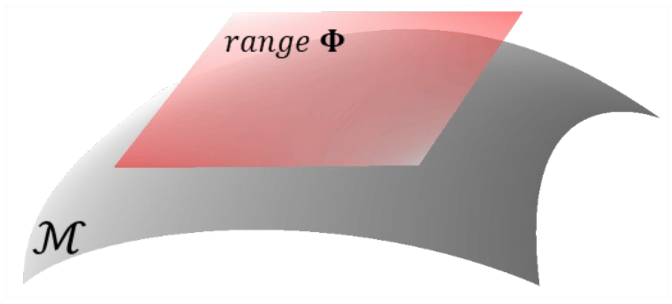
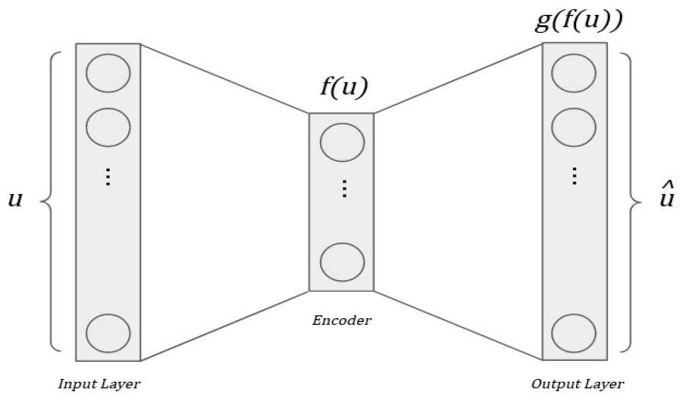
From Linear Mappings to Deep Learning for Model Reduction of Numerical Simulations of Industrial Interest:





Mr. Raul Bravo

Prof. Riccardo Rossi

Prof. Joaquin Hernandez

Mr. Carlos Roig

CIMNE

EXCELENCIA SEVERO OCHOA

KRATOS MULTI-PHYSICS

Presenting ourselves

Prof. Riccardo Rossi
UPC BarcelonaTech
CIMNE
Kratos co-founder
rrossi@cimne.upc.edu

Prof. Joaquin Hernandez
Aerospace Engineering School
UPC BarcelonaTech
CIMNE
jhortega@cimne.upc.edu

Kratos github site

Raul Bravo
PhD Student
Projection-based ROMs
jrbravo@cimne.upc.edu

Carlos Roig
PhD Student
Deep Learning ROMs
croig@cimne.upc.edu

CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Objetives of the talk

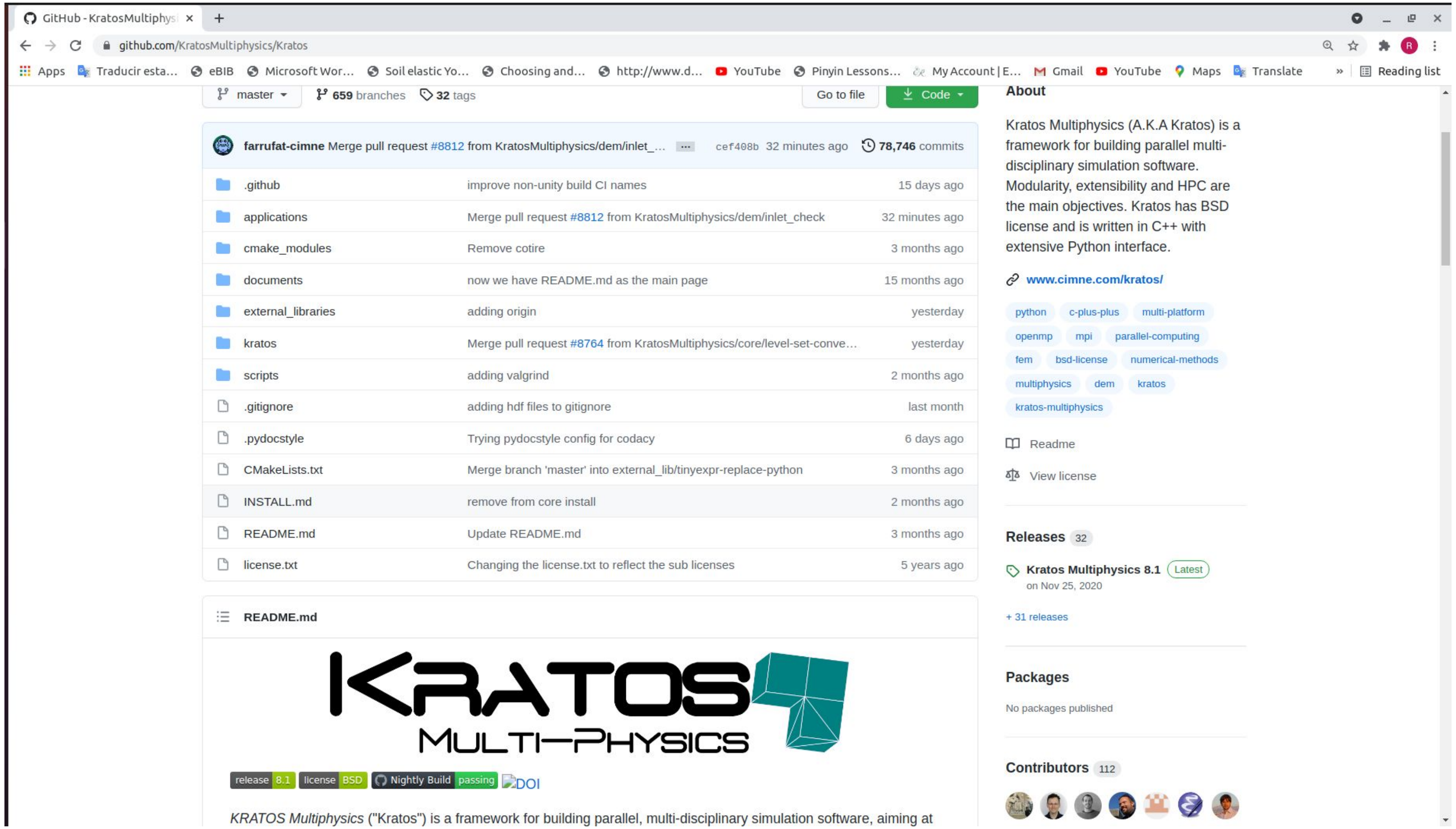
- Presenting a ROM framework implemented on a powerful open source FEM software.

FOM: $\boldsymbol{u} \in \mathbb{R}^n$ **ROM**: $\boldsymbol{q} \in \mathbb{R}^k$

- POD $\boxed{\boldsymbol{u} \approx \boldsymbol{\Phi}\, \mathbf{q}}$
- Local POD $\boxed{\boldsymbol{u}_{new} \approx \boldsymbol{u}_{old} + \boldsymbol{\Phi}^{\mathbf{i}}\, \mathbf{q}}$
- Autoencoders $\boxed{\boldsymbol{u} \approx \boldsymbol{g}(\boldsymbol{q})}$

CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Kratos Project



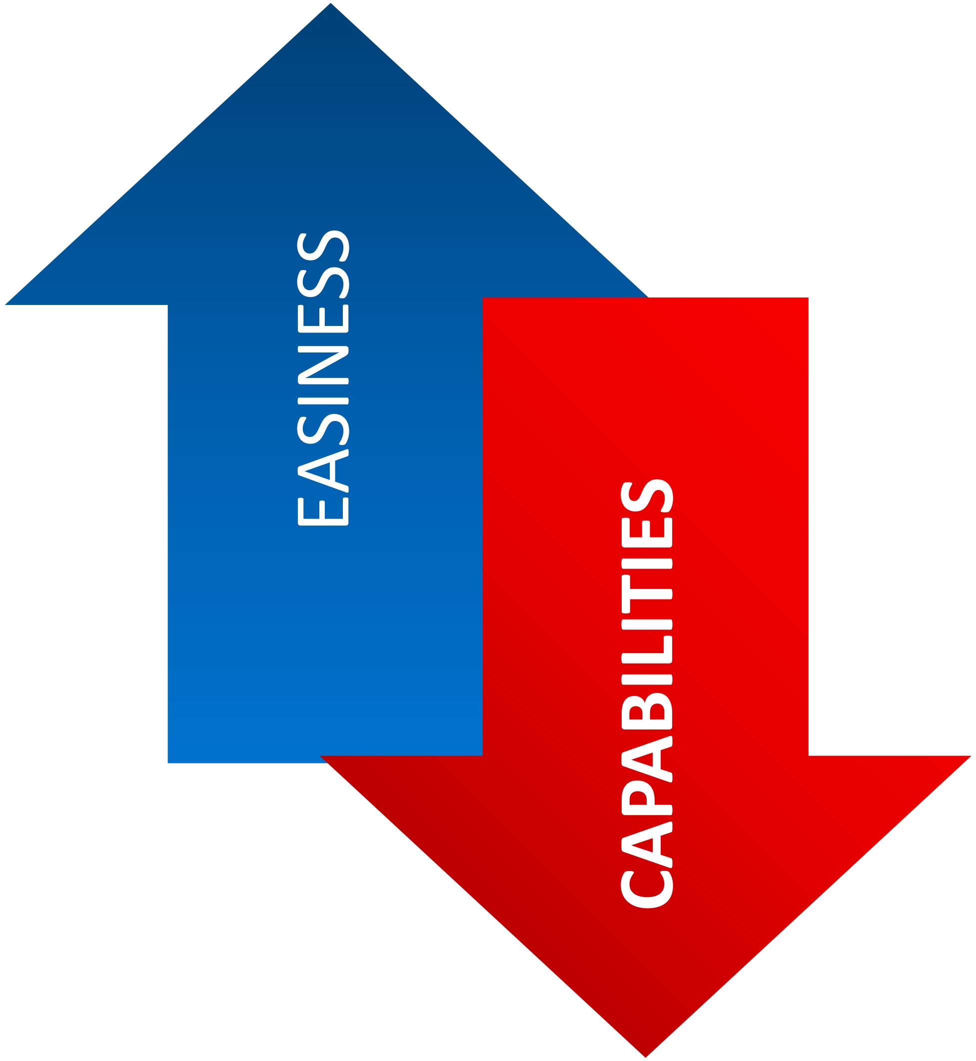
CIMNE
EXCELENCIA SEVERO OCHOA
KRATOS MULTI-PHYSICS

Using Kratos

Graphical Interface

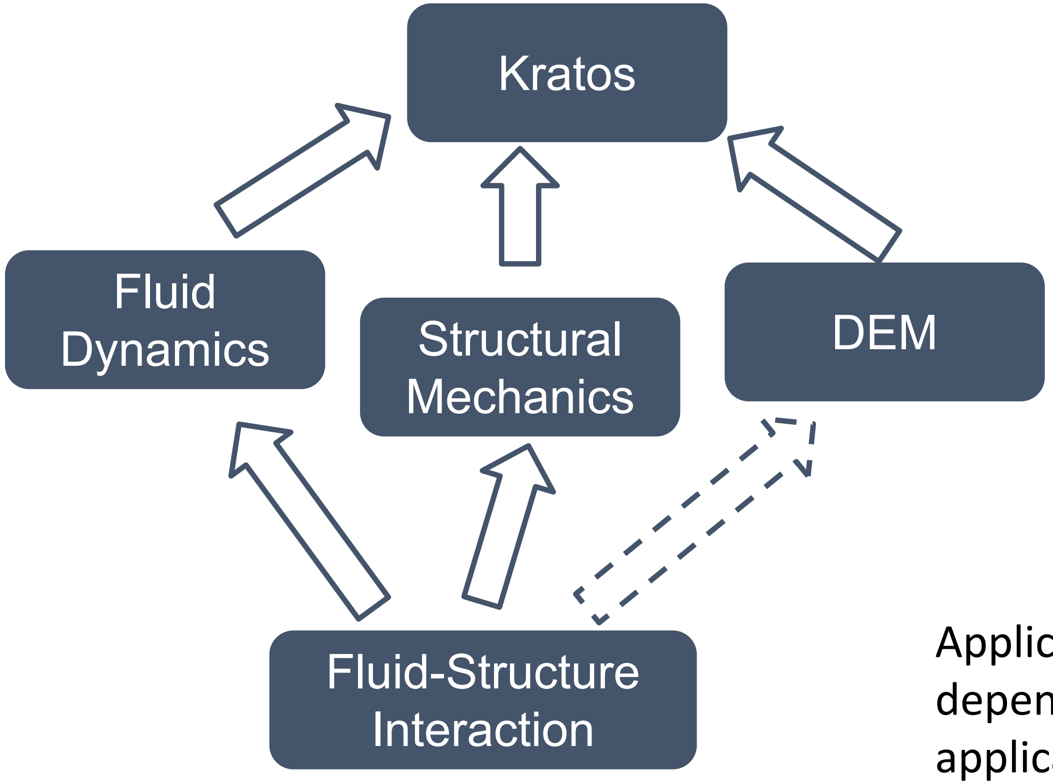
Python Interface

Applications



CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

How is Kratos structured?

