

Physics-AI model reduction for flood dynamics simulation

By M. Allabou ¹, R. Bouclier ¹, J. Monnier ¹ with P.A. Garambois ²

¹INSA - Institut de Mathématiques de Toulouse (IMT) - Institut Clément Ader (ICA)

²INRAE Aix-en-Provence.



Outline

- 1 Problem statement
- 2 POD-NN based methods
- 3 Results
- 4 Conclusion

1 Problem statement

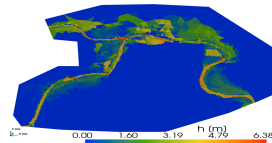
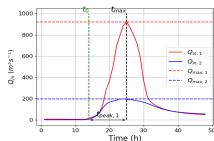
2 POD-NN based methods

3 Results

4 Conclusion

A real-like test case (Aude River, south-east France).

Case data provided by Garambois [Villenave et al., SHF 2023]. Dataset: Vigicrues, SCHAPI-DGPR, Météo-France



Plot: water elevation h [m].

Simulations using a 2D shallow-water (SW) flow model.
Finite-volume scheme, DassFlow software.

Mesh size		HR	Reduced Order Model (ROM)
$\approx 52\,000$ cells	CPU time	4 499 s	10 s
	Memory	1 219 Mo	738 Mo

Computer: standard 11th Gen Intel(R) Core(TM) i9-11950H, 2.60 GHz, 32 GB RAM. Single-core computations.

The **ROM** has been built with **inflow discharges as input parameters**.

General framework: reduction of μ -parameterized models

Given the parameter μ , find the state of the system $\mathbf{u}(\mu; x, t)$ such that:

$$\partial_t \mathbf{u}(\mu; x, t) = F(\mu; \mathbf{u}(\mu; x, t)) \quad \text{in } \Omega \times (0, T)$$

+ B.C.(μ) + I.C.

▷ For the 2D SW model, $\mathbf{u} = (h, q_x, q_y)$ ($[m]$, $[m^2/s]$, $[m^2/s]$)
 μ = parameterization of $Q_{in}(t)$.

▷ Classical numerical solvers (e.g. Finite Volumes) $\rightarrow$ High Resolution (HR) solution.
 $\hookrightarrow$ CPU-time and memory consuming.

▷ This study presents Reduced Order Models (ROMs) methods, applied to the 2D SW equations parametrized by the inflow discharge.

Offline-online strategy

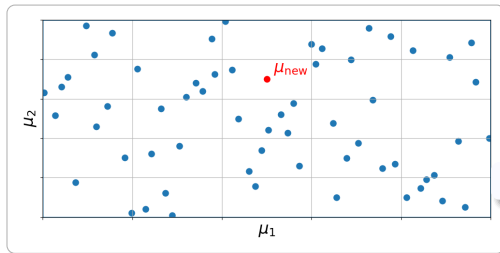
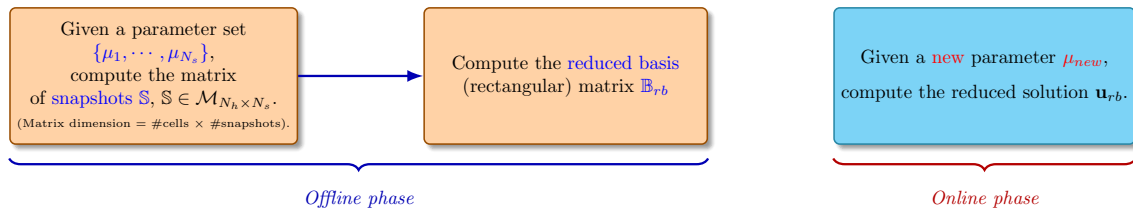
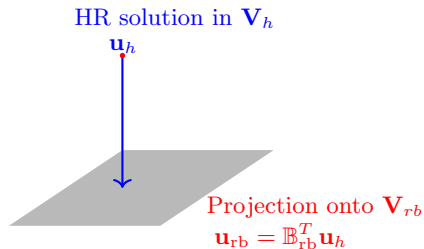


Fig.: $N_\mu = 2$

Blue points are the snapshots (HR solutions)

$N_\mu \approx 5 - 10$ at most, to be computationally tractable.

