

Parametric vector flows for registration fields in bounded domains with applications to nonlinear interpolation of shock-dominated flows

Jon Labatut, Jean-Baptiste Chapelier, Angelo Iollo, Tommaso Taddei

June 7, 2026

Abstract

We present a registration procedure for parametric model order reduction (MOR) in two- and three-dimensional bounded domains. In the MOR framework, registration methods exploit solution snapshots to identify a parametric coordinate transformation that improves the approximation of the solution set through linear subspaces. For each training parameter, optimization-based (or variational) registration methods minimize a target function that measures the alignment of the coherent structures of interest (e.g., shocks, shear layers, cracks) for different parameter values, over a family of bijections of the computational domain Ω . We consider diffeomorphisms Φ that are vector flows of given velocity fields v with vanishing normal component on $\partial\Omega$; we rely on a sensor to extract appropriate point clouds from the solution snapshots and we develop an expectation-maximization procedure to simultaneously solve the point cloud matching problem and to determine the velocity v (and thus the bijection Φ); finally, we combine our registration method with the nonlinear interpolation technique of [Iollo, Taddei, J. Comput. Phys., 2022] to perform accurate interpolations of fluid dynamic fields in the presence of shocks. Numerical results for a two-dimensional inviscid transonic flow past [symmetric and asymmetric](#) NACA airfoils and a three-dimensional viscous transonic flow past an ONERA M6 wing illustrate the many elements of the methodology and demonstrate the effectiveness of nonlinear interpolation for shock-dominated fields.

1 Introduction

The past decade has witnessed a surge in the development of nonlinear approximation strategies for model order reduction (MOR) to tackle parametric problems that feature a slow-decaying Kolmogorov n -width [9, 27, 44]. In this regard, nonlinear approximation methods based on coordinate transformations have established themselves as a prominent technique to handle parametric systems with sharp parameter-dependent coherent structures with compact support such as shocks, shear layers and cracks. If we denote by $\mathcal{M}$ the solution set associated with a given parametric partial differential equation (PDE), the problem of finding a parametric transformation Φ that improves the approximation of $\mathcal{M}$ through linear subspaces is referred to as *registration problem* [55]: these methods are inspired by registration techniques from image processing [16, 35, 41] and are also related to morphing techniques in mesh adaptation [18, 37] and shock-tracking methods in CFD [53, 60]. The aim of this work is to present a new registration method for model reduction and to apply it to nonlinear interpolation of shock-dominated fields.

We denote by Ω a Lipschitz bounded domain in $\mathbb{R}^d$ with $d = 2$ or $d = 3$; we also denote by μ a vector of model parameters in the compact parameter space $\mathcal{P}$; we further define the parametric field $U : \Omega \times \mathcal{P} \rightarrow \mathbb{R}^D$ that solves the parametric PDE model of interest; we use notation $U_\mu = U(\cdot; \mu)$ to refer to the solution to the PDE for the parameter $\mu \in \mathcal{P}$. We further introduce the set of diffeomorphisms $\text{Diff}(\bar{\Omega})$ that contains all vector fields in Ω that are differentiable up to the boundary and are bijective (one-to-one and onto) in Ω . For MOR applications, given the parameter space $\mathcal{P}$, we are interested in parametric diffeomorphisms $\Phi : \bar{\Omega} \times \mathcal{P} \rightarrow \bar{\Omega}$ such that (i) $\Phi_\mu \in \text{Diff}(\bar{\Omega})$ for all $\mu \in \mathcal{P}$, (ii) Φ is continuous with respect to the parameter, and (iii) for all $\mu \in \mathcal{P}$, Φ_μ approximately minimizes

$$\min_{\Phi \in \text{Diff}(\bar{\Omega})} \mathfrak{f}_\mu^{\text{tg}}(\Phi), \quad (1)$$

where $\mathfrak{f}_\mu^{\text{tg}}$ should measure the alignment between the coherent structures associated with the parameter μ and the deformed coherent structures associated with a reference parameter $\mu_{\text{ref}} \in \mathcal{P}$. Registration techniques should hence take as input a set of snapshots $\{U_k := U_{\mu_k} : \mu_1, \dots, \mu_{n_{\text{train}}} \in \mathcal{P}\}$ and return a parametric diffeomorphism that satisfies (i)-(ii)-(iii).

The development of registration techniques in bounded domains requires to address three major challenges: first, the choice of the target function in (1); second, the choice of a finite-dimensional parameterization $\mathbf{N} : \mathbb{R}^{N_h}$ — the subscript $(\cdot)_h$ refers to a finite-dimensional spatial discretization that shall be formally introduced in section 3 — of diffeomorphisms (or Lipschitz bijections) in Ω ; and third, the generalization of Φ to unseen

parameters. As regards the second challenge, we can show that $\text{Diff}(\overline{\Omega})$ is infinite-dimensional, and bijectivity is not stable under small perturbations in the $C^1(\Omega)$ norm (cf. [56, Lemma 3.1]): formally, given two bijections of a curved domain Ω that do not coincide everywhere on the boundary, we can show that the convex combination is not a bijection for any intermediate value of the interpolation parameter $t \in (0, 1)$. The choice of the parameterization is hence crucial to devise tractable finite-dimensional counterparts of the optimization problem (1).

In this work, we introduce vector flows [33] for parametric registration in bounded domains. Given the domain Ω , we introduce a finite element (FE) discretization $V_h \subset \text{Lip}(\Omega; \mathbb{R}^d)$; next, we define $N(\mathbf{v}) : \Omega \rightarrow \mathbb{R}^d$ as the flow of the vector field $v \in V_h$ associated with the FE vector of coefficients $\mathbf{v} \in \mathbb{R}^{N_h}$. Vector flows are diffeomorphisms in Ω under mild assumptions on the velocity v (cf. Theorem 2.1) and can be readily applied in the optimization context.

Following [20, 28], we rely on a scalar testing function to extract a point cloud $Y_\mu := \{y_j(\mu)\}_{j=1}^{Q_\mu} \subset \overline{\Omega}$ from the state U_μ that is associated with the solution features that we want to track. Next, we define the target function $f^{\text{tg}}(\Phi)$ as a weighted distance between a reference point cloud $\{\xi_i\}_{i=1}^N$ and Y_μ : since in general we cannot anticipate a one-to-one correspondence between the two point clouds, we introduce an additional matrix of unknowns $\mathbf{P} \in \mathbb{R}^{N \times Q_\mu}$ whose entries $P_{i,j}$ describe the likelihood of the point ξ_i to be mapped in the point $y_j(\mu)$. To simultaneously learn the velocity $\mathbf{v} \in \mathbb{R}^{N_h}$ and the matrix $\mathbf{P}$, we adapt the expectation maximization (EM) procedure of [41] that iteratively chooses $\mathbf{P}^*$ based on the current estimate $\mathbf{v}^*$ and then updates $\mathbf{v}^*$ by approximately solving a minimization problem of the form

$$\min_{\mathbf{v} \in \mathbb{R}^{N_h}} f_\mu^{\text{obj}}(\mathbf{v}) := f^{\text{tg}}(N(\mathbf{v}); \mathbf{P}^*, \mu) + \lambda f_{\text{pen}}(\mathbf{v}), \quad (2)$$

where f_{pen} is a regularization term that is designed to promote the smoothness of the map, and $\lambda > 0$ is a weighting parameter.

