Interpolation non linéaire pour des modèles réduits multiparamétriques

CANUM 2026

5 Juin 2026

Michel BERGMANN[1] Angelo IOLLO[1] **Mathias TRUEL[1,2]** Gilles VOGT[2]

[1]Centre Inria de l'université de Bordeaux, Equipe MONHADE

[2]Ingeliance Technologies, Direction Technique

mathias.truel@ingeliance.com

Outline

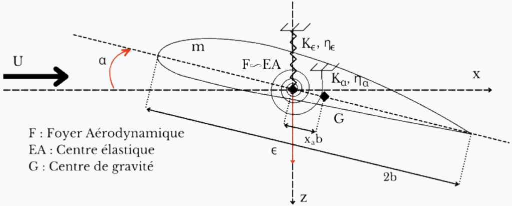
Outline

Fluid structure interaction - Flutter

Flutter demo video

Fluid structure interaction - Flutter

Dynamic instability coming from aeroelastic interactions
$\Rightarrow$ Vibrations and oscillations in the wing structure



2d flutter model

Two phenomena are studied at Ingeliance :

- Predict the flutter speed
- Predict the post flutter behavior

Two test models

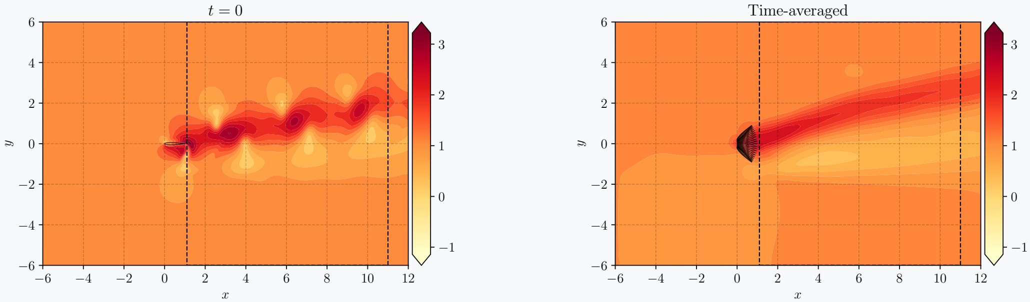
- OpenFOAM flutter model (Ingeliance)
- NaSCar forced oscillations (M. Bergmann)

Limitations of linear subspaces

Time-averaged NASCAR (domain restriction)

Test case: Time-averaged NASCAR with domain restriction.

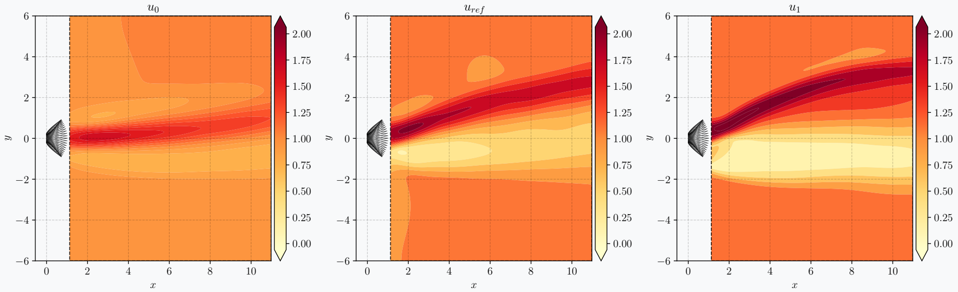
- Studied parameter: frequency F; four wing cycles are used.



First timestep of the solution

Time-averaged solution

Test case: Time-averaged NASCAR with domain restriction
Objective: Interpolate at $F = 400$.



