Multi-GPU Implementation of the Levy and Lindenbaum's Method for Streaming SVD with Application to Large Datasets of KHRTI Simulation

Lucas Lestandi[1], Yassin Ajanif[1], Aditi Sengupta[2], Bhavna Joshi[2], Luisa Rocha da Silva[1]

[1] École Centrale Nantes, Nantes Université, CNRS, GeM, UMR 6183, F-44000 Nantes, France
[2] Department of Mechanical Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand 826004, India

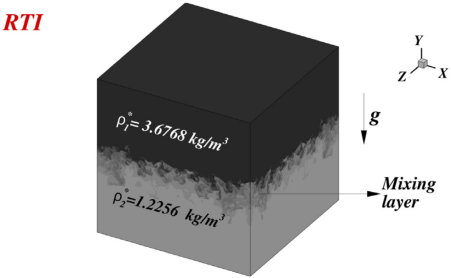
March 4, 2026

Outline

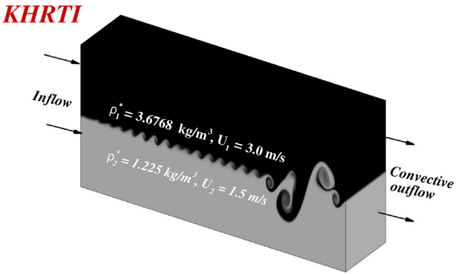
- Introduction
- A multi-GPU implementation of HLL-SVD
- Application to KHRTI simulation data

Improving the understanding of fluid flows instability

Extremely refined simulations help understanding the world albeit at the cost of skyrocketing compute time AND data size.



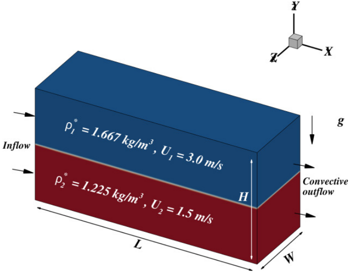
(a) 4.2 billion dofs, 1 snapshot => 157GB, 1E-7 timesteps


KHRTI
Inflow
ρ_1^* = 3.6768 kg/m^3, U_1 = 3.0 m/s
ρ_2^* = 1.225 kg/m^3, U_2 = 1.5 m/s
Convective outflow

(b) 480 millions dofs, 1 snapshot => 15GB, 1E-6 timesteps

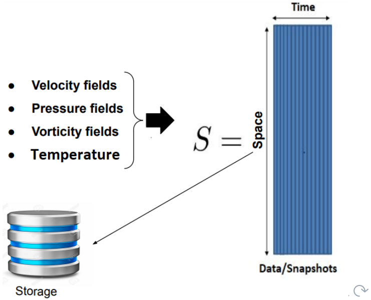
How to handle tens of terabytes of data in an international project? The KHRTI case

1. Run simulation locally
2. Store only necessary data → 2.4TB
3. Process it onsite → Develop suitable SVD
4. Share / Analyse the reduced basis to improve knowledge


Y
X
Z
g
Inflow
$\rho_1^* = 1.667$ kg/m³, $U_1 = 3.0$ m/s
$\rho_2^* = 1.225$ kg/m³, $U_2 = 1.5$ m/s
H
Convective outflow
L
W

⇒

DNS

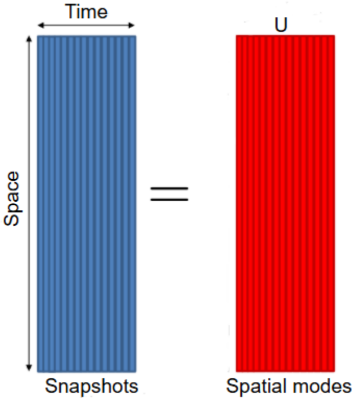


- Velocity fields
- Pressure fields
- Vorticity fields
- Temperature

$S =$

Singular Value Decomposition (SVD)

- Linked with physics (PCA, POD)
- Used as a preprocessor in ML
- Offers data compression
- Building block of ROM



