TRAINING SAMPLING VARIATIONS IN NON-INTRUSIVE DATA-DRIVEN SURROGATE MODELS FOR COUPLED VISCOUS FLOW AND TRANSPORT

Roberto Velho, Adriano Côrtes, Gabriel Barros, Fernando Rochinha, Alvaro Coutinho

High Performance Computing Laboratory - Federal University of Rio de Janeiro

2025 M2P

ECCOMAS Thematic Conference

IACM SPECIAL INTEREST CONFERENCE

Math to Product (M2P)

Valencia (Spain), 4 - 6 June 2025

Introduction: Data-driven Parametric Surrogate Model

Data-driven Parametric Surrogate models[1,2] help us on:

- 🙂 Reduce the computational cost of forward evaluations in high-fidelity models.
- 🙂 (+) Data-driven: allows a non-intrusive implementation.
- 🙂 (+) Parametric: Enable efficient *many-query* analyses, such as uncertainty quantification (UQ).
- ☹️ **Trade-off**: reduced solution accuracy.

---

[1] Guo, Hesthaven - Data-driven reduced order modeling for time-dependent problems. Computer Methods in Applied Mechanics and Engineering, vol. 345, 2018.

[2] Fresca, Manzoni - Pod-dl-rom: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. Computer Methods in Applied Mechanics and Engineering, vol. 388, 2022.

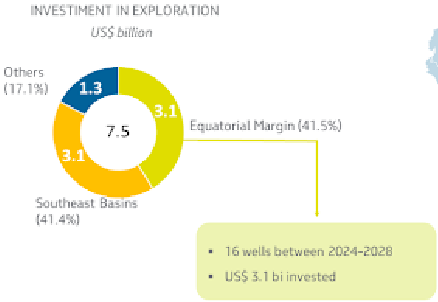
Introduction: Super-Resolution in Surrogate Models

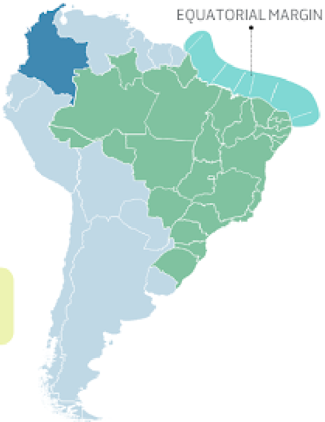
🎯 **Goal of Super-Resolution**: Enhance image quality by reconstructing finer details.

- **Proposed Approach**: Apply super-resolution techniques to improve the quality of predictions generated by a surrogate model.
- **Training Strategy**:
  - Train super-resolution model on the same dataset as the surrogate model.
  - Leverage high-quality data to boost surrogate model output fidelity.

🏆 **Expected Outcome**: Improved accuracy and visual quality in surrogate model predictions.

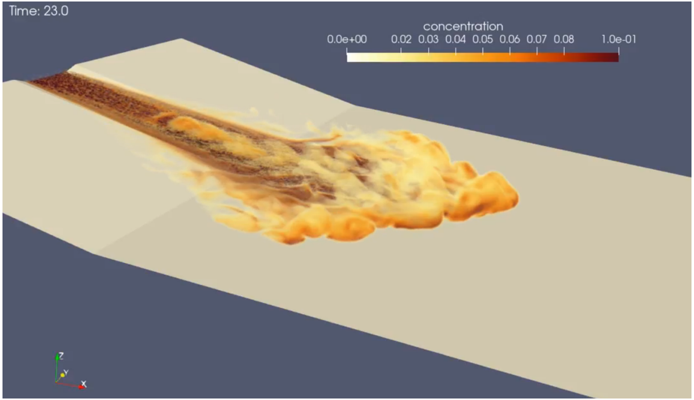
Motivation: Prediction of Sediment Deposition





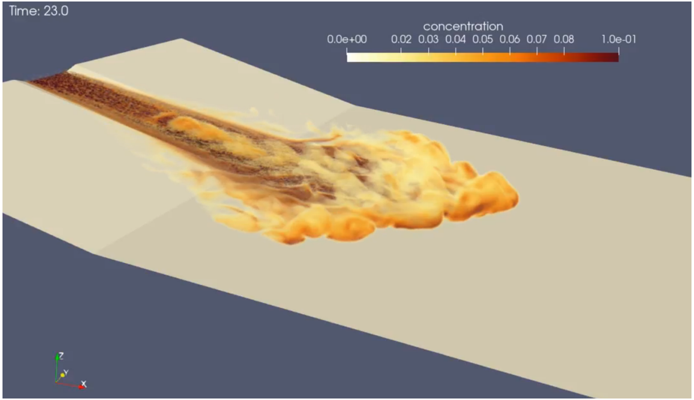
Sediment deposition is a key aspect for geologists - Oil Extraction!