NASCAR cases for three frequencies($A = 3500$, $H = 200$, $P = 0$)

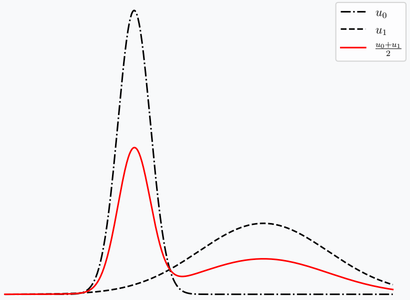
Limitations of linear subspaces

Linear interpolation video

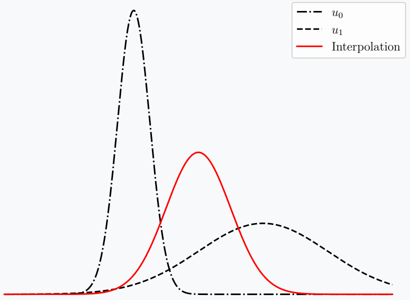
CDT interpolation video

Limitations of linear subspaces

Interpolations for advection problems :



The interpolation we get with linear methods



The interpolation we want

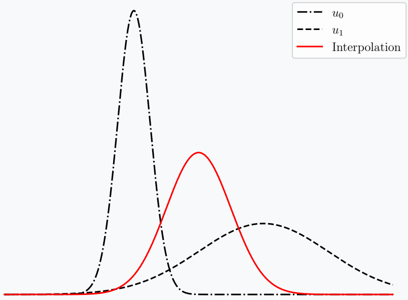
Outline

How to interpolate non-linearly?

We will use a "good" mapping to transport coherent structures.

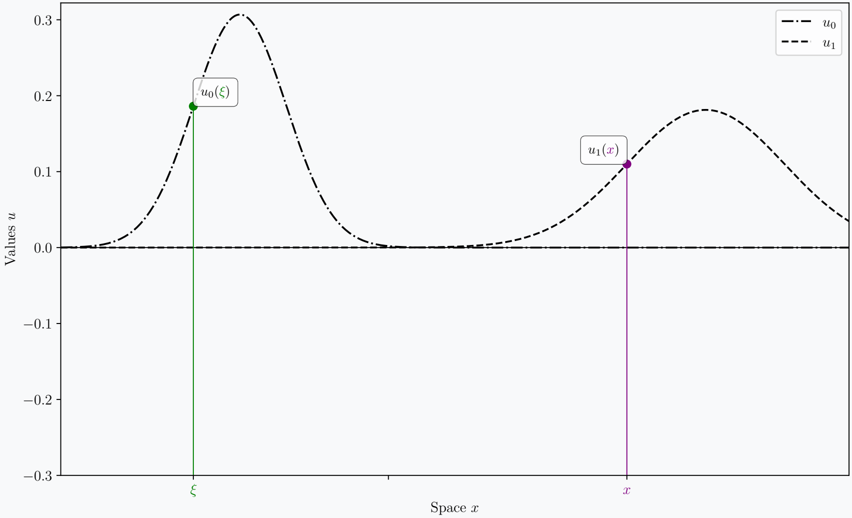
Linear interpolation on values
+ Linear interpolation on transport
= Convex Displacement Interpolation

Iollo and Taddei (2022b)