Levy and Lindenbaum's Approach

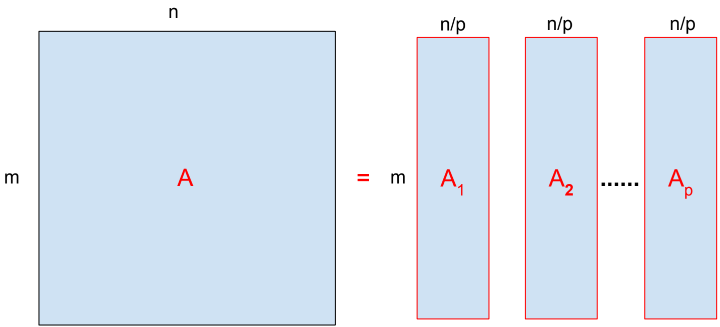
- *Approach introduced by Avraham Levy and Michael Lindenbaum in 1997*[1].
- *Recently used by Maulik and Mengaldo in PyParSVD (2021), an MPI-based implementation*[2]

[1] A Levy & M Lindenbaum. Sequential Karhunen-Loeve basis extraction and its application to images. In Proceedings of ICIP98 (Cat. No. 98CB36269), volume 2, pages 456–460. IEEE, 1998.

[2] R Maulik & G Mengaldo, PyParSVD: A Streaming, Distributed and Randomized Singular-Value-Decomposition Library, 2021, https://doi.org/10.48550/arXiv.2108.08845

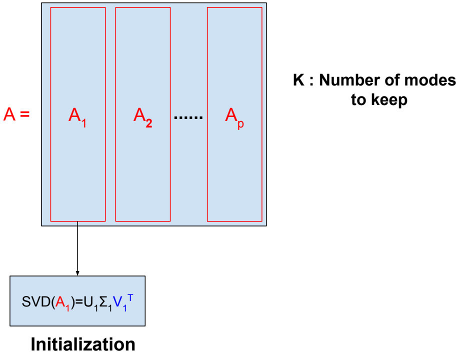
Levy and Lindenbaum’s Approach

Step 1: *Creating batches*



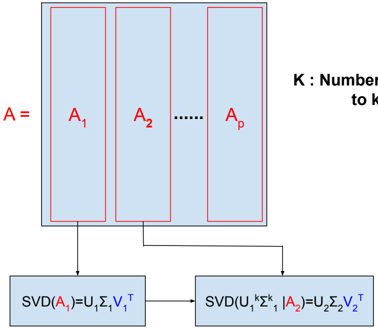
Levy and Lindenbaum's Approach

Step 2: Computation of the SVD


A =
A_1
A_2
......
A_p
K : Number of modes to keep
$\mathrm{SVD}(A_1)=U_1\Sigma_1V_1^T$
Initialization