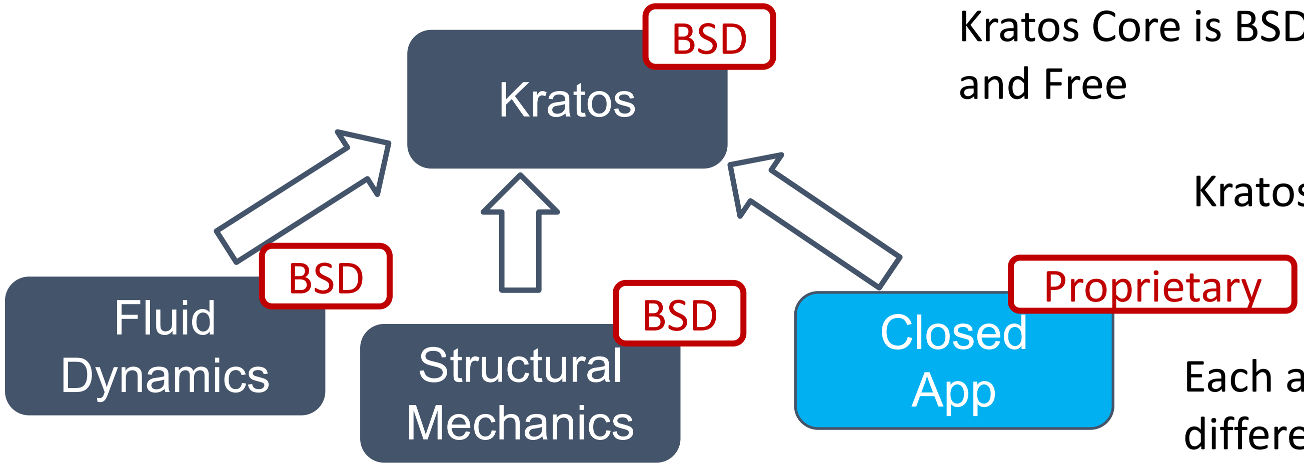


Numerical and programming core

Physics of the problem

Applications can depend on other applications

Kratos Framework
Flexible License



Kratos Core is BSD
and Free

Kratos main apps are also BSD

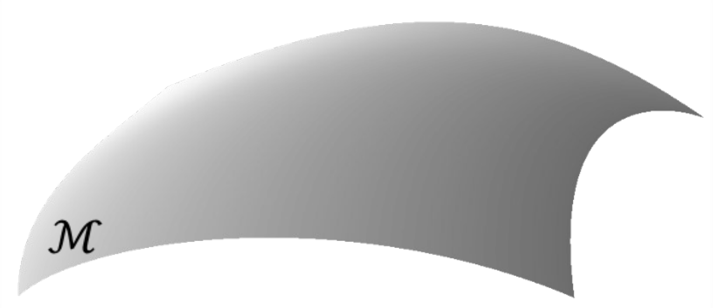
Each application can have
different license
Some can be closed

Proper Orthogonal Decomposition

Full Order Model (FOM)

$$A\,u = b$$

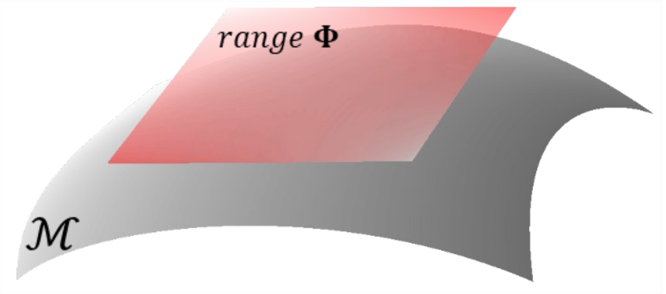
Solution manifold: $\mathcal{M} = \{\, \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \,\} \subset \mathbb{R}^n$



Let $\boldsymbol{u} \approx \boldsymbol{\Phi}\,\mathbf{q}$

Reduced Order Model (ROM)

$$\boldsymbol{\Phi}^T A\, \boldsymbol{\Phi}\, q = \boldsymbol{\Phi}^T b$$

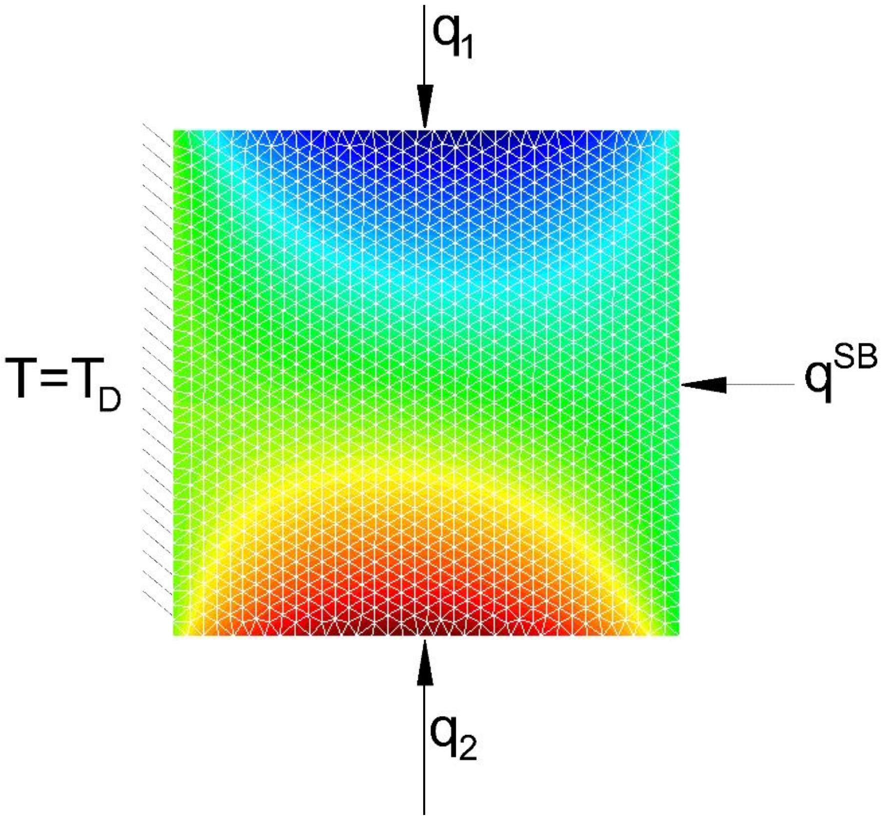


A MUCH SMALLER SYSTEM! $A^*\, q = b^*$

CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Proper Orthogonal Decomposition

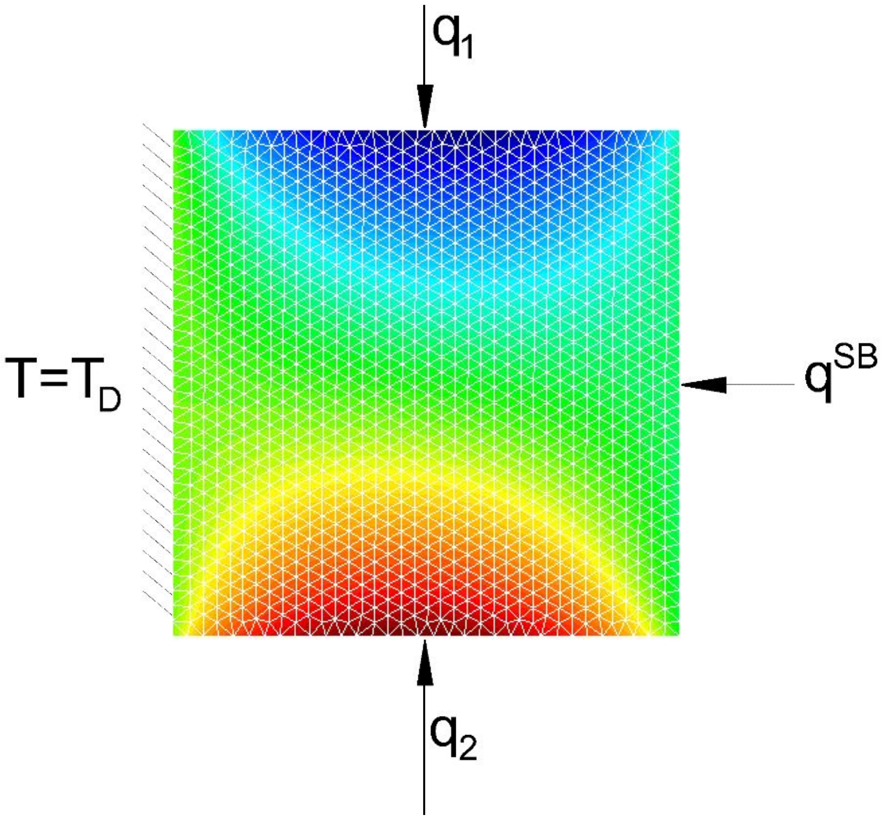
Solve the FOM using Finite Elements

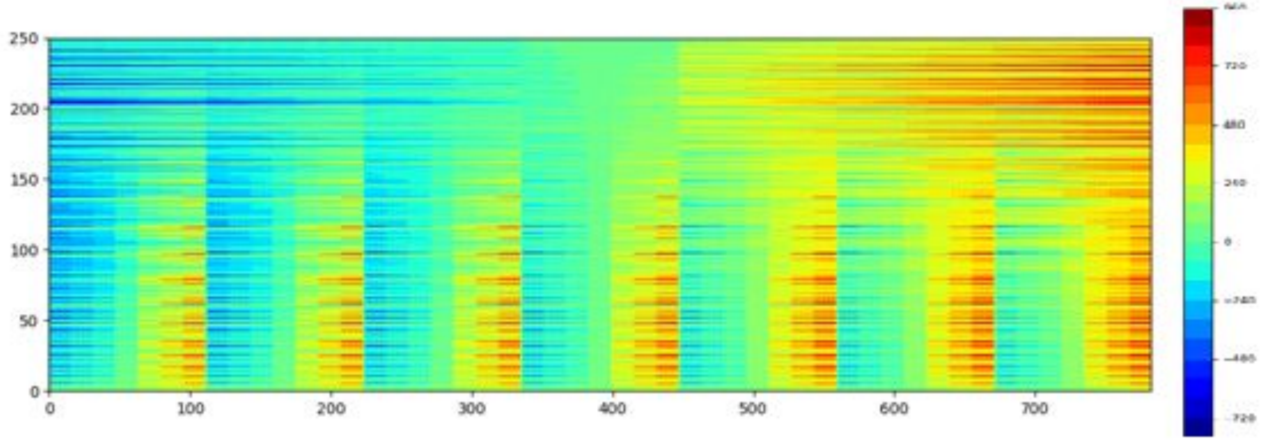


$$\boldsymbol{\mu} = (T_D, q_1, q_2, q_{SB}) \in \mathcal{P}$$

Proper Orthogonal Decomposition

- Store the nodal solution $\boldsymbol{u}$ the Snapshots matrix $\boldsymbol{S} = [\,\boldsymbol{u}_1 \boldsymbol{u}_2 \ldots \boldsymbol{u}_p\,]$

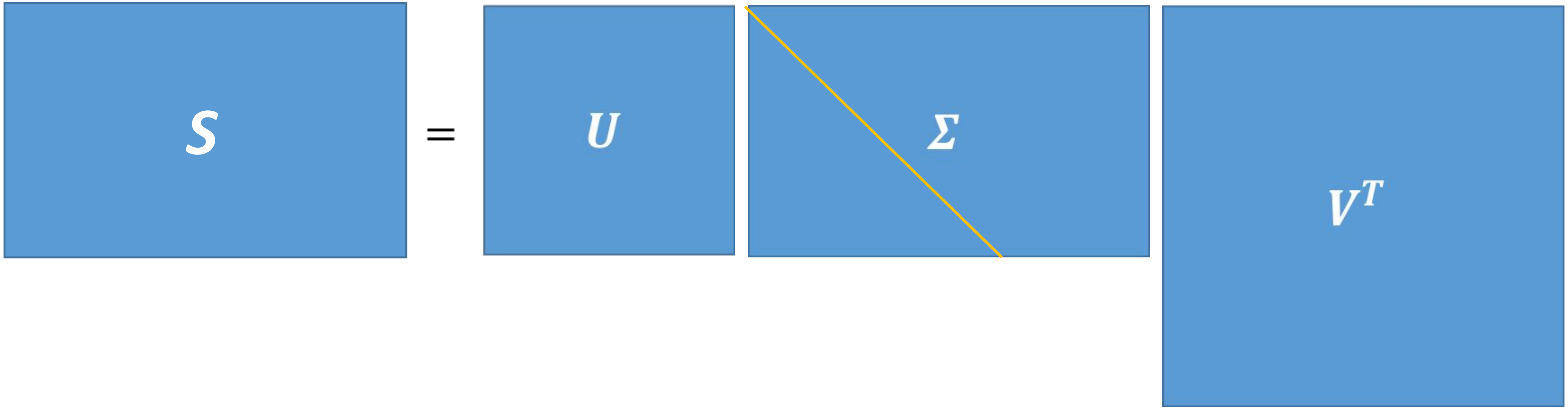




Each column collects the temperature on all the nodes in the mesh, for a given choice of the fluxes $q_1\ q_2\ q_{sb}$

Proper Orthogonal Decomposition

- Take the SVD of $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \approx \boldsymbol{U}_k\boldsymbol{\Sigma}_{\mathrm{k}}\mathbf{V}_{\mathrm{k}}^{\mathrm{T}}$



Proper Orthogonal Decomposition

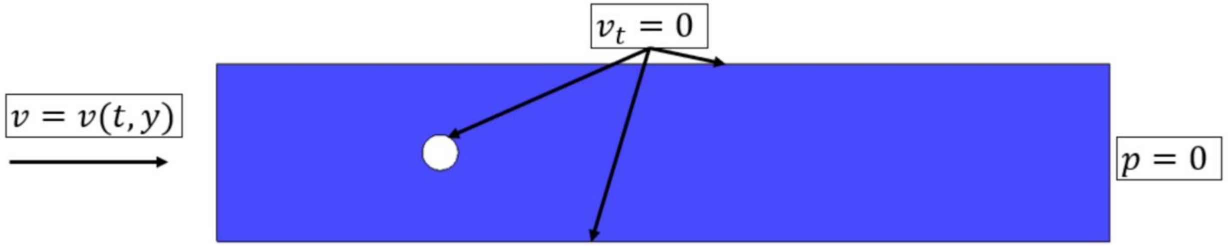
- Take the SVD of $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \approx \boldsymbol{U}_k\boldsymbol{\Sigma}_{\mathrm{k}}\mathbf{V}_{\mathrm{k}}^{\mathrm{T}}$