To demonstrate the impact of our registration method to MOR, we consider the application to shock-dominated flows in two and three dimensions. In particular, we consider the transonic flow past a NACA0012 airfoil and a NACA2314 airfoil for varying inflow Mach numbers and the three-dimensional viscous flow past a wing (Onera M6) for varying angles of attack. Towards this end, we resort to the convex displacement interpolation (CDI) method introduced in [28] and further developed in [20]: as described below, CDI can be interpreted as a convex interpolation along the characteristics associated with the vector field v , which is determined using the registration procedure.

Our method is related to several previous works.

- The registration problem (1) is related to the Monge’s problem in optimal transportation theory [51], for a suitable choice of the target function and the penalty function; the analogy with optimal transportation theory enables to guarantee the existence and the uniqueness of the solution to (1) for specific target functions. In this work, we pursue an optimization-based approach: we introduce a problem-dependent target function and a parameterization N that is amenable for computations; then, we use standard optimization tools to estimate a (possibly local) minimum of (1). Optimization-based (or variational) methods for registration were previously considered in [55, 56] for MOR, and also [11, 19] for image registration.
- Thanks to the seminal contributions by Benamou and Brenier [12] (see also [51, Chapter 6]), vector flows have been extensively studied in optimal transportation theory; furthermore, they have also been widely adopted for image registration tasks [11, 19], mesh adaptation [3, 21], parametric geometry reduction [29], and also model reduction [27]. In this regard, the main feature of the present work is the focus on bounded domains: the accurate preservation of the boundary is of critical importance for computational mechanics applications, due to the need to accurately resolve sharp boundary features (e.g., boundary layers) and also to ensure the satisfaction of boundary conditions.
- Since the work of [55], several authors have proposed to directly parameterize the displacement field rather than the velocity field for MOR applications [39, 47, 56]: a thorough comparison of these two classes of methods is beyond the scope of the present work and is the subject of the preprint [26].
- We can distinguish between two classes of nonlinear approximation methods based on coordinate transformation for MOR: Lagrangian and Eulerian methods. *Lagrangian methods* rely on the definition of a reference configuration where coherent structures of interest are approximately fixed [10, 39, 40, 43, 47]; on the other hand, *Eulerian methods* [27, 28] do not require the definition of a reference configuration and can in principle cope with multiple transport fields [13, 30, 48]. As stated above, we here rely on the CDI method for nonlinear interpolation: this method is inherently Eulerian. Nevertheless, we envision that our registration method can be applied in conjunction with other MOR techniques based on coordinate transformation.

- We (Iollo, Taddei) already considered the problem of point cloud matching in [28] and [20]: first, we applied a point set registration method for unbounded geometries to determine the matching and then we used the results to inform the bounded registration problem. The approach of this work can be interpreted as a generalization of the one in [20, 28] and, in our experience, it is significantly more robust, especially for geometric configurations with slender bodies.
- We further remark that several alternative target functions have been proposed in the literature. In the MOR framework, similarly to [11], the author of [55] minimizes the distance between a parameter-dependent mapped function u_μ and a template field $\bar{u}$, that is $\mathcal{J}^{\text{eg}}(\Phi; \mu) = \|u_\mu \circ \Phi - \bar{u}\|_{L^2(\Omega)}^2$; the authors of [57] generalize the latter to measure the L^2 distance from a low-dimensional template space. In the image processing literature, alternative objective functions can be found in [24] and the references therein.
- Finally, we remark that registration techniques critically rely on the assumption that the shock topology is fixed across the parameter range (i.e., the number and type of shocks do not change), as the diffeomorphism would theoretically become singular in the presence of merging or splitting structures. To address this issue, the authors of [58] proposed registration with clustering techniques to cope with parameter-induced shock-topology changes.

The outline of this paper is as follows. In section 2, we introduce vector flows and we review some critical properties that justify its application to registration in bounded domain; we introduce the target function that is used in the computations; and we introduce the CDI method of [28]. In section 3, we present the main elements of the methodology — namely, the FE discretization, the evaluation of the target function and its gradient, the choice of the penalty function, the choice of the sensor, and the deployment of the EM procedure. In section 4, we present the numerical results for the two model problems introduced above. Section 5 concludes the paper.

2 Formulation

2.1 Vector flows

Given the Lipschitz connected domain $\Omega \subset \mathbb{R}^d$, we denote by $\mathbf{n} : \partial\Omega \rightarrow \mathbb{S}^2 = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ the outward unit normal, and by $C^k(\bar{\Omega}; \mathbb{R}^d)$ the space of k -times differentiable functions up to the boundary, with values in $\mathbb{R}^d$. We introduce the space of time-dependent vector fields whose normal component vanishes on $\partial\Omega$ at all times,

$$\mathcal{V}_0 := \{v \in C^1(\bar{\Omega} \times [0, 1]; \mathbb{R}^d) : v(\cdot, t) \cdot \mathbf{n}|_{\partial\Omega} = 0 \ \forall t \in [0, 1]\}$$

Below, $\text{id} : \Omega \rightarrow \Omega$ denotes the identity map, $\text{id}(x) = x$, while $\mathbb{1} \in \mathbb{R}^{d \times d}$ is the identity matrix.

Let $v : \bar{\Omega} \times [0, 1] \rightarrow \mathbb{R}^d$ satisfy $v \in C(\bar{\Omega} \times [0, 1]; \mathbb{R}^d)$, $v(\cdot, t) \in C^1(\bar{\Omega}; \mathbb{R}^d)$ for all $t \in [0, 1]$, and $v(\cdot, t) \cdot \mathbf{n}|_{\partial\Omega} = 0$. Then, we say that $X : \bar{\Omega} \times [0, 1] \rightarrow \bar{\Omega}$ is the flow of the vector field v if

$$\begin{cases} \frac{\partial X}{\partial t}(\xi, t) = v(X(\xi, t), t) & t \in (0, 1], \\ X(\xi, 0) = \xi, \end{cases} \quad \forall \xi \in \bar{\Omega}. \quad (3)$$

Furthermore, we define the vector flow (VF) $F[v] : \bar{\Omega} \rightarrow \bar{\Omega}$ as the end point of the flow X , that is $F[v](\xi) := X(\xi, t = 1)$. If we restrict ourself to a finite-dimensional velocity space, we obtain the family of diffeomorphisms

$$\mathbf{N}(\xi; \mathbf{v}) := F[v(\mathbf{v})](\xi), \quad \text{where } v(x, t; \mathbf{v}) = \sum_{i=1}^{N_h} (\mathbf{v})_i \phi_i(x, t), \quad (4)$$

where $\mathbf{v} \in \mathbb{R}^{N_h}$ is a vector of coefficients, and $\phi_1, \dots, \phi_{N_h} \in \mathcal{V}_0(\Omega)$ are linearly-independent. In view of the analysis, we introduce a relevant subset of diffeomorphisms.