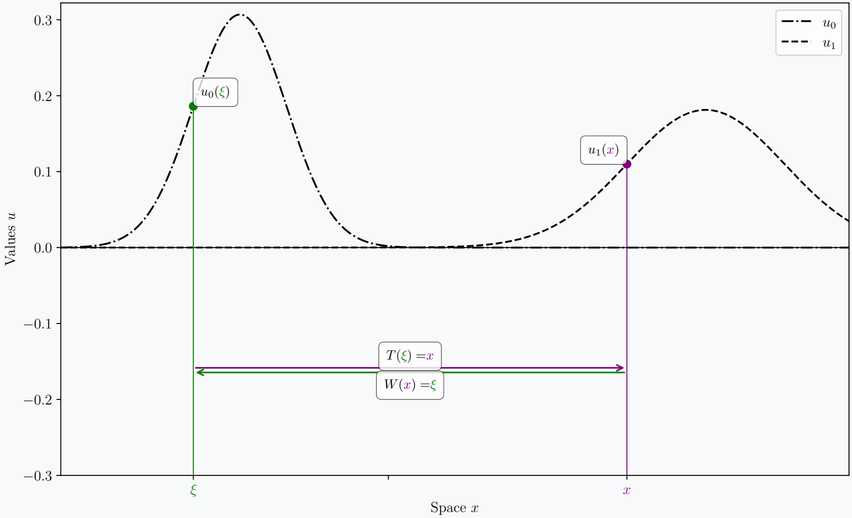


The interpolation we want

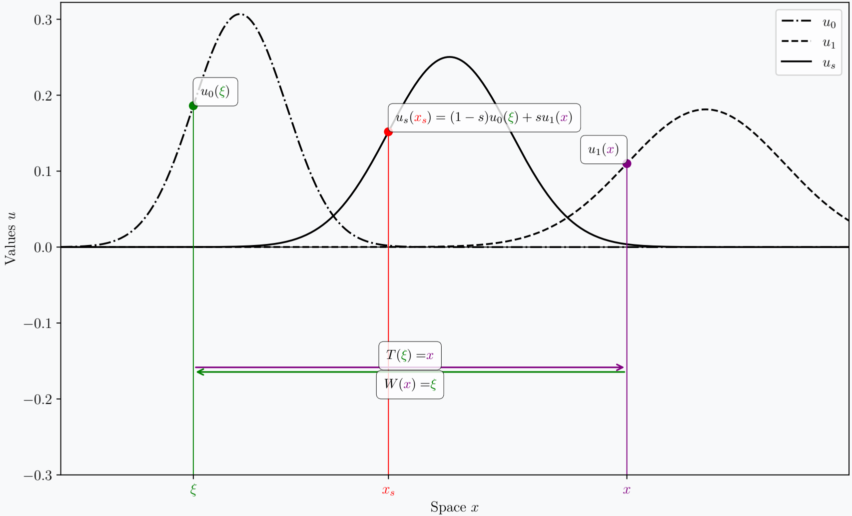
Convex Displacement Interpolation



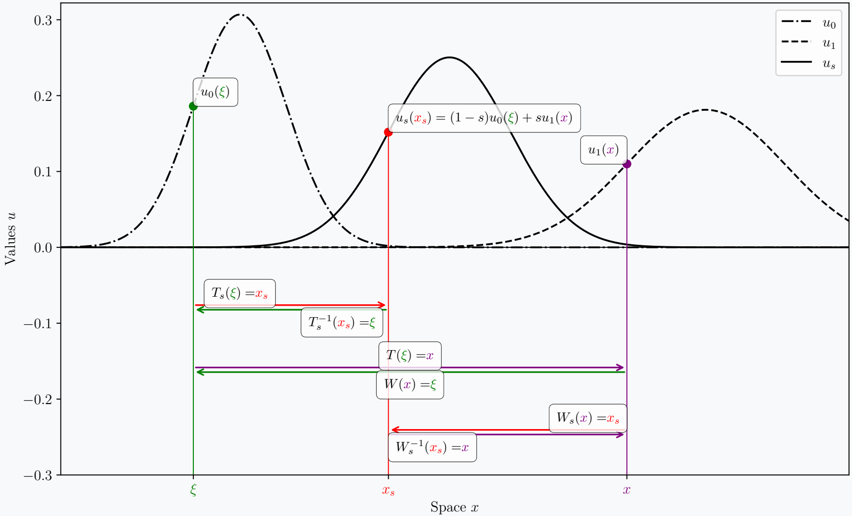
Convex Displacement Interpolation



Convex Displacement Interpolation



Convex Displacement Interpolation



Convex Displacement Interpolation

Let be, T and W, two mappings between the features.
s is an interpolation parameter.

Convex interpolation on transport :