Levy and Lindenbaum's Approach

Step 2: Computation of the SVD



$A = A_1 \; A_2 \; \ldots\ldots \; A_p$

$\mathrm{SVD}(A_1)=U_1\Sigma_1V_1^T$

$\mathrm{SVD}(U_1^k\Sigma_1^k \,|A_2)=U_2\Sigma_2V_2^T$

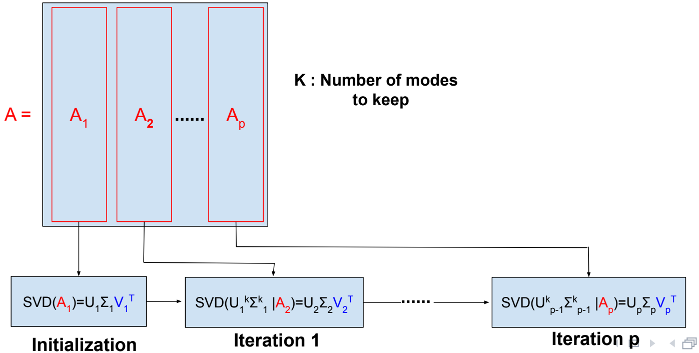
K : Number of modes to keep

Initialization

Iteration 1

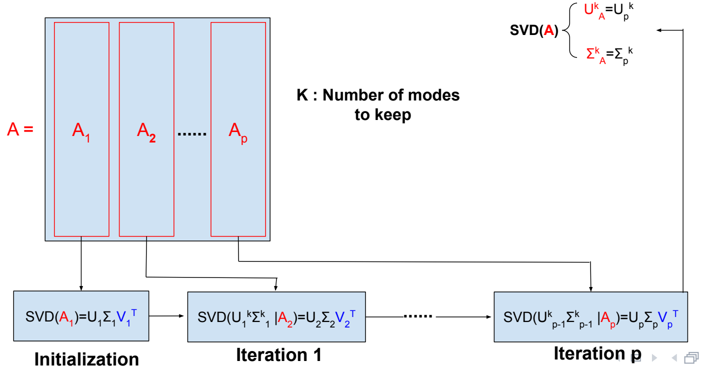
Levy and Lindenbaum's Approach

Step 2: Computation of the SVD

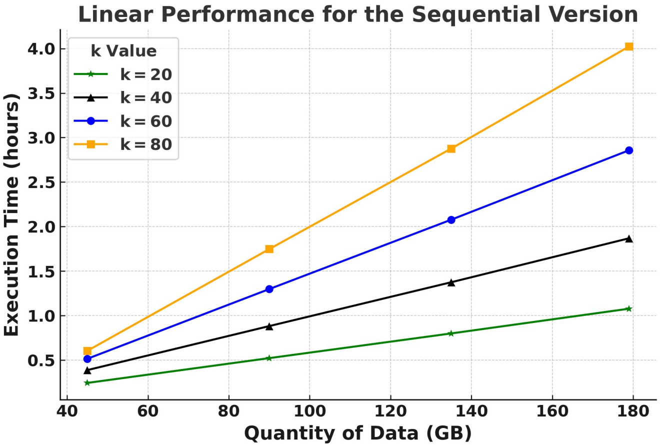

$A =$
A_1
A_2
......
A_p
K : Number of modes to keep
$SVD(A_1)=U_1\Sigma_1V_1^T$
$SVD(U_1^k\Sigma_1^k \mid A_2)=U_2\Sigma_2V_2^T$
......
$SVD(U_{p-1}^k\Sigma_{p-1}^k \mid A_p)=U_p\Sigma_pV_p^T$
Initialization
Iteration 1
Iteration p

Levy and Lindenbaum's Approach

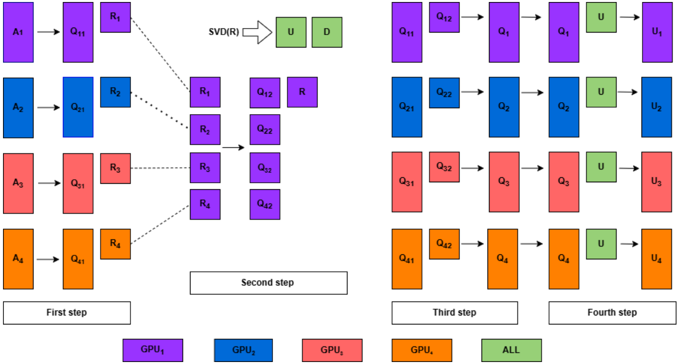
Step 2: Computation of the SVD


$A =$
A_1
A_2
......
A_p
K : Number of modes to keep
SVD(A)
$U^k_A = U_p^k$
$\Sigma^k_A = \Sigma_p^k$
$SVD(A_1)=U_1\Sigma_1V_1^T$
$SVD(U_1^k\Sigma^k_1 | A_2)=U_2\Sigma_2V_2^T$
......
$SVD(U^k_{p-1}\Sigma^k_{p-1} | A_p)=U_p\Sigma_pV_p^T$
Initialization
Iteration 1
Iteration p

The Sequential version scales linearly with time and k



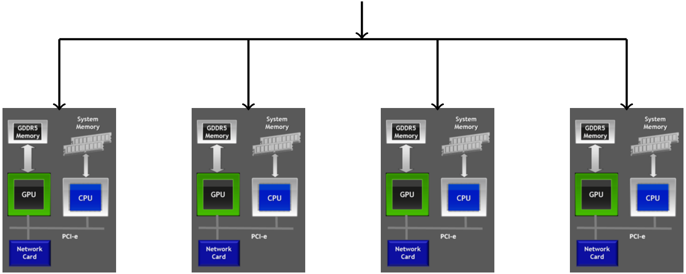
HLL-SVD :Parallelizing the batch SVD with Direct TSQR on Multiple GPUs