Definition 2.1. ([8, Chapter 1, page 3]) Let Φ be a diffeomorphism in the Lipschitz domain $\Omega \subset \mathbb{R}^d$. We say that Φ is isotopic to the identity if there exists a smooth field (isotopy) $X \in C^1(\bar{\Omega} \times [0, 1]; \mathbb{R}^d)$ such that (i) $X(\cdot, 0) = \text{id}$, (ii) $X(\cdot, 1) = \Phi$, (iii) the map $X(\cdot, t) \in \text{Diff}(\bar{\Omega})$ for all $t \in [0, 1]$. We denote by $\text{Diff}_0(\bar{\Omega})$ the subset of diffeomorphisms that are isotopic to the identity.

We can easily construct diffeomorphisms that do not belong to $\text{Diff}_0(\bar{\Omega})$ (see, e.g., [26]); nevertheless, we observe that $\text{Diff}_0(\bar{\Omega})$ is the subset of diffeomorphisms of interest for model reduction applications. Given the parameter domain $\mathcal{P}$, in model order reduction, we seek parametric mappings $\Phi : \bar{\Omega} \times \mathcal{P} \rightarrow \bar{\Omega}$ such that $\Phi(\cdot, \mu) : \bar{\Omega} \rightarrow \bar{\Omega}$ is a diffeomorphism for all $\mu \in \mathcal{P}$ [55]. Exploiting Definition 2.1, we can readily show that if (i) $\mathcal{P}$ is connected, (ii) $\Phi(\cdot, \mu')$ is equal to the identity for some $\mu' \in \mathcal{P}$, and (iii) Φ is continuous with respect to the parameter, we must have that $\Phi(\cdot, \mu) \in \text{Diff}_0(\bar{\Omega})$ for all $\mu \in \mathcal{P}$. Below, we summarize key properties of vector flows that show their relevance for registration tasks. We refer to [26] (see also [59]) for the proofs.

Proposition 2.1. Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain; we denote by $\mathcal{W}_h := \text{span}\{\phi_i\}_{i=1}^{N_h} \subset \mathcal{V}_0(\Omega)$ the approximation space associated to (4). Then, the following hold.

- **Bijectivity.** Let $v \in \mathcal{V}_0(\Omega)$. Define $\Phi : \bar{\Omega} \rightarrow \mathbb{R}^d$ such that $\Phi = F[v]$. Then, Φ is a diffeomorphism (one-to-one and onto) in $\bar{\Omega}$.
- **Approximation.** Let $\Phi \in \text{Diff}_0(\bar{\Omega})$. Then, there exists $v \in \mathcal{V}_0(\Omega)$ such that $\Phi = F[v]$. Furthermore, if we denote by L the Lipschitz constant of v , we have that

$$\inf_{\mathbf{v} \in \mathbb{R}^{N_h}} \|\Phi - \mathbf{N}(\mathbf{v})\|_{L^\infty(\Omega)} \leq \frac{e^L - 1}{L} \inf_{\phi \in \mathcal{W}_h} \|v - \phi\|_{L^\infty(\Omega \times (0,1))}. \quad (5)$$

- **Derivative.** The derivative of the VF (4) with respect to the coefficients $\mathbf{v}$ is given by:

$$\frac{\partial \mathbf{N}}{\partial a_i}(\xi; \mathbf{v}) = \nabla X(\xi, 1; \mathbf{v}) \int_0^1 (\nabla X(\xi, \tau; \mathbf{v}))^{-1} \phi_i(X(\xi, \tau; \mathbf{v}), \tau) d\tau, \quad \text{for } i = 1, \dots, M; \quad (6)$$

where the gradient $t \mapsto \nabla X(\xi, t; \mathbf{v})$ satisfies the differential equation:

$$\begin{cases} \frac{\partial \nabla X}{\partial t}(\xi, t; \mathbf{v}) = \nabla_x v(X(\xi, t; \mathbf{v}), t; \mathbf{v}) \nabla X(\xi, t; \mathbf{v}) & t \in (0, 1], \\ \nabla X(\xi, 0; \mathbf{v}) = \mathbf{1}. \end{cases} \quad (7)$$

- **Jacobian determinant.** Let $\Phi = F[v]$ for some $v \in \mathcal{V}_0(\Omega)$. Then, the Jacobian determinant $J := \det(\nabla \Phi)$ satisfies

$$J(\xi) = \exp \left(\int_0^1 \nabla_x \cdot v(X(\xi, s), s) ds \right), \quad \xi \in \Omega, t \in [0, 1]. \quad (8)$$

Some comments are in order.

- The VF (4) is guaranteed to be a diffeomorphism of $\bar{\Omega}$ for all $\mathbf{v} \in \mathbb{R}^{N_h}$; therefore, the penalty term in (2) should solely be designed to promote smoothness rather than to ensure bijectivity.
- Eq. (5) shows that any element of $\text{Diff}_0(\bar{\Omega})$ can be expressed in the form (3) and that we can explicitly control the approximation error.
- It is easy to show that the practical computation of $\frac{\partial \mathbf{N}}{\partial \mathbf{v}}(\mathbf{v})$ at the discrete level for typical (e.g., Runge Kutta, linear multi-step) time-integration schemes involves the application of the chain rule over all time steps: it is hence computationally expensive. On the other hand, Eq. (6) provides a constructive way to evaluate the continuous derivative of the mapping — and ultimately of the target function — with respect to the generalized coordinates $\mathbf{v}$ at roughly the same cost of evaluating $\mathbf{N}$.
- Eq. (8) signifies that the Jacobian determinant of VF maps is strictly positive, which implies that VF maps preserve orientation.