Projection of the High Resolution (HR) solutions $\mathbf{u}_h$ onto the reduced space V_{rb}



- $V_{rb} \subset V_h$
 $\dim(V_{rb}) \ll \dim(V_h)$

- Projection error:

$$\mathbf{e}_h = \mathbf{u}_h - \mathbb{B}_{rb} \mathbb{B}_{rb}^T \mathbf{u}_h$$

The standard POD based method

▷ Step 1: build the **snapshot matrix** $\mathbb{S} = [\mathbf{u}_h(\mu_1) \mid \cdots \mid \mathbf{u}_h(\mu_{N_s})] \in \mathcal{M}_{N_h \times N_s}$.

= HR solutions matrix of dimension $\# \text{cells} \times \# \text{snapshots}$.

▷ Step 2: the **reduced basis** (rb) matrix $\mathbb{B}_{rb}$ is constructed as $\mathbb{B}_{rb} = \text{POD}(\mathbb{S}^T \mathbb{S})$. $\mathbb{B}_{rb} \in \mathcal{M}_{N_h \times N_{rb}}$.

⇒ **Optimal representation of the snapshots** (in L^2 -norm).

↪ Efficient reduction method for linear PDEs (combined with Empirical Interpolation type methods for non-affine dependency wrt μ). However, **inaccurate - inefficient for non-linear PDEs**.

1 Problem statement

2 POD-NN based methods

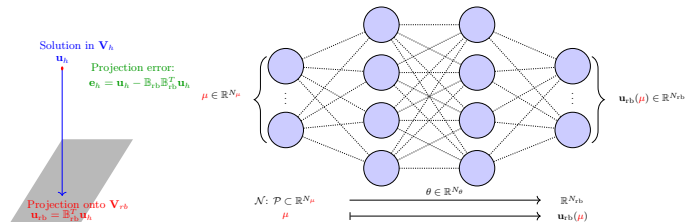
3 Results

4 Conclusion

The POD-Neural Network (NN) method as proposed in [Hesthaven-Ubbiali, JCP 2018]

- 1) Given the snapshot set $\mathbb{S}$, $\mathbb{B}_{rb}$ is built as $\mathbb{B}_{rb} = \text{POD}(\mathbb{S}^T \mathbb{S})$
 $\hookrightarrow$ **POD-based reduced basis.**
- 2) **Trick to extend the method to non-linear PDEs:**
 a NN is used to learn the snapshots coordinates in the reduced POD basis.

$$\boxed{\begin{array}{lll} \mathcal{N}^\theta & : \mathcal{P} & \rightarrow \mathbb{R}^{N_{rb}} \\ & \mu & \mapsto \mathbb{B}_{rb}^T \mathbf{u}_h(\mu) \end{array}}$$



The N_s snapshots $\mathbf{u}_h(\mu_s)$ (= the HR solutions) are learned as $\min_{\theta} J(\theta)$,

$$J(\theta) = \frac{1}{N_s} \sum_{s=1}^{N_s} \|\mathcal{N}(\theta)(\mu_s) - \mathbb{B}_{rb}^T \mathbf{u}_h(\mu_s)\|_2^2$$

Enriched version: the Error-Aware (EA) POD-NN method

proposed in [Allabou et al., CMAME 2024] (Allabou's PhD 2026)

Given N_s snapshots (HR space-time dependent solutions) $\mathbf{u}_h(\mu_s)$, $1 \leq s \leq N_s$,

- the previous NN, now denoted by $\mathcal{N}_1^\theta$, is used to estimate the same mapping as before:

$$\begin{aligned} \mathcal{N}_1^\theta &: \mathcal{P} \rightarrow \mathbb{R}^{N_{rb}} \\ \mu &\mapsto \mathbf{u}_{rb}(\mu) = \mathbb{B}_{rb}^T \mathbf{u}_h(\mu) \end{aligned}$$

- a second NN is used to estimate the **snapshots projection error** as:

$$\begin{aligned} \mathcal{N}_2^\theta &: \mathcal{P} \rightarrow \mathbb{R}^{N_{er}} \\ \mu &\mapsto \mathbf{u}_{er}(\mu) = \mathbb{B}_{er}^T \mathbf{e}_h(\mu) \end{aligned}$$

with $\boxed{\mathbf{e}_h(\mu) = \mathbf{u}_h(\mu) - \mathbb{B}_{rb} \cdot \mathbb{B}_{rb}^T \mathbf{u}_h(\mu)}.$