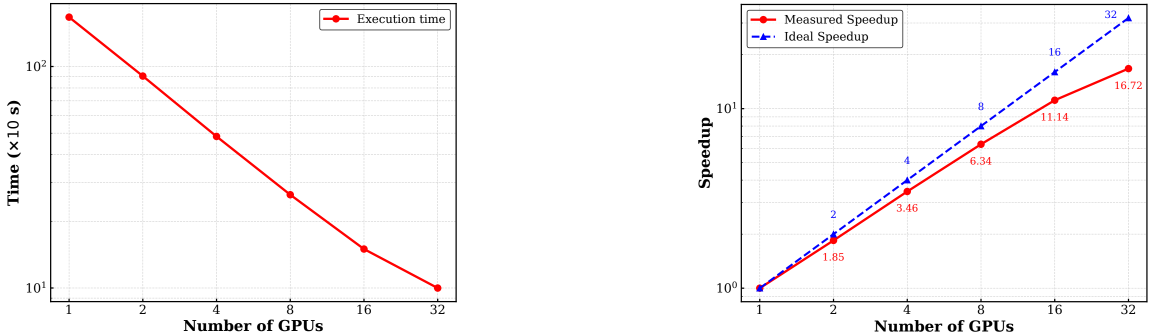
Hardware Used

*We perform our simulations on the **Jean-Zay** supercomputer, equipped with NVIDIA H100 GPUs with 80 GB of memory, two Intel Xeon Platinum 846 sockets (48 cores per socket), and 512 GB of RAM.*

```
mpirun -np 4 python3 script.py
```



Strong Scaling the HLL-SVD

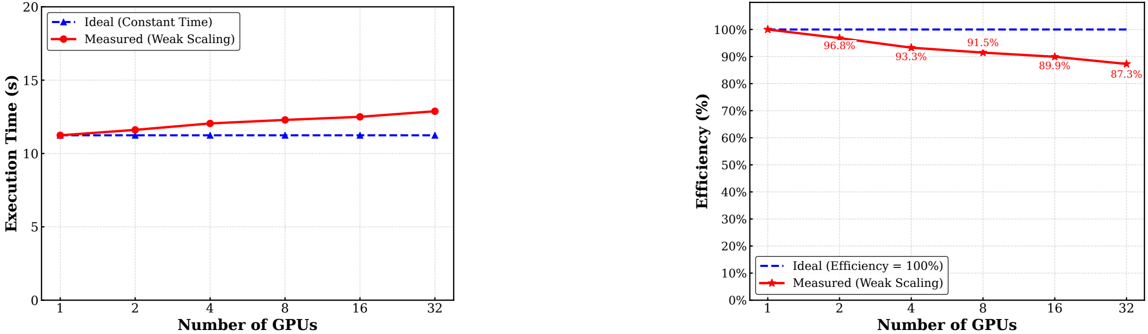


(a) Execution Time

(b) Speedup

Figure 2: Strong scaling performed on a random matrix of size $6E7 \times 3E3$ stored in single precision (32-bit), 150 batches(20 columns), k=10.

Weak scaling of the HLL-SVD



(a) Execution Time

(b) Speedup

Figure 3: Weak scaling performed on a random matrix of initial size $1E6 \times 3E3$ stored in single precision (32-bit), 150 batches(20 columns), k=10. Number of row is increased with nGPU.

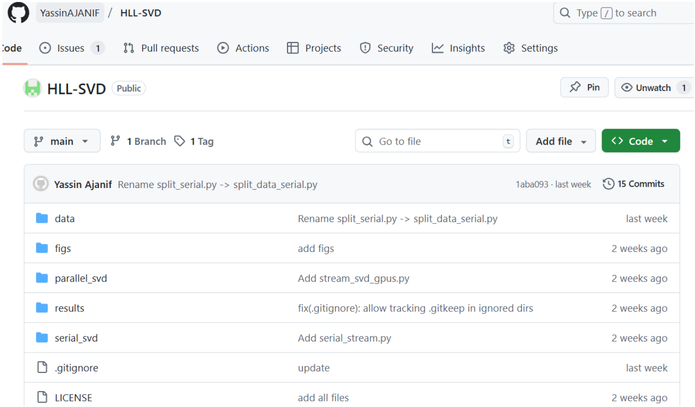
Comparing HLL-SVD and PyParSVD (MPI)

In this evaluation, we fix $k = 20$, while keeping the number of columns equal to 100. The number of rows is progressively increased in order to assess scalability.

Method	**Cores/GPUs**	10^5	10^6	10^7	10^8
PyParSVD	1 core	14.47	161.24	820.05	2651.38
PyParSVD	48 cores	0.37	10.01	57.58	113.66
PyParSVD	96 cores	0.24	4.47	28.44	57.10
HLL-SVD	1 GPU	1.38	7.83	29.06	47.45
HLL-SVD	2 GPUs	1.03	5.15	16.11	24.31

Table 1: Execution time (in seconds) as a function of the number of rows.