2.2 Target function

In this work, we resort to a discrete matching function, which is broadly used in image registration to align unmatched point clouds [41]. Given the parameter $\mu \in \mathcal{P}$, we introduce the reference (parameter-independent) point cloud $\{\xi_i\}_{i=1}^N$ and the target (parameter-dependent) cloud $\{y_j(\mu)\}_{j=1}^{Q_\mu}$. Next, we define the matrix of weights $\mathbf{P} = [P_{i,j}]_{i,j} \in \mathbb{R}^{N \times Q_\mu}$, which was already introduced in section 1, such that $P_{i,j} \in [0, 1]$, and $\sum_j P_{i,j} \leq 1$ and $\sum_i P_{i,j} \leq 1$ for all $i = 1, \dots, Q_\mu$ and $j = 1, \dots, N$. As discussed in [41] the weights $\{P_{i,j}\}_{i,j}$ describe the likelihood of the point ξ_j to be mapped in the point $y_i(\mu)$. Finally, we introduce the *pointwise objective*:

$$\mathfrak{f}^{\text{tg}}(\Phi; \mathbf{P}, \mu) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{Q_\mu} P_{i,j} \|\Phi(\xi_i) - y_j(\mu)\|_2^2. \quad (9)$$

We notice that the target function (9) is bounded from below; provided that $\lim_{\|\mathbf{v}\|_2 \rightarrow \infty} \mathfrak{f}_{\text{pen}}(\mathbf{v}) = +\infty$, we hence find that the minimization statement (2) has a (non-necessarily unique) minimum in $\mathbb{R}^{N_h}$ for any $\lambda > 0$ (cf. [26]). The point clouds in (9) are extracted from the solution field using a properly-chosen point-set sensor (cf. section 3.2); on the other hand, the weights $\mathbf{P}$ are chosen using a variant of the expectation-maximization (EM) procedure of [41] (cf. section 3.5). We postpone the discussion on the practical construction of the weights $\mathbf{P}$ and the point-set sensor for the model problems of this work to section 3.

We can exploit (6) and the chain rule to compute the derivative of the objective $E : \mathbf{v} \mapsto \mathfrak{f}^{\text{tg}}(\mathbf{N}(\mathbf{v}); \mathbf{P}, \mu)$. In more detail, we find

$$\frac{\partial E}{\partial v_i}(\mathbf{v}; \mathbf{P}, \mu) = D\mathfrak{f}^{\text{tg}}[\mathbf{N}(\mathbf{v}), \mathbf{P}, \mu] \left(\frac{\partial \mathbf{N}}{\partial v_i}(\mathbf{v}) \right), \quad i = 1, \dots, M, \quad (10a)$$

where the expression of $\frac{\partial \mathbf{N}}{\partial v_i}(\mathbf{v})$ is provided in (6) and

$$D\mathfrak{f}^{\text{tg}}[\Phi; \mathbf{P}, \mu](h) = \sum_{i=1}^N \sum_{j=1}^{Q_\mu} P_{i,j} (\Phi(\xi_i) - y_j(\mu)) \cdot h(\xi_i) = \sum_{i=1}^N \left(\Phi(\xi_i) - \sum_{j=1}^{Q_\mu} P_{i,j} y_j(\mu) \right) \cdot h(\xi_i), \quad (10b)$$

As discussed in the next section, the gradient of E is employed by the gradient-descent technique to find local minima of (2).

2.3 Convex displacement interpolation

In this work, we rely on convex displacement interpolation (CDI) to perform nonlinear interpolation of fluid dynamics fields. CDI was introduced in [28] for the interpolation of two snapshots, and further extended in [20] to deal with multiple snapshots and multi-dimensional parameterizations. Below, we briefly review the CDI as introduced in [28].

We denote by $\mu_0, \mu_1 \in \mathcal{P}$ two parameters and by $U_0, U_1 : \Omega \rightarrow \mathbb{R}^D$ the corresponding solution fields, $U_i := U(\cdot; \mu_i)$, $i = 0, 1$; CDI provides a nonlinear interpolation $\widehat{U} : \Omega \times [0, 1] \rightarrow \mathbb{R}^D$ of the solution $s \in [0, 1] \mapsto U_s = U(\cdot; (1-s)\mu_0 + s\mu_1)$. Towards this end, given the velocity field $v \in \mathcal{V}_0(\Omega)$ and $s \in [0, 1]$, we define the mappings $\Phi_{s \rightarrow 0}$ and $\Phi_{s \rightarrow 1}$ such that

$$\Phi_{s \rightarrow 0}(\xi) = Y(\xi, t = 0), \quad \Phi_{s \rightarrow 1}(\xi) = X(\xi, t = 1), \quad (11a)$$

where X and Y are the backward and the forward flows associated with the velocity v ,

$$\begin{cases} \frac{\partial X}{\partial t}(\xi, t) = v(X(\xi, t), t) & t \in (s, 1], \\ X(\xi, s) = \xi, \end{cases} \quad \begin{cases} \frac{\partial Y}{\partial t}(\xi, t) = -v(Y(\xi, t), t) & t \in (0, s], \\ Y(\xi, s) = \xi, \end{cases} \quad (11b)$$

for all $\xi \in \Omega$. Then, we introduce the CDI of the parametric field U as

$$\widehat{U}(\xi, s) = (1-s)U_0 \circ \Phi_{s \rightarrow 0}(\xi) + sU_1 \circ \Phi_{s \rightarrow 1}(\xi) \quad \xi \in \Omega, \quad s \in [0, 1]. \quad (11c)$$

As discussed in [28], we can interpret the CDI as a convex interpolation along the characteristics associated with the vector field v . Similarly, we can also view (11c) as a symmetrized McCann (or displacement) interpolation [45]: note that the latter is defined only for probability measures, while (11c) is defined for arbitrary vector-valued fields.

Recalling Proposition 2.1, we find that $\Phi_{s \rightarrow 0}$ and $\Phi_{s \rightarrow 1}$ are diffeomorphisms for all $s \in [0, 1]$: CDI is hence well-defined for any $v \in \mathcal{V}_0(\Omega)$, provided that U_0, U_1 are measurable. Furthermore, CDI ensures interpolation at end points, and satisfies maximum and minimum principles [20]. Finally, we notice that, if v is time-independent, $\Phi_{s \rightarrow 0} = X'(\cdot, s)$ where X' is the flow associated with $-v$ and $\Phi_{s \rightarrow 1} = X(\cdot, 1-s)$.

3 Methodology

We present the elements of the methodology.

- In section 3.1, we introduce the finite element (FE) discretization of the velocity field. The FE method provides a finite-dimensional discretization V_h of the space of stationary admissible stationary velocities $\mathcal{U}_0 = \{v \in C^1(\overline{\Omega}; \mathbb{R}^d) : v \cdot \mathbf{n}|_{\partial\Omega} = 0\}$.
- In section 3.2, we define the scalar testing functions that are used to extract the point cloud $Y_\mu = \{y_j(\mu)\}_{j=1}^{Q_\mu}$ for the state U_μ (see (9)).
- In section 3.3, we discuss the evaluation of the target (9) $E(\mathbf{v}) = \mathfrak{f}^{\text{tg}}(\mathbf{N}(\mathbf{v}); \mathbf{P}, \mu)$ and its gradient for any choice of the weights $\mathbf{P}$ and the point clouds.
- In section 3.4, we comment on the choice of the penalty term in (2) that ensures the smoothness of the map and the satisfaction of the constraints $v \cdot \mathbf{n}|_{\partial\Omega} = 0$.
- In section 3.5, we present the complete EM procedure for the computation of the velocity v and the weights $\mathbf{P}$. The EM algorithm requires the introduction of a probabilistic model for the target points.

