COPPE
UFRJ

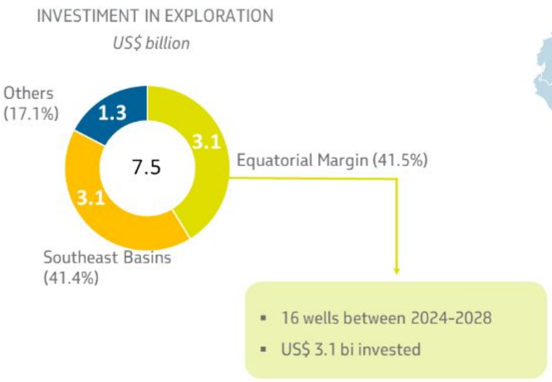
Turbidity Currents Simulations in Channels with Different Slopes Using ROM-NN Models

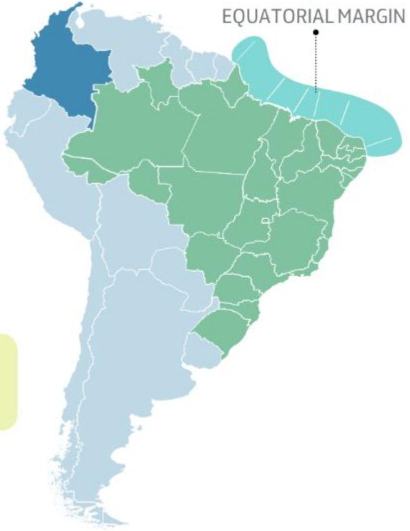
R. Velho[1], A. Côrtes[1], G. Barros[1], J. Camata[2], G. Guerra[3], R. Elias[1], F. Rochinha[1], A. Coutinho[1]

1- Federal University of Rio de Janeiro, 2- Federal University of Juiz de Fora, 3- Fluminense Federal University

Motivation

New exploratory frontier:
Equatorial Margin





Petrobras hits oil for second time this year in ultra-deep waters of Brazil's Equatorial Margin

EXPLORATION & PRODUCTION

April 19, 2024, by Melisa Čavčić

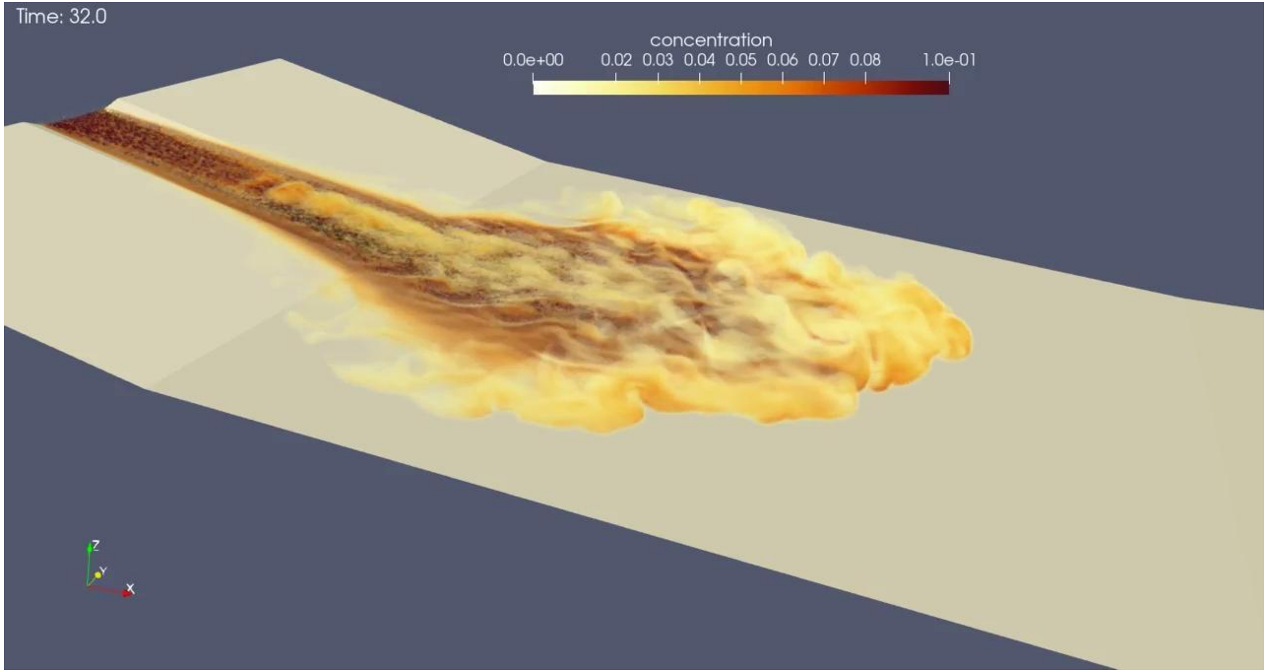
Brazilian state-owned oil and gas giant Petrobras has made an oil discovery at another well in the Equatorial Margin, which stretches along the Brazilian coast from Rio Grande do Norte to Amapá.

According to Pellegrini and Ribeiro, plays are **turbidites**

https://petrobras.com.br/en/quem-somos/novas-fronteiras

Pellegrini B da S, Ribeiro HJPS. Exploratory plays of Pará-Maranhão and Barreirinhas basins in deep and ultra-deep waters, Brazilian Equatorial Margin. Braz J Geol [Internet]. 2018 Jul;48(3):485–502. Available from: https://doi.org/10.1590/2317-4889201820180146

Turbidite currents simulation with EdgeCFD[1]



Channel slope: 11º;
Basin slope: 4º

Re= 11100

Slope matters[2]

[1]Cortes, AMA, et al. "EdgeCFD: a parallel residual-based variational multiscale code for multiphysics." International Journal of Computational Fluid Dynamics 34.7-8 (2020): 529-548.
[2]Birman, V. K., Battandier, B. A., Meiburg, E., & Linden, P. F. (2007). Lock-exchange flows in sloping channels. Journal of Fluid Mechanics, 577, 53-77.