Code Repository on GitHub[3]



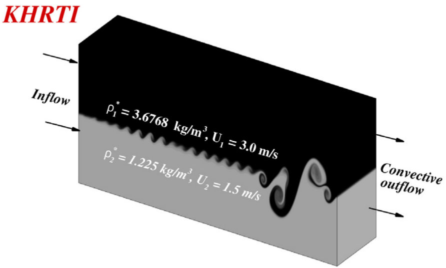
Yassin Ajanif Rename split_serial.py -> split_data_serial.py 1aba093 · last week 15 Commits

data	Rename split_serial.py -> split_data_serial.py	last week
figs	add figs	2 weeks ago
parallel_svd	Add stream_svd_gpus.py	2 weeks ago
results	fix(.gitignore): allow tracking .gitkeep in ignored dirs	2 weeks ago
serial_svd	Add serial_stream.py	2 weeks ago
.gitignore	update	last week
LICENSE	add all files	2 weeks ago

[3] https://github.com/YassinAJANIF/HLL-SVD

Application to KHRTI Data

- *DNS data of the Kelvin–Helmholtz Rayleigh–Taylor instability (KHRTI)* [a] *.*
- *Study only the pressure perturbation field* p'.
- SVD on **875** snapshots, each containing **480** million dofs.
- $k = 10$
- Single precision (32-bit).
- Run on 8 GPUs.


KHRTI
Inflow
$\rho_1^* = 3.6768$ kg/m^3, $U_1 = 3.0$ m/s
$\rho_2^* = 1.225$ kg/m^3, $U_2 = 1.5$ m/s
Convective outflow

[a] Joshi et al., Comparing the highly-resolved onset of Rayleigh–Taylor and Kelvin–Helmholtz Rayleigh–Taylor instabilities, European Journal of Mechanics - B/Fluids, 115, 2026, 204382, ISSN 0997-7546, https://doi.org/10.1016/j.euromechflu.2025.204382.

Evolution of the pressure disturbance

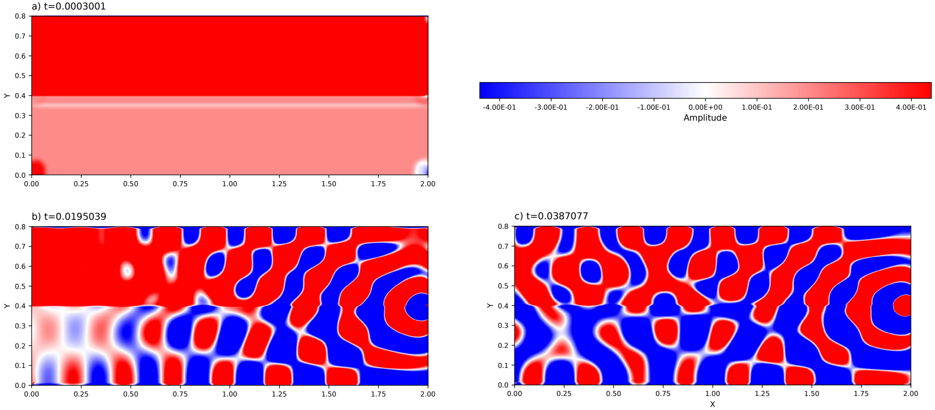
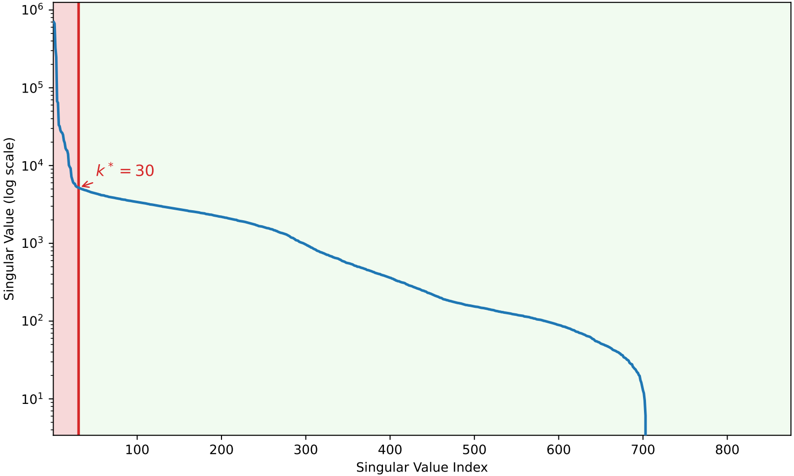
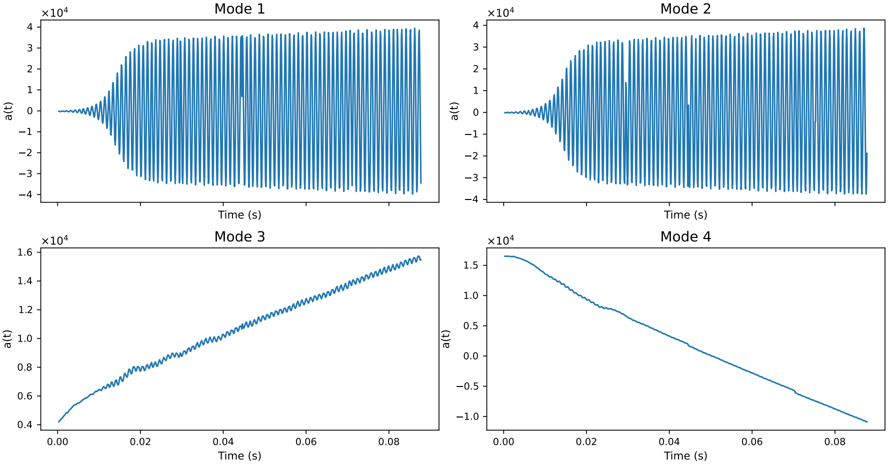


