SAND2021-9303C

ASME IDETC-CIE 2021

International Design Engineering Technical Conferences & Computers and Information in Engineering Conference

VIRTUAL CONFERENCE AUG 17–20

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.

A MACHINE LEARNING FRAMEWORK FOR ALLEVIATING BOTTLENECKS OF PROJECTION-BASED REDUCED ORDER MODELS

Vlachas Konstantinos, Dept. of Civil, Environmental and Geomatic Engr., ETH Zurich (Presenter)
T. Simpson, Dept. of Civil, Environmental and Geomatic Engr., ETH Zurich
C. Martinez, A. R. Brink, Component Sciences and Mechanics, Sandia National Laboratories, Albuquerque, New Mexico
Prof. Dr. E. Chatzi, Dept. of Civil, Environmental and Geomatic Engr., ETH Zurich

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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. SAND2021-3704 A

Problem Statement

Physics-Based Reduced Order Modelling (ROM)

Real life system

Complex Dynamics

Nonlinear behaviour

Parametric dependency on:

- *Geometric features*
- *Material properties*
- *Environmental conditions*
- *Operational conditions*
- *Excitation*

High fidelity Finite Element model

Capture underlying dynamics

Reproduce physical behaviour

Retain parametric dependencies

Efficient evaluation

Virtual Representation

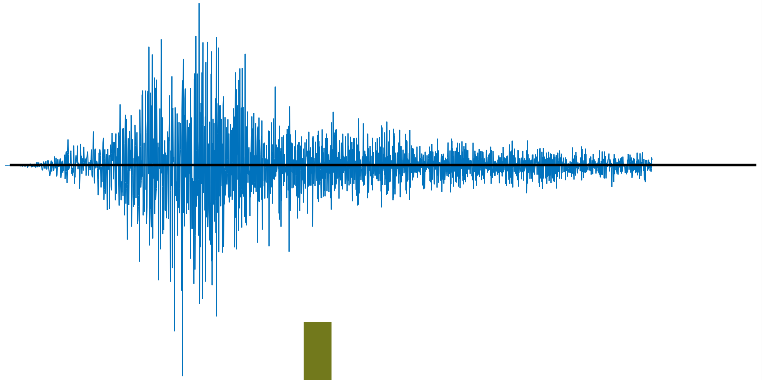
Reduced Order Model (ROM)

Problem Statement

Physics-Based Reduced Order Modelling (ROM)

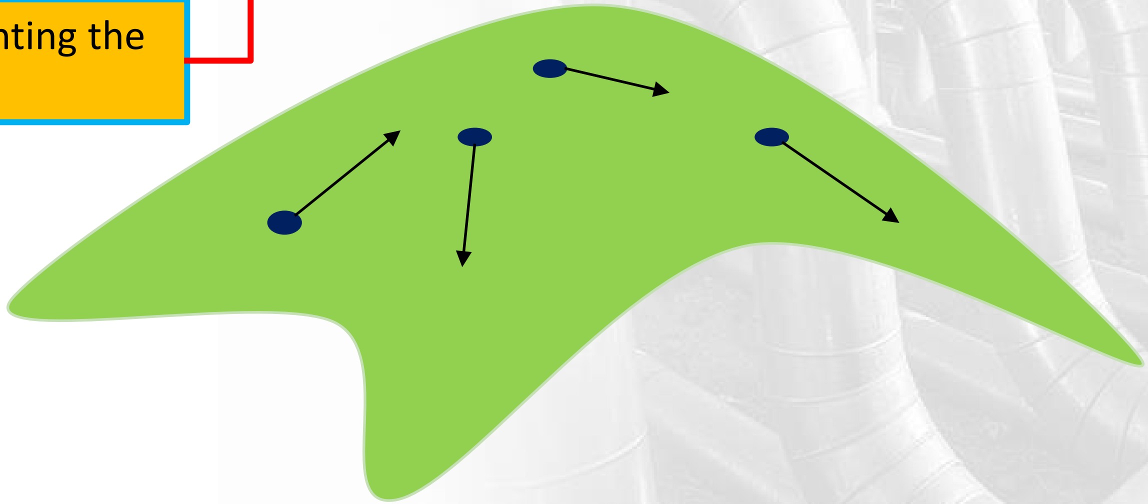
Real life system

Dynamic structural response



Try to extract components representing the **solution manifold S**

Physical Interpretation:
Dynamic response under any parametric state spans low-dimensional subspace ***S***



Problem Statement
Projection-Based Reduction

Assuming a general a nonlinear, parametric, dynamical system:

$$\mathbf{M}(\mathbf{p})\ddot{\mathbf{u}}(t) + \mathbf{g}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}\right) = \mathbf{f}(t, \mathbf{p})$$

$$\mathbf{u}(t) \in \mathbb{R}^n, \mathbf{M}(\mathbf{p}) \in \mathbb{R}^{n \times n}, \mathbf{f}(t, \mathbf{p}) \in \mathbb{R}^n, \mathbf{g}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t)\right) \in \mathbb{R}^n$$

Parametric dependency on *k* parameters denoted by: $\mathbf{p} = [p_1, ..., p_k]^T \in \Omega \subset \mathbb{R}^k$

Relevant notation:

M is the system mass matrix

f is the vector of external loads

u is the response time history

g are the nonlinear, state-dependent internal forces

Problem Statement

Projection-Based Reduction

*The **goal of parametric ROM is to generate a low-dimensional, equivalent system** such that the underlying physics along with the parametric dependencies of interest are further retained.*

$$\mathbf{M}_{\mathrm{r}}(\mathbf{p}_j)\ddot{\mathbf{u}}_{\mathrm{r}}(t) + \mathbf{g}_{\mathrm{r}}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right) = \mathbf{f}_{\mathrm{r}}(t, \mathbf{p}_j)$$

$$\mathbf{M}_{\mathrm{r}}(\mathbf{p}_j) \in \mathbb{R}^{r\times r}, \mathbf{g}_{\mathrm{r}}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right) \in \mathbb{R}^{r}, \mathbf{f}_{\mathrm{r}}(t, \mathbf{p}_j) \in \mathbb{R}^{r}$$

$$r \ll n$$

Galerkin Projection Basis

$$\mathbf{u}(t) = \mathbf{V}(\mathbf{p}_j)\mathbf{u}_{\mathrm{r}}(t)$$

$$\mathbf{M}_{\mathrm{r}}(\mathbf{p}_j) = \mathbf{V}(\mathbf{p}_j)^T\mathbf{M}(\mathbf{p}_j)\mathbf{V}(\mathbf{p}_j)$$

$$\mathbf{f}_{\mathrm{r}}(\mathbf{p}_j) = \mathbf{V}(\mathbf{p}_j)^T\mathbf{f}(t, \mathbf{p}_j)$$

$$\mathbf{g}_r(\mathbf{p}_j) = \mathbf{V}(\mathbf{p}_j)^T\mathbf{g}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right)$$

Problem Statement

Algorithmic Framework

Step 1: Parametric input states

$$\forall \mathbf{p}_k, k \in [1, N_s]$$

Step 2: Time Integration of Full Model

$$\forall t_i, i \in [0, N_t]$$

Notation:

- n : Full-order dimension
- N_s : Number of training samples
- N_t : Number of simulated timesteps
- $\mathbf{M}$: Mass matrix
- $\mathbf{f}$: External forcing
- $\mathbf{u}$: Response solution

For each parametric state:

- Assemble **system matrices**
  stiffness K / mass M / damping C / ***Excitation f***
- Evaluate the **time domain response** (integration)

The full-order, high fidelity finite element model **depends on a parametric input state.**

The parametric states are first sampled. The respective **parameters may represent**:

- ***system properties:*** *yield stress, hysteretic damping coeffs.*
- ***excitation traits:*** *amplitude of ground motion, frequency content*

Problem Statement

Algorithmic Framework

Training / Offline Phase

Step 1: Parametric input states

$$\forall \mathbf{p}_k, k \in [1, N_s]$$

Step 2: Time Integration of Full Model

$$\forall t_i, i \in [0, N_t]$$

Step 3: Assemble matrices and evaluate Equations of Motion

$$\mathbf{M}(\mathbf{p})\ddot{\mathbf{u}}(t_i) + \mathbf{g}_i(\mathbf{u}(t_i), \dot{\mathbf{u}}(t_i), \mathbf{p}) - \mathbf{f}(\mathbf{t_i}, \mathbf{p}) = \mathbf{0}$$

$$\mathbf{u}(t) \in \mathbb{R}^n, \mathbf{M}(\mathbf{p}) \in \mathbb{R}^{n \times n}, \mathbf{f}(t, \mathbf{p}) \in \mathbb{R}^n, \mathbf{g}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t)\right) \in \mathbb{R}^n$$

Nonlinear terms *includes stiffness and damping*

Step 4: Compute Residual on Equations and "predict" correction

$$if \quad \mathbf{R}_i(\mathbf{u}_r(t_i)) > tol => \tilde{\mathbf{u}}_r(t_i)$$

Step 5: Employ updated solution to re-assemble nonlinear terms & matrices

$$\tilde{\mathbf{u}}(t_i) => \mathbf{g}_i(\tilde{\mathbf{u}}(t_i), \dot{\tilde{\mathbf{u}}}(t_i), \mathbf{p})$$

Notation:
- N_s: Full-order dimension
- n : Number of training samples
- N_t: Number of simulated timesteps
- $\mathbf{M}$: Mass matrix
- $\mathbf{f}$: External forcing
- $\mathbf{u}$: Response solution

Problem Statement

Algorithmic Framework

POD - Projection-based Reduction

Assemble POD Basis

Proper Orthogonal Decomposition

$$\mathbf{u}(t) = \mathbf{V}(\mathbf{p})\mathbf{u}_{\mathrm{r}}(t) \qquad \mathbf{U} \equiv [\mathbf{u}(t_1)...\mathbf{u}(t_N)] = \mathbf{W}\boldsymbol{\Sigma}\mathbf{R}^T$$

$$\mathbf{V} \equiv \mathbf{W}_k = \mathbf{W}(:, 1:\mathrm{k})$$

Each training point has a *projection basis.*

principal components

As a next step, to evaluate the pROM :

- Assemble the ***full-order matrices*** of the validation parametric state
- Assemble a ***suitable basis*** representing the solution subspace (***Interpolation / Clustering***)
- ***Project*** the system matrices on the reduced dimension
- ***Integrate the reduced order matrices*** forward in time to evaluate the dynamic response
- ***Map*** the reduced response ***back*** to full coordinates

Problem Statement

Bottlenecks/Limitations

POD - Projection-based Reduction

Assemble POD Basis

Proper Orthogonal Decomposition

$$\mathbf{u}(t) = \mathbf{V}(\mathbf{p})\mathbf{u}_{\mathrm{r}}(t) \qquad \mathbf{U} \equiv [\mathbf{u}(t_1)...\mathbf{u}(t_N)] = \mathbf{W\Sigma R}^T$$

$$\mathbf{V} \equiv \mathbf{W}_k = \mathbf{W}(:, 1:\mathrm{k})$$

Limitations:

- **POD is a linear operator**
  Linearization in neighbourhood of stable points is assumed to address nonlinearities
- **Accuracy** for new parametric states *relies on clustering or interpolation between POD bases*

Problem Statement

Algorithmic Framework

ROM Evaluation / Online Phase

Step 1: Parametric input states

$$\exists \mathbf{p}_v, v \notin [1, N_s]$$

Step 2: Time Integration of ROM

$$\forall t_i, i \in [0, N_t]$$

Step 3: Assemble matrices and evaluate Equations of Motion

$$\mathbf{M}_r(\mathbf{p})\ddot{\mathbf{u}}_r(t_i) + \mathbf{g}_{ri}(\mathbf{u}(t_i), \dot{\mathbf{u}}(t_i), \mathbf{p}) - \mathbf{f}_\mathrm{r}(\mathbf{t_i}, \mathbf{p}) = \mathbf{0}$$

$$\mathbf{M}_\mathrm{r}(\mathbf{p}_j) \in \mathbb{R}^{r\times r}, \mathbf{g}_\mathrm{r}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right) \in \mathbb{R}^r, \mathbf{f}_\mathrm{r}(t, \mathbf{p}_j) \in \mathbb{R}^r$$

Step 4: Compute Residual on Equations and "predict" correction

$$if \quad \mathbf{R}_i(\mathbf{u}_r(t_i)) > tol => \tilde{\mathbf{u}}_r(t_i)$$

Step 5: Employ updated solution to re-assemble nonlinear terms & matrices

$$\tilde{\mathbf{u}}(t_i) = \mathbf{V}\tilde{\mathbf{u}}_\mathbf{r}(t_i)$$

$$\tilde{\mathbf{u}}(t_i) => \mathbf{g}_i(\tilde{\mathbf{u}}(t_i), \dot{\tilde{\mathbf{u}}}(t_i), \mathbf{p})$$

$$\mathbf{g}_{ri} = \mathbf{V}^T\mathbf{g}_i$$

Problem Statement

Bottlenecks/Limitations

ROM Evaluation / Online Phase

Step 1: Parametric input states

$$\exists \mathbf{p}_v, v \notin [1, N_s]$$

Step 2: Time Integration of ROM

$$\forall t_i, i \in [0, N_t]$$

Step 3: Assemble matrices and evaluate Equations of Motion

$$\mathbf{M}_r(\mathbf{p})\ddot{\mathbf{u}}_r(t_i) + \mathbf{g}_{ri}(\mathbf{u}(t_i), \dot{\mathbf{u}}(t_i), \mathbf{p}) - \mathbf{f_r}(\mathbf{t_i}, \mathbf{p}) = \mathbf{0}$$

$$\mathbf{M}_\mathrm{r}(\mathbf{p}_j) \in \mathbb{R}^{r\times r}, \mathbf{g}_\mathrm{r}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right) \in \mathbb{R}^r, \mathbf{f}_\mathrm{r}(t, \mathbf{p}_j) \in \mathbb{R}^r$$