Proper Orthogonal Decomposition

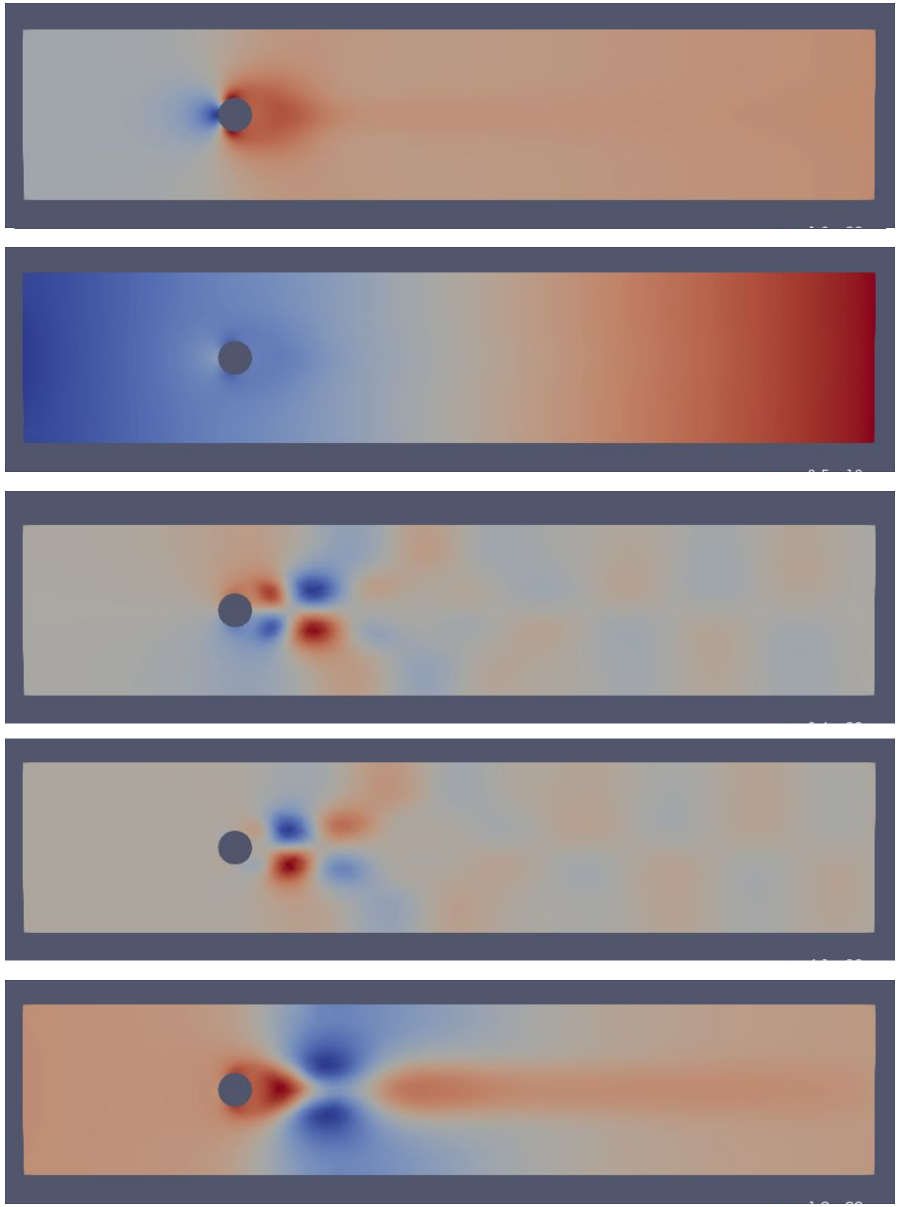
- Take the SVD of $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \approx \boldsymbol{U}_k\boldsymbol{\Sigma}_{\mathbf{k}}\mathbf{V}_{\mathbf{k}}^{\mathrm{T}}$

$$\boldsymbol{\Phi} := \boldsymbol{U}_k$$

Example in CFD



$\mathbf{\Phi} =$

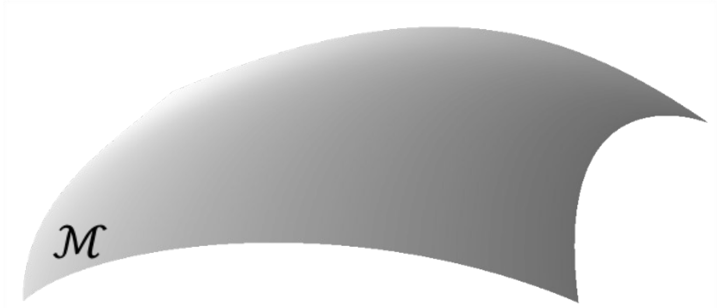


Proper Orthogonal Decomposition

Full Order Model (FOM)

$$A \, u = b$$

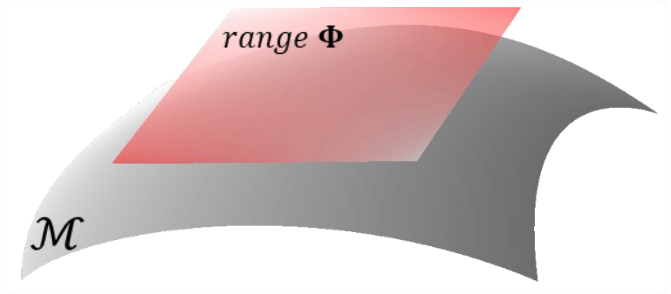
Solution manifold: $\mathcal{M} = \{\, \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \,\} \subset \mathbb{R}^n$



Let $\boldsymbol{u} \approx \boldsymbol{\Phi} \, \mathbf{q}$

Reduced Order Model (ROM)

$$\boldsymbol{\Phi}^T \, A \, \boldsymbol{\Phi} \, q = \boldsymbol{\Phi}^T \, b$$



A MUCH SMALLER SYSTEM! $A^* \, q = b^*$

PROBLEM: **STILL EXPENSIVE TO MOUNT THE SYSTEM**

CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Hyper-reduction

- In general, a second reduction layer is required. The goal is to find a **subset of elements and corresponding weights** by solving an optimization problem [1].

$$(\boldsymbol{E}, \boldsymbol{W}) = \arg min \ \| \ \boldsymbol{J}(\boldsymbol{R}, \boldsymbol{\Phi}) \ \|_2^2$$
$$subject\ to\ \boldsymbol{W} \succ \boldsymbol{0}$$

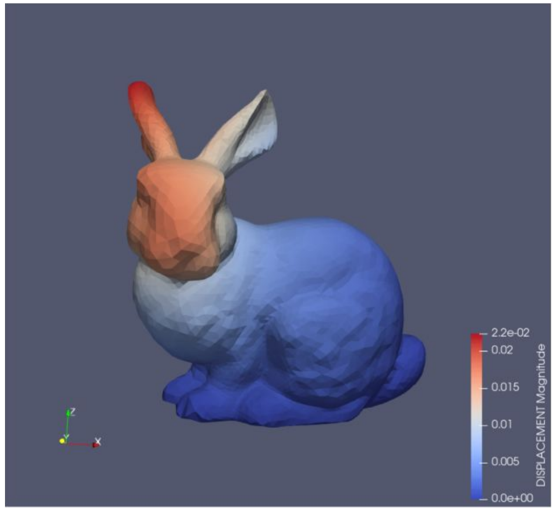
NP-HARD. Solving via greedy procedure

Hyper-reduction

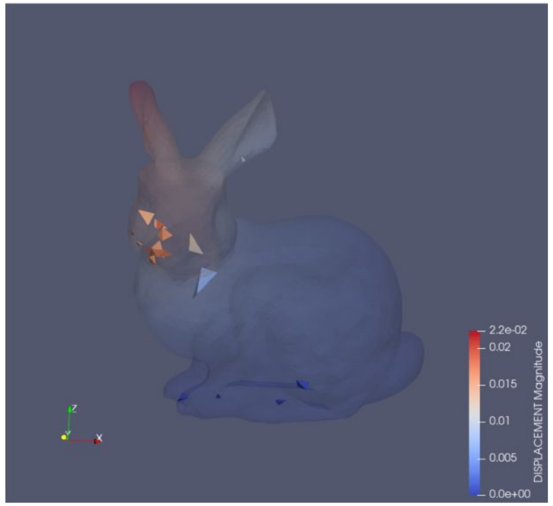
Assembly comparison ROM vs HROM:

$$\left(\prod_{e=1}^{n\,elem} A_e\right)\boldsymbol{u} = \prod_{e=1}^{n\,elem} b_e \quad \Longrightarrow \quad \left(\sum_{e\in E} \boldsymbol{\Phi}_e^T \boldsymbol{A}_e \boldsymbol{\Phi}_{\mathbf{e}}\, \omega_e\right)\boldsymbol{q} = \sum_{e\in E} \boldsymbol{\Phi}_{\mathbf{e}}^T \boldsymbol{b}_e \omega_e$$