Governing Equations

Navier-Stokes equations with the Boussinesq hypothesis coupled with N sediment transport equations describe the incompressible turbulent fluid flows carrying dilute suspensions of sediments

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\sqrt{Gr}} \Delta \mathbf{u} + c_T \, \mathbf{e}^g \quad \text{in} \quad \Omega \times [0, t_f]$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega \times [0, t_f]$$

$$\frac{\partial c_i}{\partial t} + (\mathbf{u} + u_{S_i} \, \mathbf{e}^g) \cdot \nabla c_i = \nabla \cdot \left(\frac{1}{Sc_i \sqrt{Gr}} \nabla c_i \right) (i = 1, .., N) \text{ in } \Omega \times [0, t_f]$$

$$c_T = \textstyle\sum_{i=1}^{N} c_i \qquad \mathbf{e}^g = \{\sin(\theta), -\cos(\theta)\}$$

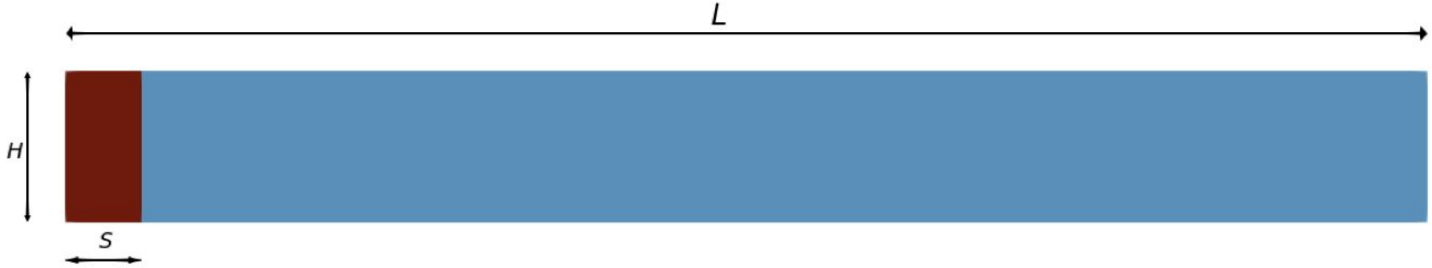
Guerra, Gabriel M., et al. "Numerical simulation of particle-laden flows by the residual-based variational multiscale method." International Journal for Numerical Methods in Fluids 73.8 (2013): 729-749

Simple Parametric Case: 2D Lock-Exchange with Varying Slope

FOM: Fully coupled (u-p-c) RBVMS FEM in FEniCS[1]

BCs: NS no-slip, Transport no-flux, $\Delta t = 0.01, T = [0, 30], u_S = 0, \theta \in \{0, 2, 4, 6, 8, 10\}$

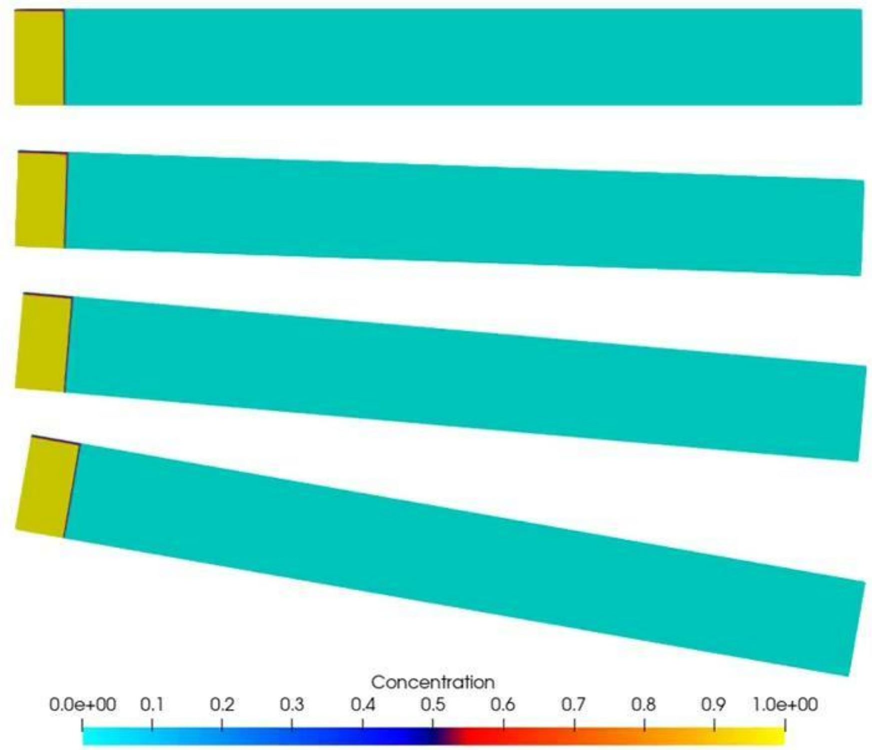
$L = 18, H = 2, S = 1, Gr = 5 \times 10^6, Sc = 1, c_0 = 1 \text{ in } H \times S, c_0 = 0 \text{ in } (L - S) \times H$



Mesh: 700X100 cells, each cell with 2 linear triangles

[1]Barros, Gabriel F., et al. "Dynamic mode decomposition in adaptive mesh refinement and coarsening simulations." Engineering with Computers 38.5 (2022): 4241-4268.

FOM Simulations for Slope of 0, 2, 5, 10 degrees



Fitting and Prediction Attempt

- 6 different angles of 0, 2, 4, 6, 8, and 10 degrees;

- 180 snapshots (one every 5 time steps of FOM).

Can we predict the dynamics for an unseen angle at multiple time instants?

At which computational cost? And with which level or error?

Fitting and Prediction Attempt

Training Set

- 6 different angles of 0, 2, 4, 6, 8, and 10 degrees;
- 180 snapshots (one every 5 time steps of FOM).

Can we predict the dynamics for an unseen angle at multiple time instants?

At which computational cost? And with which level or error?

Test Set

- Angle of 5 degrees;
- 180 predictions at same instants of FOM.

Fitting and Prediction Attempt

Training Set

- 6 different angles of 0, 2, 4, 6, 8, and 10 degrees;
- 180 snapshots (one every 5 time steps of FOM).

Can we predict the dynamics for an unseen angle at multiple time instants?

At which computational cost? And with which level or error?

Test Set

- Angle of 5 degrees;
- 180 predictions at same instants of FOM.

Difficulty

- 6 x 180 = 1080 observations
- Each snapshot has ~ **70k** mesh points
- We need a mapping from dimension 2 to ~70k:
  (“angle”, “time”) —> concentration for each mesh point.

Fitting and Prediction Attempt

Snapshot Matrix

$$Y = \begin{bmatrix} | & | & \cdots & | & | & \cdots & | & \cdots & | \\ c(t_1,\theta_0) & c(t_2,\theta_0) & \cdots & c(t_{180},\theta_0) & c(t_1,\theta_2) & \cdots & c(t_{180},\theta_2) & \cdots & c(t_{180},\theta_{10}) \\ | & | & \cdots & | & | & \cdots & | & \cdots & | \end{bmatrix}$$

Training Set

- 6 different angles of 0, 2, 4, 6, 8, and 10 degrees;
- 180 snapshots (one every 5 time steps of FOM).

Can we predict the dynamics for an unseen angle at multiple time instants?

At which computational cost? And with which level or error?

Test Set

- Angle of 5 degrees;
- 180 predictions at same instants of FOM.

Difficulty

- 6 x 180 = 1080 observations
- Each snapshot has ~ 70k mesh points
- We need a mapping from dimension 2 to ~70k:
  (“angle”, “time”) —> concentration for each mesh point.