Nonlinear terms
*still **scale** with **full dimension***

Step 4: Compute Residual on Equations and "predict" correction

$$if \quad \mathbf{R}_i(\mathbf{u}_r(t_i)) > tol => \tilde{\mathbf{u}}_r(t_i)$$

Step 5: Employ updated solution to re-assemble nonlinear terms & matrices

$$\tilde{\mathbf{u}}(t_i) = \mathbf{V}\tilde{\mathbf{u}}_\mathbf{r}(t_i)$$

$$\tilde{\mathbf{u}}(t_i) => \mathbf{g}_i(\tilde{\mathbf{u}}(t_i), \dot{\tilde{\mathbf{u}}}(t_i), \mathbf{p})$$

$$\mathbf{g}_{ri} = \mathbf{V}^T\mathbf{g}_i$$

Problem Statement

Bottlenecks/Limitations

Step 2: Time Integration of ROM

$$\forall t_i, i \in [0, N_t]$$

Step 3: Assemble matrices and evaluate Equations of Motion

$$\mathbf{M}_r(\mathbf{p})\ddot{\mathbf{u}}_r(t_i) + \mathbf{g}_{ri}(\mathbf{u}(t_i), \dot{\mathbf{u}}(t_i), \mathbf{p}) - \mathbf{f}_\mathbf{r}(\mathbf{t_i}, \mathbf{p}) = \mathbf{0}$$

$$\mathbf{M}_\mathrm{r}(\mathbf{p}_j) \in \mathbb{R}^{r\times r}, \mathbf{g}_\mathrm{r}\left(\mathbf{u}(t), \dot{\mathbf{u}}(t), \mathbf{p}_j\right) \in \mathbb{R}^r, \mathbf{f}_\mathrm{r}(t, \mathbf{p}_j) \in \mathbb{R}^r$$

Nonlinear terms
*still **scale** with **full dimension***

Step 4: Compute Residual on Equations and "predict" correction

$$if \quad \mathbf{R}_i(\mathbf{u}_r(t_i)) > tol => \tilde{\mathbf{u}}_r(t_i)$$

Step 5: Employ updated solution to re-assemble nonlinear terms & matrices

$$\tilde{\mathbf{u}}(t_i) = \mathbf{V}\tilde{\mathbf{u}}_\mathbf{r}(t_i)$$

$$\tilde{\mathbf{u}}(t_i) => \mathbf{g}_i(\tilde{\mathbf{u}}(t_i), \dot{\tilde{\mathbf{u}}}(t_i), \mathbf{p})$$

$$\mathbf{g}_{ri} = \mathbf{V}^T\mathbf{g}_i$$

- The evaluation of the nonlinear terms still ***scales with the full order dimension.***
- For every solution increment we need to:
  - ***Project*** displ./vel. ***back to full-order***
  - Evaluate nonlinear terms
  - ***Update*** forces and stiffness matrix
  - **Project** updated matrices ***back to reduced-order*** coordinates.

*This **back-and-forth projection is a major computational bottleneck**.*

Especially in large scale systems
where time integration savings cannot outweigh
the projection & evaluation.

*To address this, we rely on **hyper-reduction, a second-tier approximation** of the nonlinear contributions.*

Problem Statement

Bottlenecks/Limitations

- **Back & forth projection** to update nonlinear terms **compromises efficiency**

- Hyper-reduction is introduced

  ⇒*Several alternatives available (ECSW, DEIM, GNAT, EQM)*

  ⇒***ECSW**: Sparse evaluation of full-order nonlinear terms based on energy contributions coupled with weighting scheme*

  ⇒***DEIM:** Additional POD-based reduction coupled with selection scheme*

  ✓ Hyper-reduction **is essential for efficiency**

❖ Introduces an additional source of error that usually outweighs the POD projection error

=> Accuracy bottleneck/threshold for the ROM

Step 5: Employ updated solution to re-assemble nonlinear terms & matrices

$$\tilde{\mathbf{u}}(t_i) = \mathbf{V}\tilde{\mathbf{u}}_\mathbf{r}(t_i)$$

$$\tilde{\mathbf{u}}(t_i) => \mathbf{g}_i(\tilde{\mathbf{u}}(t_i), \dot{\tilde{\mathbf{u}}}(t_i), \mathbf{p})$$

$$\mathbf{g}_{ri} = \mathbf{V}^T\mathbf{g}_i$$

Working Approaches

Reconstruction Error Minimization

Tackle projection-based reduction error

=> Replace POD with AE-driven process

Argumentation:
POD is a linear operator / Accuracy relies on linearization assumptions

- *Employ mapping techniques based on nonlinear (feature) transformations*
- *Explore potential nonlinear kernels*
- *Check if orthogonality conditions need to be retained*

=> Improve pROM accuracy

- Initial approaches yield questionable results
- Additional bottleneck on how to propagate ROM dynamics after projection

=> Further and deeper research is needed

Working Approaches

Data-driven Mapping for Hyper-Reduction

Address ROM's nonlinear mapping limitations

⇒ ***Replace* Hyper-Reduction with *data-driven surrogates***

Argumentation:
Hyper-Reduction guarantees efficiency however is the largest error source of the pROM

- *Couple POD basis assembly process with data-driven method to learn the nonlinear mapping directly in ROM coordinates*
- *Every iteration of the training state contributes nonlinear mapping training data*
  => Thousands of data available for a single training realization
- *Parametric dependencies on the nonlinear law might be explored as a next step*

+ Potentially substantial improvement in efficiency
+ Achieve real-time evaluations
- Potential trade-off with accuracy
- Parametric dependency is challenging

Numerical Validation

Case study description

Two-story shear frame with nodal connections modeled employing hysteretic links

Multisine **stochastic ground motion excitation**

Hysteretic links response model

- *Total restoring force:*

$$\mathbf{R} = \mathbf{R}_{linear} + \mathbf{R}_{hysteretic} = \alpha k \mathbf{u} + (1-\alpha) k \mathbf{z}$$

- *Bouc-Wen equation with degradation/deterioration effects:*

$$\dot{\mathbf{z}} = \frac{A\dot{\mathbf{u}} - \nu(t)(\beta|\dot{\mathbf{u}}|\mathbf{z}|\mathbf{z}|^{w-1} - \gamma\dot{\mathbf{u}}|\mathbf{z}|^{w})}{\eta(t)}$$

$$\nu(t) = 1.0 + \delta_\nu \epsilon(t), \quad \eta(t) = 1.0 + \delta_\eta \epsilon(t), \quad \epsilon(t) = \int_0^t \mathbf{z}\dot{\mathbf{u}} \delta t$$