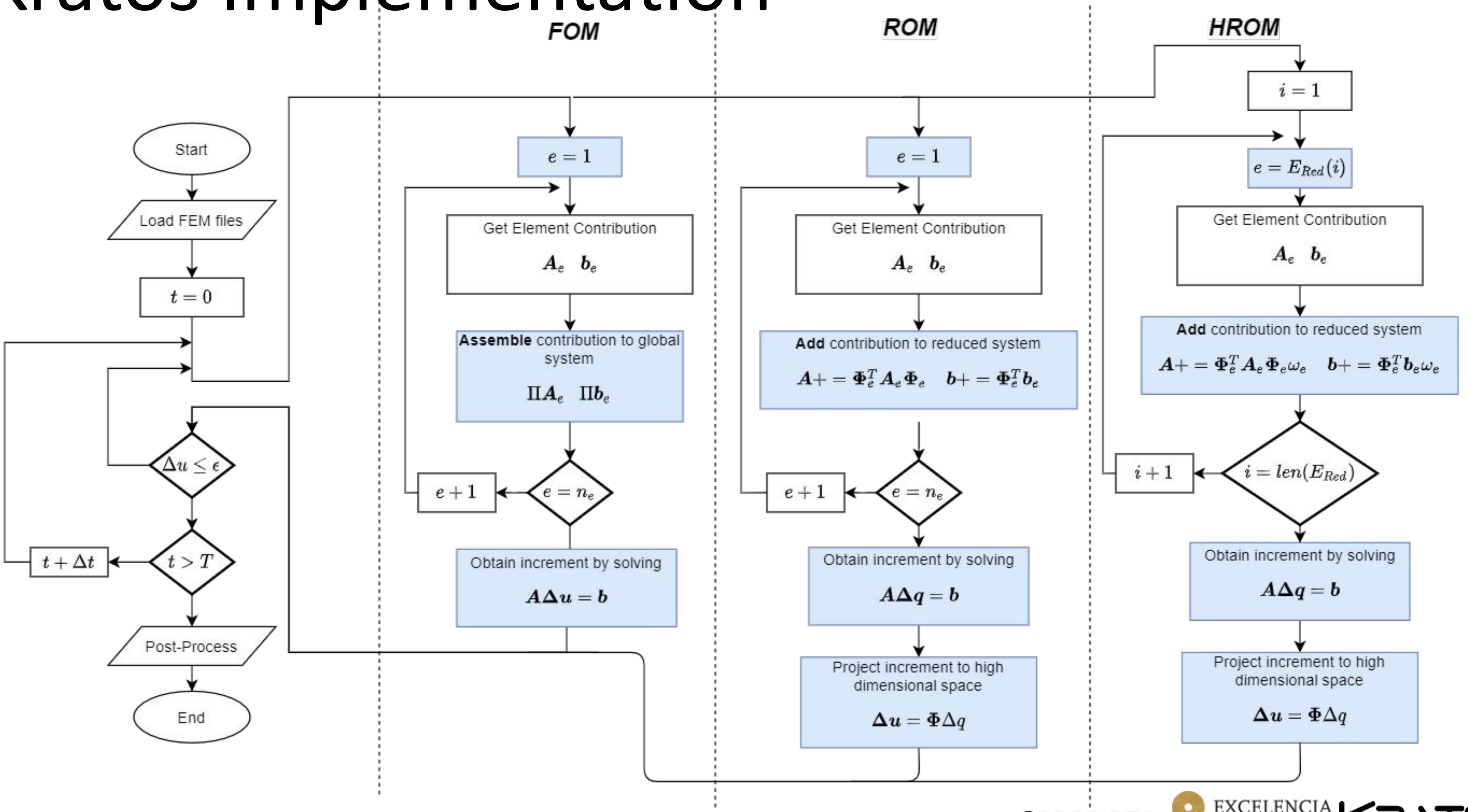
FOM Simulation



HROM Simulation

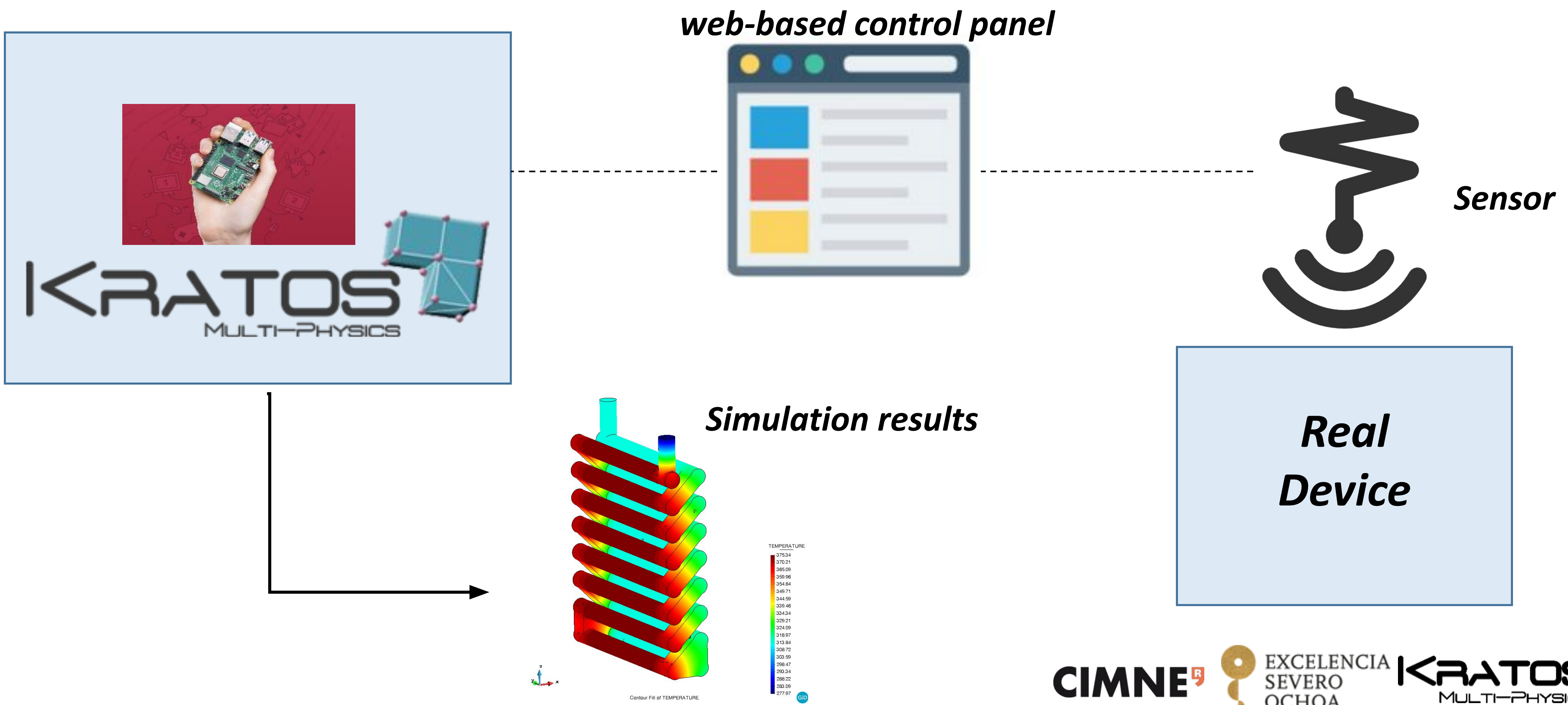


Kratos Implementation



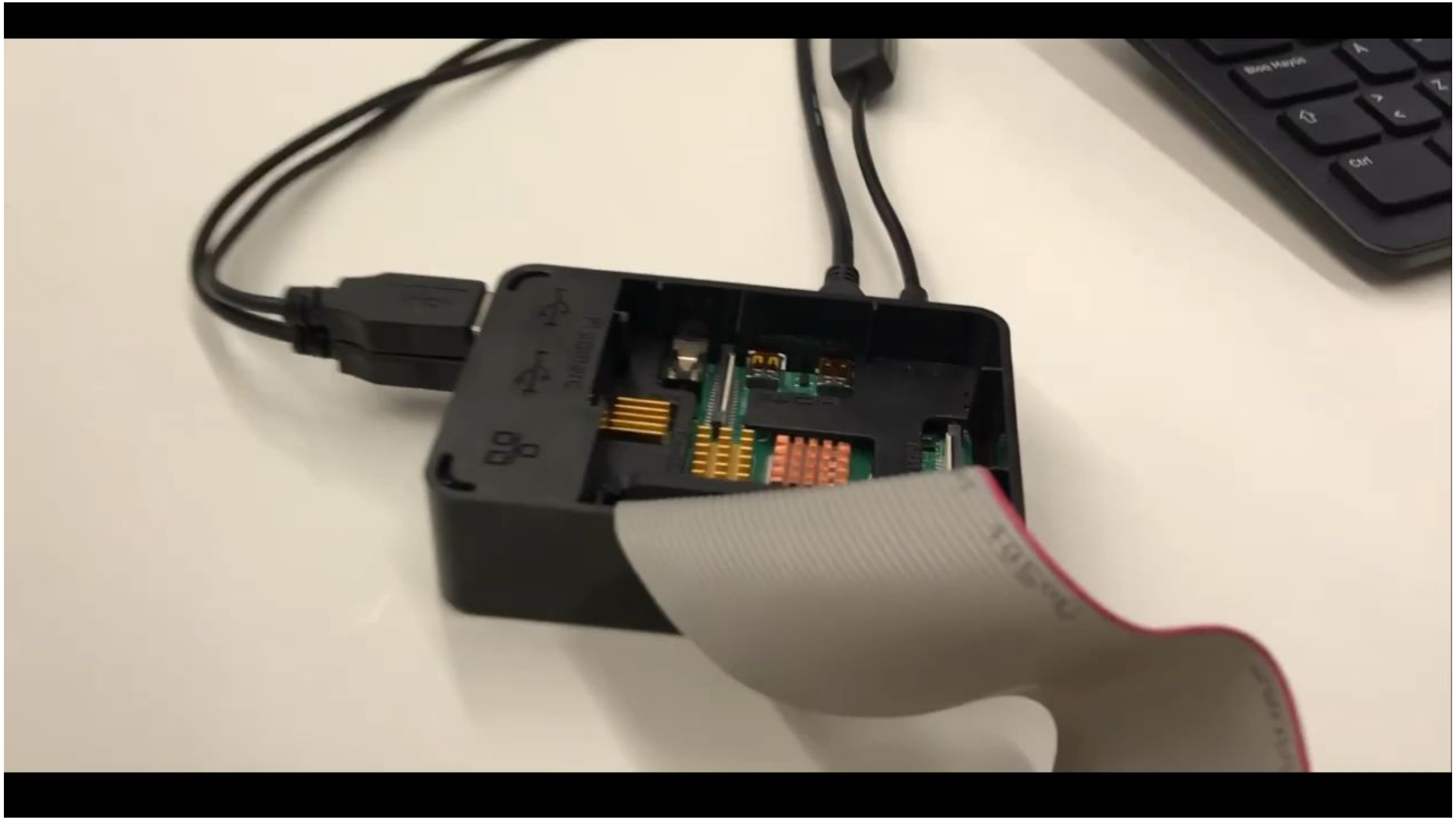
Link to the external world

Kratos provides an interface to retrieve data from sensors placed in situ.



CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Kratos ROM on a Raspberry Pi



CIMNE
EXCELENCIA SEVERO OCHOA
KRATOS MULTI-PHYSICS

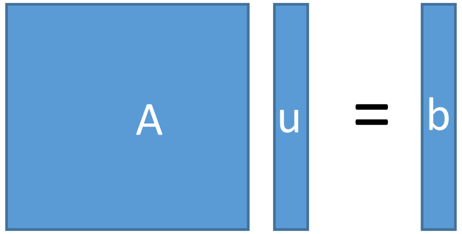
POD weaknesses and strengths

- Straightforward procedure for training and inference

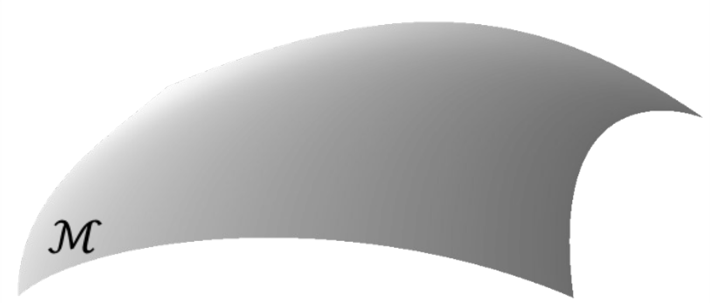
- Not ideal for certain problems(convection dominated, highly nonlinear)

Local POD

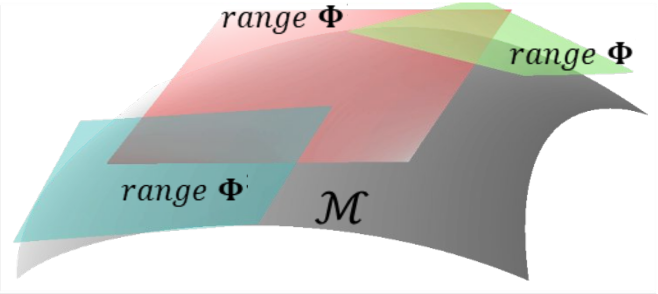
Full Order Model (FOM)



Solution manifold: $\mathcal{M} = \{\, \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \,\} \subset \mathbb{R}^n$

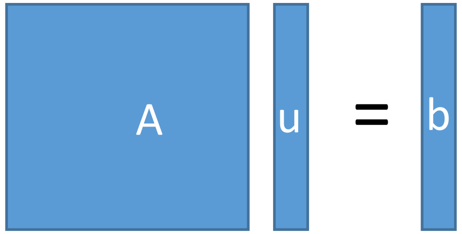


Let $\boldsymbol{u}_{new} \approx \boldsymbol{u}_{old} + \boldsymbol{\Phi}^{\mathbf{i}}\, \mathbf{q}$

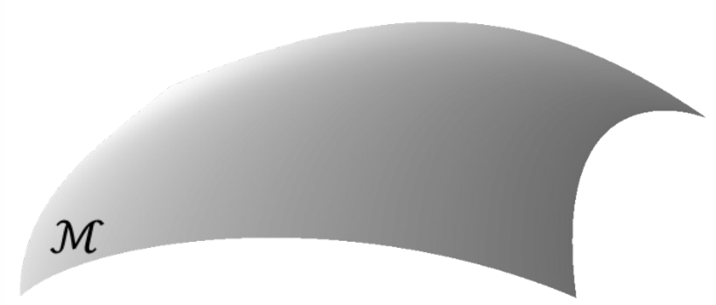


Local POD

Full Order Model (FOM)

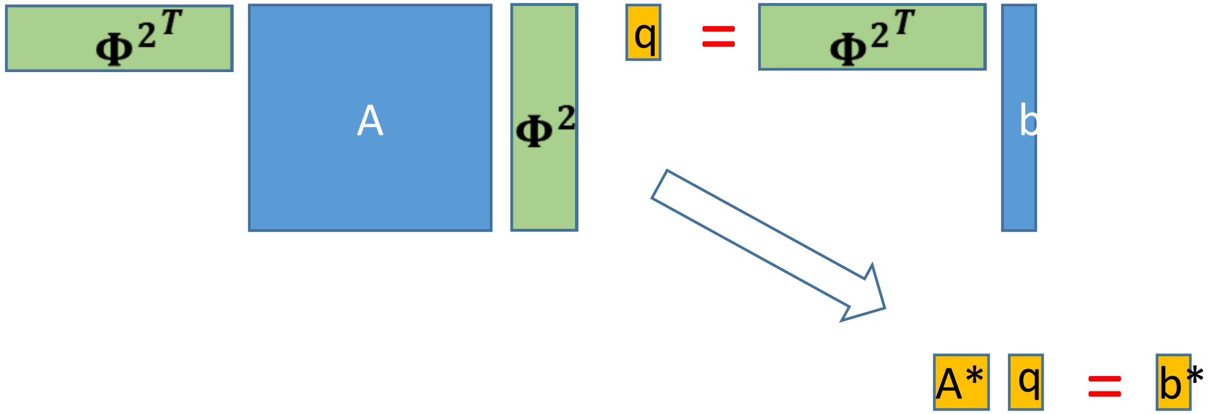


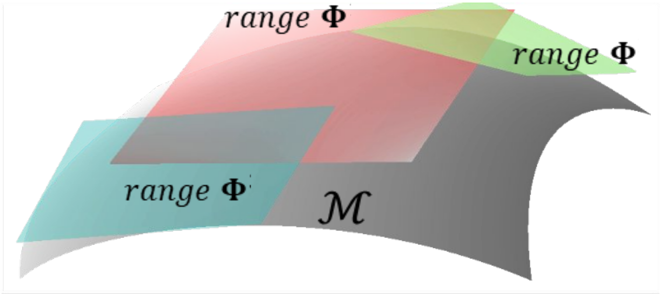
Solution manifold: $\mathcal{M} = \{ \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \} \subset \mathbb{R}^n$



Let $\boldsymbol{u}_{new} \approx \boldsymbol{u}_{old} + \boldsymbol{\Phi}^{\mathbf{i}}\, \mathbf{q}$

Reduced Order Model (ROM)

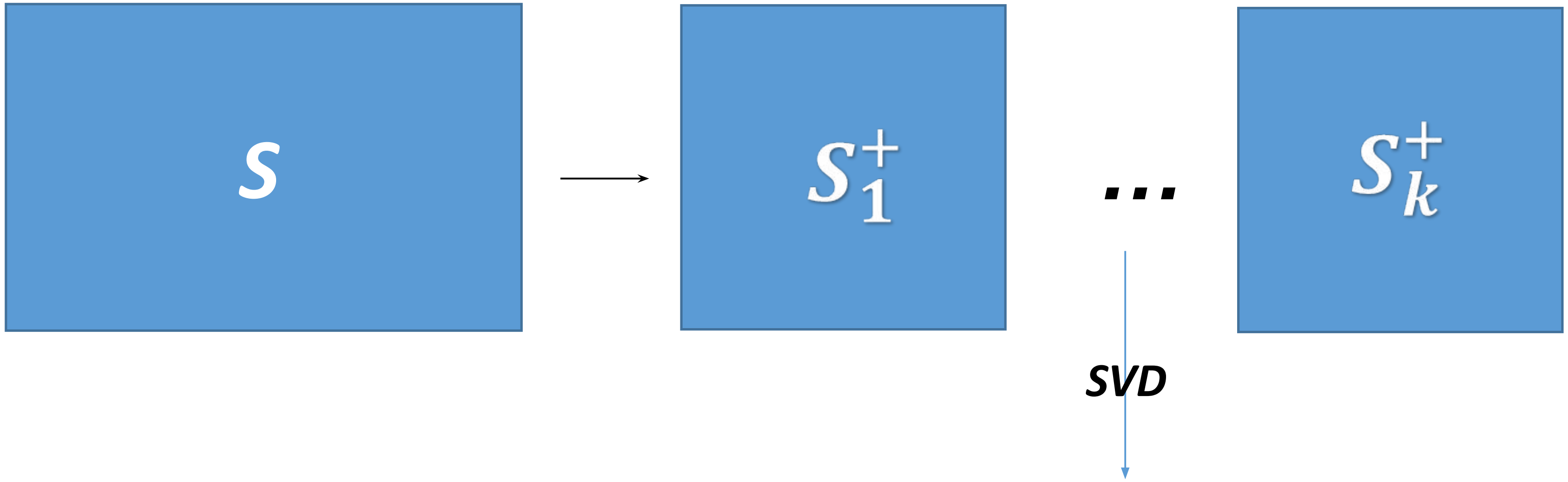




Local POD

- Use an unsupervised learning method to build clusters
- **1.** $\boldsymbol{S_i = kmeans(S)}$ **2.** Add overlapping $\boldsymbol{S_i^+ = overlap(S_i)}$. See ref. [2]