Motivation: Prediction of Sediment Deposition



We measure the concentration at the bottom of the basin.

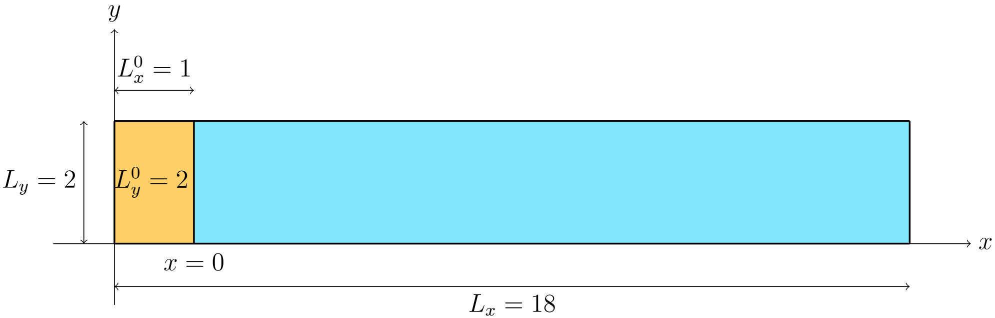
Motivation: Prediction of Sediment Deposition



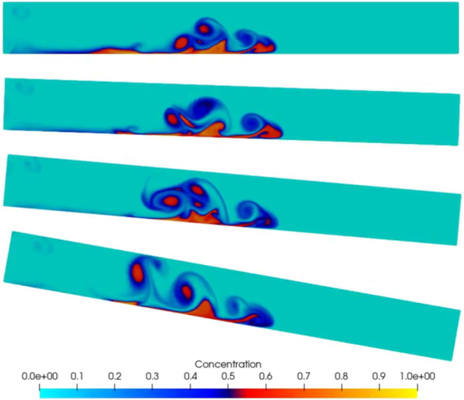
We measure the concentration at the bottom of the basin.

Target: Layers of deposition material.

ROM for Turbidity Currents under Lock-Exchange Setup



Lock-Exchange Setup in 2D



Evolution of Concentration.

Governing Equations

Navier–Stokes equations in non-dimensional form in their non-conservative formulation:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{1}{\sqrt{Gr}} \Delta \boldsymbol{u} + c\, \boldsymbol{e}_\theta^g, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1b}$$

$$\frac{\partial c}{\partial t} + \left(\boldsymbol{u} + u_S\, \boldsymbol{e}_\theta^g\right) \cdot \nabla c = \nabla \cdot \left(\frac{1}{Sc\sqrt{Gr}} \nabla c\right). \tag{1c}$$

- Domain: $\Omega = [0, 18] \times [0, 2]$
- BC: no-slip condition ($\boldsymbol{u} = \mathbf{0}$) on all solid boundaries
- BC: zero-flux Neumann condition for sediment concentration
- ICs: $c = 1$ for $x < 0$ and $c = 0$ Ow.,

- c: normalized sediment concentration
- $\boldsymbol{u}$: velocity field, u_s: cte. sedimentation vel.
- $Gr = \sqrt{5} \times 10^3$ Grashof, $Sc = 1.00$ Schmidt
- p: pressure, θ: channel's inclination angle
- Gravit. force: $c\, \boldsymbol{e}_\theta^g$, $\boldsymbol{e}_\theta^g = (\sin\theta, -\cos\theta)$

Dataset Generation

Simulation Setup - Full Order Model (FOM):

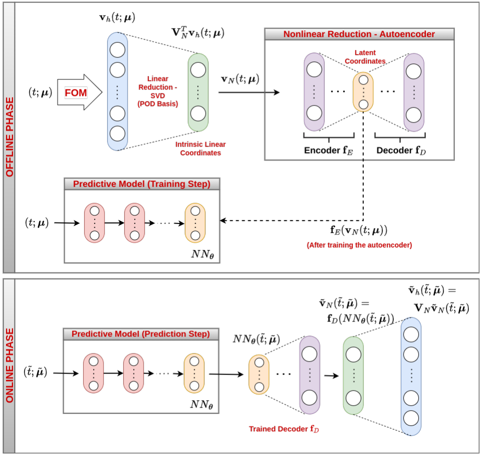
- Stabilized FEM - residual-based var. multiscale method[1] - FEniCS framework
- $\Omega = [0, 18] \times [0, 2]$, 701×101 nodes, yielding 70801 concentration values.
- Time interval $[0, 22]$ with timestep $\Delta t = 5 \times 10^{-3}$;
- Concentration snapshots every 0.05 time units, giving 440 snapshots.