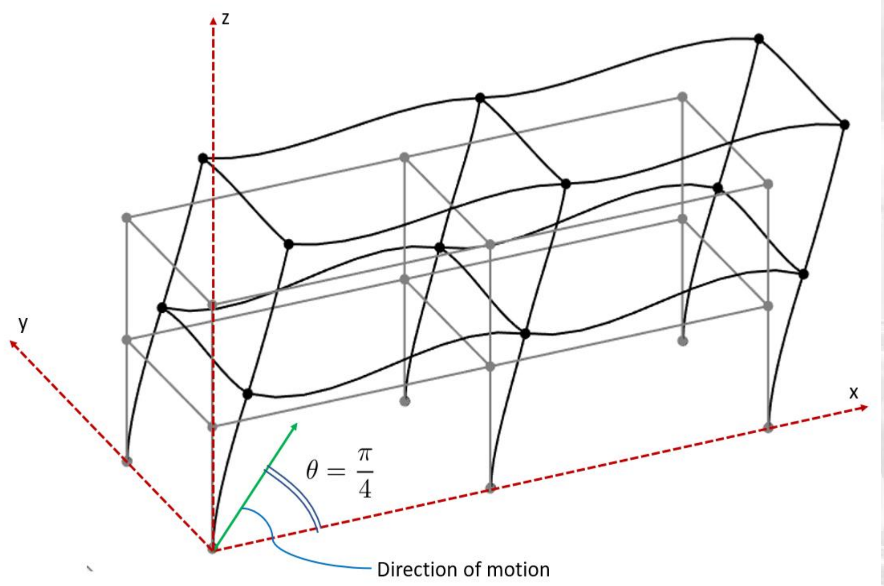
Characteristics of the Bouc-Wen links:

β, γ, A, w : *Smoothness and shape of hysteresis curve*

δ_ν, δ_η : *Degradation/Deterioration effects*

a, k : *Linear/Hysteretic contribution weighting*

=> Parametric dependencies of the hysteretic links



Benchmark example featured in:

- Vlachas K. et al. "A local basis approximation approach for nonlinear parametric model order reduction." *Journal of Sound and Vibration* 502 (2021): 116055.
- Vlachas K. et al. " Two-story frame with Bouc-Wen hysteretic links as a multi-degree of freedom nonlinear response simulator." Workshop on non-linear system identification benchmarks (2021)

Numerical Validation

Data-driven Mapping for Hyper-Reduction

Address ROM's nonlinear mapping limitations
$\Rightarrow$ ***Replace* Hyper-Reduction with *data-driven surrogates***

Argumentation:
Hyper-Reduction guarantees efficiency however is the largest error source of the pROM

Detailed Task Formulation

Input:
- *Reduced-Order Displacements in current iteration (and previous ones)* $\rightarrow \mathbf{U} \in \mathrm{R}^{16,(t-k):t}$
- *Reduced-Order Force terms in previous iteration(s)* $\rightarrow \mathbf{F} \in \mathrm{R}^{16,(t-k):(t-1)}$
- *Reduced-Order Stiffness terms in previous iteration(s)* $\rightarrow \mathbf{K} \in \mathrm{R}^{16\times16,(t-k):(t-1)}$

Output:
- *Reduced-Order Force terms in current iteration* $\rightarrow \mathbf{F} \in \mathrm{R}^{16,1}$
- *Reduced-Order Stiffness terms in current iteration* $\rightarrow \mathbf{K} \in \mathrm{R}^{16\times16,1}$

Numerical Validation

Network details and potential improvements

Neural Network Details

- 1D (Temporal) Convolution layers
- Spatial local correlations ignored => Assumes only time correlations
- Activation function: CELU, Tanh
- Train / Validation ratio: 0.75
- Temporal width: 8
- Dropout: 0.01
- Scaling: Normalization/ Minimum=0, Maximum=1
- Kernel size = 3
- Batch size = 64 / Epochs = 10000
- Adaptive learning rate with overfitting patience enabled
- Network size: ~500,000 parameters

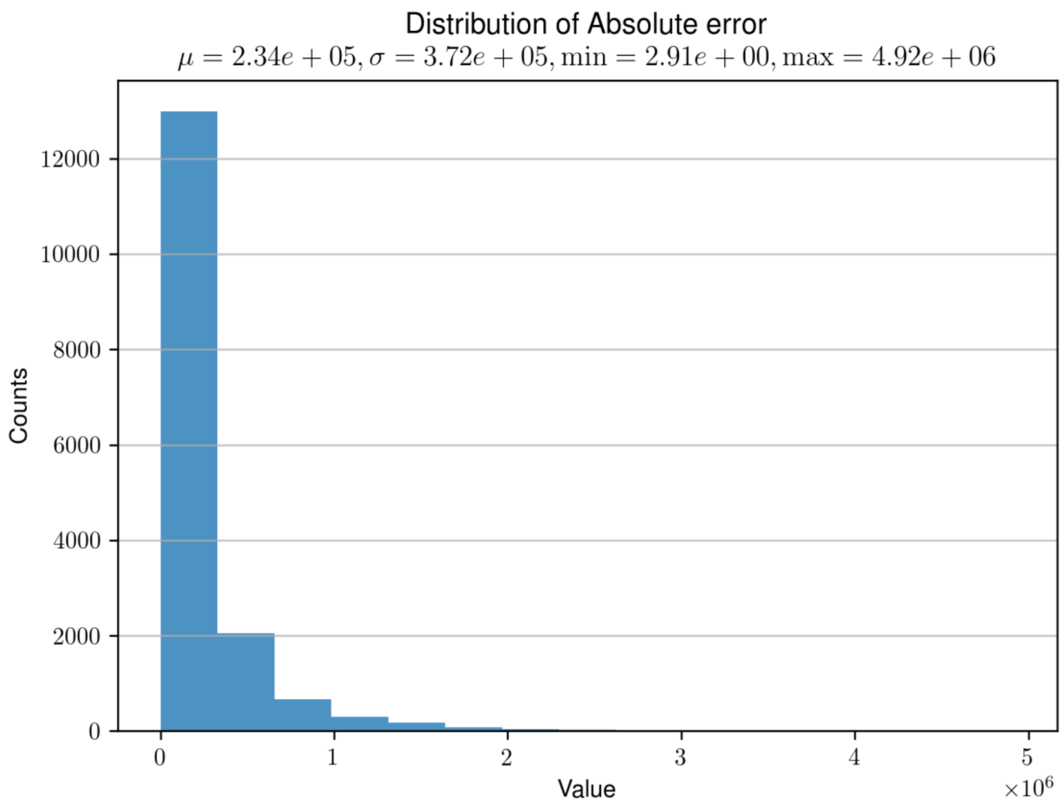
Potential improvements:

- ✓ Increase temporal width parameter
- ✓ Improve post-processing and error diagnostics
- ✓ Explore the trade-off between deep CNNs and efficiency
- ✓ Explore MLP network potential