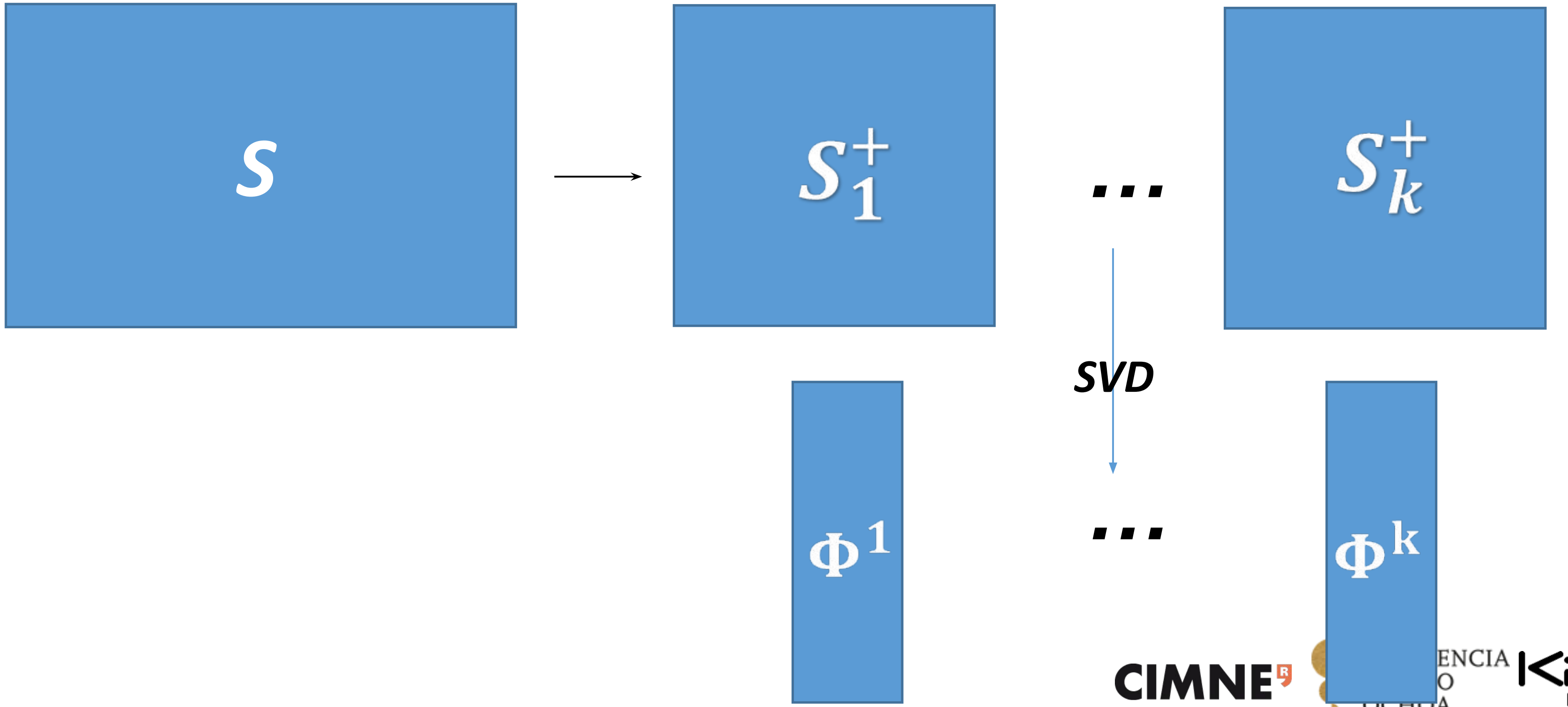
- **1.** $\boldsymbol{S_i^+ = fuzzy\text{-}c - means(S_i)}$ **.** See ref. [3]



Local POD

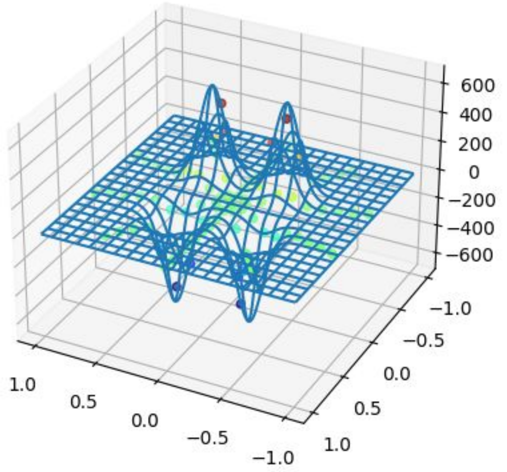
- Use an unsupervised learning method to build clusters
- **1.** $\boldsymbol{S_i = kmeans(S)}$ **2.** Add overlapping $\boldsymbol{S_i^+ = overlap(S_i)}$. See ref. [2]

- **1.** $\boldsymbol{S_i^+ = fuzzy\text{-}c - means(S_i)}$ **.** See ref. [3]

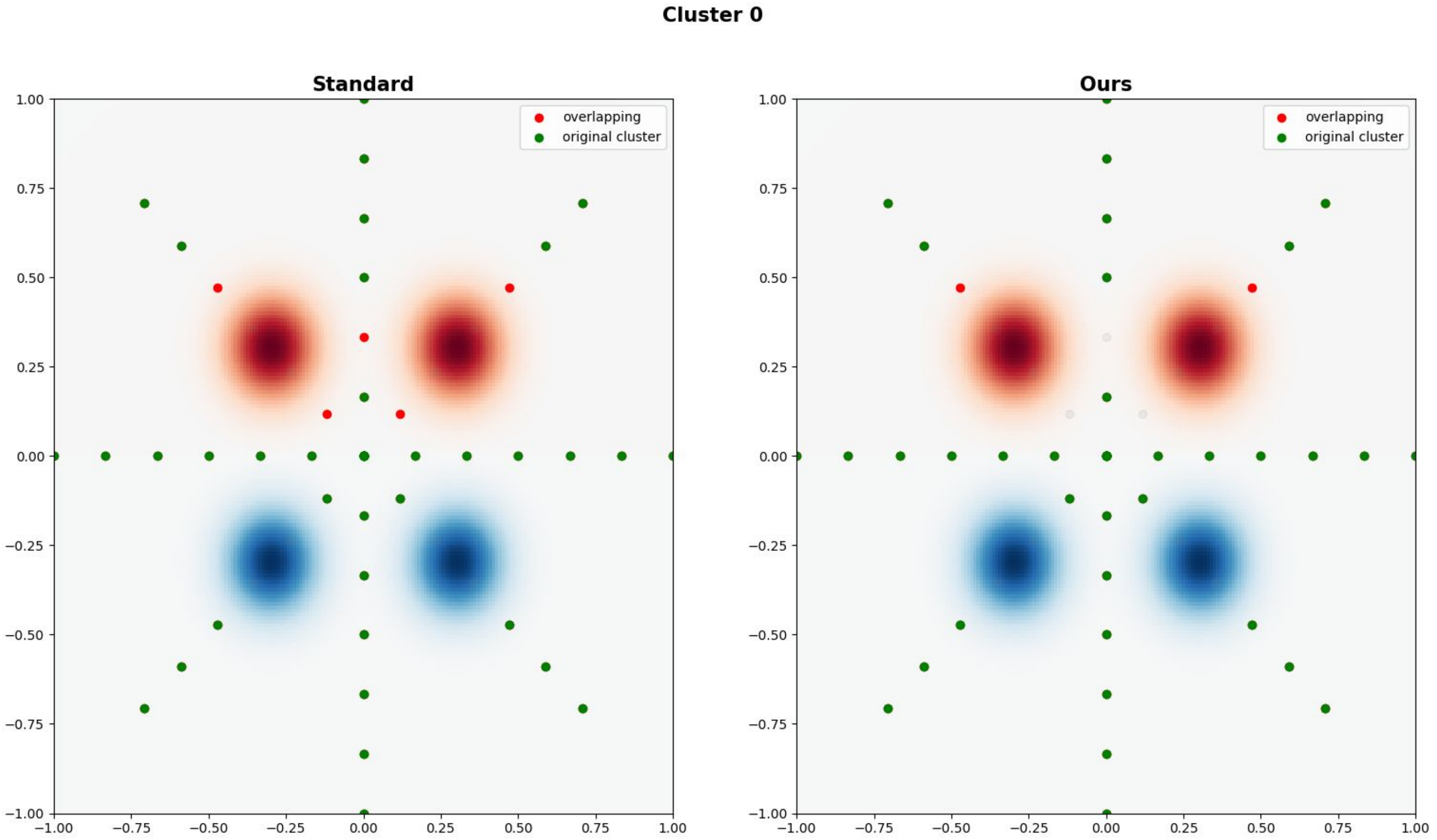


CIMNE KRATOS MULTI-PHYSICS

Local POD. The importance of overlapping

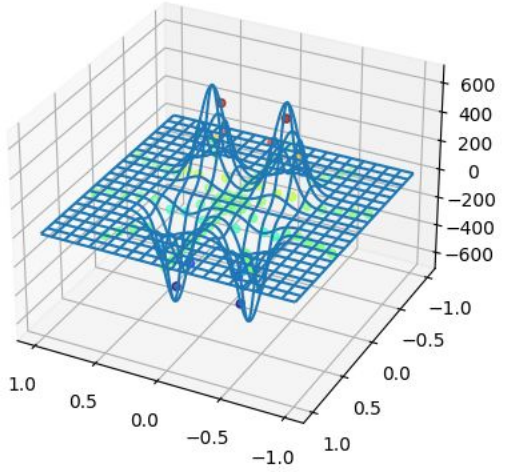


Synthetic 2-Manifold

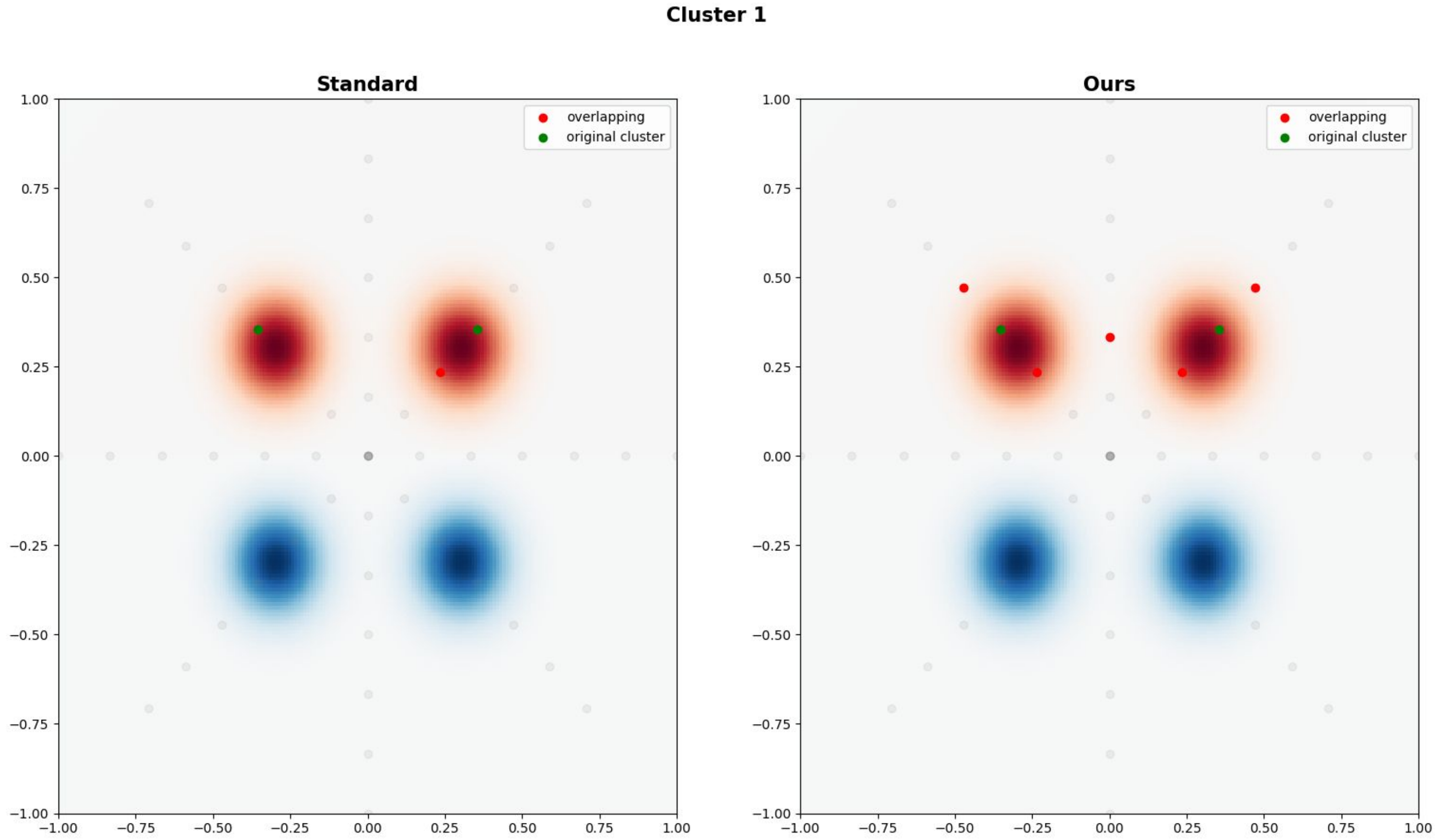


CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

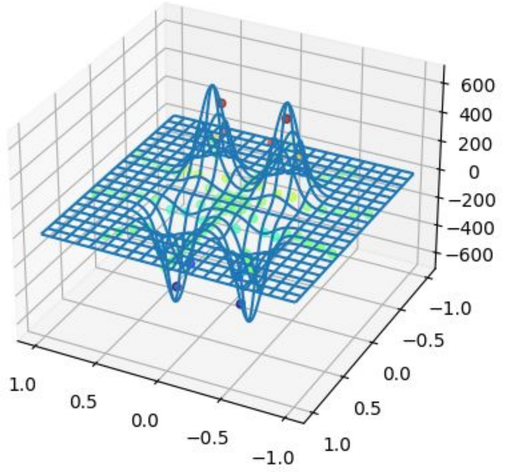
Local POD. The importance of overlapping