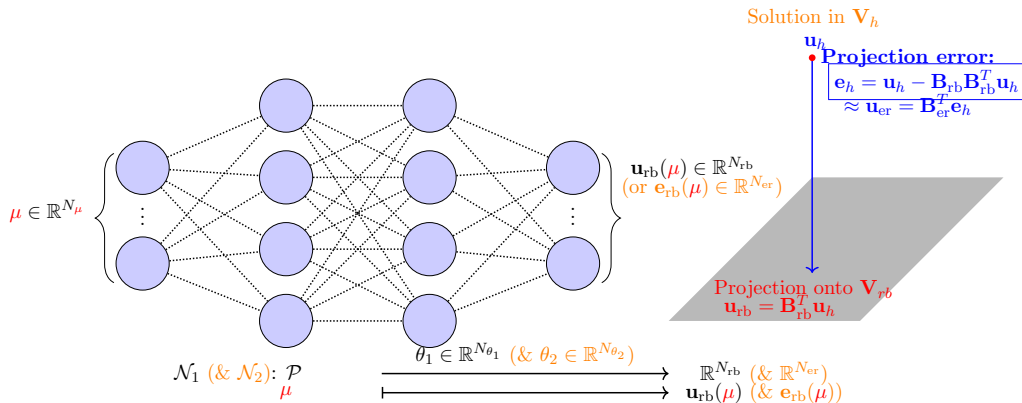
$\Rightarrow$ Matrix of errors: $\mathbb{E} = [\mathbf{e}_h(\mu_1) | \dots | \mathbf{e}_h(\mu_{N_s})] \in \mathcal{M}_{N_h \times N_s}.$

Next, we build: $\mathbb{B}_{er} = \mathbf{POD}(\mathbb{E}^T \mathbb{E})$, $\mathbb{B}_{er} \in \mathcal{M}_{N_h \times N_{er}}.$

$\triangleright$ **The final approximation of $\mathbf{u}_h(\mu)$ is reconstructed as:** $\boxed{\tilde{\mathbf{u}}_h(\mu) = \mathbb{B}_{rb} \mathbf{u}_{rb}(\mu) + \mathbb{B}_{er} \mathbf{u}_{er}(\mu)}.$

The Error-Aware POD-NN method: two NNs trained in parallel.

[Allabou et al. CMAME'24]



$$\mathcal{J}_1(\theta_1) = \frac{1}{N_{train}} \sum_{s=1}^{N_{train}} \left\| \mathcal{N}_1(\theta_1)(\mu_s) - \mathbb{B}_{rb}^T \mathbf{u}_h(\mu_s) \right\|_2^2 \quad ; \quad \mathcal{J}_2(\theta_2) = \frac{1}{N_{train}} \sum_{s=1}^{N_{train}} \left\| \mathcal{N}_2(\theta_2)(\mu_s) - \mathbb{B}_{er}^T \mathbf{e}_h(\mu_s) \right\|_2^2.$$

1 Problem statement

2 POD-NN based methods

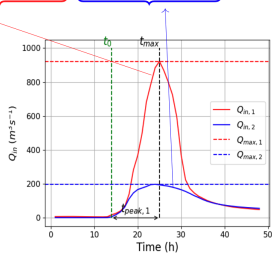
3 Results

4 Conclusion

The real-like test case. $\mu \in \mathbb{R}^{N_\mu}$ with $N_\mu = (4 + 1) = 5$

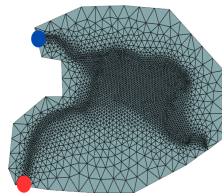
Case data provided by Garambois [Villenave et al., SHF 2023]. Dataset: Vigicrues, SCHAPI-DGPR, Météo-France

$$\mu = (\underbrace{t_{\text{peak},1}, Q_{\text{max},1}}_{\text{red}}, \underbrace{t_{\text{peak},2}, Q_{\text{max},2}, t_s}_{\text{blue}}) \in \mathbb{R}^5$$



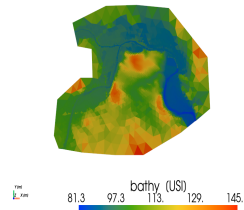
2 inflow hydrographs

(red) for the Aude river, (blue) for the Fresquel river, each parameterized by 2 parameters: $(t_{\text{peak}}, Q_{\text{max}})$.