Reduction

POD (270 modes - 99.99999% EV):

Map 2D -> 270D

POD (270m.) + NL autoencoder (16D):

Map 2D -> 16D

Mapping is done via a Neural Net

Nonlinear Manifold ROM

+ Sequence of linear and nonlinear reductions

+ Followed by predictive model via neural network

+ Implemented in PyTorch (SnapFlow[1])

+ Based on:

Romor, F., Stabile, G., and Rozza, G. Non-linear manifold ROM with Convolutional Autoencoders and Reduced Over-Collocation method (2023)

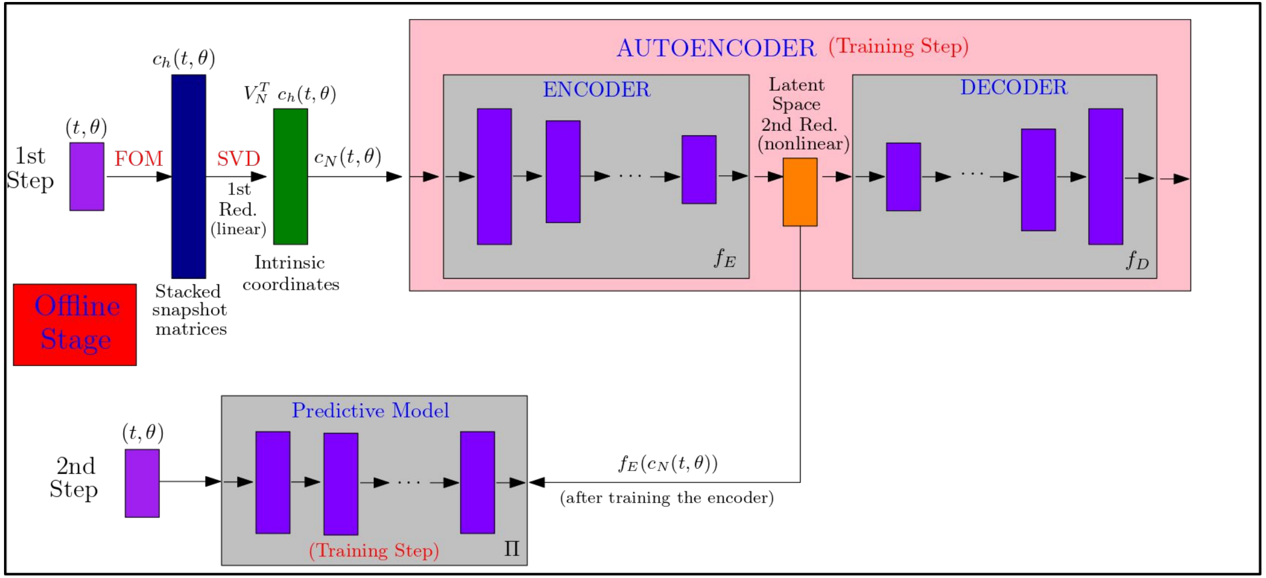
Fresca, Dede, and Manzoni - “A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs” (2021)

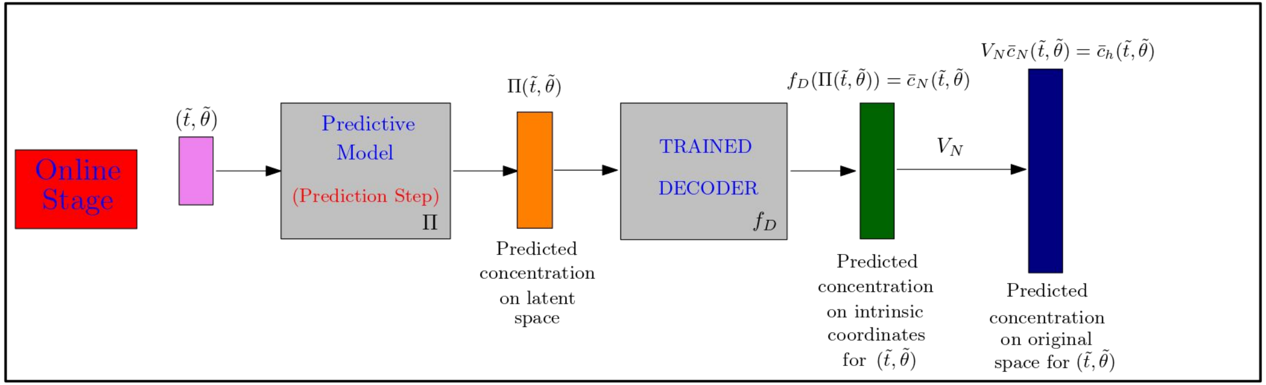
+ Related Works

Guo and Hesthaven - “Data-driven reduced order modeling for time-dependent problems” (2019)

Lee and Carlberg - “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders” (2020)

Wang, Vinuesa, et al. - “Towards optimal β-variational autoencoders combined with transformers for reduced-order modelling of turbulent flows” (2024)





[1]https://github.com/gf-barros/snapflow

Networks Parameters (after ablation studies)

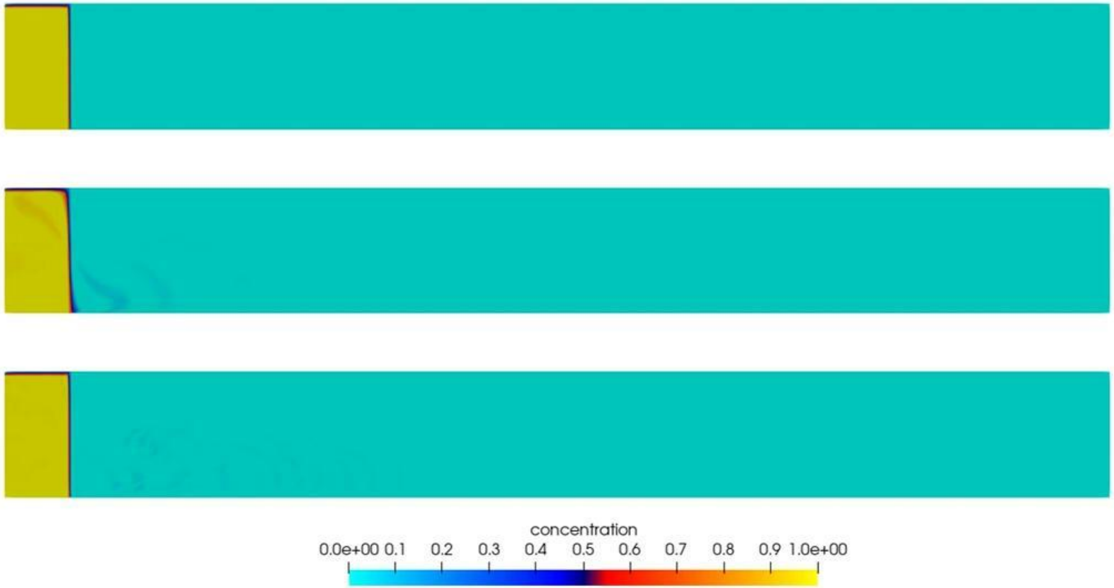
	Autoencoder	Neural Network
Epochs	5k	2k
Learning Rate	1×10^{-4}	$5 \times 10^{-3}(\times 0.75)/50$ ep.
Optimizer	ADAM	ADAM
Seq. Layers(Neurons)	$256, 128, 64, 32, 16$	$50, 50, 50, 50$
Activation fct. (hidden layers)	Leaky ReLU(0.1)	Sigmoid
Activation fct. (output layer)	Sigmoid	Sigmoid
Weight Initialization	Kaiming Normal	Kaiming Normal
Loss function	Smooth $L^1(\beta = 2.0)$	RMSE
Input dim.	270	270
Output dim.	2	16(POD-DL) or 270(POD)
Batch Size	25	6