Data Generation:

- Angles of $0, 2, 4, 6, 8, 10$ for training the ROM Model;
- Angles of 5 and 12 for testing - interpolation and extrapolation.

---

[1] G. Guerra, S. Zio, J. Camata, F. Rochinha, R. Elias, P. Paraizo, and A. Coutinho - **Numerical simulation of particle-laden flows by the residual-based variational multiscale method** - International Journal for Numerical Methods in Fluids, 2013.



Reduced-order Model (ROM) Setup

[1] R. Velho, A. Côrtes, G. Barros, F. Rochinha. A. Coutinho - **Advances in Data-Driven Reduced Order Models using Two-Stage Dimension Reduction for Coupled Viscous Flow and Transport** - Finite Elements in Analysis & Design 2025.

Denoising Diffusion Probabilistic Model (DDPM)

Overview:

- DDPMs are generative models that generate data by reversing a gradual noise-adding process.
- Data is iteratively corrupted with Gaussian noise, then reconstructed by learning a denoising sequence.

Key Concepts:

- **Forward Process**: Gradually adds noise to data to create a noisy sequence.
- **Reverse Process**: Learns to remove noise, restoring the original data from a noisy state.
- **Applications**: High-quality image generation, **super-resolution**, inpainting, and more.

Denoising Diffusion Probabilistic Model (DDPM)

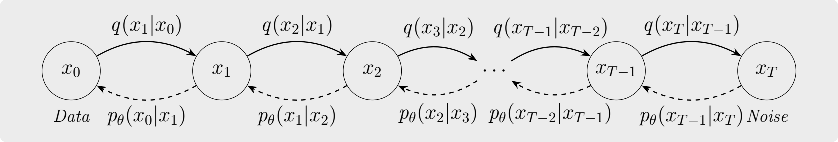


Figure: Markovian process in DDPM showing forward diffusion q and reverse denoising p_θ transitions.

Denoising Diffusion Probabilistic Model (DDPM)

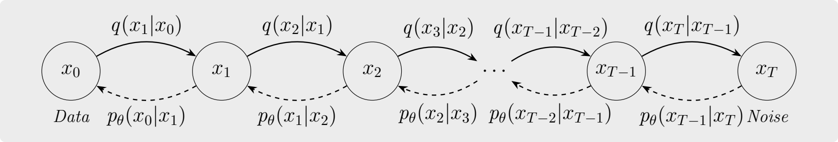


Figure: Markovian process in DDPM showing forward diffusion q and reverse denoising p_θ transitions.

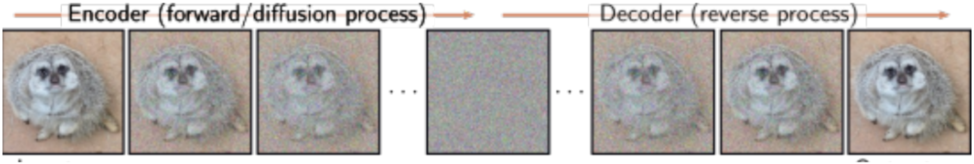


Figure: S.J.D. Prince, *Understanding Deep Learning*, The MIT Press, 2023.

Super-Resolution(SR): Training Setup

- **GPU**: NVIDIA RTX A4000 (16GB RAM)
- **Framework**: PyTorch with Adam optimizer
- **Epochs**: 300 (approx. 16 hours)
- **Learning Rate**: Fixed at 2×10^{-4}
- **Batch Size**: 12
- **Number of diffusion time steps**: 1000
- **Linear schedule for β_t: from $\beta_1 = 10^{-4}$ to $\beta_{1000} = 0.02$**

[1] D. Shu, Z. Li, A. Farimani - **A physics-informed diffusion model for high-fidelity flow field reconstruction** - Journal of Computational Physics, 2023.

[2] D. Shu, Z. Li, A. Farimani - **Diffusion-based-Fluid-Super-resolution** - https://github.com/BaratiLab/Diffusion-based-Fluid-Super-resolution/tree/main_v1.