In this work we consider stationary velocity fields to reduce computational costs. Our implementation is based on the FE library FEniCS [5], and on the linear algebra and optimization software PETSC [7].

3.1 Finite element discretization

We resort to a continuous FE discretization of the velocity field. Given the domain $\Omega \subset \mathbb{R}^d$, we introduce the mesh $\mathcal{T} = (\{x_j\}_{j=1}^{N_v}, \mathbf{T})$ with nodes $\{x_j\}_{j=1}^{N_v}$ and connectivity matrix $\mathbf{T} \in \mathbb{N}^{N_e \times n_{lp}}$, where N_e is the number of elements in the mesh and n_{lp} is the number of local degrees of freedom. In this work, we consider linear triangular meshes for $d = 2$ -dimensional domains and linear tetrahedral meshes for $d = 3$ -dimensional domains: we denote by $\{\mathbf{D}_k\}_{k=1}^{N_e}$ the elements of the mesh and by $\{\mathbf{e}_j\}_{j=1}^{N_f}$ the facets of the mesh. We further denote by $h > 0$ the characteristic size of the mesh,

$$h := \max_{k=1, \dots, N_e} \sqrt[d]{|\mathbf{D}_k|},$$

and we introduce the approximate domain $\Omega_h := \bigcup_{k=1}^{N_e} \mathbf{D}_k$ with outward normal $\mathbf{n}_h : \partial\Omega_h \rightarrow \mathbb{S}_2$.

Exploiting this notation, we introduce the FE space of degree p associated with the mesh $\mathcal{T}$:

$$V_h = \{v \in C^0(\overline{\Omega}_h; \mathbb{R}^d) : v|_{\mathbf{D}_k} \in [\mathbb{P}_p(\mathbf{D}_k)]^d, k = 1, \dots, N_e\}, \quad (12)$$

where $\mathbb{P}_p(\mathbf{D}_k)$ is the space of polynomials in $\mathbf{D}_k$ of degree less or equal to p . We further introduce the Lagrangian basis $\{\ell_i\}_{i=1}^{N_v}$ associated with the nodes $\{x_j\}_{j=1}^{N_v}$ of $\mathcal{T}$ and the canonical basis $\mathbf{e}_1, \dots, \mathbf{e}_d$ of $\mathbb{R}^d$. Given the FE field $v \in V_h$, we denote by $\mathbf{v} \in \mathbb{R}^{N_h}$ with $N_h = N_v d$ the corresponding FE vector such that

$$v(x) := \sum_{j=1}^{N_h} (\mathbf{v})_j \phi_j(x), \quad \text{where } \phi_{i+(\ell-1)N_v}(x) := \ell_i(x) \mathbf{e}_\ell, \quad i = 1, \dots, N_v, \ell = 1, \dots, d. \quad (13)$$

We notice that the FE discretization introduces several sources of approximation that should be taken into account. First, the space V_h does not embed the constraint $v \cdot \mathbf{n}|_{\partial\Omega} = 0$; furthermore, the elements of V_h do not belong to C^1 . We should hence rely on the penalty function of (2) to impose these constraints. Second, if the domain Ω is curved, the use of linear meshes leads to a geometric error $\text{dist}_H(\Omega_h, \Omega)$ of order $\mathcal{O}(h)$, where the Hausdorff distance $\text{dist}_H(\Omega_h, \Omega)$ between the domains Ω_h and Ω is given by

$$\text{dist}_H(\Omega_h, \Omega) := \max\left\{ \inf_{x \in \Omega_h} \sup_{y \in \Omega} \|x - y\|_2, \inf_{y \in \Omega} \sup_{x \in \Omega_h} \|x - y\|_2 \right\}.$$

Furthermore, in the absence of a proper post-treatment, the outward normal $\mathbf{n}_h$ to Ω_h differs from the outward normal $\mathbf{n}$ to the domain Ω : this discrepancy induces an approximation in the imposition of the constraint $v \cdot \mathbf{n}|_{\partial\Omega} = 0$. In the numerical experiments, we assess the impact of these approximations to the results. [As discussed at the beginning of section 3, we rely on FEniCS \(v0.7.2\) for the finite element discretization of the mapping: since this version does not support higher-order meshes, we only consider low-order discretizations \(\$p = 1\$ or \$p = 2\$ \).](#)

3.2 Choice of the sensor

Given the parametric fluid state $\mu \in \mathcal{P} \mapsto U_\mu$, we should extract the appropriate point cloud $Y_\mu := \{y_j(\mu)\}_{j=1}^{Q_\mu} \subset \overline{\Omega}$ that is associated with the flow features of interest (e.g., shocks, shear layers, or vortices). Following [28], we perform this task by first introducing a finite-dimensional discretization Ω_{discr} of the domain Ω and a scalar testing function $\mathfrak{T} : \Omega \times \mathcal{P} \rightarrow \mathbb{R}$, and then by defining Y_μ as

$$Y_\mu := \{x \in \Omega_{\text{discr}} : \mathfrak{T}(x, \mu) > 0\}. \quad (14)$$

In the numerical experiments, we define Ω_{discr} as the union of all the cell centroids of the mesh which is used for CFD calculations; on the other hand, in order to define the function $\mathfrak{T}$, we first introduce the sensor

$$\varsigma_s(x; U_\mu) = \frac{1}{a_\mu(x) \|\nabla p_\mu(x)\|_2} (v_\mu(x)^\top \nabla p_\mu(x)), \quad (15a)$$

where a_μ is the sound speed, p_μ is the pressure, and v_μ is the fluid velocity. The sensor ς_s is a broadly-used shock detector for compressible flows¹. The corresponding scalar testing function is given by

$$\mathfrak{T}_s(x, \mu) = \varsigma_s(x; U_\mu) - \text{tol}, \quad (15b)$$

where $\text{tol} \in (0, 1)$ is a tunable parameter.