Test set results: slope 5 degrees

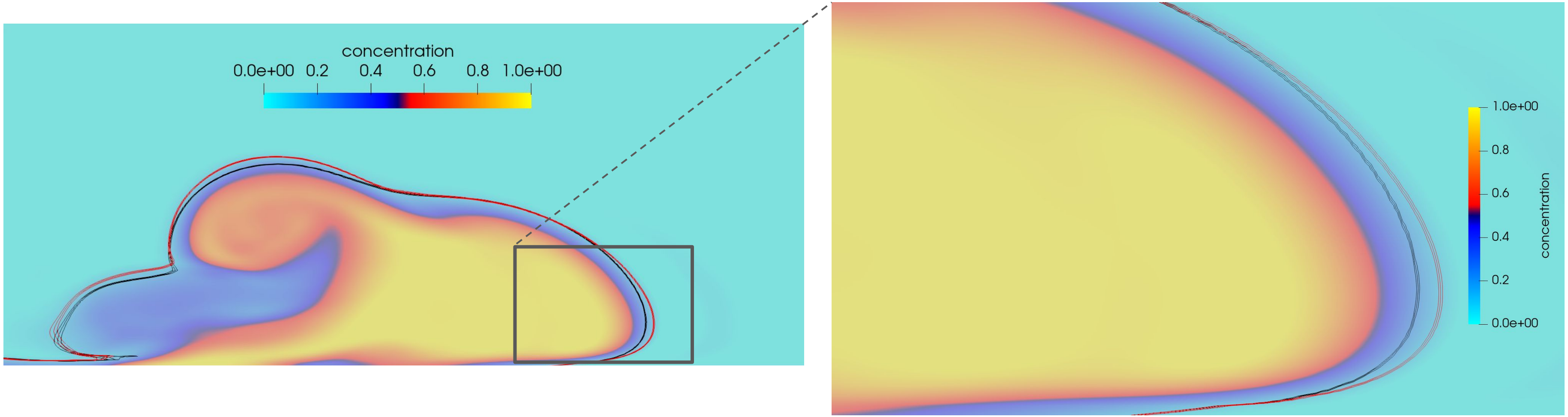
FOM
(Ground truth)

POD

POD-DL



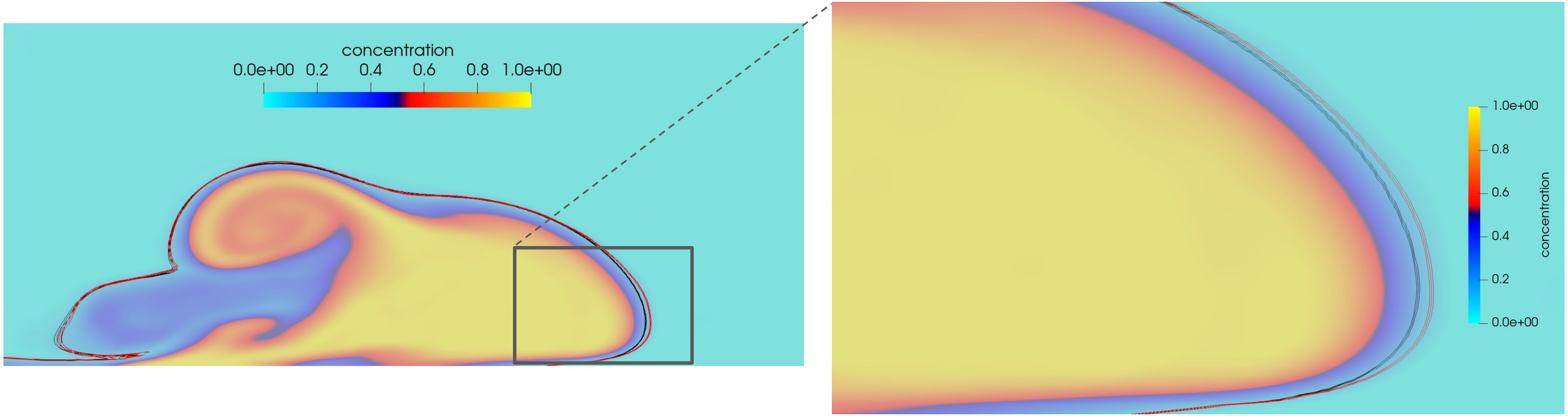
Isolines Prediction versus Ground Truth (POD only)



Figures: Isolines for the concentration values 0.09, 0.1, 0.11 @ time 6 - Test angle of 5 degrees

Black lines (**ground truth**) , **Red lines** (**POD prediction** - in background).

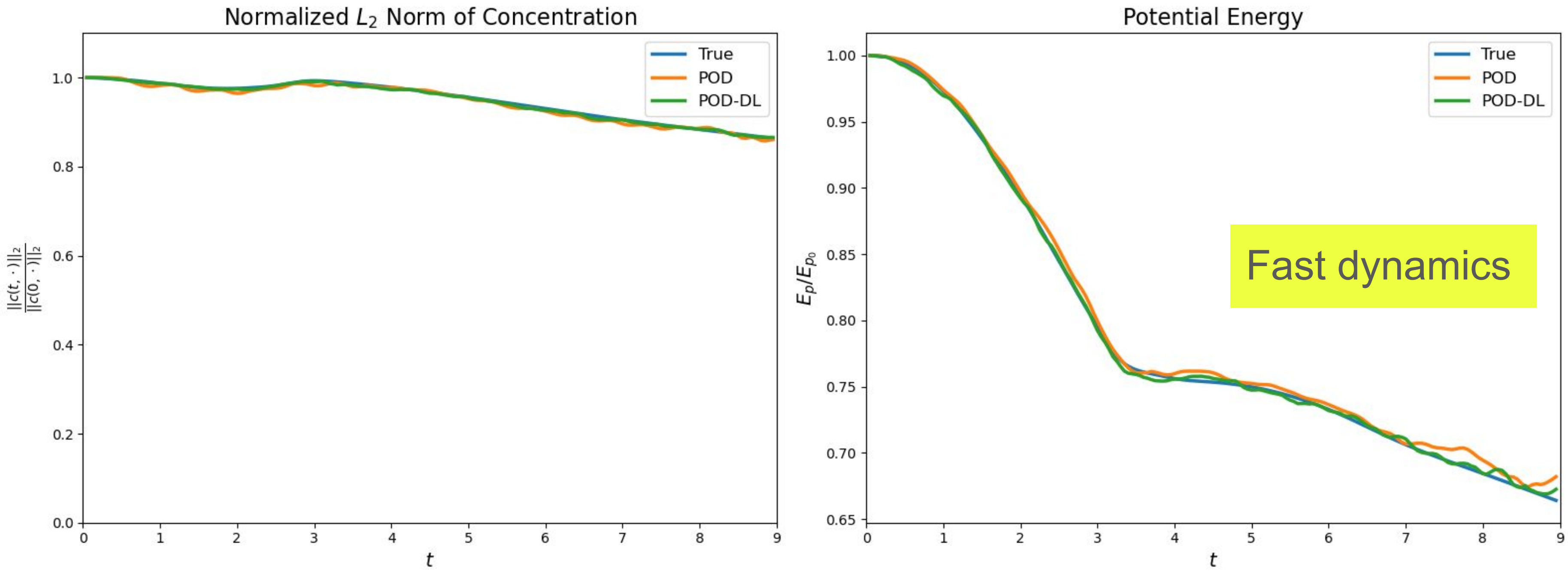
Isolines Prediction versus Ground Truth (POD-DL)