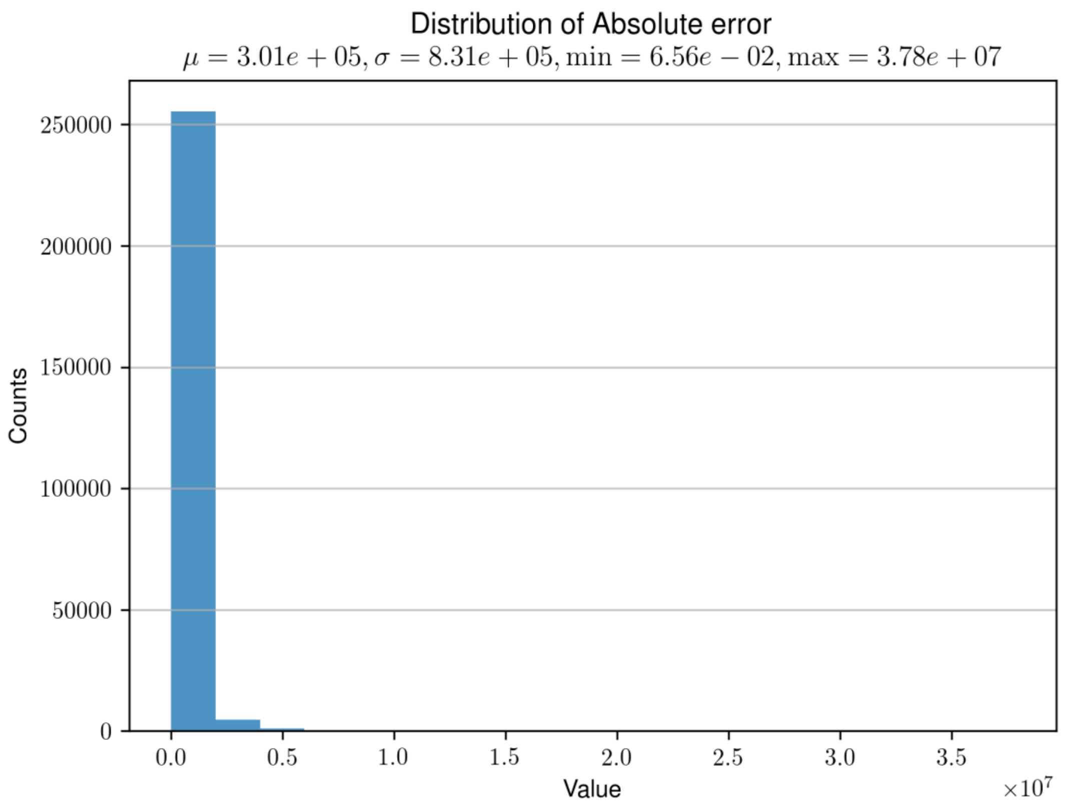
Numerical Validation

Absolute errors distribution

Internal Forces F



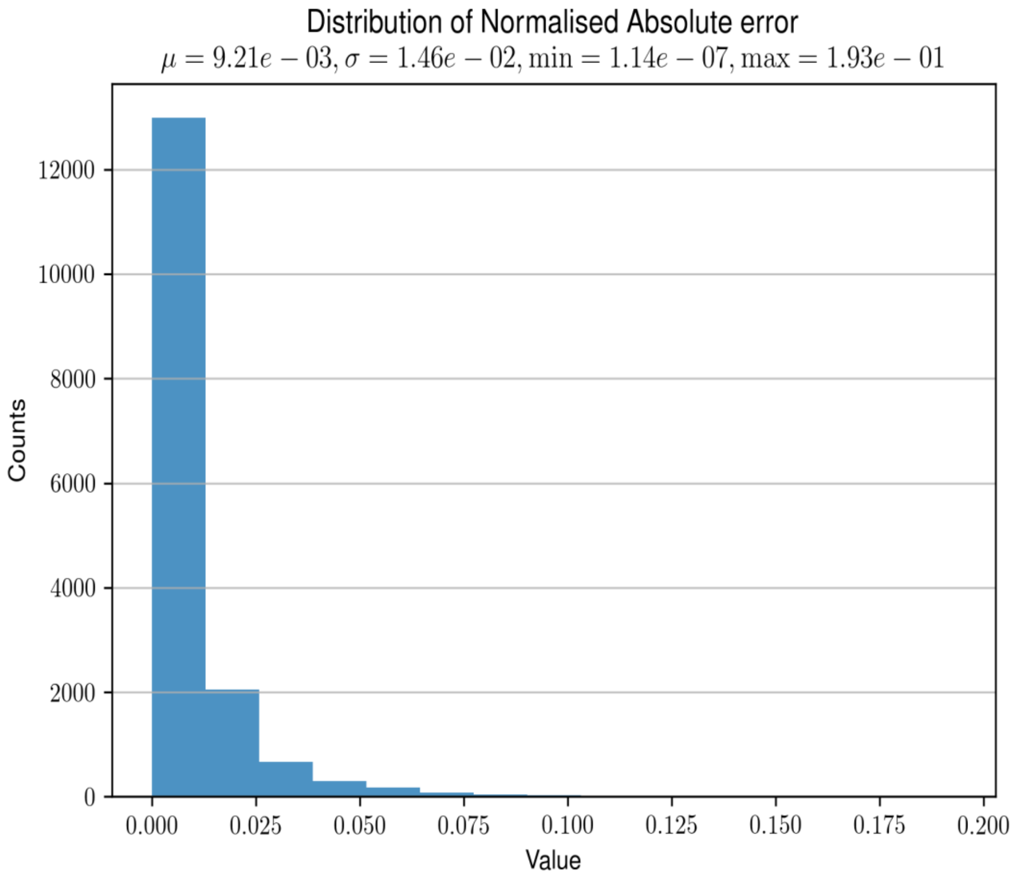
Stiffness Matrix K



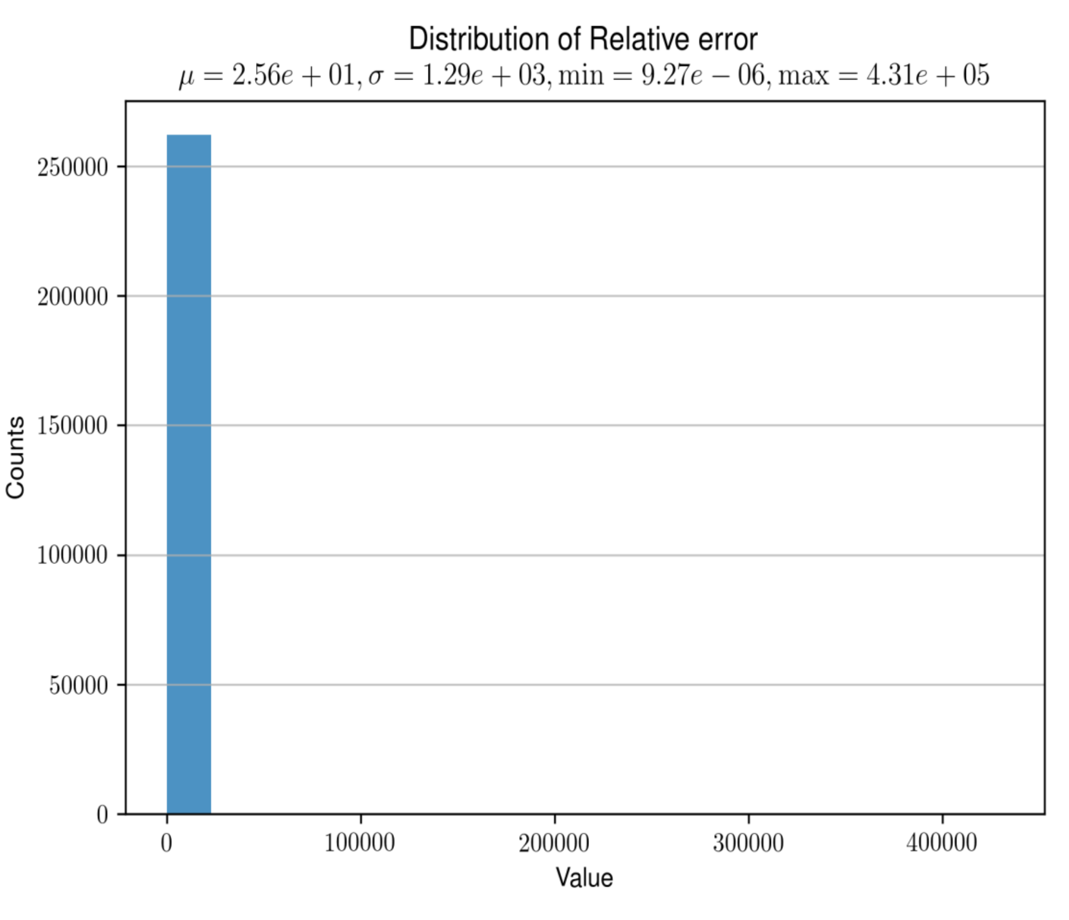
Numerical Validation

Relative errors distribution

Internal Forces F

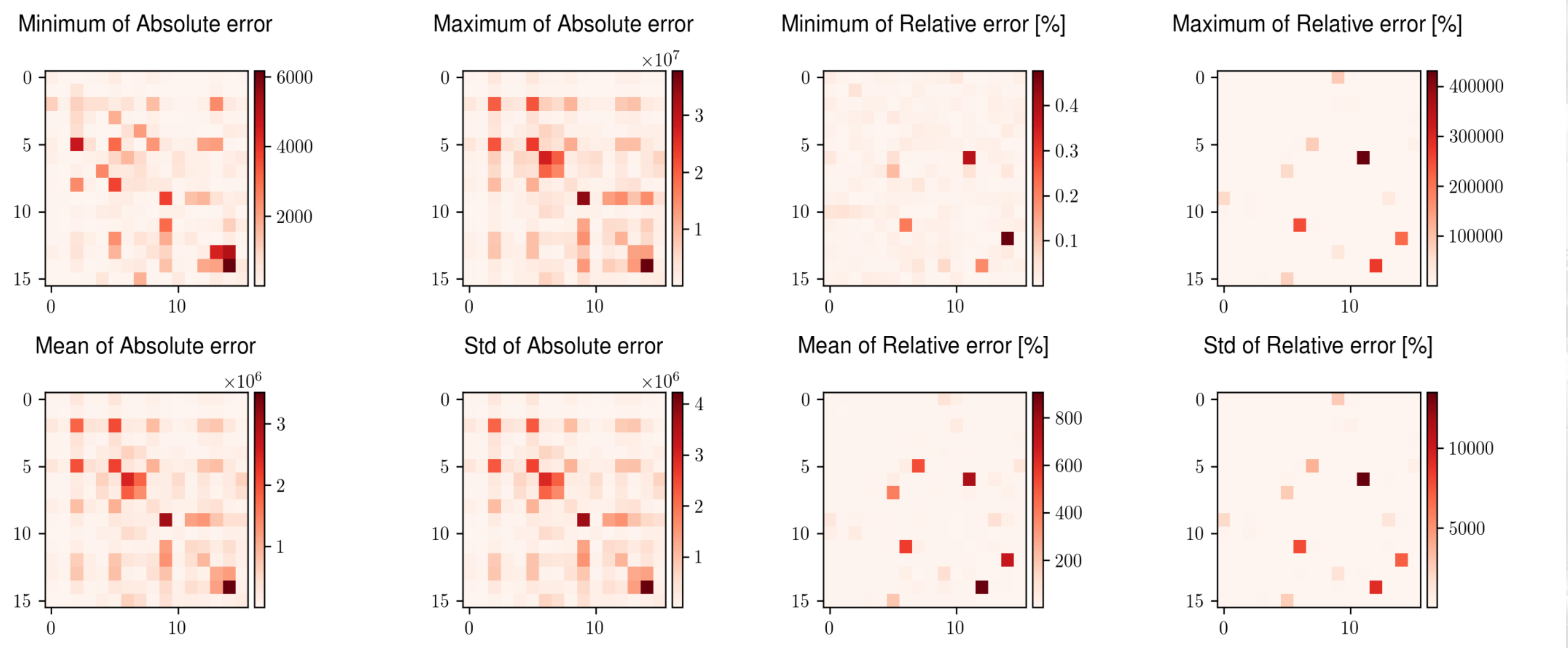


Stiffness Matrix K



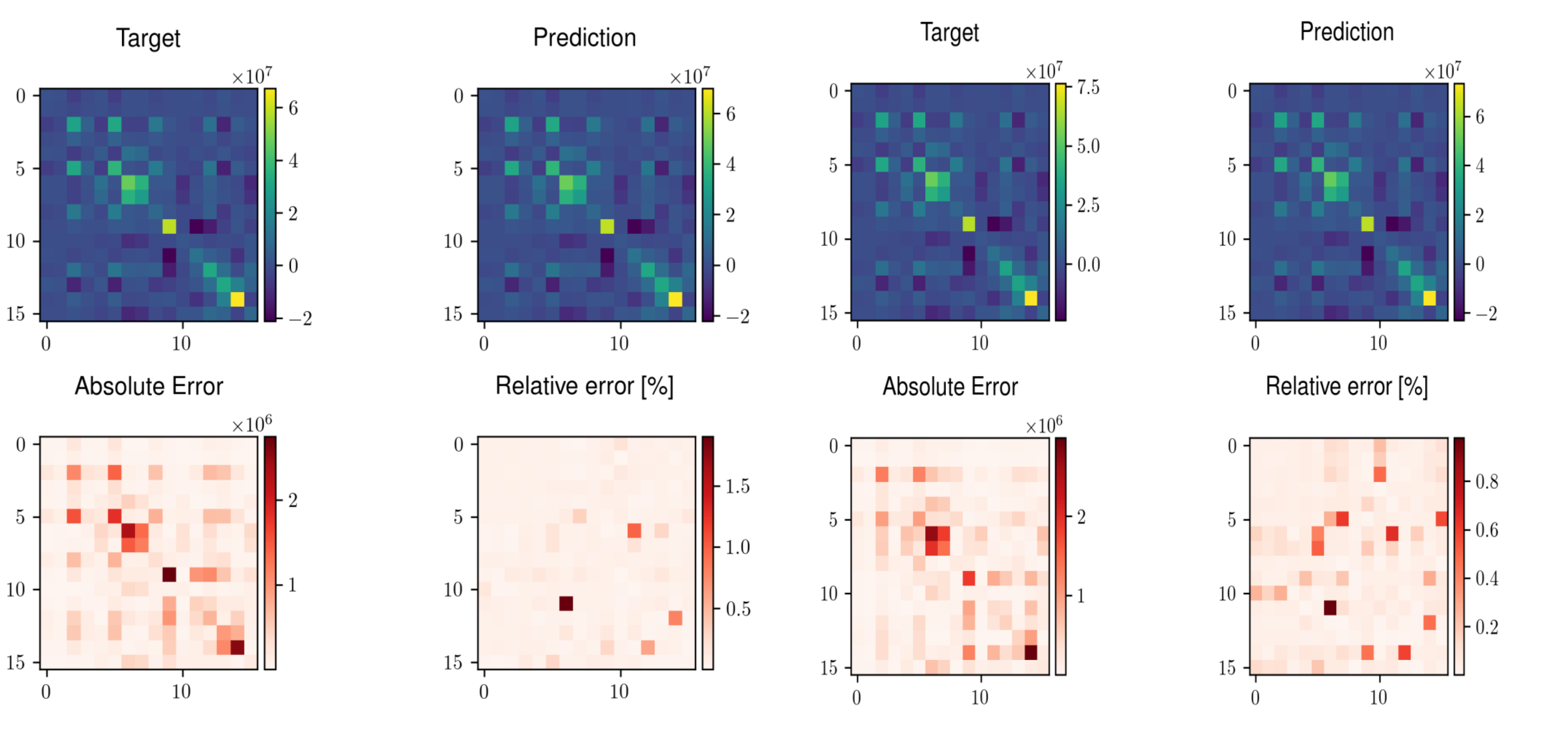