In the numerical experiments, we also tested the scalar testing function

$$\mathfrak{T}_{\text{iso}}^\pm(x, \mu) = \pm \varsigma_{\text{iso}}(x; U_\mu) - \text{tol}, \quad (16a)$$

¹See the User's manual of Tecplot 360, Release 1 (2013), page 580 of 598.

that aims to align selected isolines of the normalized density field,

$$\text{S}_{\text{iso}}(x; U_\mu) := \begin{cases} \frac{\rho_\mu(x) - \bar{\rho}_\mu}{\rho_{\text{max},\mu} - \bar{\rho}_\mu}, & \text{if } \rho_\mu(x) - \bar{\rho}_\mu > 0, \\ \frac{\rho_\mu(x) - \bar{\rho}_\mu}{\bar{\rho}_\mu - \rho_{\text{min},\mu}}, & \text{if } \rho_\mu(x) - \bar{\rho}_\mu < 0. \end{cases} \quad (16b)$$

where $\bar{\rho}_\mu$, $\rho_{\text{min},\mu}$, $\rho_{\text{max},\mu}$ are the mean, the minimum and the maximum of the density ρ_μ over Ω . This alternative scalar testing function is designed to identify strong compression and expansion regions, for aerodynamic external flows.

3.3 Evaluation of the target function

As discussed in section 2, the evaluation of the flow X requires the integration of the system of d nonlinear ODEs (3); furthermore, the evaluation of the derivatives of the flow with respect to the velocity degrees of freedom requires the solution to the linear system of $d \times d$ ODEs (7). Algorithm 1 summarizes the procedure for the computation of the target (9) and its derivative (10b): for simplicity, in Algorithm 1, we consider the explicit Euler method for time integration; furthermore, we introduce the set of “active” degrees of freedom,

$$\mathbf{I}_{\text{act}}(\{x_j\}_{j=1}^N) := \{i \in \{1, \dots, N_h\} : x_j \in \text{spt}(\phi_i), \text{ for some } j \in \{1, \dots, N\}\}, \quad (17)$$

where $\text{spt}(\phi_i)$ denotes the support of the function ϕ_i .

Algorithm 1: Evaluation of the target (9) and its derivative (10b).

Data: $\mathbf{P} \in \mathbb{R}_+^{N \times Q_\mu}$ sets of weights; $\{\xi_i\}_{i=1}^N$, $\{y_j\}_{j=1}^{Q_\mu}$ reference and target point clouds; $\mathbf{v} \in \mathbb{R}^{N_h}$ velocity field; $(\Delta t, N_t)$ time grid.

Result: $E(\mathbf{v})$, $\nabla E(\mathbf{v})$.

- 1 *Initialization:* set $X_i^{(0)} \leftarrow \xi_i$, $\nabla X_i^{(0)} \leftarrow \mathbb{1}$, $\frac{\partial \mathbf{N}_i}{\partial v_\ell} = 0$, for $i = 1, \dots, N$, $\ell = 1, \dots, N_h$.
 - 2 **for** $k = 0, \dots, N_t - 1$ **do**
 - 3 Compute $v_i^{(k)} := v(X_i^{(k)})$, $\nabla v_i^{(k)} := \nabla v(X_i^{(k)})$ for $i = 1, \dots, N$;
 - 4 Update positions and gradients: $X_i^{(k+1)} \leftarrow X_i^{(k)} + \Delta t v_i^{(k)}$, $\nabla X_i^{(k+1)} \leftarrow \nabla X_i^{(k)} + \Delta t \nabla v_i^{(k)} \nabla X_i^{(k)}$, for $i = 1, \dots, N$;
 - 5 Identify active DOFs: $\mathbf{I}_{\text{act}}^{(k)} := \mathbf{I}_{\text{act}}(\{X_i^{(k)}\}_i \cup \{X_i^{(k+1)}\}_i)$;
 - 6 Update $\frac{\partial \mathbf{N}_i}{\partial v_\ell} = \frac{\partial \mathbf{N}_i}{\partial v_\ell} + \frac{\Delta t}{2} \left((\nabla X_i^{(k+1)})^{-1} \phi_\ell(X_i^{(k+1)}) + (\nabla X_i^{(k)})^{-1} \phi_\ell(X_i^{(k)}) \right)$ for all $\ell \in \mathbf{I}_{\text{act}}^{(k)}$.
 - 7 Return $E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{Q_\mu} P_{i,j} \|X_i^{(N_t)} - y_j(\mu)\|^2$
 - 8 $\frac{\partial \mathbf{N}_i}{\partial v_\ell} \leftarrow \nabla X_i^{(N_t)} \frac{\partial \mathbf{N}_i}{\partial v_\ell}$, for $i = 1, \dots, N$, $\ell \in \bigcup_k \mathbf{I}_{\text{act}}^{(k)}$.
 - 9 Return $\frac{\partial E}{\partial v_\ell} = \sum_{i=1}^N \left(X_i^{(N_t)} - \sum_{j=1}^{Q_\mu} P_{i,j} y_j(\mu) \right) \cdot \frac{\partial \mathbf{N}_i}{\partial v_\ell}$, for $\ell \in \bigcup_k \mathbf{I}_{\text{act}}^{(k)}$.
-

Some comments are in order.

- At each time step, we evaluate the FE field $v \in V_h$ and its gradient at the points $\{X_i^{(k)}\}_i$ (cf. Line 3). If $X_i^{(k)} \in \Omega_h$, we first find the element of the mesh to which $X_i^{(k)}$ belongs using an octree search and then we use the functional expression of the velocity field to determine its pointwise value (cf. FEniCS documentation). If $X_i^{(k)} \notin \Omega_h$, we consider a zero-th order extrapolation: we replace $X_i^{(k)}$ with the nearest vertex of the mesh and then we use the same expression as before to evaluate v . We rely on k -dimensional (KD, [34]) trees to rapidly identify the nearest mesh node.
- The approximation of $\{\frac{\partial \mathbf{N}_i}{\partial v_\ell}\}_{i,\ell}$ in Algorithm 1 corresponds to a second-order approximation of the integral (6) based on the trapezoidal rule. We notice that the procedure does not require the storage of the whole trajectories $\{X_i^{(k)}\}_{i,k}$ and $\{\nabla X_i^{(k)}\}_{i,k}$. Furthermore, it requires exactly N octree searches at each time iteration to determine the elements of the mesh $\mathbf{I}_{\text{el}}^{(k)} \subset \{1, \dots, N_e\}$ where the points $\{X_i^{(k)}\}_{i=1}^N$ lie: for this reason, to improve efficiency, we only update the estimate of $\{\frac{\partial \mathbf{N}_i}{\partial v_\ell}\}_{i,\ell}$ for the active degrees of freedom (17)

(cf. Line 9). Note that the active degrees of freedom can be readily identified from the sampled elements $\mathbf{I}_{\text{el}}^{(k)}$, the definition of the Lagrangian basis and the connectivity matrix $\mathbf{T}$.