$$W_s(x) = (1-s)x + sW(x)$$
$$T_s(\xi) = (1-s)\xi + sT(\xi)$$

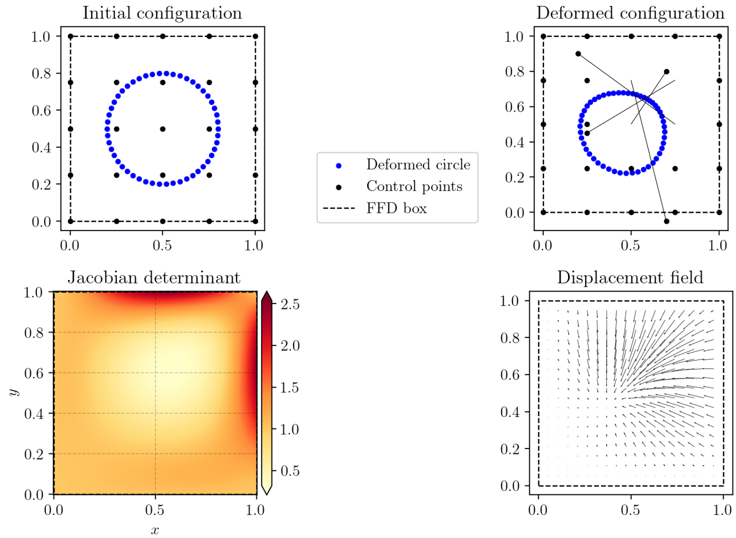
The Convex Displacement Interpolation (CDI) interpolates solutions :

$$u_s(x_s) = (1-s)u_0(T_s^{-1}(x_s)) + su_1(W_{1-s}^{-1}(x_s))$$

The goal is to **find a good mapping**.

Outline

A parametrized mapping : Free-Form Deformation



Finding mappings via minimisation

Minimisation of two parametrized maps through displacement of $2 \times N_{cp}$ control points. We minimise the functional:

$$
\begin{aligned}
J(T, W) = &\ L_2 \text{ alignment between } u_0 \text{ and } u_1 \\
&+ \text{Bijectivity of } T \text{ and } W \\
&+ \text{Wasserstein based terms for large displacements} \\
&+ \dots
\end{aligned}
$$

Finding mappings via minimisation: L_2 alignment

We minimise the alignment of u_0 and u_1 by the mappings:

$$\begin{aligned} J(T, W) = {} & \alpha_0 \| u_0(W(x)) - u_1(x) \|_{l_2} + \alpha_1 \| u_0(\xi) - u_1(T(x)) \|_{l_2} \\ & + \text{Bijectivity of } T \text{ and } W \\ & + \text{Wasserstein based terms for large displacements} \\ & + \dots \end{aligned}$$

- Sensor Free alignment: No need to work on densities, or use markers

Finding mappings via minimisation: Bijectivity terms

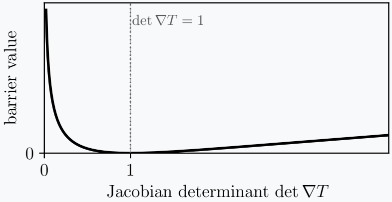
We ensure bijectivity with two terms:

$$\begin{aligned} J(T,W) = {} & \alpha_0 \|u_0(W(x)) - u_1(x)\|_{l_2} + \alpha_1 \|u_0(\xi) - u_1(T(x))\|_{l_2} \\ & + \alpha_2 \|W \circ T(\xi) - \xi\|_{l_2} + \alpha_2 \|T \circ W(x) - x\|_{l_2} \\ & + \alpha_3 \|\text{barrier}(\det(\nabla T))\|_{l_2} + \alpha_3 \|\text{barrier}(\det(\nabla T))\|_{l_2} \\ & + \text{Wasserstein based terms for large displacements} \end{aligned}$$

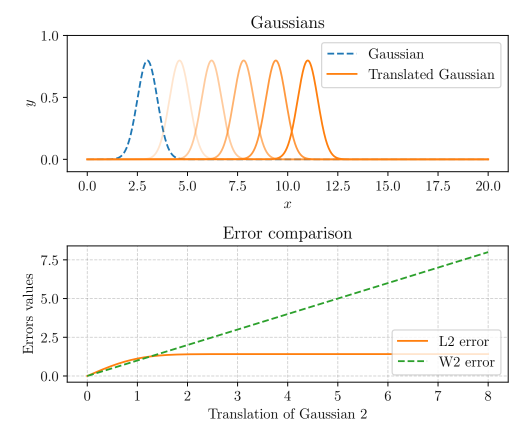
Numerically cheap term
T and W are inverses of each other.