Mesh: 7,267 cells

Bathymetry elevation



Bathymetry.

Numerical experiment set up [Allabou et al., 2024]

Offline phase

- Time interval $[0, T]$, $T = 60$ h, snapshots every $\Delta t_s = 36$ min.
- Snapshots matrix $\mathbb{S} = [\mathbf{h}, \mathbf{u}, \mathbf{v}]^T \in \mathcal{M}_{N_h \times N_s}$ with $N_h = 21\,801$, and $N_s = 2\,025 \times N_t$, $N_t = 89$ time steps $\Rightarrow N_s = 180\,225$.
- **Reduced dimension** $N_{rb} = 5$ ($\varepsilon_{POD} = 10^{-2}$) and $N_{er} = 16$ ($\varepsilon_{er} = 10^{-1}$).
- NN architectures: $\mathcal{N}_1$ with 10 hidden layers with 20 neurons per layer, $\mathcal{N}_2$ with 10 hidden layers with 30 neurons per layer.
- **CPU-times:** eigenvectors computation (≈ 30 min) plus the two NNs training (1h10) (with training number $N_{tr} \approx 0.8N_s$).

Training done in parallel $\Rightarrow$ similar time as for the original POD-NN method.

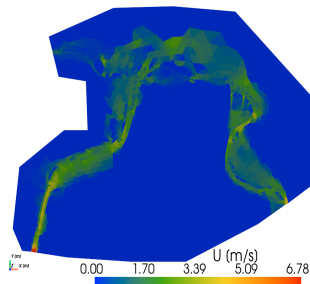
Online phase

- **Prediction CPU-time:** ≈ 5 s vs HF simulation ≈ 118 s for $\#cells \approx 7,000$ ($N_h = 21\,801$).

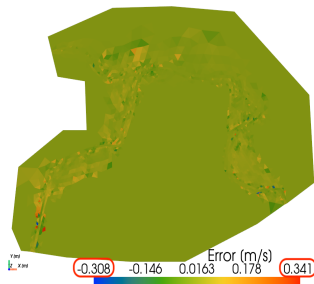
Computer: laptop, 11th Gen Intel(R) Core(TM) i9-11950H, 2.60 GHz, 32 GB of RAM.

Prediction (on-line computations in sec)

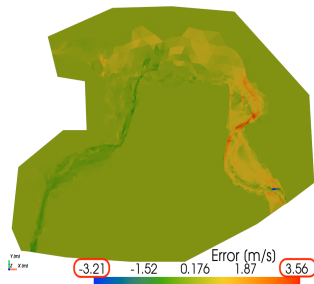
Plots: velocity norm $\|\mathbf{v}_{HF}\|$ [ms^{-1}] at the peak instant



(L) HF solution = the reference.
 $[ms^{-1}]$.
 $\|\mathbf{v}_{HF}\|$ [ms^{-1}].



(M) EA - POD-NN method.

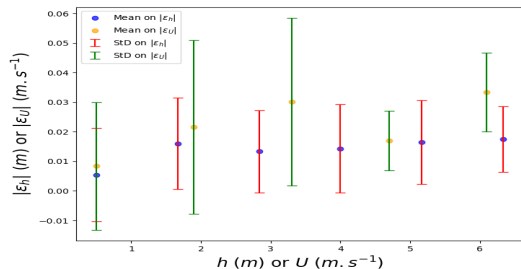


(R) POD-NN method.

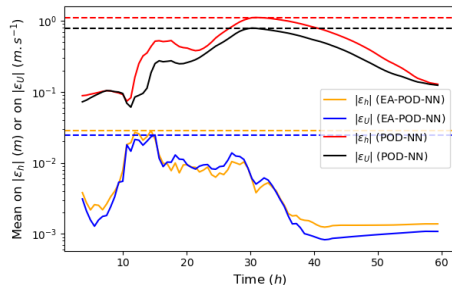
Note the ≈ 10 factor between the two scales.

