Non-linear reduction for data-driven non-intrusive reduced-order models with parallel autoencoders

Karim Yehia Aly, Nicola Demo, Giovanni Stabile and Gianluigi Rozza

Mathematics Area, mathLab, SISSA, International School of Advanced Studies, Trieste, Italy

Introduction

The completely **data-driven non-intrusive ROM** using **linear reduction** methods such as the **POD** fails for problems characterised by a slow decay of the Kolmogorov N-width. In this ongoing research, we are exploring the usage of a **non-linear reduction** approach such as the **autoencoder** neural network instead of the linear POD, to tackle various types of problems within the **HPC** framework, taking the advantage of **data parallelism** techniques.

1 - The traditional data-driven non-intrusive ROM

Using Proper Orthogonal Decomposition with Interpolation (PODI):

Starting from a **parametric** geometrical model; if we have a database of **parameter** values $\mathbf{\Xi} = [\boldsymbol{\mu}_1 \dots \boldsymbol{\mu}_N]$ and a database of **snapshots** $\mathbf{\Theta} = [u(\boldsymbol{\mu}_1) \dots u(\boldsymbol{\mu}_N)]$ as in [1], we can apply the **Singular Value Decomposition (SVD)** to the **snapshots matrix** $\mathbf{\Theta}$:

$$\mathbf{\Theta} = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Phi}^T$$

where $\mathbf{\Psi}$ and $\mathbf{\Phi}$ are the left and right singular vectors matrices of $\mathbf{\Theta}$, and $\mathbf{\Sigma}$ is the diagonal matrix containing the singular values in decreasing order. The columns of Ψ are called the **POD modes** ψ and the **modal coefficients** α can be obtained through $\alpha = \mathbf{\Psi}^T\mathbf{\Theta}$ as mentioned in [2].

Then the **reduced solution** of the problem can be viewed as a **linear combination** of the POD modes ψ multiplied by the modal coefficients α as follows:

$$u^N = \sum_{i=0}^{N} \alpha_i \psi_i$$

Q: How to get value of the modal coefficients α for each new parameter?

In the offline phase: We **compute the full-order solution** using the high-fidelity solver for some parameter points $\boldsymbol{\mu}_k \in \mathbf{\Xi}$. Since the high fidelity solution $u(\boldsymbol{\mu}_k)$ and the reduced solution $u^N(\boldsymbol{\mu}_k)$ are assumed to be equal by construction, we have the following:

$$\forall \boldsymbol{\mu}_k \in \mathbf{\Xi} : u(\boldsymbol{\mu}_k) = u^N(\boldsymbol{\mu}_k) = \sum_{i=0}^{N} \alpha_i(\boldsymbol{\mu}_k)\psi_i$$

Now for the set of parameters $\boldsymbol{\mu}_k$, we can compute the corresponding coefficients $\alpha(\boldsymbol{\mu}_k)$.

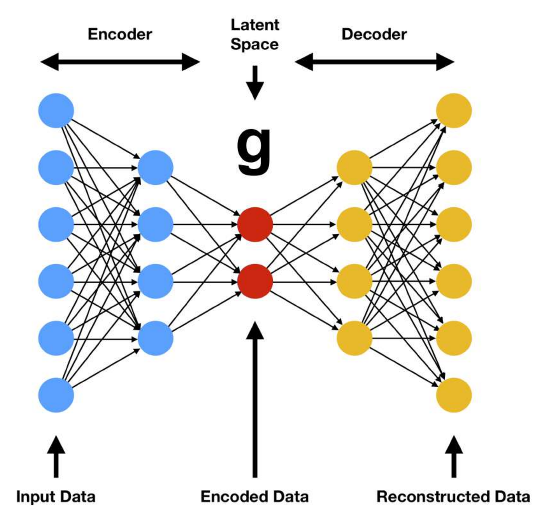
In the online phase: Since we obtained pairs of $(\boldsymbol{\mu}_k, \alpha(\boldsymbol{\mu}_k))$, for each new parameter $\boldsymbol{\mu}_{\text{new}}$, we **interpolate the previously computed coefficients** $\alpha(\boldsymbol{\mu}_k)$ to find the new coefficients $\alpha(\boldsymbol{\mu}_{\text{new}})$. The new reduced solution can be obtained by:

$$u^N_{new} = \sum_{i=0}^{N} \alpha_i(\boldsymbol{\mu}_{new})\psi_i$$

2 - From Linear to non-linear reduction

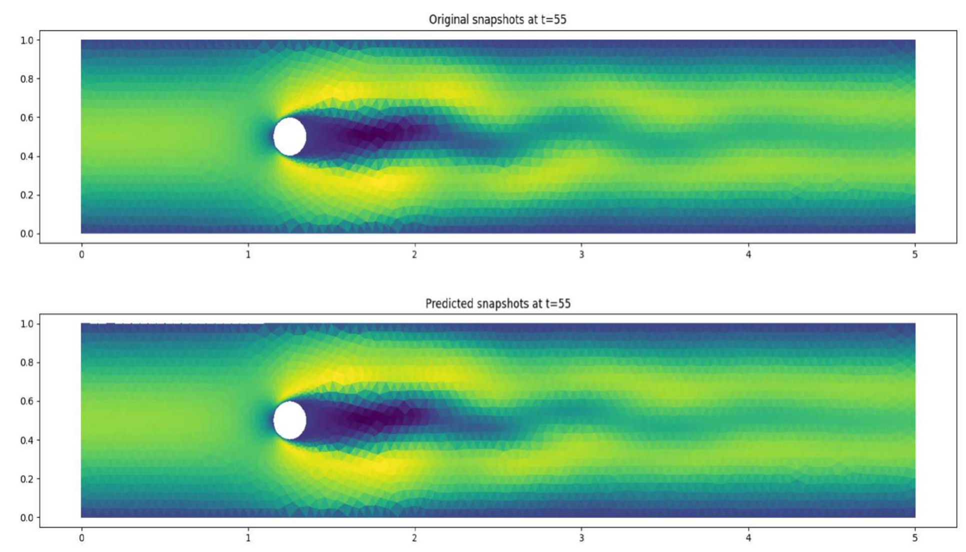
Linear reduction: SVD $\rightsquigarrow$ $u(\boldsymbol{\mu}) \approx \sum_{i=0}^{N} \alpha_i(\boldsymbol{\mu})\psi_i$

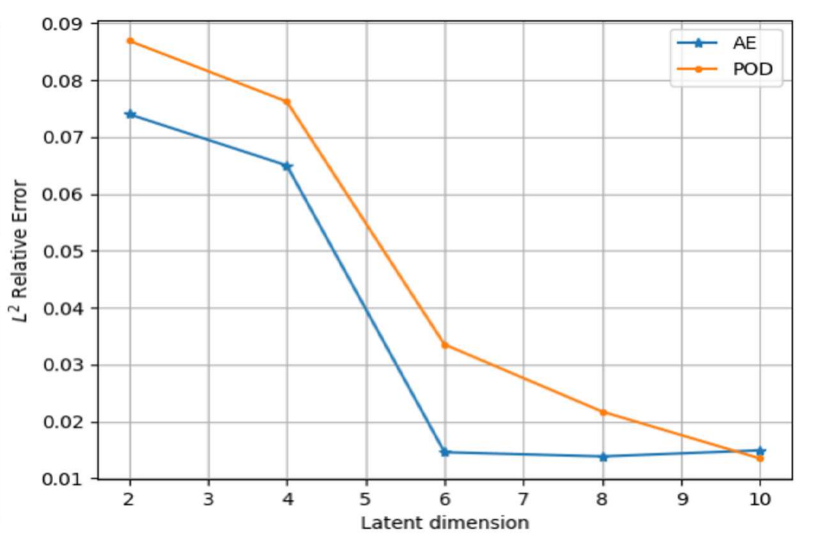
Non-linear activation $\left(\sum_{i=1}^{N}(x_i w_i) + \text{bias}\right)$



Non-linear reduction: AE $\rightsquigarrow$ $u(\boldsymbol{\mu}) \approx f(\boldsymbol{g})$

3 - Preliminary results



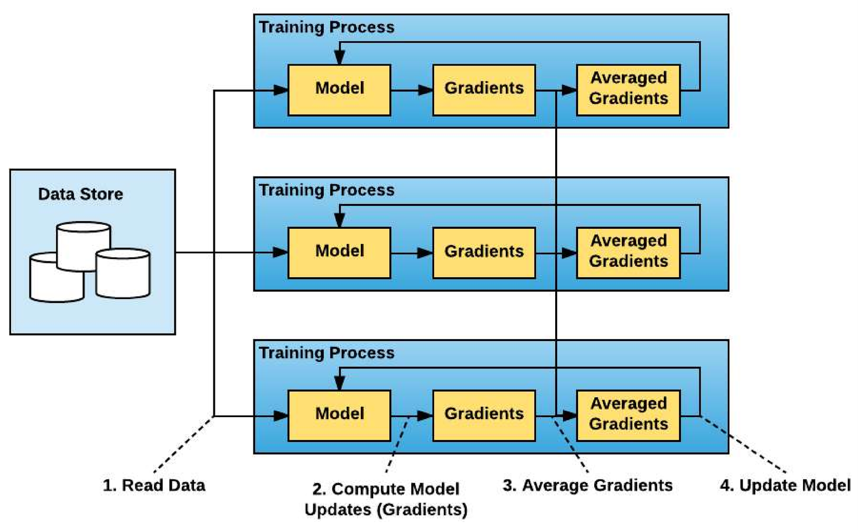


Note:
The non-linear reduction can achieve **smaller** solution manifolds with **better** accuracy.