Figures: Isolines for the concentration values 0.09, 0.1, 0.11 @ time 6 - Test angle of 5 degrees

Black lines (ground truth), **Red lines (POD-DL prediction** - in background)

Time History of Global Quantities of Interest



FOM Computation Time (data generation): **30 h**
Offline Training Time: POD ~ **0.5 h**; POD-DL ~ **1.5 h**

On-line Prediction Speed-up: one angle - all times: **500X**
one angle - one time: **10,000X**

Time Data-Partitioning (after Cracco et al[1])

As an example, suppose we have 17 time steps for each angle:

$$[t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}]$$

We can define 3 partitions:

$$[t_1, t_2, t_3, t_4, t_5, t_6] \qquad [t_7, t_8, t_9, t_{10}, t_{11}, t_{12}] \qquad [t_{13}, t_{14}, t_{15}, t_{16}, t_{17}]$$

Also with **overlapping partitions**[2] (here with ⅓ of the original size for each side):

$$[t_1, t_2, t_3, t_4, t_5, t_6, {\color{red}t_7}, {\color{red}t_8}] \qquad [{\color{red}t_5}, {\color{red}t_6}, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, {\color{red}t_{13}}, {\color{red}t_{14}}] \qquad [{\color{red}t_{11}}, {\color{red}t_{12}}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}]$$

[1]Cracco, Stabile, et al., "Deep learning-based reduced-order methods for fast transient dynamics" arXiv:2212.07737v2 (2022);
[2]Diaz, Choi, and Heinkenschloss, "A fast and accurate domain decomposition nonlinear manifold reduced order model" Comput. Methods Appl. Mech. Engrg. 425 - 2024.

Time partitioning

$$[\theta_1, \theta_2, \ldots, \theta_M]$$

$$[\mathrm{t}_1, t_2, t_3, t_4, t_5, t_6] \qquad [t_7, t_8, t_9, t_{10}, t_{11}, t_{12}] \qquad [t_{13}, t_{14}, t_{15}, t_{16}, t_{17}]$$

$$\mathrm{Y}_1 = \begin{bmatrix} | & | & \ldots & | & | & \ldots & | & \ldots & | \\ c(t_1, \theta_1) & c(t_2, \theta_1) & \ldots & c(t_6, \theta_1) & c(t_1, \theta_2) & \ldots & c(t_6, \theta_2) & \ldots & c(t_6, \theta_M) \\ | & | & \ldots & | & | & \ldots & | & \ldots & | \end{bmatrix}$$

$$\mathrm{Y}_2 = \begin{bmatrix} | & | & \ldots & | & | & \ldots & | & \ldots & | \\ c(t_7, \theta_1) & c(t_8, \theta_1) & \ldots & c(t_{12}, \theta_1) & c(t_7, \theta_2) & \ldots & c(t_{12}, \theta_2) & \ldots & c(t_{12}, \theta_M) \\ | & | & \ldots & | & | & \ldots & | & \ldots & | \end{bmatrix}$$

$$\mathrm{Y}_3 = \begin{bmatrix} | & | & \ldots & | & | & \ldots & | & \ldots & | \\ c(t_{13}, \theta_1) & c(t_{14}, \theta_1) & \ldots & c(t_{17}, \theta_1) & c(t_{13}, \theta_2) & \ldots & c(t_{17}, \theta_2) & \ldots & c(t_{17}, \theta_M) \\ | & | & \ldots & | & | & \ldots & | & \ldots & | \end{bmatrix}$$

Time partitioning - with Overlap

$$[\theta_1, \theta_2, \ldots, \theta_M]$$

$$[\mathrm{t}_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8] \quad [t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}] \quad [t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}]$$

$$\mathrm{Y}_1 = \begin{bmatrix} | & | & \ldots & | & | & | & | & \ldots & | & | & | & \ldots & | \\ c(t_1,\theta_1) & c(t_2,\theta_1) & \ldots & c(t_6,\theta_1) & c(t_7,\theta_1) & c(t_8,\theta_1) & c(t_1,\theta_2) & \ldots & c(t_6,\theta_2) & c(t_7,\theta_2) & c(t_8,\theta_2) & \ldots & c(t_8,\theta_M) \\ | & | & \ldots & | & | & | & | & \ldots & | & | & | & \ldots & | \end{bmatrix}$$

$$\mathrm{Y}_2 = \begin{bmatrix} | & | & | & | & \ldots & | & | & | & | & \ldots & | & \ldots & | \\ c(t_5,\theta_1) & c(t_6,\theta_1) & c(t_7,\theta_1) & c(t_8,\theta_1) & \ldots & c(t_{12},\theta_1) & c(t_{13},\theta_1) & c(t_{14},\theta_1) & c(t_5,\theta_2) & \ldots & c(t_{14},\theta_2) & \ldots & c(t_{14},\theta_M) \\ | & | & | & | & \ldots & | & | & | & | & \ldots & | & \ldots & | \end{bmatrix}$$

