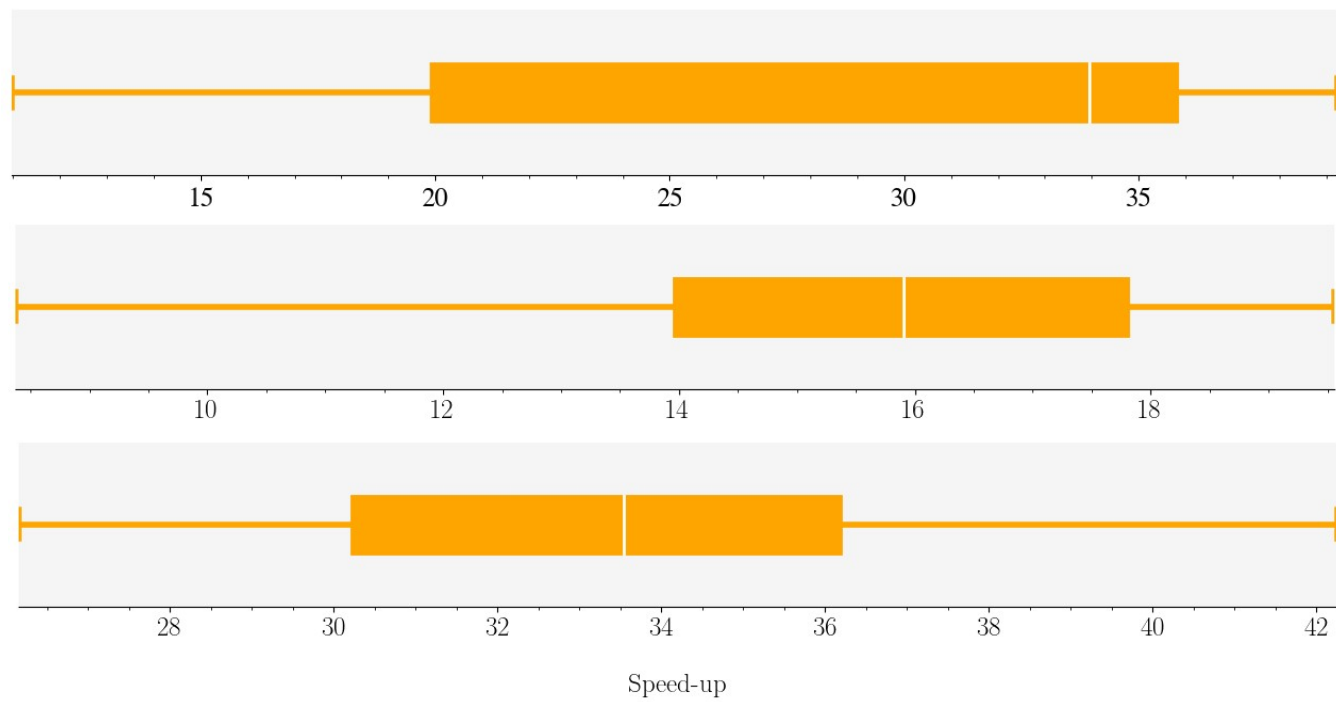


Active elements: 0.7 %

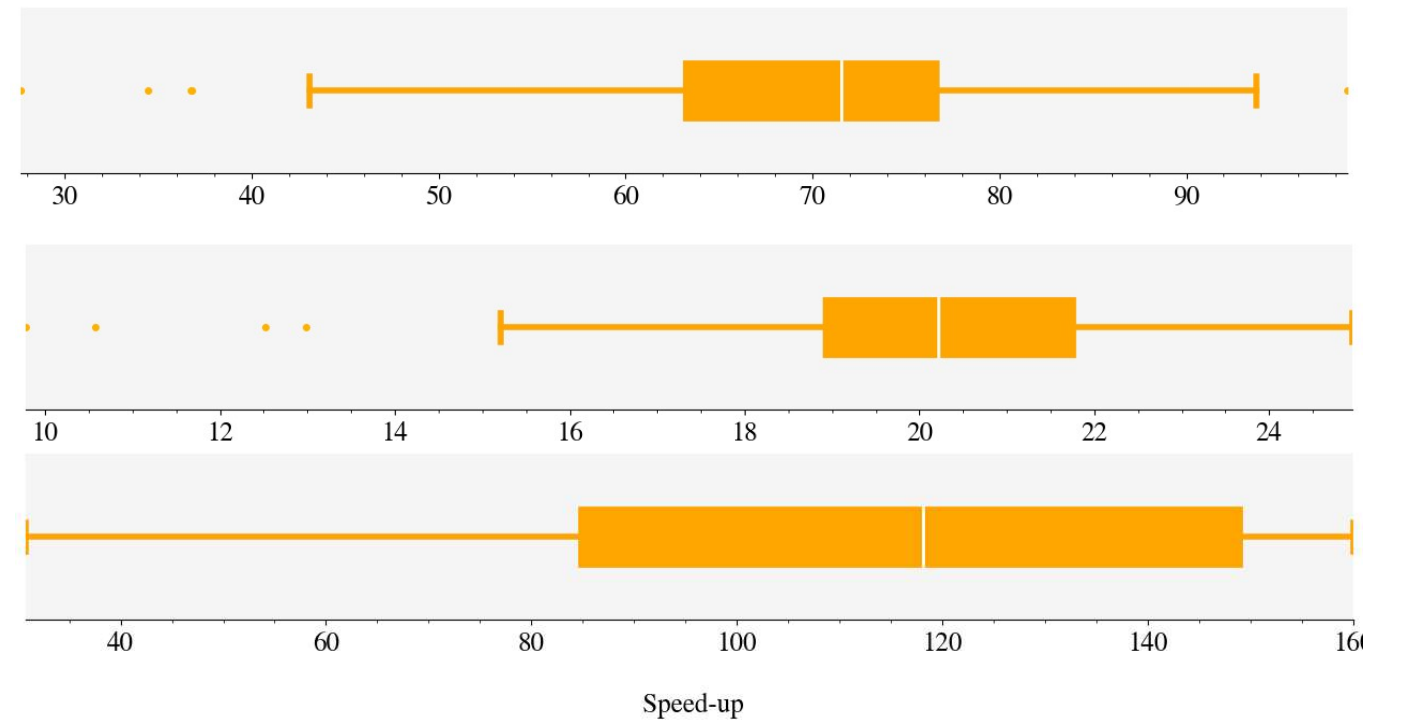


Comparison of speed-up

DEIM



ECSW

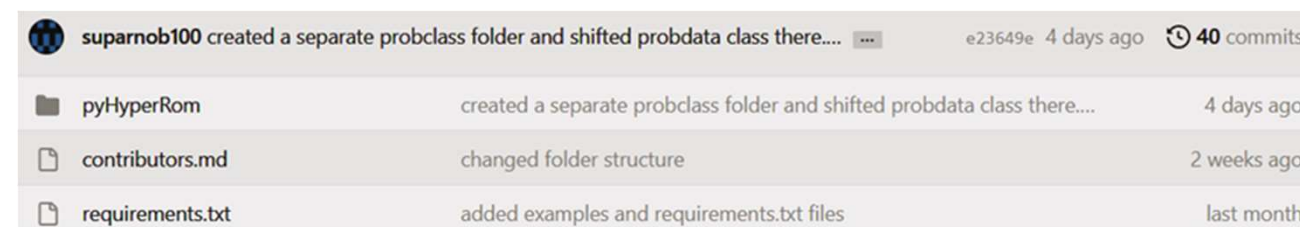


Conclusion

- Preliminary results on hyper-Reduced ROMs for steady-state nonlinear heat conduction problem.
- Comparison of the two popular hyperreduction techniques: DEIM, ECSW.
- Preliminary findings suggest ECSW-based hyperreduction is more effective than DEIM with similar ROM-error for these problems.

Future work

- Extend to time-dependent radiation transport models, and flow and transport models in porous media.
- Building a Github repository, which focuses on hyper-reduced ROMs that can be used for educational and computational purposes.



The screenshot shows a GitHub commit history for the user 'suparnob100'. The top bar indicates a commit 'e23649e' made '4 days ago' with '40 commits' in total. Below this, a table lists recent changes:

File	Commit Message	Time
pyHyperRom	created a separate probclass folder and shifted probdata class there....	4 days ago
contributors.md	changed folder structure	2 weeks ago
requirements.txt	added examples and requirements.txt files	last month

- Explore the scope of developing SciML-based hyper-reduction approaches.
- Repository of benchmark problems to evaluate hyper-reduction algorithms.
- **We are open to collaboration!**

suparnob@tamu.edu

Extra Slides

DEIM Optimization problem

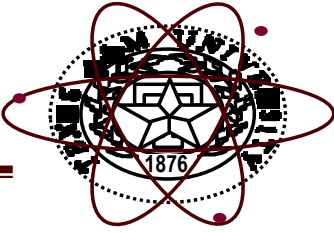
$$\hat{\mathbf{f}} = \arg \min_{\mathbf{y} \in \mathbb{R}^f} ||\mathbf{P}^T \mathbf{f} - \mathbf{P}^T \Phi_f \mathbf{y}||_2^2$$