Strict but expensive term : Ensures positive Jacobian det

$$\text{barrier}(\det \nabla T) = (-\varepsilon \log_{10}(\det \nabla T))^2$$



Finding mappings via minimisation: Wasserstein based terms



The analytical Gaussian Wasserstein distance

1. Transform the fields into positive densities with a sensor
2. Fit two gaussians $N(m_0, \Sigma_0)$ and $N(m_1, \Sigma_1)$ on u_0 and u_1
3. Compute the analytical Wasserstein distance between the two gaussians

$$W(x) = m_1 + \Sigma_0^{-1/2}\left(\Sigma_0^{1/2}\Sigma_1\Sigma_0^{1/2}\right)\Sigma_0^{-1/2}(x - m_0)$$

Fernández and Gerbeau (2009) :

This W2 with a sensor is only used for the first iterations.

Solver iterations

Three outer iterations are performed:

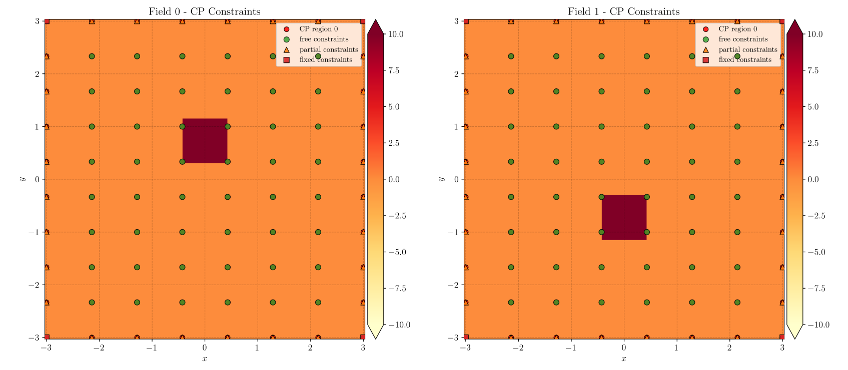
1. $W2(u_0, u_1)$ alignment without L_2 alignment, with $||T \circ W = I||$ for bijectivity
2. L_2 alignment with $||T \circ W = I||$ for bijectivity
3. L_2 alignment with barrier method on jacobian determinant, until $\det(\nabla T) > 0.0$

Usually only 1 iteration with jacobian is needed. To check that $\det(\nabla T) > 0.0$ indeed.

Control points constraints

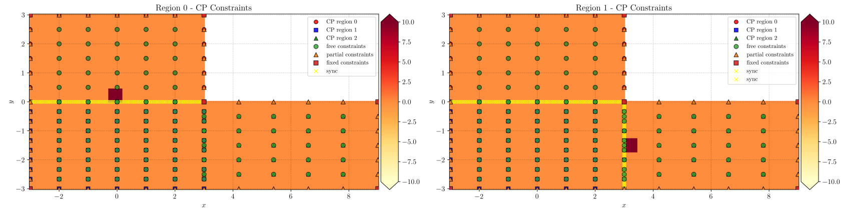
Slip boundary conditions:

$$\lambda \cdot n = 0 \quad \text{on } \Gamma_\Omega$$



Control point constraints: source and target fields

Piecewise definition of the mapping



Control point constraints: source and target fields

JAX implementation

FFD is implemented using the **JAX** library.

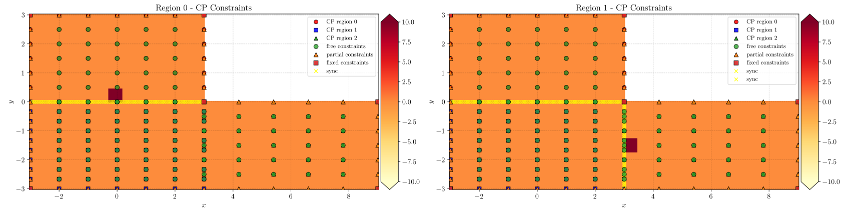
- Automatic differentiation for gradient computation
- GPU acceleration

Allow the use of an **L-BFGS** optimizer (Optax).