Maps = "errors" (differences) $\varepsilon_V = (\|\mathbf{v}_{HF}\| - \|\mathbf{v}_{rb}\|)$

EA-POD-NN method: "errors" $|\varepsilon_h|$ and $|\varepsilon_V|$



(L) **EA-POD-NN vs HR reference on h and $\|v\|$.**
Differences at peak flooding.



(R) Mean differences in m and $(m.s^{-1})$ vs time.
for the Error Aware (EA) and the original POD-NN methods.

▷ **The EA - POD-NN method provides quite accurate solutions both in h [m] and q [m²/s].**

① Problem statement

② POD-NN based methods

③ Results

④ Conclusion

Conclusion

Summary

- ▷ POD-NN based reduction methods: a non-intrusive approach, applicable to FE and FV solvers.
 - Basic principle. Offline phase: CPU demanding training & HF solver runs;
Online phase enables predictions in seconds. Ref. [Hesthaven et al.].
 - ⊕ Well-suited to nonlinear PDEs (parabolic, FE, [Hesthaven et al.]),
including nonlinear hyperbolic systems solved by FV [Allabou et al.].
 - ⊖ Inaccurate if the snapshots are not "numerous enough" or not "properly sampled" in $\mathcal{P}$.
- ▷ The enriched version "Error Aware" EA - POD-NN
 - ⊕ enables to obtain much better accuracies (with same experiment set up) [Allabou et al.].
 - ↔ enables to consider much less snapshots (HR solutions), therefore lower CPU-times and smaller NNs too.
 - ⊖ The snapshots still need to be properly sampled to accurately capture non-regular solutions or transitions.

Conclusion

Summary

- ▷ POD-NN based reduction methods: a non-intrusive approach, applicable to FE and FV solvers.
 - Basic principle. Offline phase: CPU demanding training & HF solver runs;
Online phase enables predictions in seconds. Ref. [Hesthaven et al.].
 - ⊕ Well-suited to nonlinear PDEs (parabolic, FE, [Hesthaven et al.]),
including nonlinear hyperbolic systems solved by FV [Allabou et al.].
 - ⊖ Inaccurate if the snapshots are not "numerous enough" or not "properly sampled" in $\mathcal{P}$.
- ▷ The enriched version "Error Aware" EA - POD-NN
 - ⊕ enables to obtain much better accuracies (with same experiment set up) [Allabou et al.].
 - ↪ enables to consider much less snapshots (HR solutions), therefore lower CPU-times and smaller NNs too.
 - ⊖ The snapshots still need to be properly sampled to accurately capture non-regular solutions or transitions.

On-going studies

- Improve the accuracy and/or reduce the number of required snapshots (with similar accuracy) by considering:
 - i) **enriched NNs with physics information**; ii) **adaptive parameter samplings**.
↪ *on-going PhD of T.-V. N'Guyen's PhD, INSA-IMT Toulouse.*
- Development of a complete reduction chain, for more complex input parameters than those presented, with application to **large dimensional real cases**.
↪ *on-going PhD CIFRE of D. Gomez, CS Lab / INSA-IMT Toulouse.*
- **Uncertainty Quantification (UQ)** considerations.

On this topic (and more), see our [AI Cluster ANITI, Chaire "PILearnWater"], Toulouse.

Thank you for your attention.

References

- The original POD-NN method:
S. Hesthaven and S. Ubbiali. "Non-intrusive reduced order modeling of nonlinear problems using neural networks". *Journal of Computational Physics (JCP)*, 2018.
- The enriched version named "Error Aware - POD-NN":
A. Allabou, R. Bouclier, P.-A. Garambois, J. Monnier. "Reduction of the Shallow Water System by an Error Aware POD-Neural Network Method: Application to Floodplain Dynamics". *Comput. Meth. Applied Mech. Eng. (CMAME)*, 2024.
- A chain of model reduction with application to an urban-like configuration:
A. Allabou, R. Bouclier, P.-A. Garambois, J. Monnier. In preparation.
- * *Defence of the PhD of M. Allabou (INSA-IMT Toulouse) on March 24th, 2026.*
- * AI Cluster ANITI, Chaire "PILearnWater" (<https://aniti.univ-toulouse.fr>)