Radiation transport is a much more complex equation that devolves to a diffusion operator in the limit of high amount of scattering, which is not an unreasonable approximation here.

The thermal fluid flow is typically solve using low-Mach fluid solver, so there's a pressure-Poisson solver (against an elliptic operator) hidden in there.

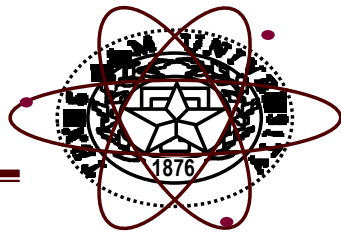
Energy conservation in the solid is a nonlinear elliptic operator.

Parallel NNLS



- The solution to the assembly problem is to domain decompose the problem and solve NNLS on each of the N_d sub-domains
- We take $\widetilde{N}_e = N_e/N_d$ and assemble
$$[C_1, C_2, \dots, C_{N_d}], C_i \in \mathbb{R}^{N_s \times \widetilde{N}_e}$$
- We then form and solve the NNLS problem on each sub-domain
$$[C_1 \xi_1^* \approx d_1, C_2 \xi_2^* \approx d_2, \dots, C_{N_d} \xi_{N_d}^* \approx d_{N_d}]$$
- After solving for $[\xi_1^*, \xi_2^*, \dots, \xi_{N_d}^*]$ we can form ξ^* that gives the weighting for ECSW over the entire domain
- Farhat et. al. (2015) proved that you can maintain a global tolerance ε by having a tolerance on each sub-domain of
$$\varepsilon_i \equiv \left(\frac{\|d\|_2}{N_s \|d_i\|} \right) \varepsilon$$
- We should be able to calculate NNLS in parallel with no loss in accuracy

Parallel NNLS Results



	Serial	Parallel N=2	Parallel N=5
Error	1.66e-06 %	3.73e-06 %	3.72e-06 %
% Cells Retained	10.6%	20.0%	40.2%
NNLS Speedup	1x	1.76x	4.03x
ROM Speedup	9.27x	4.47x	2.06x

- Speedup is achieved in the NNLS by the parallelization
- The number of cells retained goes up however in the case where there is not a tolerance
- This tradeoff means that the system matrix must be very large before it is economical to consider parallel NNLS