$\Longrightarrow$ **Fast optimization** (120s independent of N_{cp})

Outline

Results: Alignment of squares



Control point constraints: source and target fields

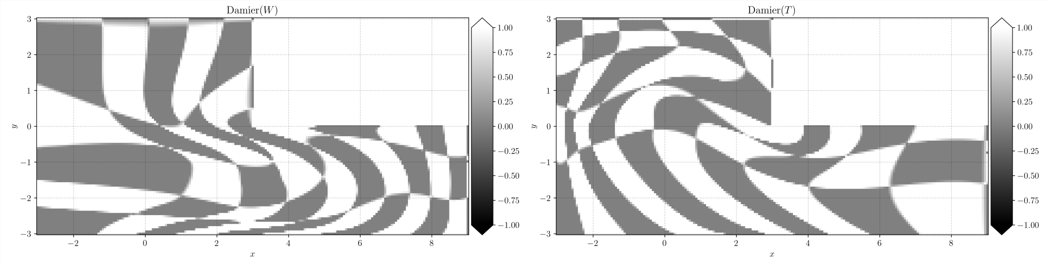
Results: Alignment of squares

Cnr interpolation video

Results: Linear interpolation

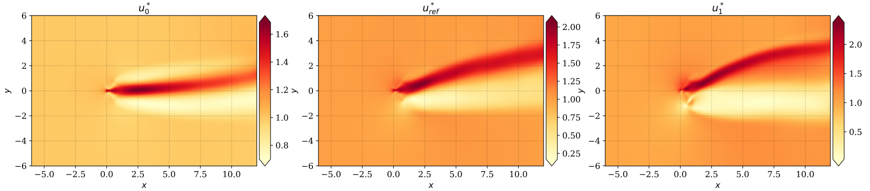
linear interpolation video

Results: Alignment of squares



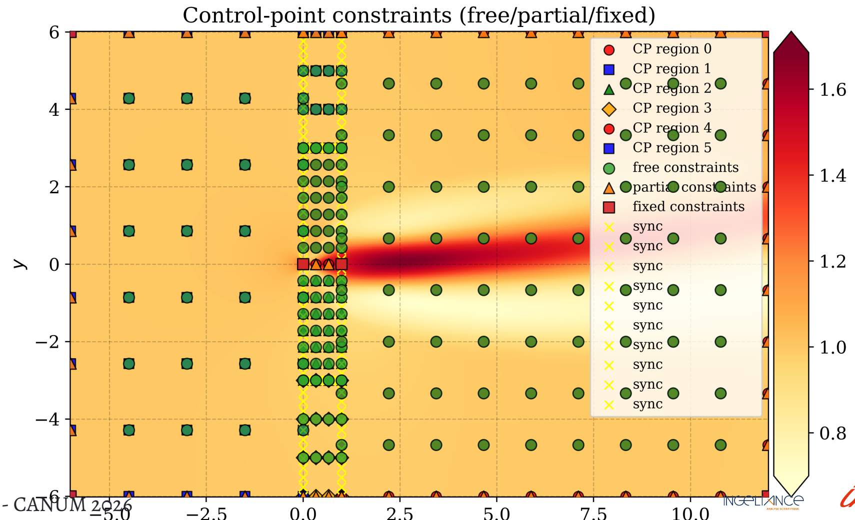
Piecewise FFD checkerboards

Results: Alignment of NACA

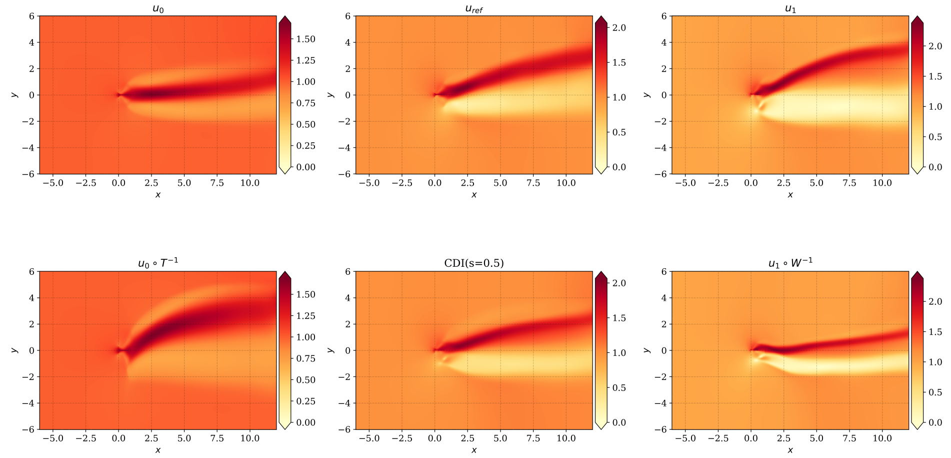


NACA source, reference and target fields

Results: Alignment of NACA



Results: Alignment of NACA



NACA source, reference and target fields and aligned fields

Outline

Limitations of linear subspace

Non linear interpolation

Mapping minimisation

Numerical Results

Conclusion and future works

Conclusion and future works

A tool to align fields:

- Does not depend on sensor to model fields
- Flexible and fast implementation

Perspectives for the method:

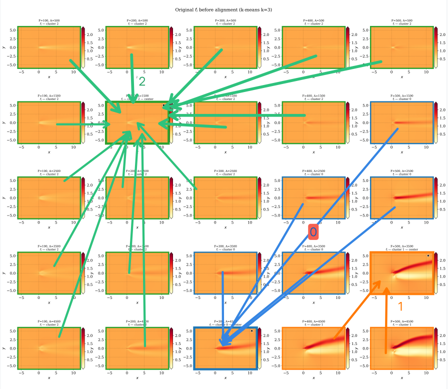
- Try other parametrized mappings like RBF functions
- Use improved FFD formulations
- Solve the mapping velocity instead displacement: Handle obstacles and complex geometries

Ongoing works with Reduced Order Models:

- A multi-parametric and multi-field version of the interpolation method: A Wasserstein Barycenter like approach
- A clustered approach to construct local ROM bases on aligned fields
- Predict instationary fields

Clustered ROM - Methodology proposal

- Cluster fields
- Learn the mappings in a cluster
- Construct a POD on the aligned fields
- Construct a ROM on the aligned fields



K-means clustering of field samples ($k = 3$)

Thanks you for your attention!

References:

Yongxin Chen, Tryphon T. Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2019. doi: 10.1109/ACCESS.2018.2889838.

Simona Cucchiara, Angelo Iollo, Tommaso Taddei, and Haysam Telib. Model order reduction by convex displacement interpolation, 2023.

Miguel Angel Fernández and Jean-Frédéric Gerbeau. Algorithms for fluid-structure interaction problems. In *Fluid-Structure Interaction Problems*, volume 1, page 307. Springer Verlag, 2009. URL https://inria.hal.science/inria-00542757.

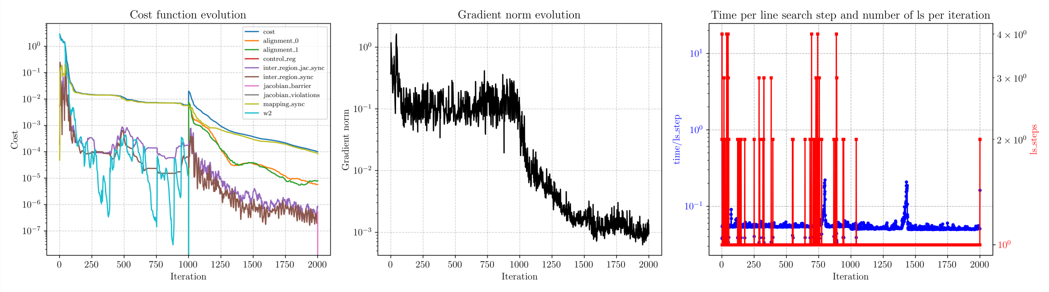
Rui Hua, Jose M. Pozo, Zeike A. Taylor, and Alejandro F. Frangi. Multiresolution eXtended free-form deformations (XFFD) for non-rigid registration with discontinuous transforms. *Medical Image Analysis*, 36:113–122, February 2017. ISSN 1361-8415. doi: 10.1016/j.media.2016.10.008. URL https://www.sciencedirect.com/science/article/pii/S1361841516301888.