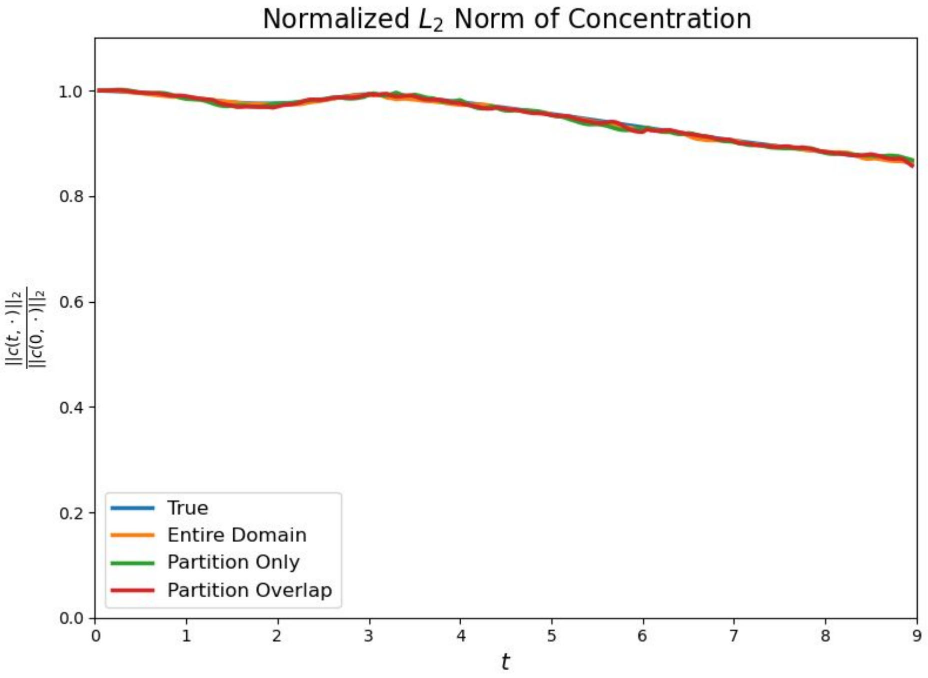
$$\mathrm{Y}_3 = \begin{bmatrix} | & | & | & \ldots & | & | & | & | & \ldots & | & | & | \\ c(t_{11},\theta_1) & c(t_{12},\theta_1) & c(t_{13},\theta_1) & \ldots & c(t_{17},\theta_1) & c(t_{11},\theta_2) & c(t_{12},\theta_2) & c(t_{13},\theta_2) & \ldots & c(t_{17},\theta_2) & \ldots & c(t_{17},\theta_M) \\ | & | & | & \ldots & | & | & | & | & \ldots & | & | & | \end{bmatrix}$$

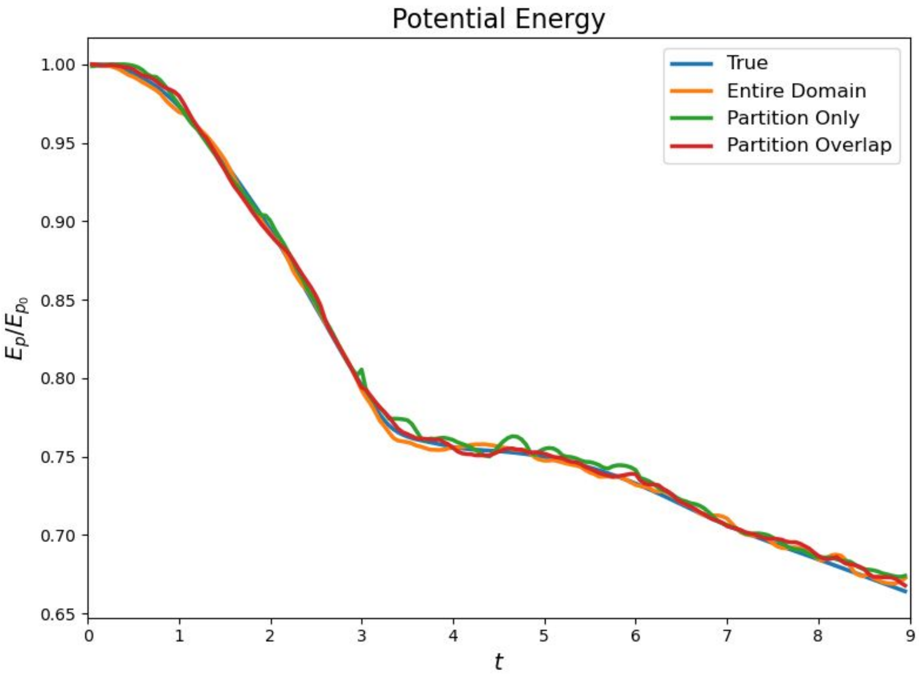
Network Parameters for Time Partitioning Case (after ablation studies)

	Part.1	Part.2	Part.3	Part.1 wO.	Part.2 wO.	Part.3 wO.
POD Modes	45	95	151	56	136	180
Layers AE	32, 16, 8, 4	64, 32, 16, 8	128, 64, 32, 16	32, 16, 8, 4	128, 64, 32, 16	128, 64, 32, 16

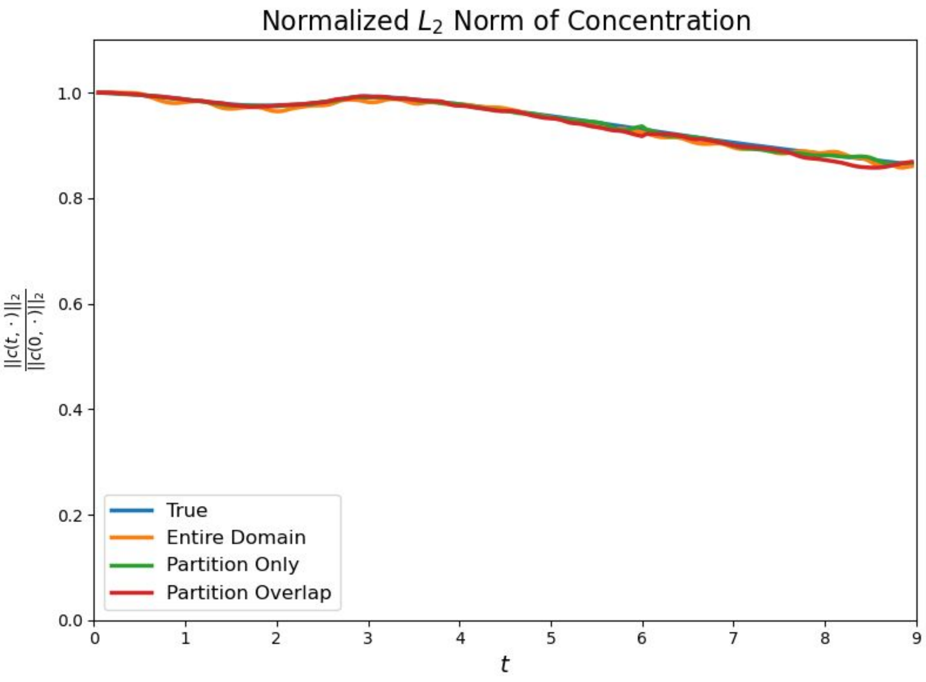
- As before, number of POD modes such that E.V.= 99.99999%
- Thus we may need to change network architecture when using overlapping
- Offline **Training** of each partition completely **independent** (possible **Parallelization**)
- Speed up on Training Networks based on the number of partitions
- On-line Prediction could also be done **independently**
- Each partition has a simpler surrogate model than original one
- Simpler hardware requirements for training models