Figure 4: The evolution of pressure disturbances in $z = 10$ plane during early onset of the KHRTI.

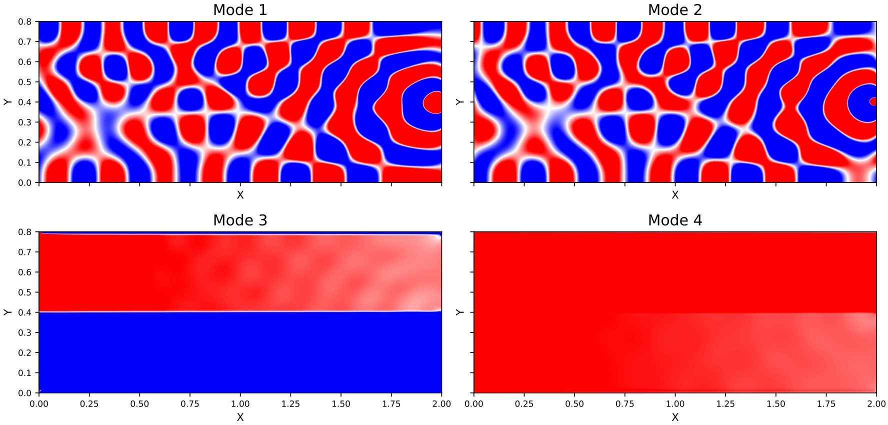
Singular values of the KHRTI data, computed using the snapshot method



Temporal Modes : $a_i = \sigma_i v_i^T$



Spatial Modes Φ_i



Reconstruction with 10 Modes — Snapshot at $t = 0.02050410$

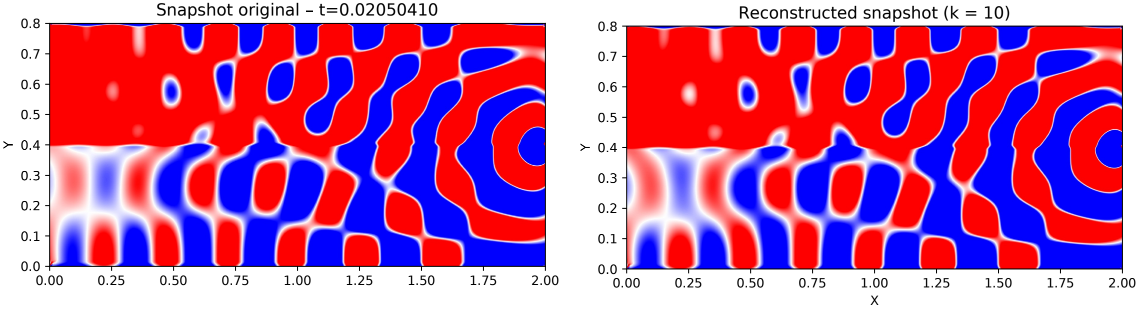


Figure 5: Reconstruction of a single snapshot at $z = 10$ with relative error $\varepsilon_{\mathrm{rel}} = 7.5584\%$.