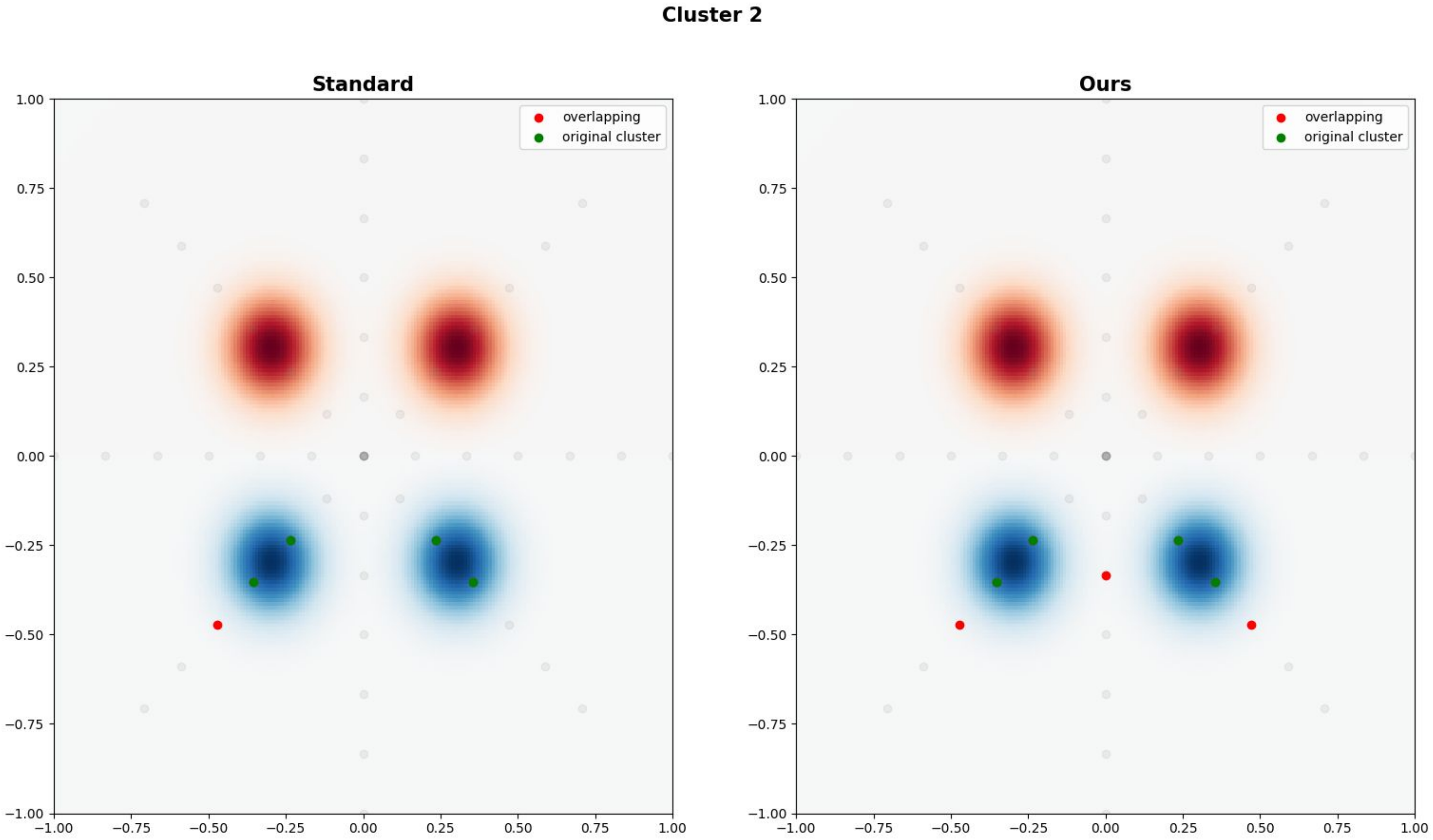
Synthetic 2-Manifold



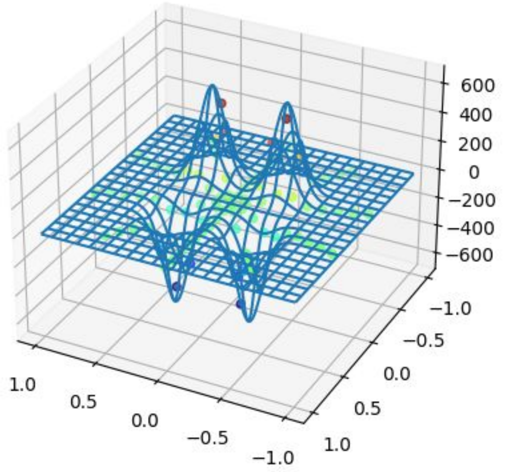
Local POD. The importance of overlapping



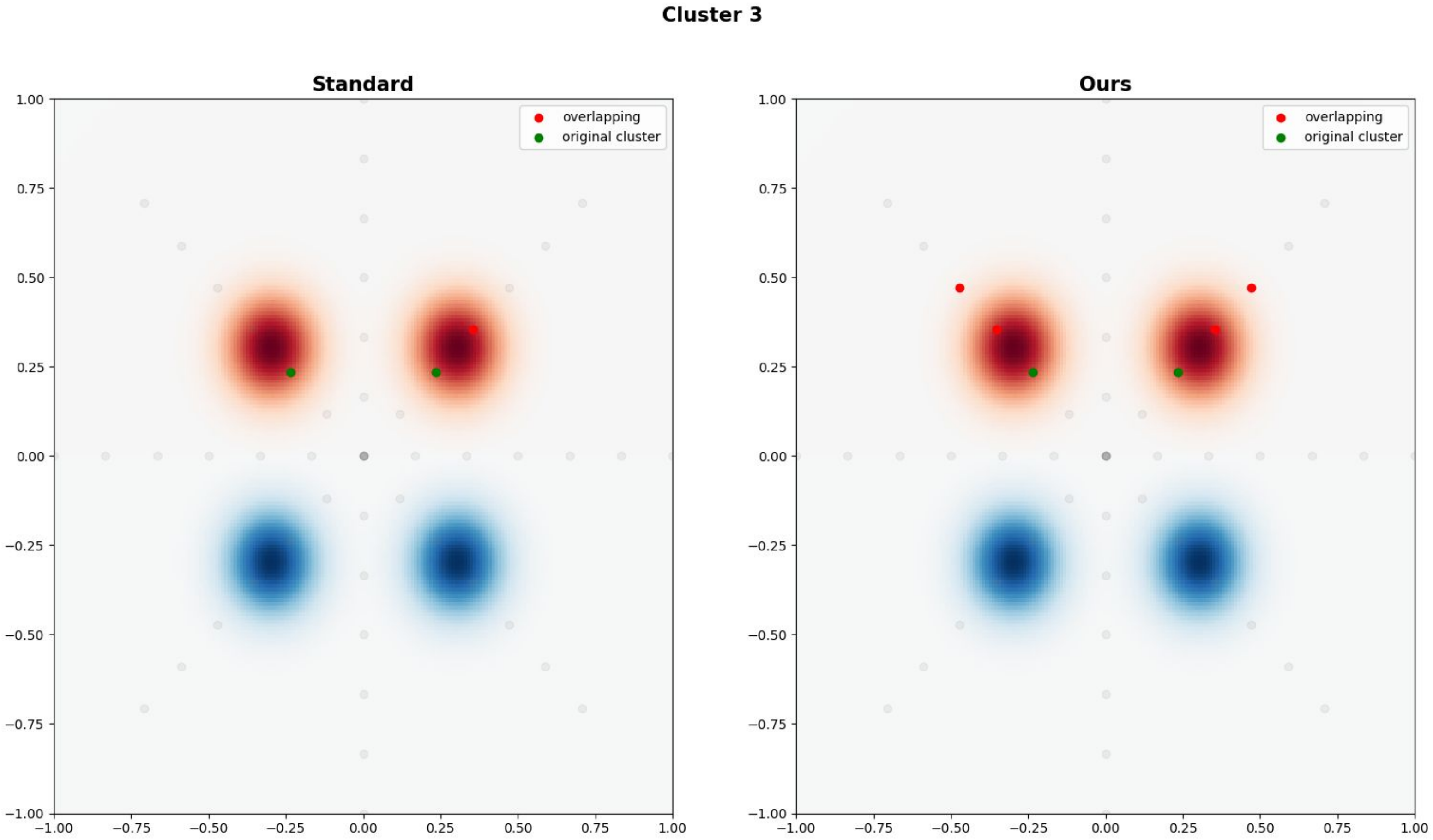
Synthetic 2-Manifold



Local POD. The importance of overlapping

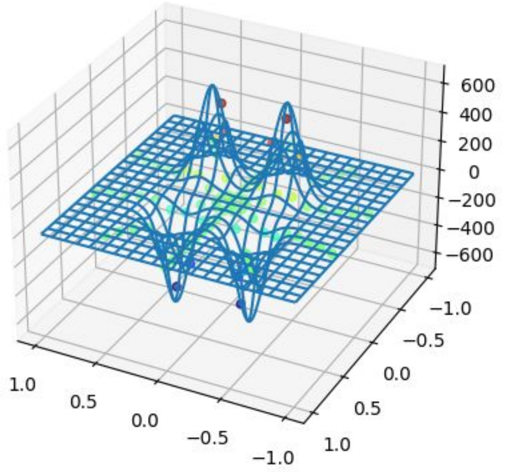


Synthetic 2-Manifold

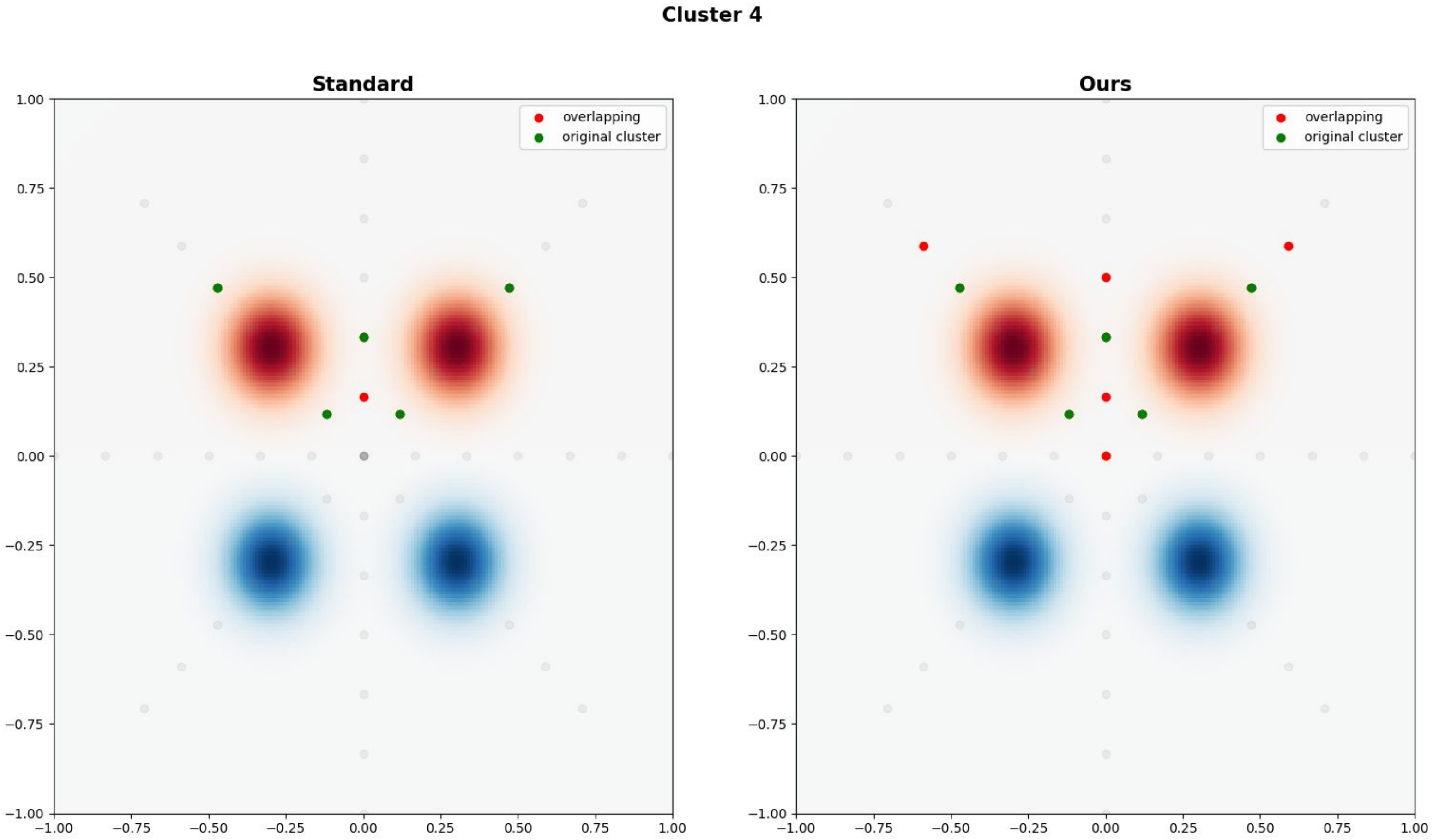


CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Local POD. The importance of overlapping