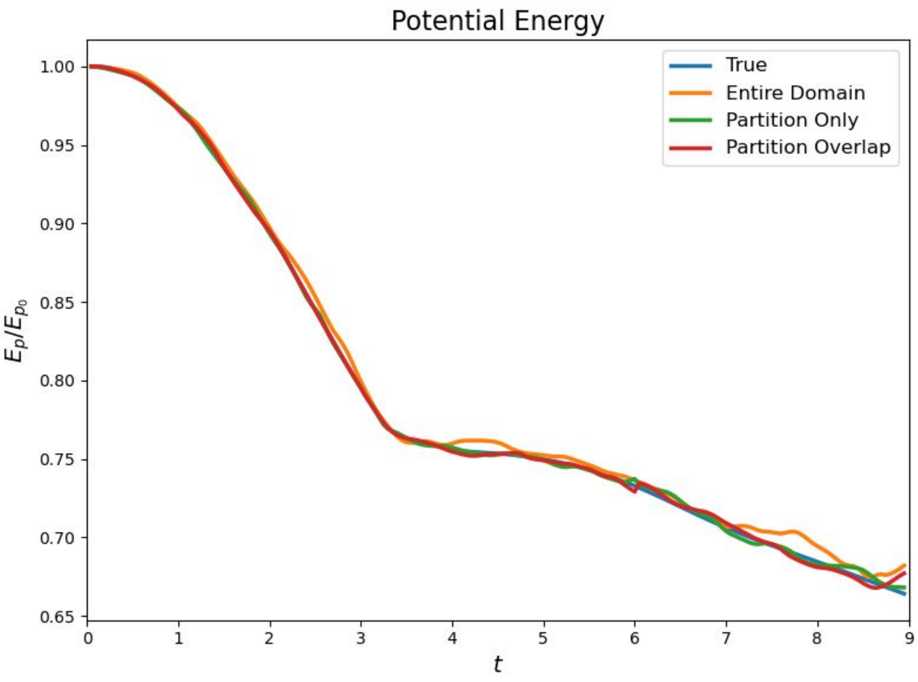
Time History of Global Quantities of Interest (POD)





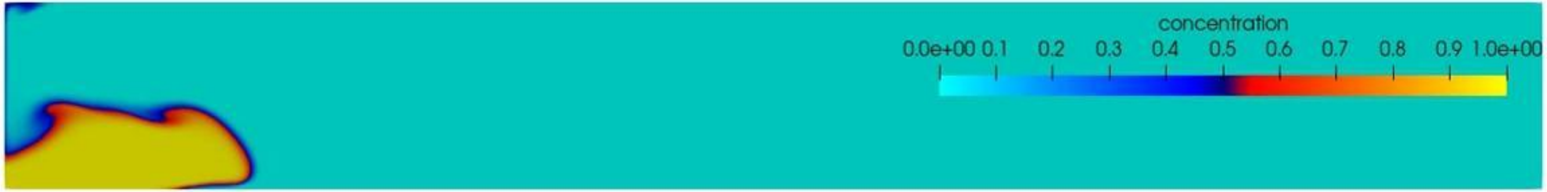
Time History of Global Quantities of Interest (POD-DL)



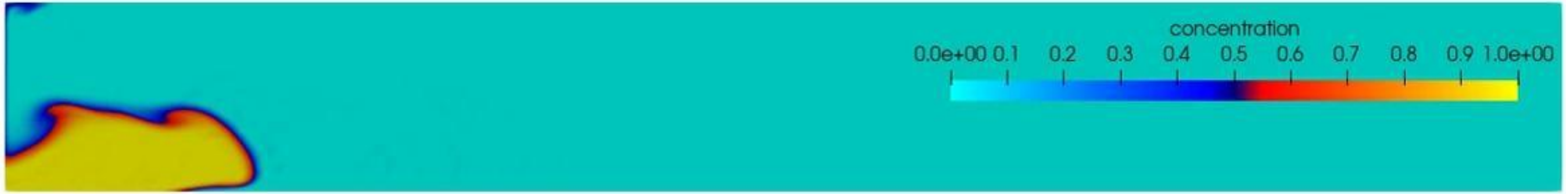


Test Case - slope of 5 degrees (Partition 2)

FOM



POD-DL (entire domain)



POD-DL (partition)

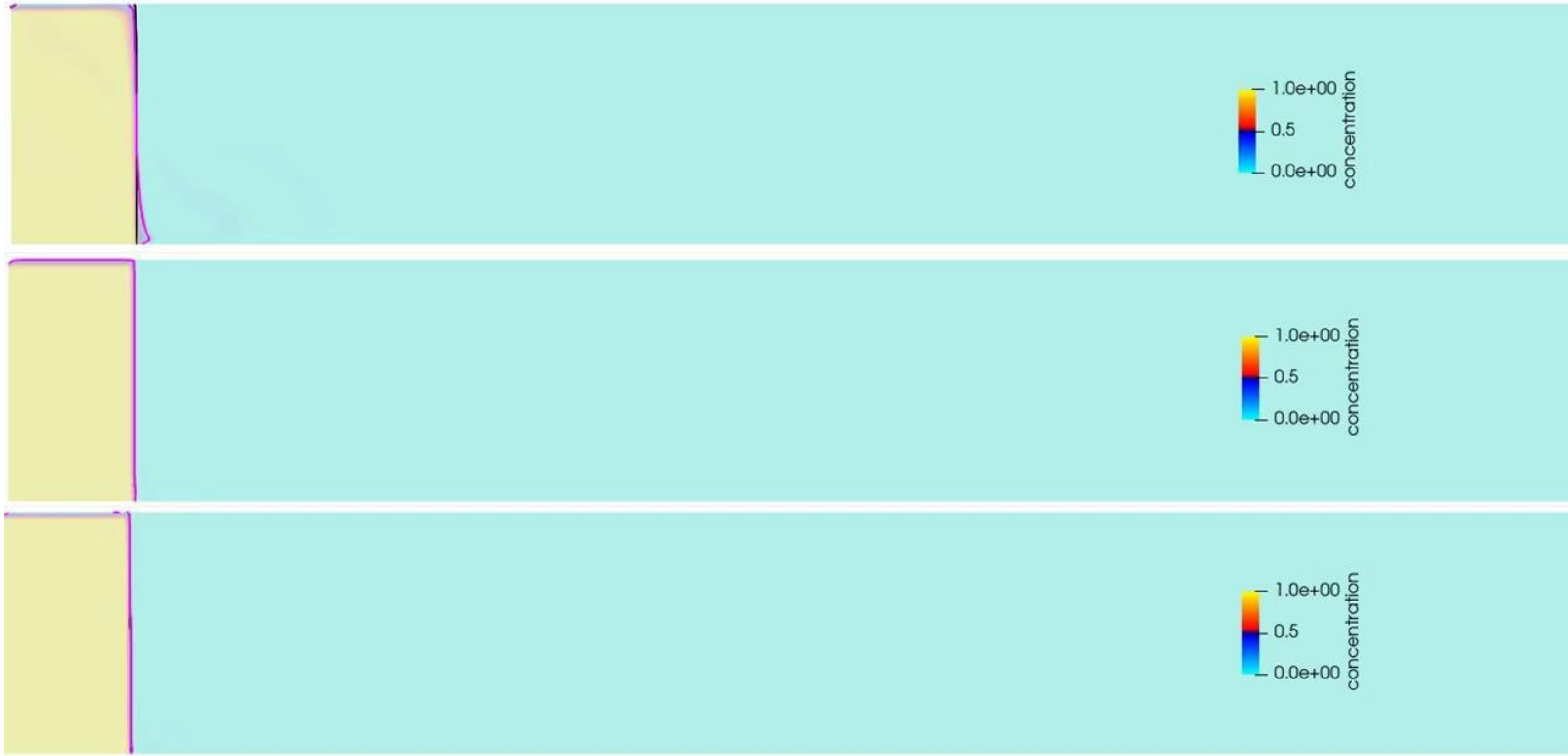
POD-DL (partition w/ overlap)