Applying the trained SR model:

- For evaluation, we applied the trained diffusion super-resolution model to the ROM outputs for angles not included in the training set;

- ROM output is not Gaussian noise; it still contains the principal characteristics expected in the final solution;

- Thus we cannot input the ROM at the final step of the diffusion chain (1000) for denoising. We must introduce it at an intermediate step $n_s < 1000$;

Applying the trained SR model:

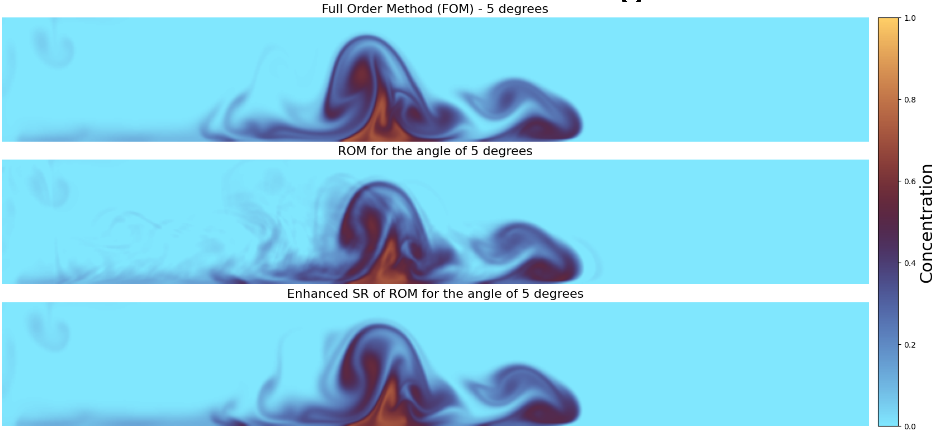
- When the selected starting time is insufficient, the Enhanced ROM results exhibit numerous numerical artifacts, indicating inadequate denoising;

- Conversely, an excessively large starting time can lead to overcorrection, potentially eliminating smaller secondary dynamics and resulting in information loss;

- Through empirical testing, we determined that a starting time of $n_s = 200$ **produces optimal reconstruction quality**;

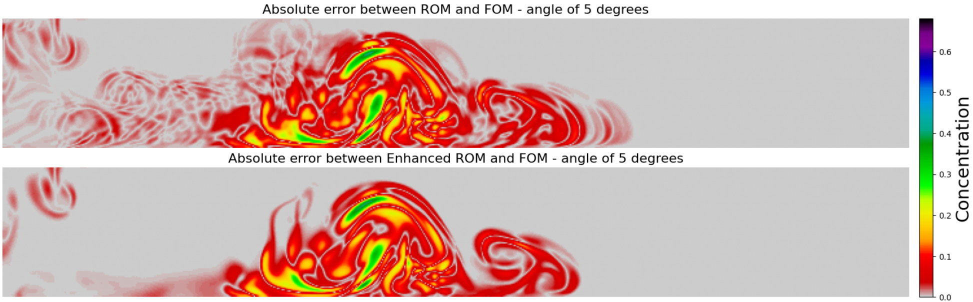
- Computational Time: 1 time/image - 10s. Entire dynamics (440 times) - 27 minutes.

Visual of the reconstruction - θ=5 degrees at 20s



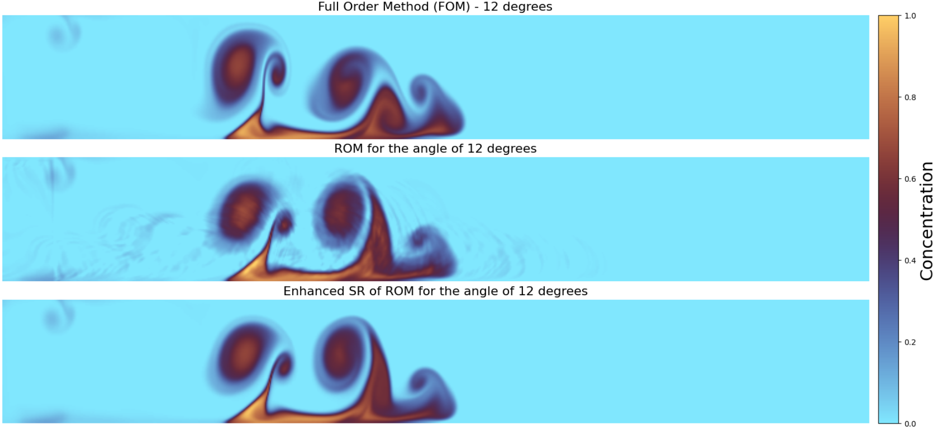
Comparison between a Finite Element Method data (top), Reduced order model data(middle), and the super-resolution enhanced data (bottom) at time 20s.