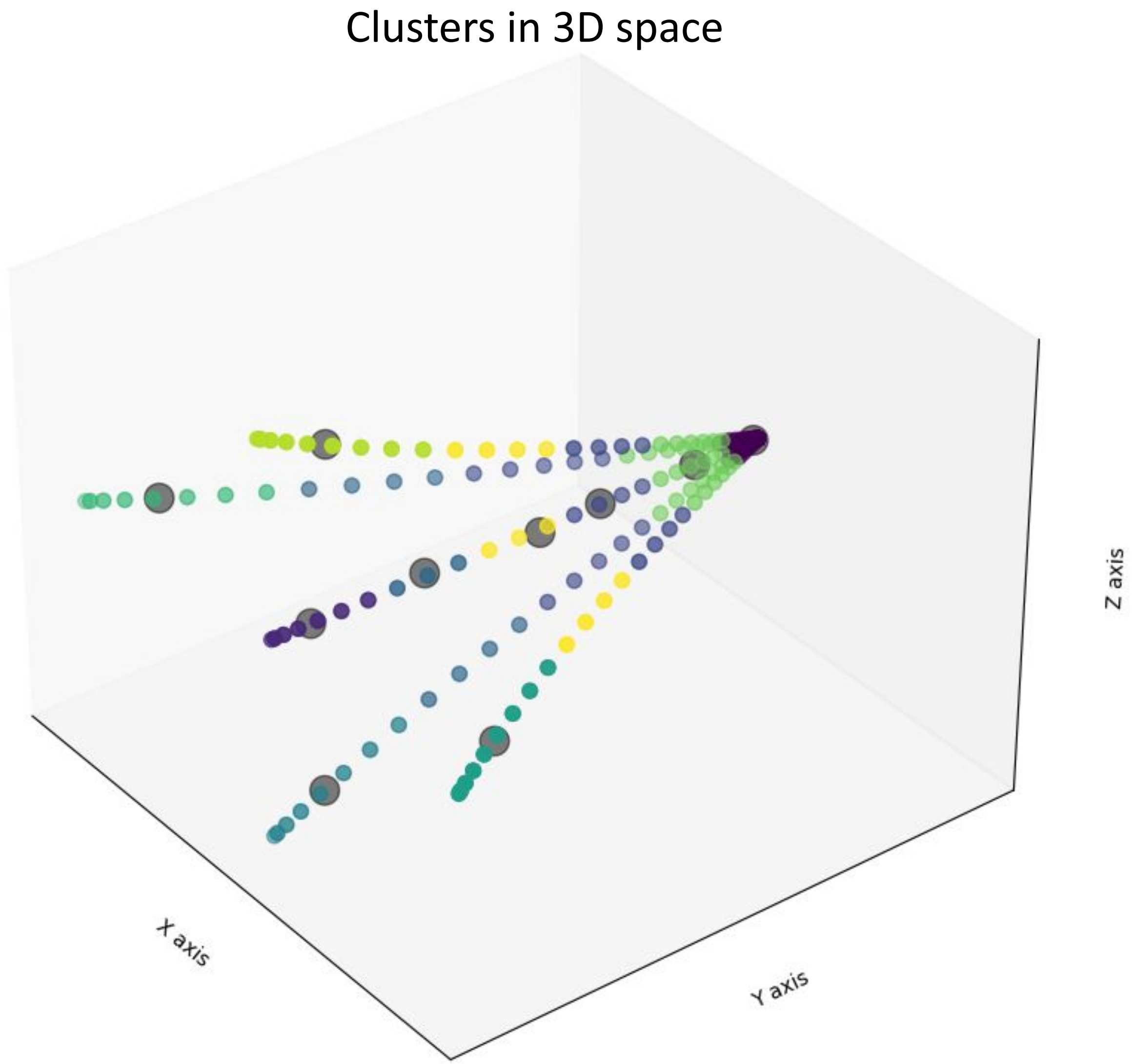


Synthetic 2-Manifold



CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

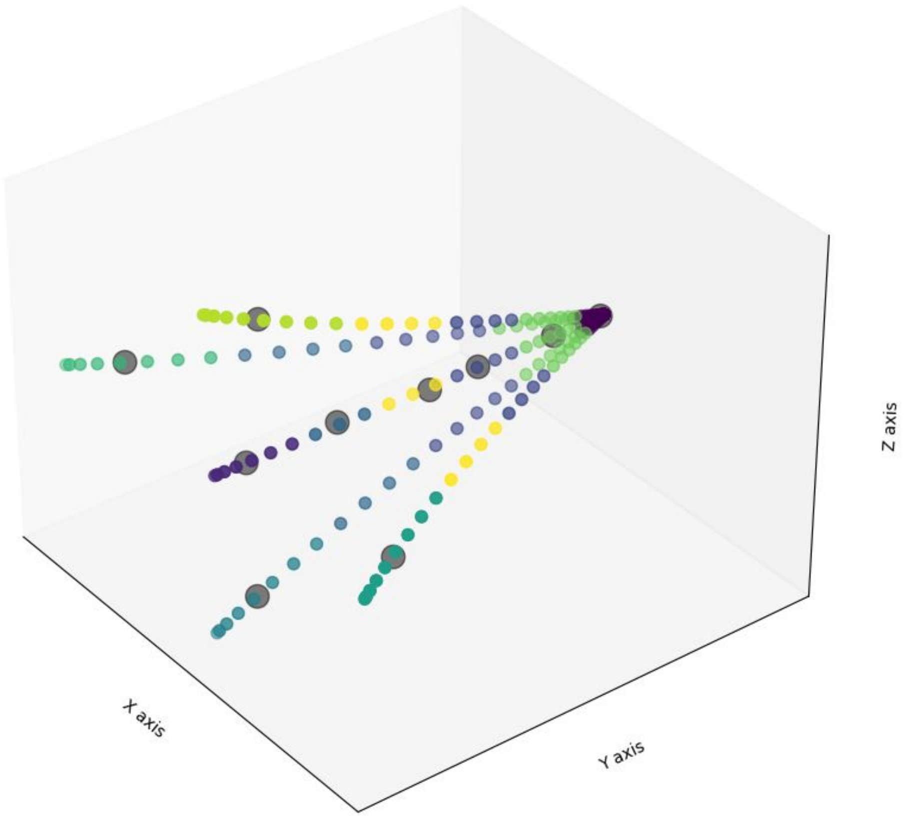
Local POD. Example



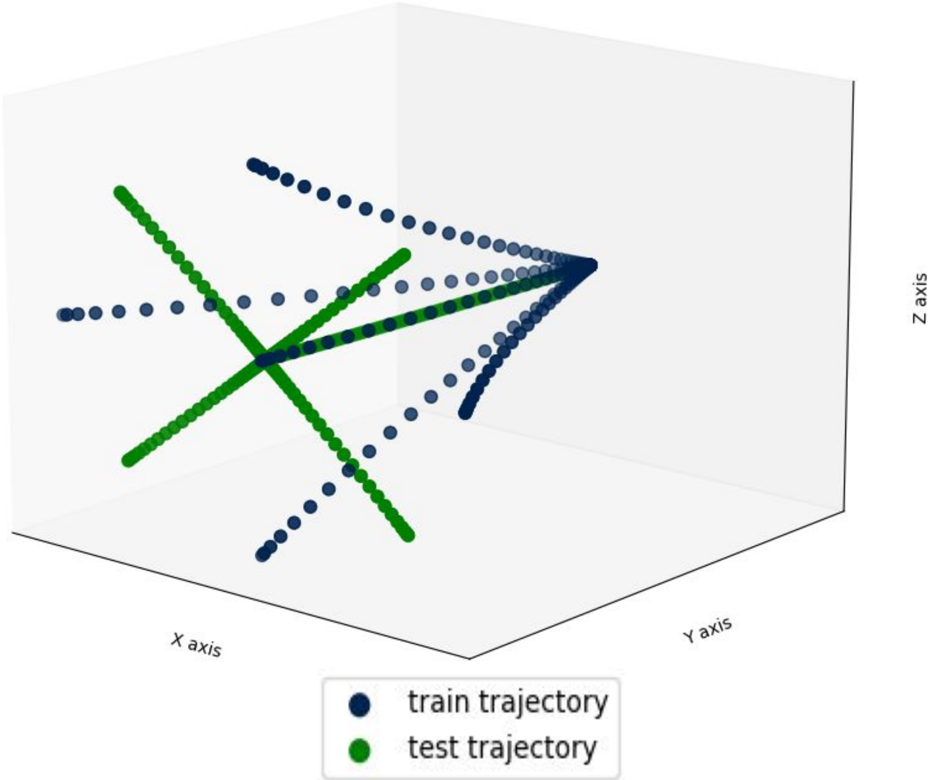
EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Local POD. Example

Clusters in 3D space

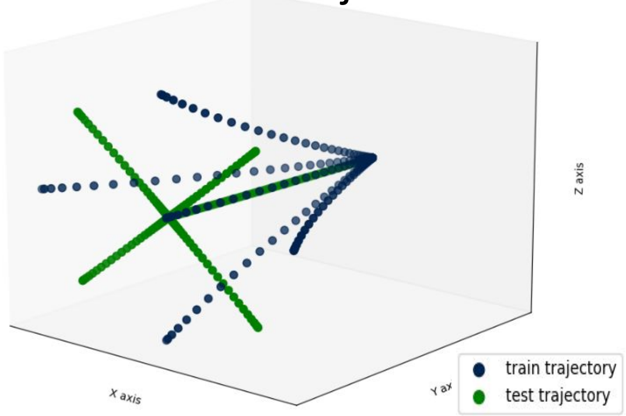


Train and Test trajectories

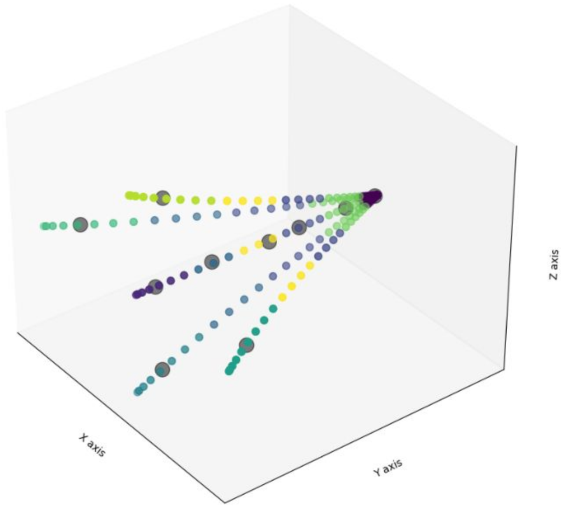


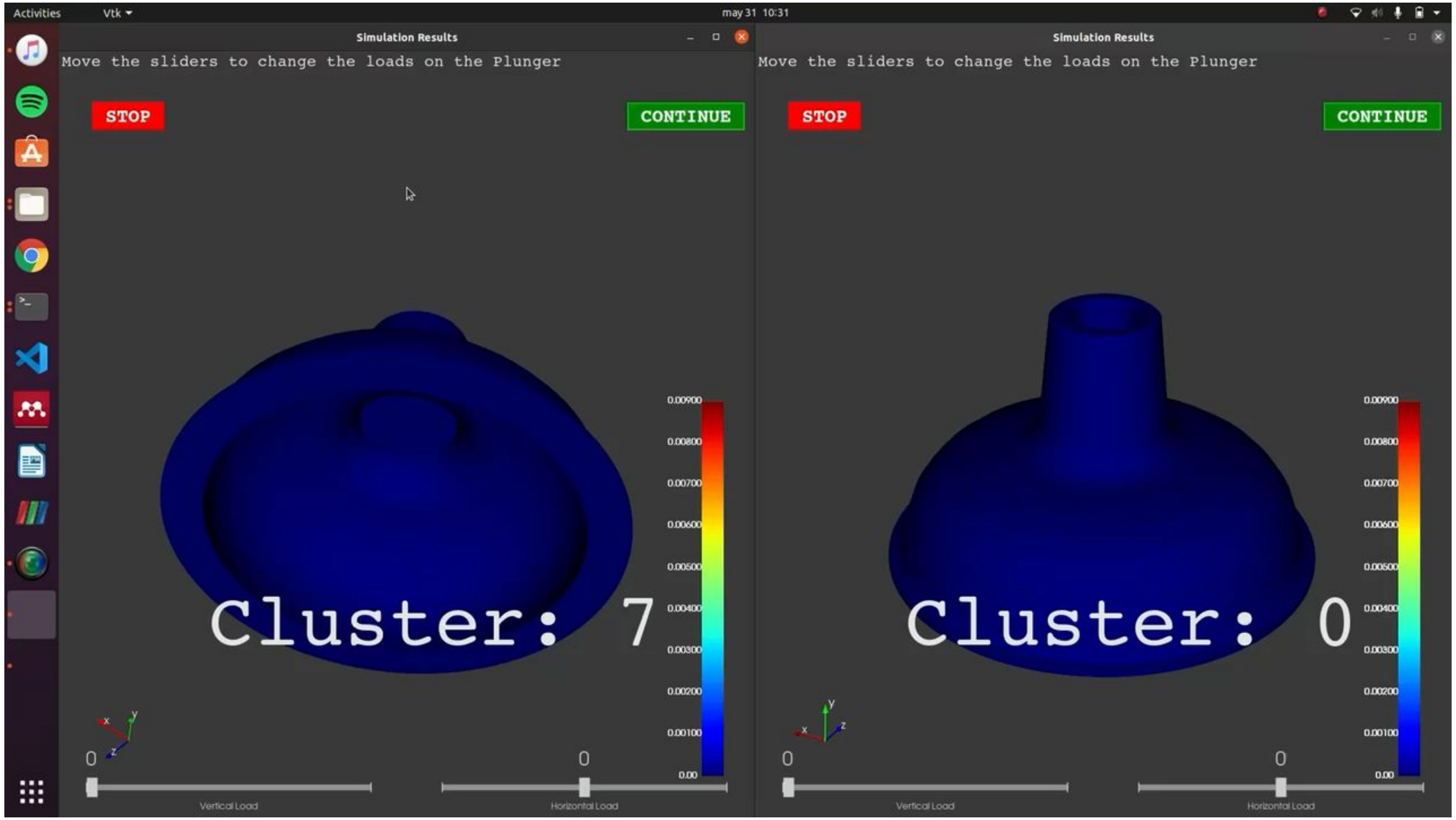
Local POD. Example

Train and Test trajectories



Clusters in 3D space

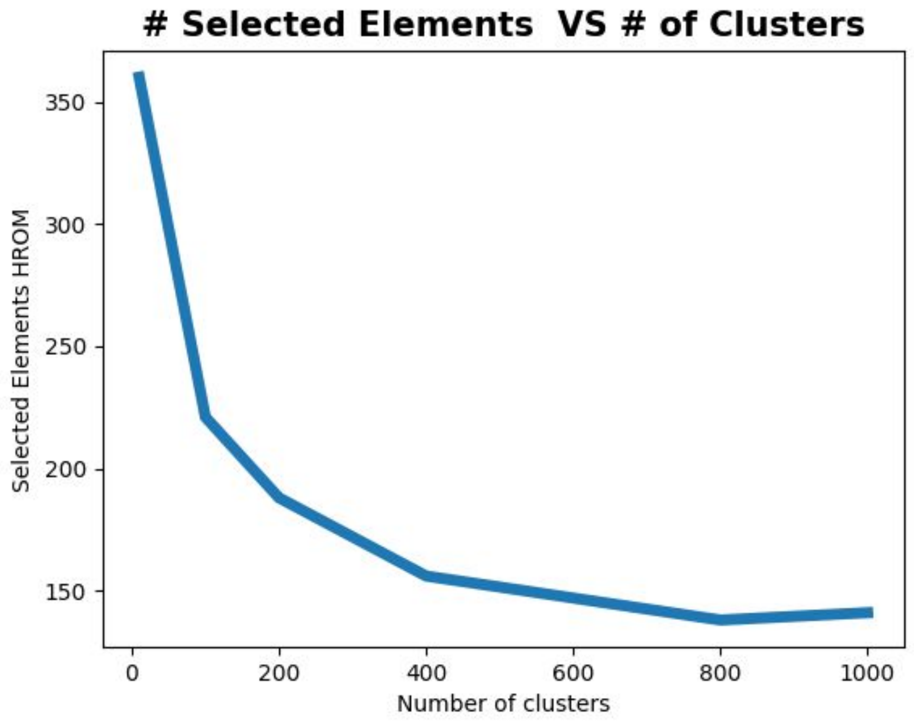




CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

Local POD. Improved hyper-reduction

$$(\boldsymbol{E}, \boldsymbol{W}) = \arg min \ \| \ \boldsymbol{J}(\boldsymbol{R}, \boldsymbol{\Phi}) \ \|_2^2$$
$$subject\ to\ \boldsymbol{W} \succ \boldsymbol{0}$$