- In the numerical experiments, if not specified otherwise, we resort to the explicit midpoint method, which is a second-order two-stage Runge Kutta scheme: given the ODE system $\dot{y} = f(t, y)$ and the time grid $\{t^{(k)} = k\Delta t\}_{k=0}^{N_t}$, the method reads as

$$y^{(k+1)} = y^{(k)} + \Delta t f\left(t^{(k)} + \frac{\Delta t}{2}, y^{(k)} + \frac{\Delta t}{2} f\left(t^{(k)}, y^{(k)}\right)\right), \quad k = 0, 1, \dots \quad (18)$$

We can readily adapt Algorithm 1 to cope with other Runge Kutta explicit schemes by properly modifying Line 4. On the other hand, the development of effective adaptive methods is more involved due to the need to synchronize the N independent systems of ODEs and is beyond the scope of the present work.

- As anticipated in section 2, Algorithm 1 returns an estimate of ∇E based on the continuous derivative of $\mathbf{v} \mapsto \mathbf{N}(\mathbf{v})$ in (6), which differs from the actual gradient of the discrete objective: in the numerical experiments, we assess the accuracy of this approximation for typical choices of the time step Δt .

3.4 Choice of the penalty term

The penalty term $\mathbf{f}_{\text{pen}}$ should be designed to promote smoothness and to ensure the constraint $v \cdot \mathbf{n}|_{\partial\Omega} = 0$. Recalling the Rellich-Kondrachov embedding theorem (cf. [1, Chapter 6.3]), we find that the Sobolev space $H^s(\Omega)$ is contained in $C^p(\Omega)$ if $s > \frac{d}{2} + p$. It is hence natural to bound the $H^s(\Omega)$ norm of the velocity for $s = 2$ or $s = 3$ — for $d = 2$ or $d = 3$, the former ensures continuity of the velocity, while the latter also ensures continuity of its derivatives. Since in this work we resort to a standard FE discretization which is not H^2 -conforming, the definition of the discrete counterpart of such norms requires care and is outlined below.

Towards this end, we introduce the bilinear forms $b_0, b, m : V_h \times V_h \rightarrow \mathbb{R}$ such that

$$\begin{cases} b_0(u, v) = \int_{\Omega} \frac{1}{\kappa_0^2} \nabla u : \nabla v + u \cdot v \, dx, & m(u, v) = \int_{\Omega} u \cdot v \, dx; \\ b(u, v) = b_0(u, v) + \sum_{\mathbf{e}_h \in \mathcal{E}_h^{\text{bnd}}} \int_{\mathbf{e}_h} \left(\frac{C_{\text{pen}}}{|\mathbf{e}_h|} (u \cdot \mathbf{n}_h)(v \cdot \mathbf{n}_h) - (\mathbf{n}_h^\top \nabla u \, \mathbf{n}_h)(v \cdot \mathbf{n}_h) - (\mathbf{n}_h^\top \nabla v \, \mathbf{n}_h)(u \cdot \mathbf{n}_h) \right) ds \end{cases} \quad (19)$$

where $C_{\text{pen}}, \kappa_0 > 0$ are user-defined constants and $\mathcal{E}_h^{\text{bnd}}$ denotes the set of boundary facets of the mesh $\mathcal{T}$. The interior penalty term ensures that the bilinear form is coercive: in the experiments, we set $C_{\text{pen}} = 100$ (which is slightly more conservative than the choice recommended in [46, Chapter 11]). The parameter κ_0 can be interpreted as a cutoff frequency: large values of κ_0 allow the velocity field to capture high-frequency details and align point sets with higher accuracy; at the price of higher risks of overfitting and creating near-zero Jacobian determinants. In our experiments, we set $\kappa_0 = 4.5$ for the two-dimensional test cases and $\kappa_0 = 1$ for the three-dimensional test cases.

We further define the corresponding matrices $\mathbf{B}_0, \mathbf{B}, \mathbf{M}$ such that $(\mathbf{B}_0)_{i,j} = b_0(\varphi_j, \varphi_i)$, $(\mathbf{B})_{i,j} = b(\varphi_j, \varphi_i)$ and $(\mathbf{M})_{i,j} = m(\varphi_j, \varphi_i)$ for $i, j = 1, \dots, N_h$. It is straightforward to verify that $\mathbf{B}_0, \mathbf{M}$ are symmetric positive definite; provided that C_{pen} is sufficiently large, it can be shown that $\mathbf{B}$ is also symmetric positive definite [42]. Exploiting this notation, we consider the quadratic penalty:

$$\mathbf{f}_{\text{pen}}(\mathbf{v}) = \frac{1}{2} \mathbf{v}^\top \mathbf{K} \mathbf{v} \quad \text{where} \quad \mathbf{K} = (\mathbf{B} \mathbf{M}^{-1})^{s-1} \mathbf{B}, \quad (20)$$

and $s \in \mathbb{N}$, $s \geq 2$.

To justify (20), we provide a functional interpretation of the solution to the linear system $\mathbf{K} \mathbf{u} = \mathbf{F}$ where $\mathbf{F} \in \mathbb{R}^{N_h}$ is the FE vector associated with the functional $F \in V_h'$. The solution $\mathbf{u}$ can be obtained using the iterative procedure: first, for $k = 1, \dots, s-1$, we solve

$$\mathbf{B} \mathbf{w}_k = \begin{cases} \mathbf{F} & \text{if } k = 1, \\ \mathbf{M} \mathbf{w}_{k-1} & \text{if } k > 1; \end{cases} \quad (21)$$

then, we solve the system:

$$\mathbf{B} \mathbf{u} = \mathbf{M} \mathbf{w}_{s-1}. \quad (22)$$

We observe that, if F is the discretization of a functional in $H^{-1}(\Omega)$, the linear system $\mathbf{B} \mathbf{w}_1 = \mathbf{F}$ in (21) is a FE element discretization of the differential problem:

$$\begin{cases} -\frac{1}{\kappa_0^2} \Delta u + u = F & \text{in } \Omega; \\ u \cdot \mathbf{n} = 0, \quad \partial_n u \cdot \mathbf{t}_\ell = 0, \quad \ell = 1, \dots, d-1 & \text{on } \partial\Omega; \end{cases}$$

where the first equation should be intended in the sense of distributions and $\mathbf{t}_1, \dots, \mathbf{t}_{d-1}$ are orthonormal tangent vectors on $\partial\Omega$. Note that the Dirichlet boundary condition $u \cdot \mathbf{n} = 0$ is enforced weakly using the Nitsche's method [42]. Similarly, the linear system $\mathbf{B}u = \mathbf{M}\mathbf{w}_{k-1}$ in (22) is a FE element discretization of the differential problem:

$$\begin{cases} -\frac{1}{\kappa_0^2} \Delta u + u = w_{k-1} & \text{in } \Omega; \\ u \cdot \mathbf{n} = 0, \quad \partial_n u \cdot \mathbf{t}_\ell = 0, \quad \ell = 1, \dots, d-1 & \text{on } \partial\Omega. \end{cases}$$