Conclusions and Perspectives

- HLL-SVD is a new SVD implementation capable of processing large-scale datasets from fluid mechanics, leveraging heterogeneous HPC architectures (CPU/GPU).
- Future work : develop a reduced-order model tailored to very large datasets using HLL-SVD.

Ajanif et al. Preprint submitted to Comp. Phys. Comm.

Acknowledgements

This work was carried out under Indo-French **CEFIPRA-69T05-1** in collaboration with Prof. Aditi Sengupta at IIT-ISM Dhanbad.
Several computing centers were used to complete this work : **GLiCID**, **MesoNet**, and **Jean-Zay**.

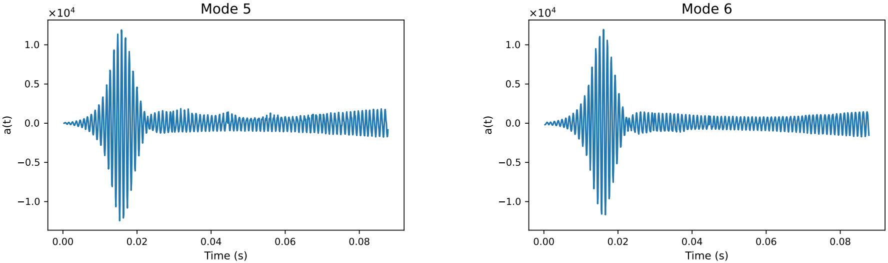
GLiCID

MESONET
le mésocentre des mésocentres

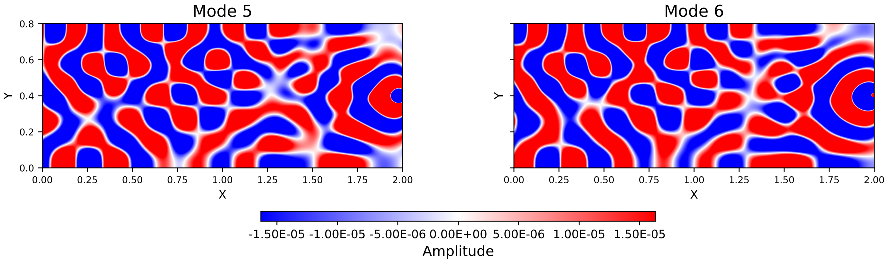
CEFIPRA

Thank You for Your Attention!

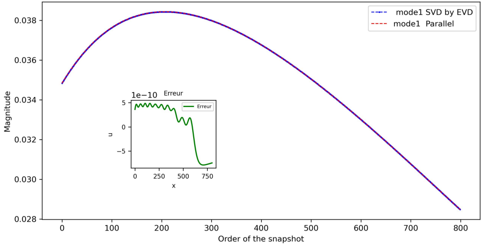
Temporal Modes : $a_i = \sigma_i v_i^T$

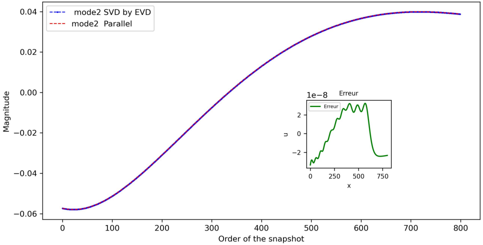


Spatial Modes Φ_i

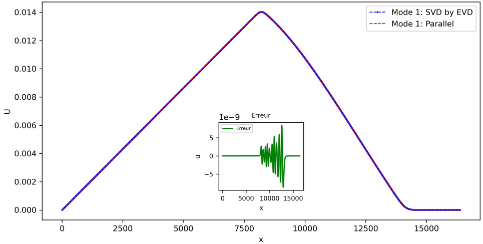


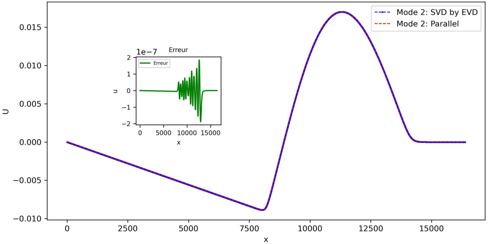
Method precision - temporal modes





Method precision - Spatial modes





GPU H100 details

- Memory : 80 GB.
- Memory Type : HBM3