Numerical Validation

Error visualization in stiffness matrix



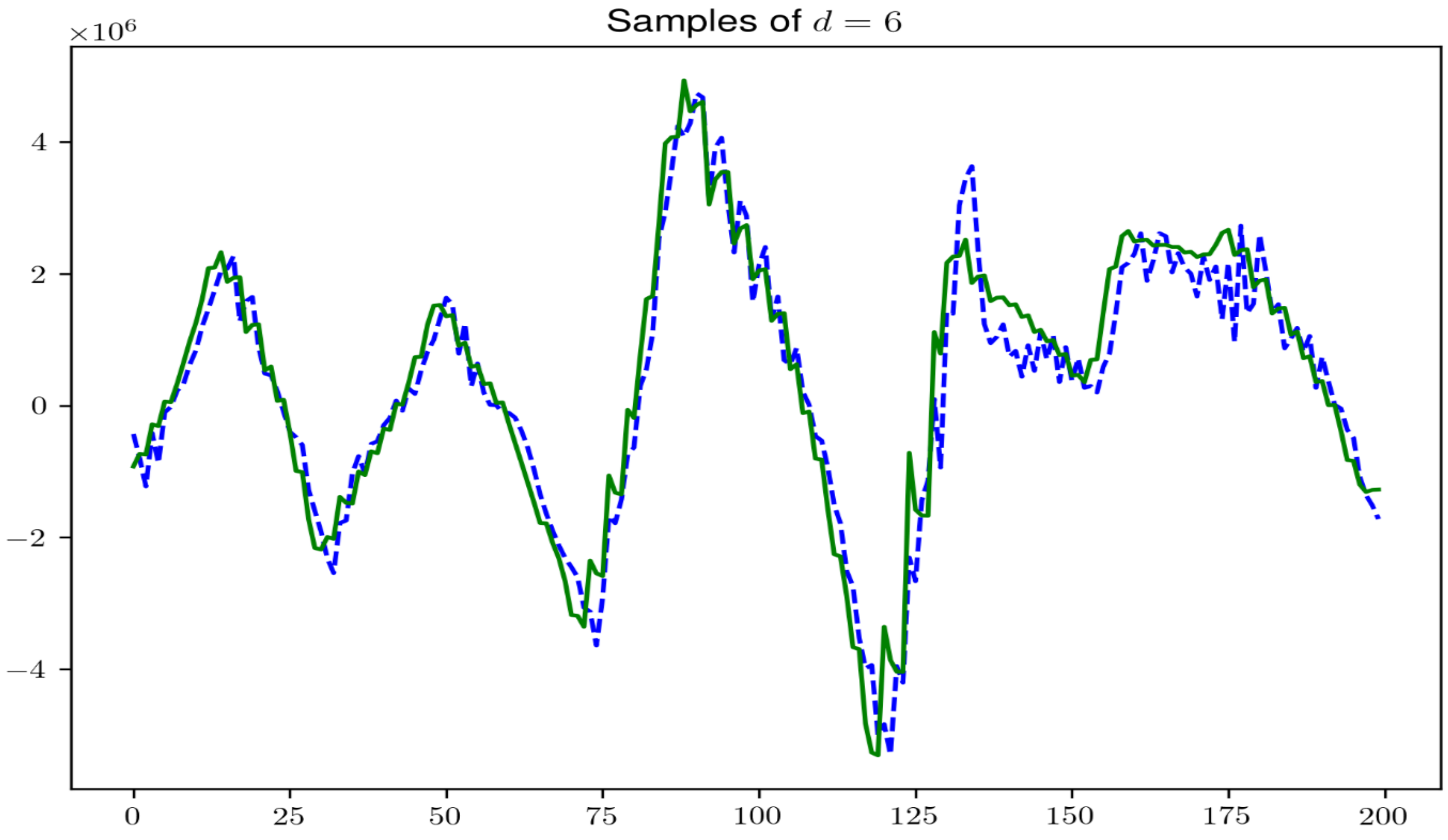
Numerical Validation

Prediction visualization in stiffness matrix



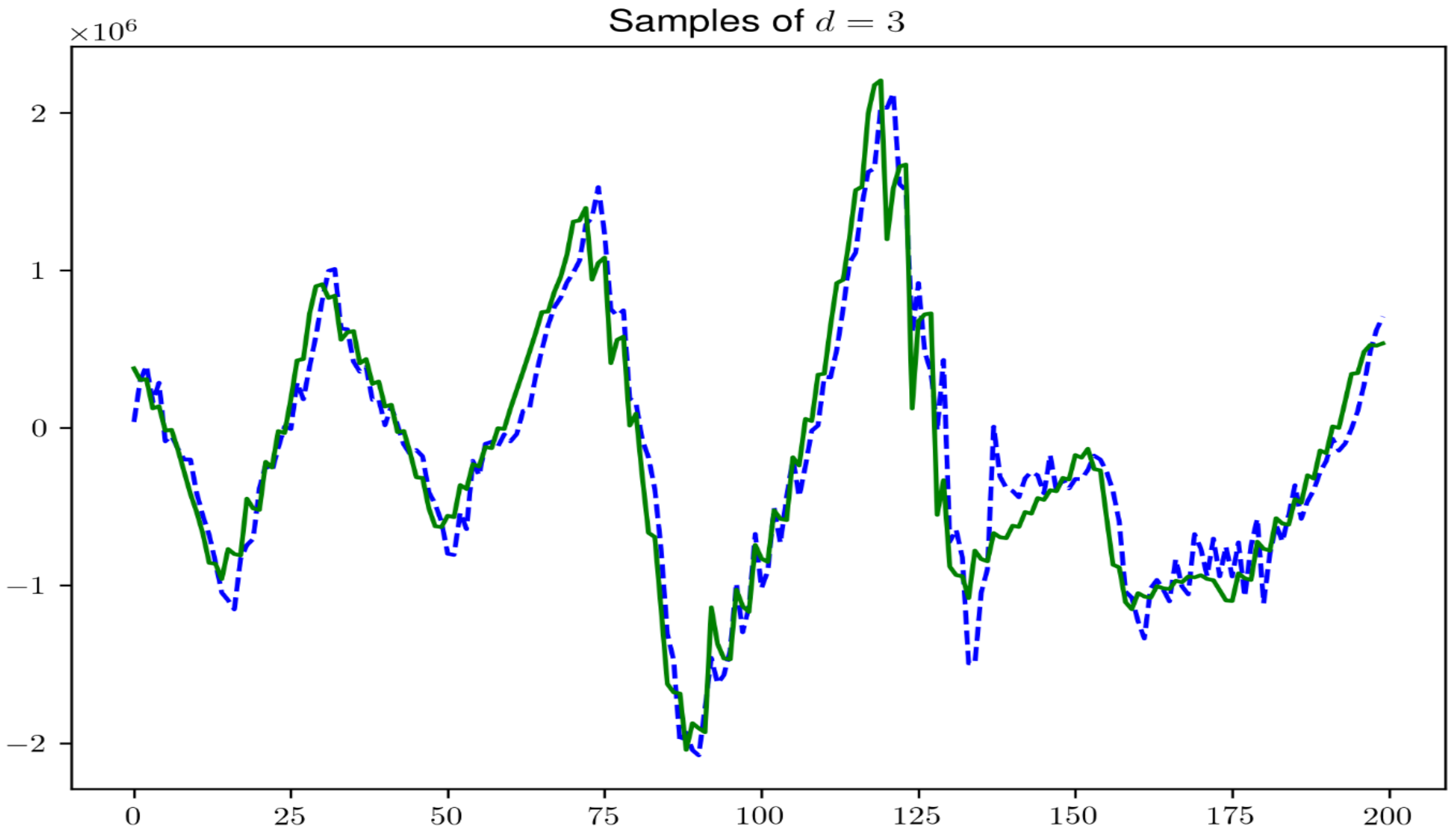
Numerical Validation

Prediction visualization in internal forces



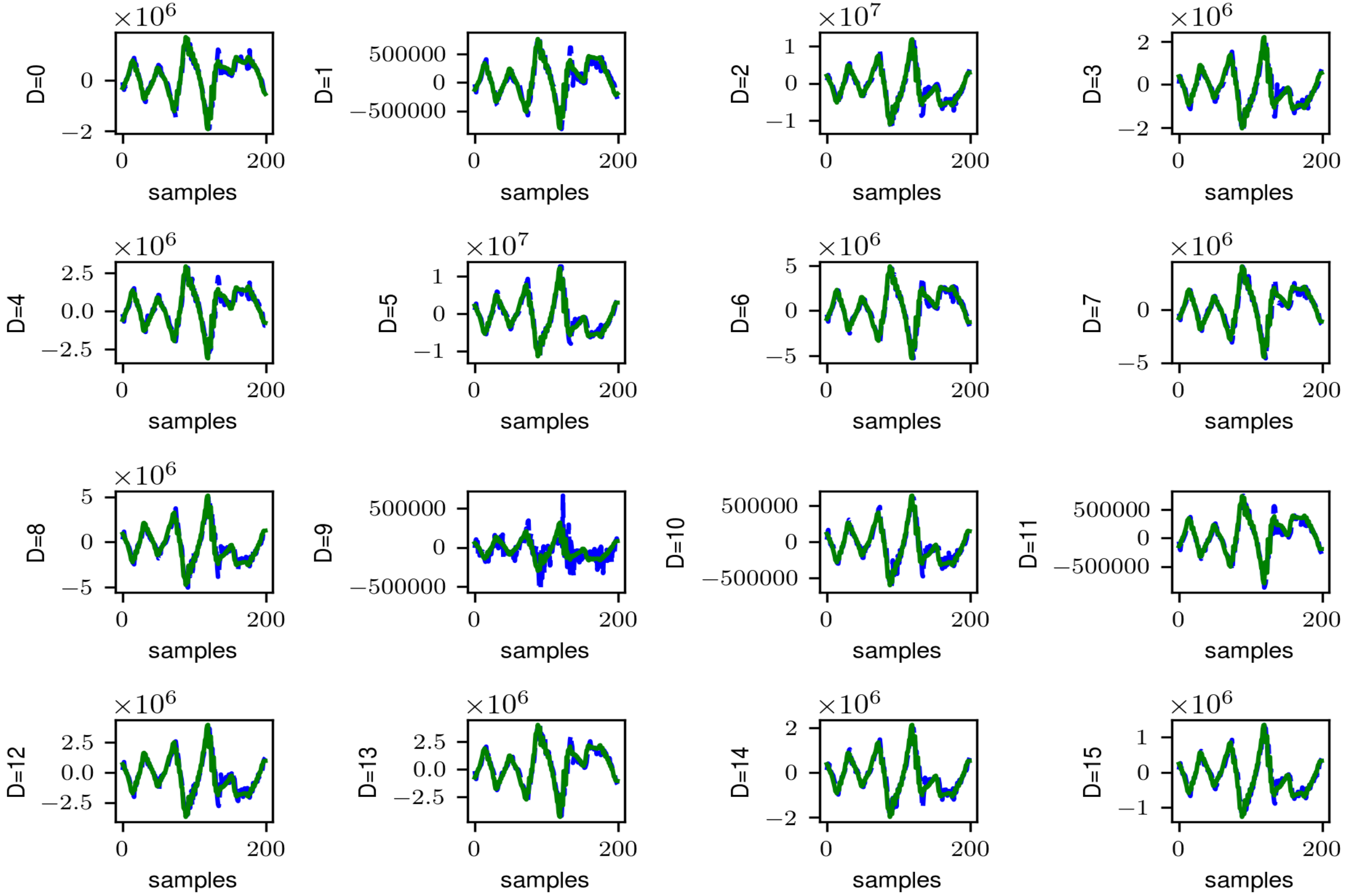
Numerical Validation

Prediction visualization in internal forces



Numerical Validation

Prediction visualization in internal forces



Concluding Remarks

ROM Performance and Generalization / Outlook

Hyper-Reduction Based POD-ROM vs Data-driven assisted POD-ROM

Accuracy performance of the pROM

*Median Error = **~2%** vs **~20%***

*Max Error = **5%** vs **25%***

Efficiency performance of the pROM (Evaluation in GPU)

*Average speed-up factor = **~7** vs **~50***

Parametric dependency treatment

***One hyper-reduction scheme** per POD cluster / Approximately **3 NN models** per cluster*

Suffers from ***error propagation***

=> Improvement by training in sequences

Remarkably ***greater efficiency***

=> Explore deeper networks

Generalization needs improvement

=> Include dependency as input variable

=> Ongoing Research