Error Analysis of the reconstruction - θ=5 degrees at 20s



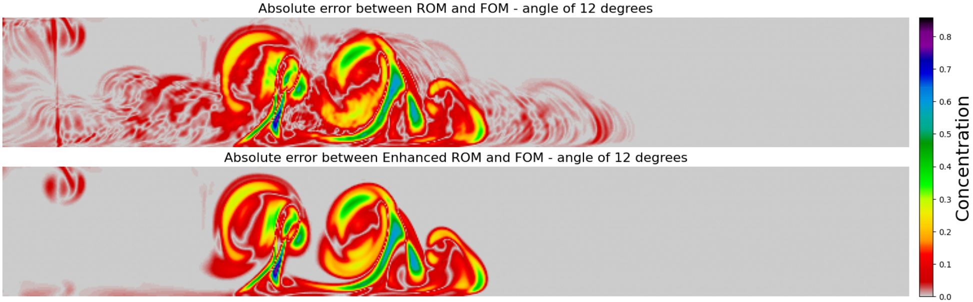
Comparison between the absolute error between the ROM and the FEM data (top) and between SR and FEM data (bottom) at time 20s.

Visual of the reconstruction - θ=12 degrees at 15s



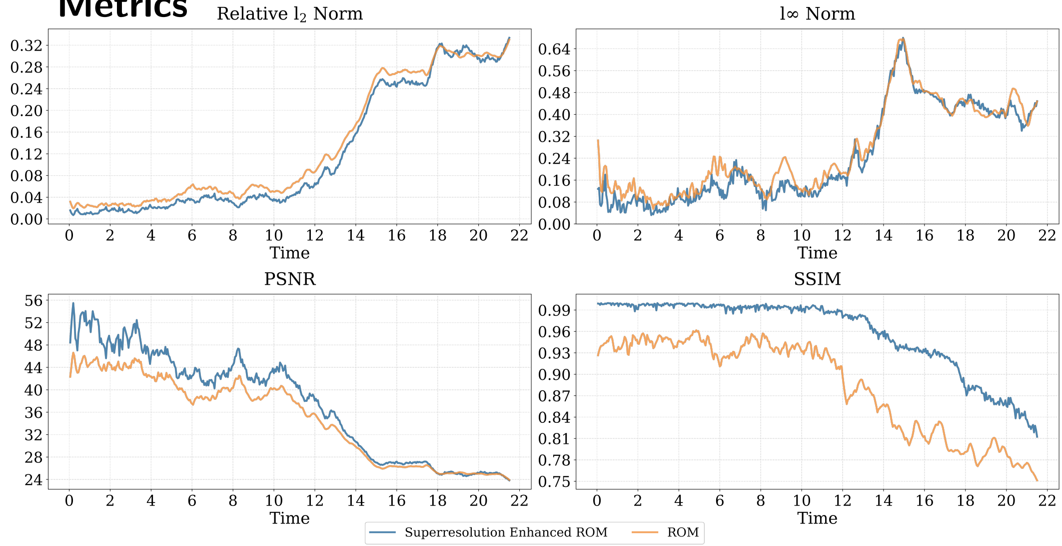
Comparison between a Finite Element Method data (top), Reduced order model data(middle), and the super-resolution enhanced data (bottom) at time 15s.

Error Analysis of the reconstruction - θ=12 degrees at 15s



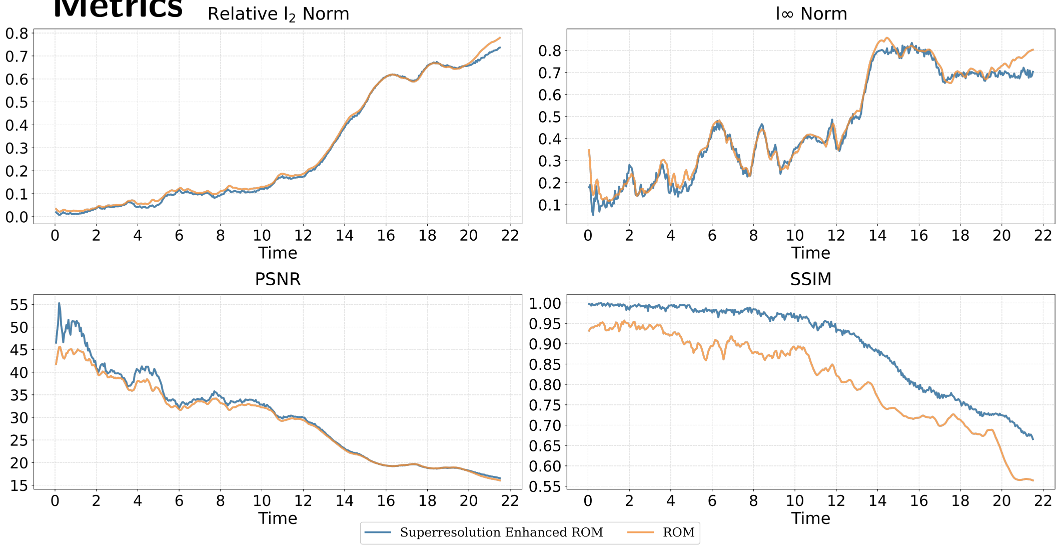
Comparison between the absolute error between the ROM and the FEM data (top) and between SR and FEM data (bottom) at time 15s.