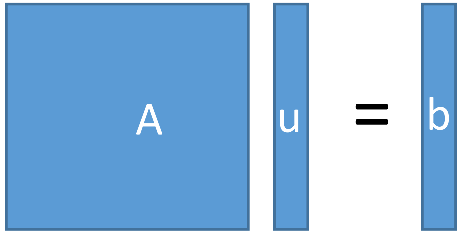
Local POD. Strengths and weaknesses

- Reasonable overhead in training and negligible in inference
- Smaller elements sets, therefore faster ROMs

- Easy to overfit to training trajectories

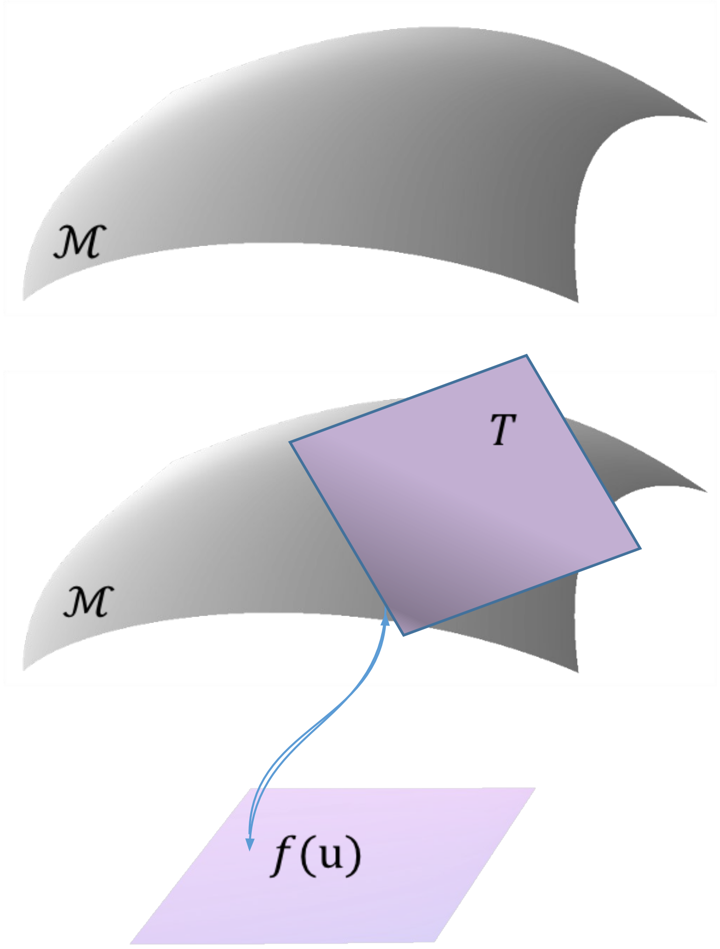
Deep autoencoders

Full Order Model (FOM)



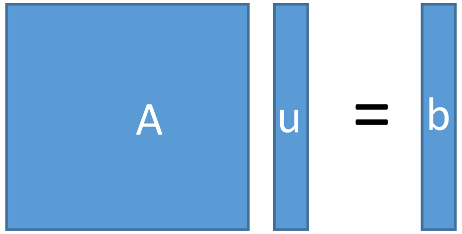
Solution manifold: $\mathcal{M} = \{\, \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \,\} \subset \mathbb{R}^n$

$$\boxed{\text{Let } \boldsymbol{u} \approx \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{u}))}$$

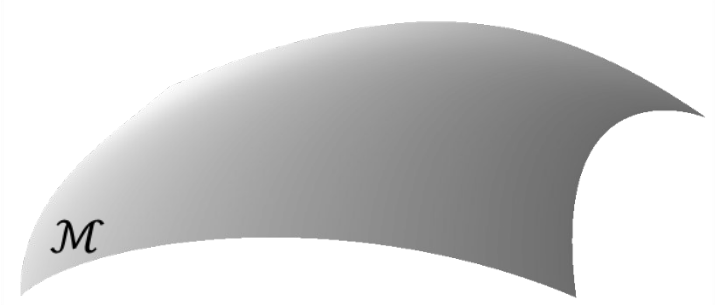


Deep autoencoders

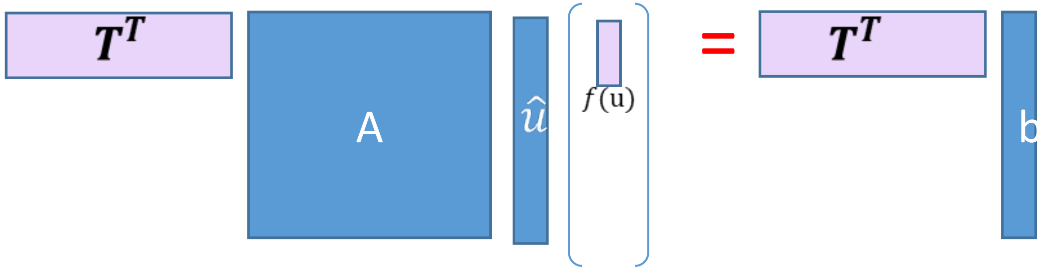
Full Order Model (FOM)

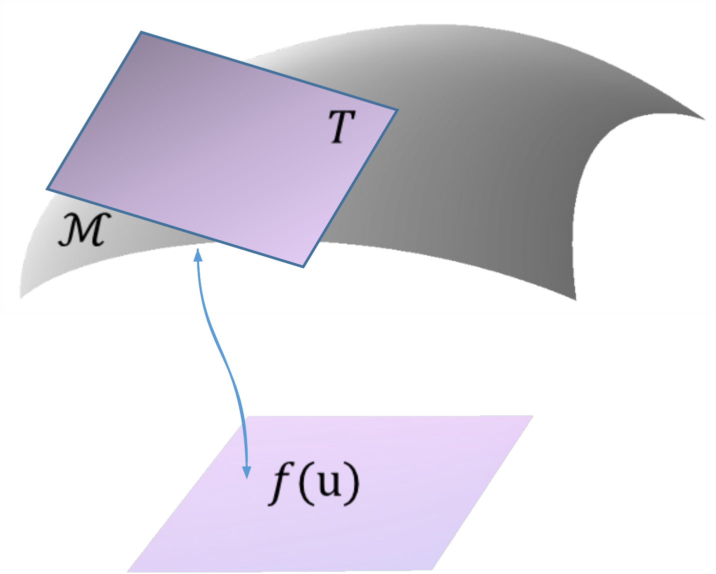


Solution manifold: $\mathcal{M} = \{\, \boldsymbol{u}(t;\boldsymbol{\mu}) \mid t \in (0,T], \boldsymbol{\mu} \in \mathcal{P} \,\} \subset \mathbb{R}^n$



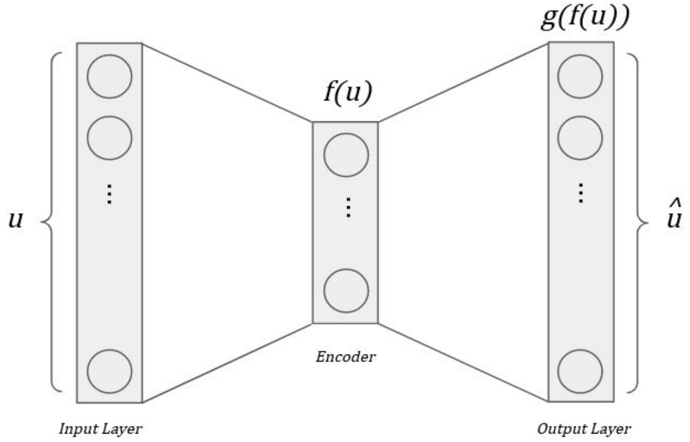
Let $\boldsymbol{u} \approx \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{u}))$





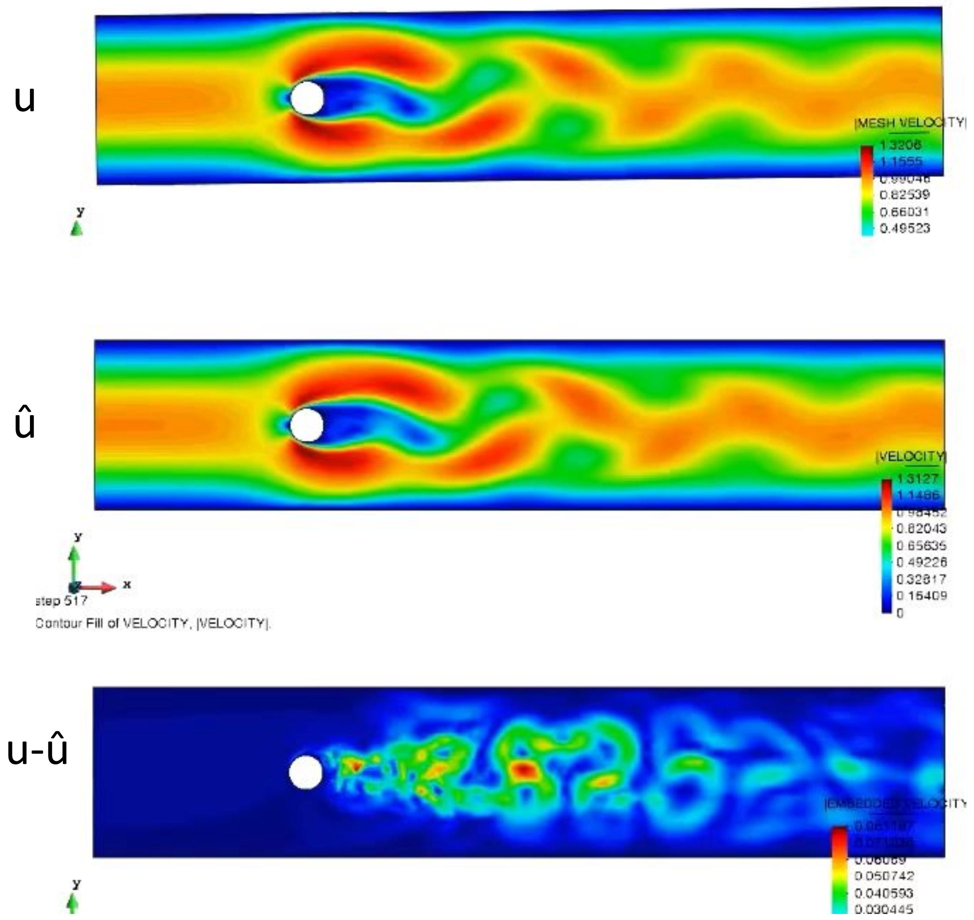
*be careful, slight oversimplifications

Deep autoencoders



$$minimize \; \|\boldsymbol{u} - \widehat{\boldsymbol{u}}\|_2^2$$

TensorFlow



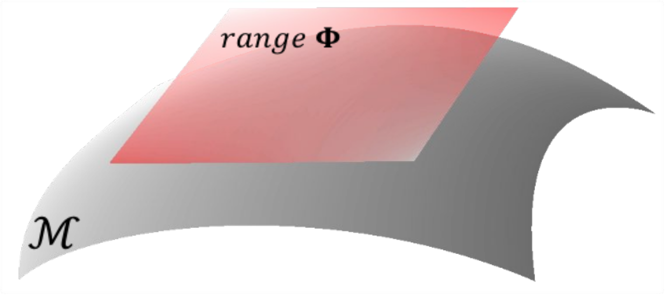
Deep autoencoders. Some comments

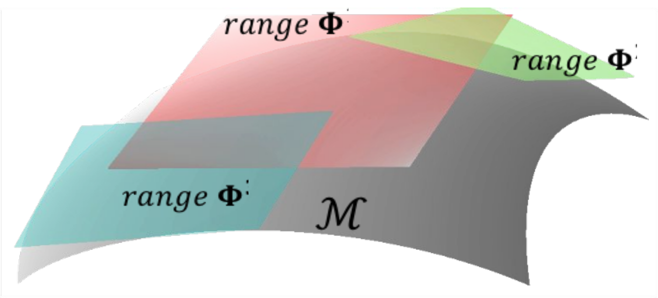
- Not difficult to integrate within Kratos (Python libraries)

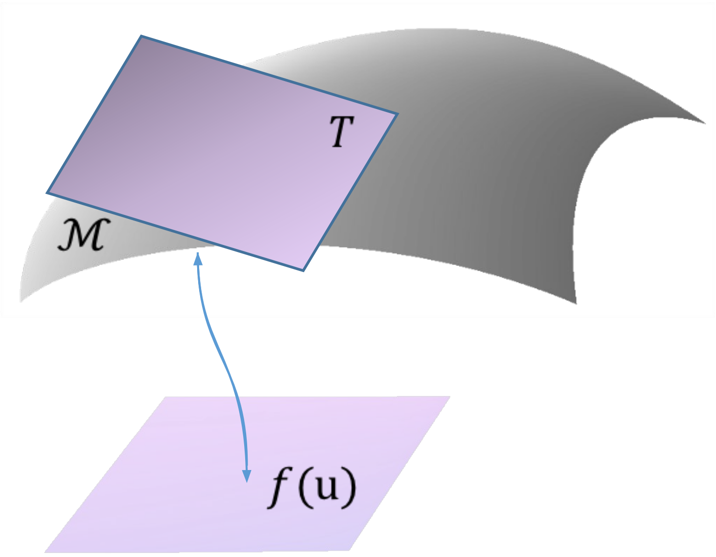
- Long training time
- Not much literature on nonstructured meshes FEM

General conclusions

- The ROM capabilities of Kratos
- Promising results and exciting challenges







THANK YOU

GRATEFUL TO:

EdgeTwins HPC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 946009

Link to Kratos github site

CIMNE EXCELENCIA SEVERO OCHOA KRATOS MULTI-PHYSICS

References:

[1] Hernández, J. A. (2020). A multiscale method for periodic structures using domain decomposition and ECM-hyperreduction. *Computer Methods in Applied Mechanics and Engineering*, *368*, 113192.

[2]Bezdek J, Ehrlich R, Full W. FCM: the fuzzy c-means clustering algorithm. Comput Geosci. 1984;10(2–3):191-203

[3]Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. Int J Numer Methods Eng. 2012;92(10):891-916