Concentration Time History (POD)

POD (entire domain)

POD (partition)

POD (partition w/ overlap)



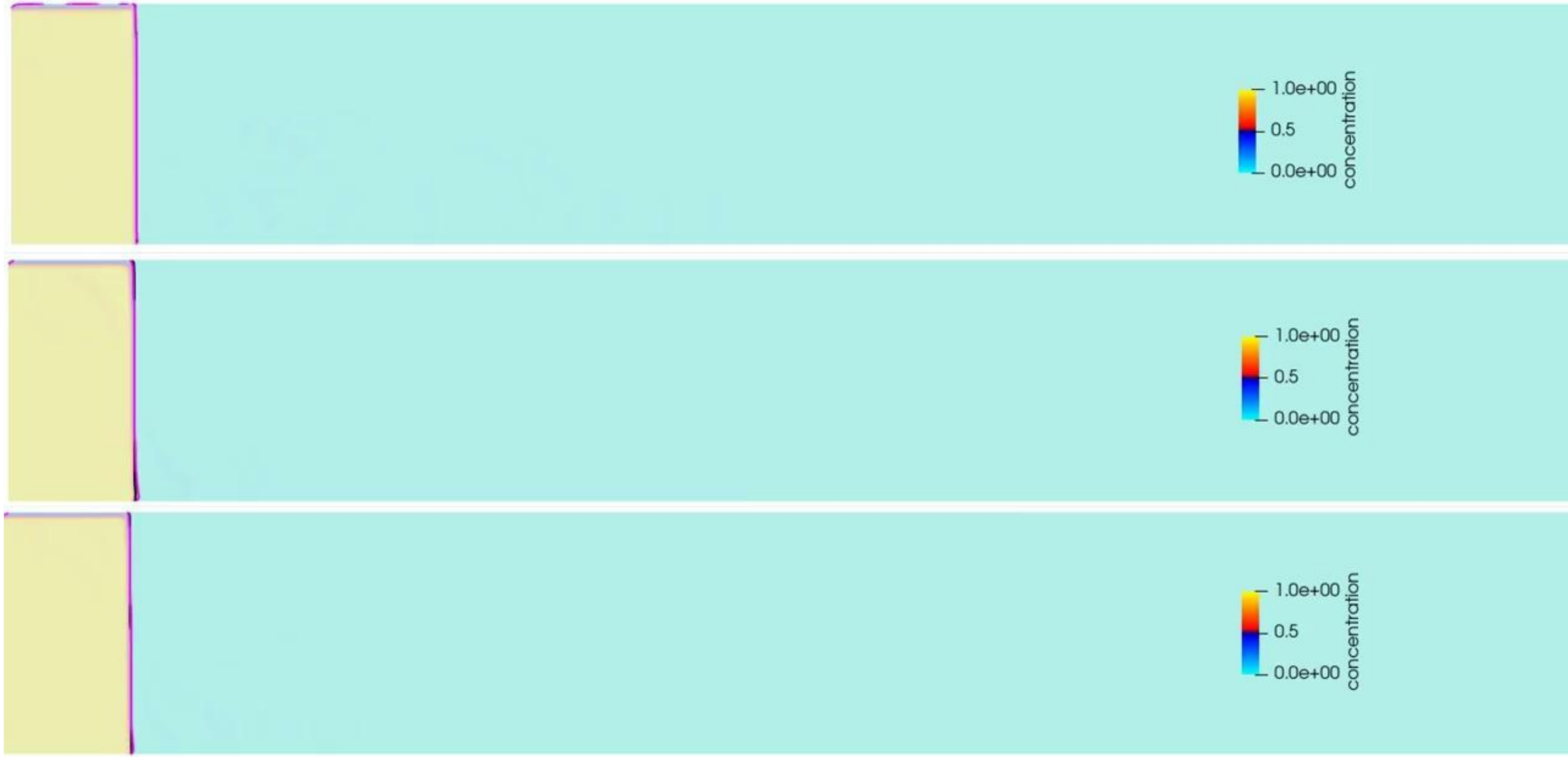
Isolines c=0.2 - **Black: FOM**, **Purple: ROM**

Concentration Time History (POD-DL)

POD-DL (entire domain)

POD-DL (partition)

POD-DL (partition w/ overlap)



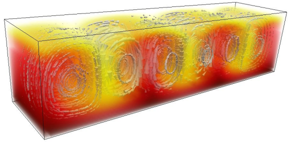
Isolines c=0.2 - **Black: FOM**, **Purple: ROM**

Final Comments and Conclusions

- Methodology demonstrated good performance in predicting sediment concentration for unseen angles and (times) (inside/outside);

- Errors are acceptable, smaller using partitioning (even better with overlapping):
  - This is confirmed by several measures: L2 norm, Isolines, Potential Energy;
  - Time partitioning maintains prediction capabilities, besides allowing focus on parts of the dynamics and parallelization;

- Difficult problem - coupled turbulent fluid flow and sediment transport:
  - Possible inclusion of lab. data - since technique is completely **data driven**;
  - Possible application to other coupled problems: Rayleigh-Benard (in preparation);
  - Develop a ROM for the deposition maps as a surrogate for UQ[1] (in preparation).
  - Test Latent Dynamics Networks[2]



[1]Guerra, Gabriel M., et al. "Uncertainty quantification in numerical simulation of particle-laden flows." *Computational Geosciences* 20 (2016): 265-281;
[2]Regazzoni, Francesco, et al. "Learning the intrinsic dynamics of spatio-temporal processes through Latent Dynamics Networks." Nature Communications 15.1 (2024): 1834.

Acknowledgements

CAPES

CNPq

BR PETROBRAS

FAPERJ
Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro

Slides and Videos available in: https://github.com/rmvelho/Eccomas2024

Contact: roberto.velho@nacad.ufrj.br