Angelo Iollo and Tommaso Taddei. Mapping of coherent structures in parameterized flows by learning optimal transportation with gaussian models. *Journal of Computational Physics*, 471:111671, 2022a. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111671. URL https://www.sciencedirect.com/science/article/pii/S0021999122007343.

Angelo Iollo and Tommaso Taddei. Mapping of coherent structures in parameterized flows by learning optimal transportation with gaussian models. *Journal of Computational Physics*, 471:111671, December 2022b. ISSN 00219991. doi: 10.1016/j.jcp.2022.111671. URL http://arxiv.org/abs/2109.08769.

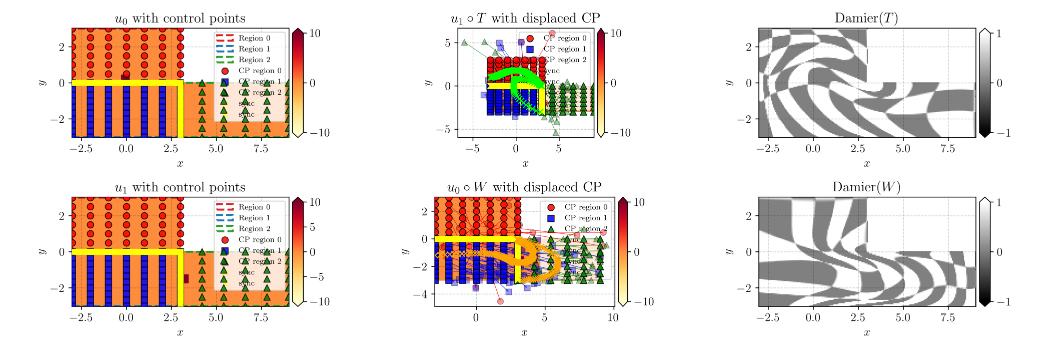
Andriy Myronenko and Xubo Song. Point-set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, December 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.46. URL http://arxiv.org/abs/0905.2635.

Annex: Alignment of squares costs



Cost function evolution during optimization

Results: Alignment on Square cps



Piecewise FFD control points displacements

Free-Form Deformation (FFD)

$$\begin{aligned} T : \mathbb{R}^{N_{cp}} \times \mathbb{R}^3 &\longrightarrow \mathbb{R}^3 \\ \lambda, \xi &\longmapsto T_{\lambda_0}(\xi) = x \end{aligned}$$

In the coordinate system $\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}$
$\mathbf{x} = \mathbf{x_0} + t_1\mathbf{e_1} + t_2\mathbf{e_2} + t_3\mathbf{e_3}$
The mapping is defined as:

$$FFD(t_1, t_2, t_3) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} B_i^l(t_1) B_j^m(t_2) B_k^n(t_3) P_{i,j,k} \tag{1}$$

Here, $P_{i,j,k}$ are the control points in the deformed lattice, and l, m, n denote the degree of the polynomial in each direction.

The (t_1, t_2, t_3) coordinates can be calculated using:

$$t_1 = \frac{(\mathbf{e_2} \times \mathbf{e_3}) \cdot (X - X_0)}{(\mathbf{e_2} \times \mathbf{e_3}) \cdot \mathbf{e_1}} \tag{2}$$

$$t_2 = \frac{(\mathbf{e_1} \times \mathbf{e_3}) \cdot (X - X_0)}{(\mathbf{e_1} \times \mathbf{e_3}) \cdot \mathbf{e_2}} \tag{3}$$

$$t_3 = \frac{(\mathbf{e_1} \times \mathbf{e_2}) \cdot (X - X_0)}{(\mathbf{e_1} \times \mathbf{e_2}) \cdot \mathbf{e_3}} \tag{4}$$

Where a Bernstein polynomial is defined as:

$$B_i^l(t) = \binom{l}{i} t^i (1 - t)^{l-i} \tag{5}$$