Metrics



Metrics for the interpolation angle of 5 degrees.

Metrics



Metrics for the extrapolation angle of 12 degrees.

Partial Conclusions

- SR model is able to increase fidelity of ROM, fixing numerical artifacts and reducing error;
- But increase in fidelity is limited by quantity of information in ROM, since the model is data-driven;
- Achieved robust image reconstruction in ROM reconstruction;
- Trainable with the same data used to build a surrogate;
- Training the SR model is a lot more expensive than training the ROM one;
- The evaluation of the SR model takes in the order of minutes while the ROM one is about miliseconds.

---

[1] R. Sousa, R. Velho, N. Matos, A. Côrtes, G. Barros, A. Coutinho - **Data-driven diffusion-based super-resolution for improvement of reduced-order model predictions in fluid dynamics** - Preprint, 2025

Sampling with different time steps

Before we train the SR model with different sampling factors, let us observe the influence of sampling in the ROM one.

For each instance of the parameter, we collect:

- SF1: 5×10^{-3} (FOM time step) - generating 4400 snapshots;
- SF2: 1×10^{-2} - generating 2200 snapshots;
- SF10: 5×10^{-2} - generating 440 snapshots;
- SF20: 1×10^{-1} - generating 220 snapshots;
- SF100: 5×10^{-1} - generating 44 snapshots;

Disk Allocation for the training snapshot matrix:
SF1 (5.27 GB), SF2 (2.64 GB), SF10 (528 MB), SF20 (265 MB), SF100 (54.4 MB).

Criterion for First Reduction (99.9999% of Explained Variance for SVD).

Rank - SF1: (?), SF2: 13183 (out of 13200), SF10: 2640 - Full, SF20: 1320 - Full, SF100: 264 - Full.

SVD Components (% of Rank) - SF1: (?), SF2: 1125 (8.5%), SF10: 1121 (42.46%), SF20: 996 (75.45%), SF100: 236 (89.39%).

Criterion for First Reduction (99.9999% of Explained Variance for SVD).

Rank - SF1: (?), SF2: 13183 (out of 13200), SF10: 2640 - Full, SF20: 1320 - Full, SF100: 264 - Full.

SVD Components (% of Rank) - SF1: (?), SF2: 1125 (8.5%), SF10: 1121 (42.46%), SF20: 996 (75.45%), SF100: 236 (89.39%).

We then gave up using SF1 and still want observe SF2 despite the clear ”saturation” on the SVD/POD components.

Encoder Architecture: Powers of 2 - starting form the immediate one below the number of SVD/POD components.

- SF2 and SF10: 1024, 512, 256, 128, 64, 32
- SF20: 512, 256, 128, 64, 32, 16
- SF100: (180) - 128, 64, 32, 16 (8, 4)

Criterion for First Reduction (99.9999% of Explained Variance for SVD).

Rank - SF1: (?), SF2: 13183 (out of 13200), SF10: 2640 - Full, SF20: 1320 - Full, SF100: 264 - Full.

SVD Components (% of Rank) - SF1: (?), SF2: 1125 (8.5%), SF10: 1121 (42.46%), SF20: 996 (75.45%), SF100: 236 (89.39%).

We then gave up using SF1 and still want observe SF2 despite the clear "saturation" on the SVD/POD components.

Encoder Architecture: Powers of 2 - starting form the immediate one below the number of SVD/POD components.

- SF2 and SF10: 1024, 512, 256, 128, 64, 32
- SF20: 512, 256, 128, 64, 32, 16
- SF100: (180) - 128, 64, 32, 16 (8, 4)

We then gave up on using SF100, at least for a more "automated" Encoder architecture construction.