4 - Data parallelism

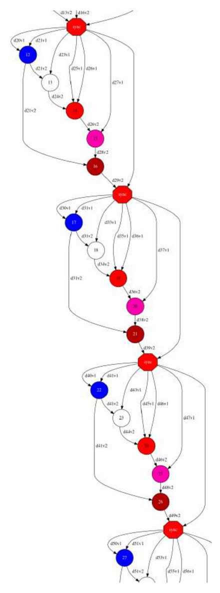
Input of large-sized batches from FOM:

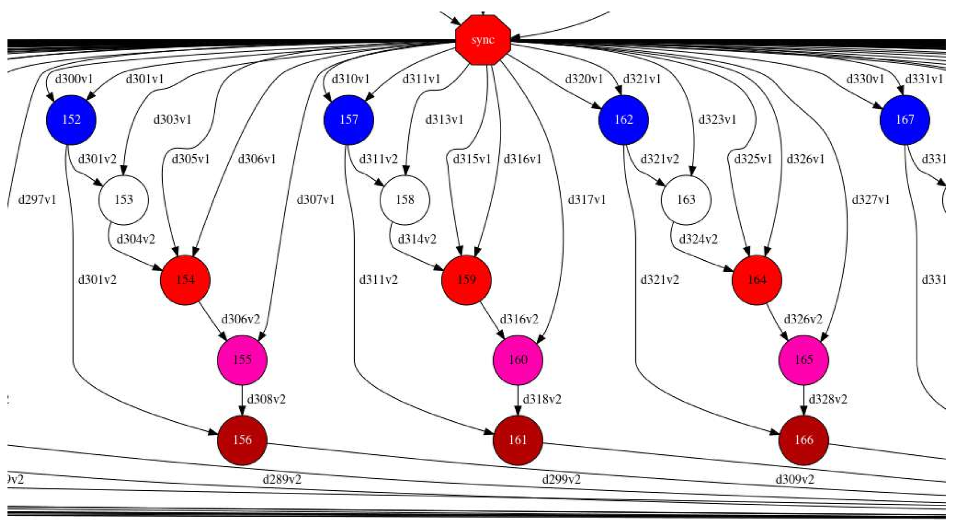
1. **Divide** into **mini-batches** and **distribute** them across processing units.
2. **Replicate the full model** on each processing units to treat a mini-batch.
3. Perform **training locally** and **synchronise local gradients**.
4. The **average** of the local gradients is used to **update** the local models.



5 - Parallel execution

Simultaneous execution of the same functions for multiple predictions or error calculations:





6 - Computational science and engineering softwares: `mathlab.sissa.it/cse-software`

EZyRB
`github.com/mathLab/EZyRB`
`mathlab.github.io/EZyRB`

EZyRB is a python library for data-driven (non-intrusive) model order reduction with linear and non-linear reduction and different approximation methods.

COMPSs
`github.com/bsc-wdc/compss`
`compss-doc.readthedocs.io/en/stable/index.html`

COMPSs is a programming model to ease the development of applications for distributed infrastructures, such as Clusters, Clouds and Containerised Platforms.

EDDL
`github.com/deephealthproject/eddl`
`deephealthproject.github.io/eddl/index.html`

EDDL is an optimised tensor library for distributed deep learning with hardware transparency support for CPUs, GPUs and FPGAs.

References and Acknowledgements

[1] F. Salmoiraghi, A. Scardigli, H. Telib, and G. Rozza, "Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation," *International Journal of Computational Fluid Dynamics*, vol. 32, no. 4-5, pp. 233–247, 2018.

[2] F. Garotta, N. Demo, M. Tezzele, M. Carraturo, A. Reali, and G. Rozza, "Reduced order isogeometric analysis approach for pdes in parametrized domains," in *Quantification of Uncertainty: Improving Efficiency and Technology*, pp. 153–170, Springer, 2020.

[3] M. Tezzele, N. Demo, A. Mola, and G. Rozza, "An integrated data-driven computational pipeline with model order reduction for industrial and applied mathematics," in *Novel Mathematics Inspired by Industrial Challenges*, pp. 179–200, Springer, 2022.

[4] N. Demo, M. Tezzele, A. Mola, and G. Rozza, "A complete data-driven framework for the efficient solution of parametric shape design and optimisation in naval engineering problems," *arXiv preprint arXiv:1905.05982*, 2019.

This work is developed as a part of the **eFlows4HPC** project in cooperation with 16 partners from seven different countries with the aim to deliver a workflow software stack and an additional set of services to enable the integration of HPC simulations and modelling with big data analytics and machine learning in scientific and industrial applications. The software stack will allow for the creation of innovative adaptive workflows that efficiently use computing resources considering novel storage solutions.