Neural Network (regression) Architecture:

- SF2: 9 (5) hidden layers of 50 neurons;
- SF10, SF20: 5 hidden layers of 50 neurons.

Neural Network (regression) Architecture:

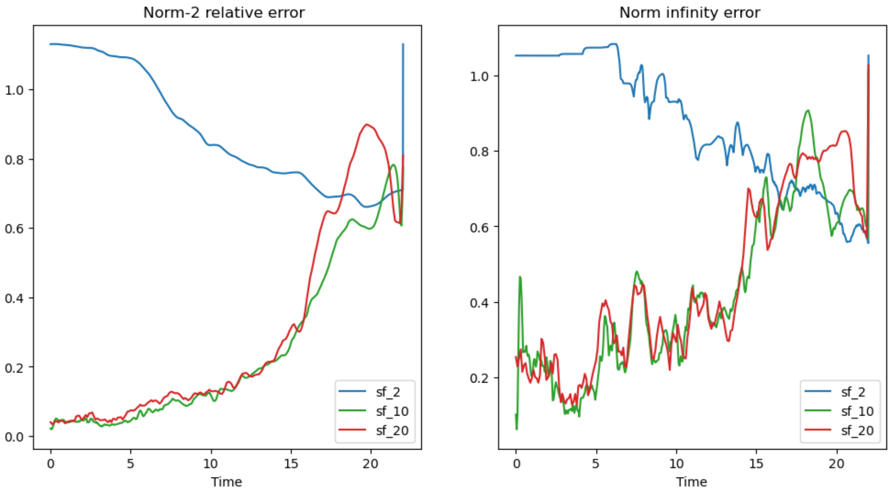
- SF2: 9 (5) hidden layers of 50 neurons;
- SF10, SF20: 5 hidden layers of 50 neurons.

RAM allocation: about 10 to 20 times disk allocation.

Training time increases with sampling factor!

For ($5k$ epochs for NN):

- SF2: 42079s - 11.7h
- SF10: 16625s - 4.6h
- SF20: 4378s - 1.2h



Errors for different sampling factors - test angle of 5 degrees.

Some Lessons

- Let the data speak by itself. Saturation on SVD components with refinement of time sampling;
- Though trying to reach the saturation (with a large number of POD components) may create problems for the autoencoder - producing a large latent space;
- Surely we could perform AE reduction without POD first. Could we learn the saturation effect? Using POD first also reduces hardware requirements;
- We could also train the AE and the NN together. Would we be able to understand the ideal latent space? Also with a functional model for the AE architecture or just brute force?

Some Lessons

- Recent papers on error bounds. Could we really design the architectures given the expressions for the error?

- Small time steps create huge data, hard to reduce;

- Large time steps may not provide enough examples to fit the NN (classical ML knowledge) but also difficult to construct the AE - for fast dynamics - increasing layers require more data, few layers do not capture the dynamics;

- Not impossible to tune models for smaller time steps - but either reduction will ”not work” or tunning the regression will be time costly.

Future Directions

- Experiment with alternative models, like Denoising Diffusion Restoration Models;
- Insert embedding (angle) in the diffusion model;
- Predictor for starting step of the reverse process;
- Effect of time splitting ROMS[1];
- Explore physics-informed guidance for improved fidelity;
- Training the ROM and the SR models with different sampling factors (eventually even different for each time partition).

[1] R. Velho, A. Côrtes, G. Barros, F. Rochinha. A. Coutinho - **Advances in Data-Driven Reduced Order Models using Two-Stage Dimension Reduction for Coupled Viscous Flow and Transport** - Finite Elements in Analysis & Design 2025.

Acknowledgments

- CAPES
- CNPq
- FAPERJ
- Petrobras - Industrial Partner

Any question?

roberto.velho@nacad.ufrj.br

Slides, preprint, movies - https://github.com/rmvelho/m2p2025

Forward Diffusion (Encoder) Process in DDPM

Overview:

- Gradually adds Gaussian noise to data over T steps, transforming data into near-random noise.
- Enables learning of the data distribution through progressive noise addition.

Mathematics:

- Transition: $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t I)$
- After t steps: $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} \cdot x_0, (1-\bar{\alpha}_t)I)$
- $\beta_t \in [0, 1]$: noise schedule, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$

Result:

- Produces noisy images $\{x_1, x_2, ..., x_T\}$, which the reverse model learns to denoise.

Reverse Process (Decoder) and Loss in DDPM

Reverse Process:

- Removes noise added in the forward process, reconstructing data from noise.
- Modeled as: $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$, $\{\sigma_t^2\}$ predetermined.

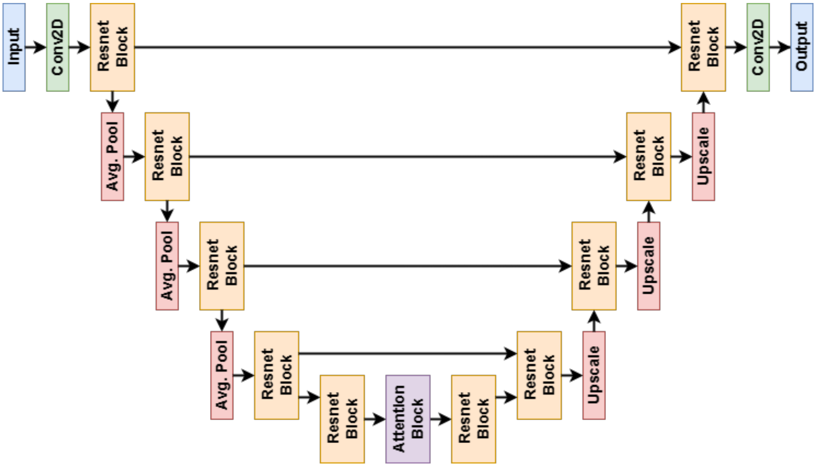
Neural Network Architecture:

- U-Net with skip connections is used to predict $\mu_\theta(x_t, t)$, as it efficiently captures multi-scale information, refining noise to match the original data structure.

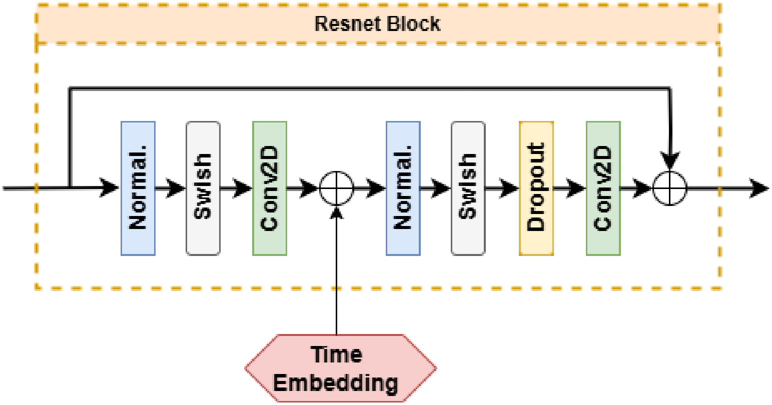
Loss Function:

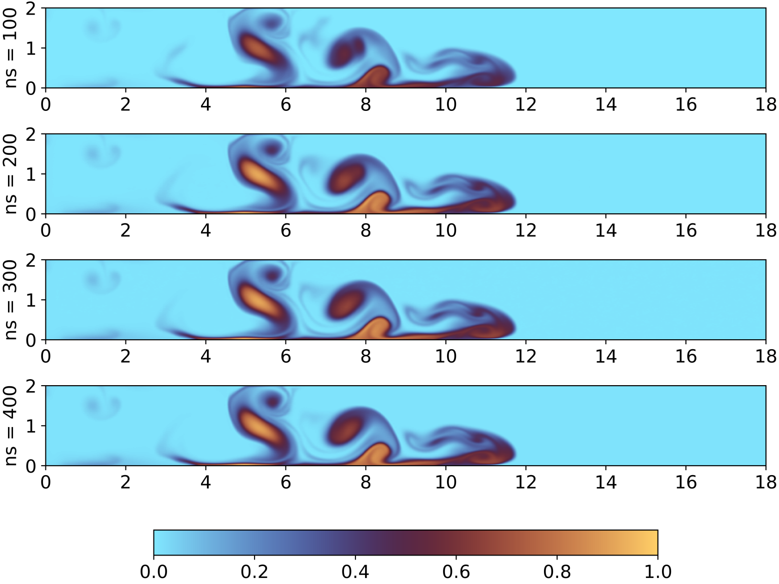
- Minimize the noise prediction error: $\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$.
- Trains model ϵ_θ to accurately predict and remove noise step-by-step.

Super-Resolution U-net Architecture



The Resnet Block




ns = 100
ns = 200
ns = 300
ns = 400
0
2
4
6
8
10
12
14
16
18
0.0
0.2
0.4
0.6
0.8
1.0