In conclusion, the action of $\mathbf{K}^{-1}$ can be interpreted as the solution to a Laplace problem with mixed boundary conditions followed by $s-1$ Laplacian smoothing steps. Since the Laplace operator is a linear bounded operator from H^α to $H^{\alpha-2}$, we can interpret the matrix $\mathbf{K}$ as the discretization of an operator from H^α to $H^{\alpha-2s}$. We can thus loosely interpret the penalty function in (20) as a proxy of the H^s norm. [Recalling the Sobolev injection theorem, we choose \$s = \lceil \frac{d}{2} \rceil + 1\$ to weakly enforce a \$C^1\$ velocity field relative to the physical dimension \$d\$.](#)

We might also attempt to interpret the system $\mathbf{K}u = \mathbf{F}$ as a mixed FE discretization (see [32] and the references therein) of the polyharmonic operator (see, e.g., [6]) for a suitable choice of the boundary conditions. A rigorous formalization of this intuition is, however, beyond the scope of the present work.

3.5 Expectation-maximization procedure

We remove dependence on the parameter $\mu \in \mathcal{P}$ to shorten notation; furthermore, we denote by Q (as opposed to Q_μ) the total number of target points $\{y_j\}_{j=1}^Q \subset \bar{\Omega}$. Since the weights $\mathbf{P}$ in (9) are typically unknown, we should optimize (2) with objective (9) and penalty (20) for both the velocity v and the weights $\mathbf{P}$. Towards this end, following [41], we resort to an expectation-maximization (EM) procedure.

The EM algorithm is a well-established technique in machine learning, which was proposed in [22] to find maximum likelihood solutions for statistical models with latent (unobservable) variables. We refer to [15, Chapter 9] and [25, Chapter 8] for a thorough review of the methodology and for a rigorous theoretical justification: in particular, we refer to [15, Chapter 9.4] and [25, Chapter 8.5.3] for the interpretation of the EM method as an alternating minimization algorithm.

We introduce the basic EM procedure in section 3.5.1; next, in section 3.5.2, we introduce the probabilistic model that is considered for the registration task; in section 3.5.3, we review the complete procedure and we comment on the termination condition; finally, in section 3.5.4, we discuss the computational procedure for the optimization problem (2) that is solved at each EM iteration.

3.5.1 Abstract procedure

The goal of the EM algorithm is to find maximum likelihood (ML) solutions for models with latent variables. We denote by $\mathbf{Y} = [y_1, \dots, y_Q] \in \mathbb{R}^{d \times Q}$ the observed data, and by $\mathbf{Z} = (z_1, \dots, z_Q) \in \{1, \dots, N\}^Q$ the latent variables for some $N \in \mathbb{N}$, and by $\theta \in \Theta$ the set of model parameters. We introduce the probability model $\theta \mapsto \Pr(\mathbf{Y}, \mathbf{Z} | \theta)$ and the corresponding marginal distribution $\theta \mapsto \Pr(\mathbf{Y}, \mathbf{Z} | \theta)$ obtained by integrating the joint density over $\mathbf{Z}$,

$$\Pr(\mathbf{Y} | \theta) = \sum_{\mathbf{Z} \in \{1, \dots, N\}^Q} \Pr(\mathbf{Y}, \mathbf{Z} | \theta) \quad (23a)$$

We can also consider continuous latent variables by replacing the sum with an integral over a continuous measure. Then, we seek solutions to the maximization problem:

$$\theta^* \in \arg \max_{\theta \in \Theta} \Pr(\mathbf{Y} | \theta) = \arg \max_{\theta \in \Theta} \log(\Pr(\mathbf{Y} | \theta)). \quad (23b)$$

Direct computation of the solution to (23) is computationally challenging due to the fact that the summation over $\mathbf{Z}$ in (23) appears inside the logarithm; however, for many probability models of interest, the estimation of the solution to (23) becomes significantly easier if $\mathbf{Z}$ is known. For this reason, the EM algorithm features an iterative procedure in which we first estimate the probability of $\mathbf{Z}$ given $\mathbf{Y}$ and the current estimate θ^{old} of the parameters θ (*E-step*), and then we determine the new estimate of θ by maximizing the expected log-likelihood (*M-step*),

$$\mathcal{Q}(\theta | \theta^{\text{old}}) := \sum_{\mathbf{Z}} \Pr(\mathbf{Z} | \mathbf{Y}, \theta^{\text{old}}) \log(\Pr(\mathbf{Y}, \mathbf{Z} | \theta)).$$

Provided that $y_1, \dots, y_Q$ and $z_1, \dots, z_Q$ are independent realizations of the random variables Y and Z , respectively, we have that $\Pr(\mathbf{Z} | \mathbf{Y}, \theta^{\text{old}}) = \prod_{j=1}^Q \Pr(Z = z_j | Y = y_j, \theta^{\text{old}})$ and $\Pr(\mathbf{Y}, \mathbf{Z} | \theta^{\text{old}}) = \prod_{j=1}^Q \Pr(Y = y_j, Z = z_j | \theta^{\text{old}})$. Therefore, the latter reduces to

$$\mathcal{Q}(\theta | \theta^{\text{old}}) := \sum_{i=1}^N \sum_{j=1}^Q P_{i,j} \log(\Pr(Y = y_j, Z = i | \theta)), \quad \text{where } P_{i,j} = \Pr(Z = i | Y = y_j, \theta^{\text{old}}). \quad (24)$$

We notice that the EM method aims to improve $\mathcal{Q}(\theta|\theta^{\text{old}})$ rather than directly improving $\Pr(\mathbf{Y}|\theta)$ at each iteration. As rigorously shown in [22], exploiting Jensen’s inequality, we can verify that

$$\log(\Pr(\mathbf{Y}|\theta)) - \log(\Pr(\mathbf{Y}|\theta^{\text{old}})) \geq \mathcal{Q}(\theta|\theta^{\text{old}}) - \mathcal{Q}(\theta^{\text{old}}|\theta^{\text{old}}). \quad (25)$$

Therefore, improvements in $\theta \mapsto \mathcal{Q}(\theta|\theta^{\text{old}})$ lead to improvements in $\theta \mapsto \log(\Pr(\mathbf{Y}|\theta))$. Estimate (25) shows that the EM procedure fits in the framework of bound optimizer algorithms, for which extensive analysis can be carried out [50].

Algorithm 2 summarizes the EM procedure. Meng and Rubin [38] proposed to partition θ into groups and then break down the M-step into multiple steps each of which involves the optimization with respect to one subgroup while the remainders are held fixed; furthermore, we can adapt the EM procedure to tackle maximum a posteriori (MAP) estimates by augmenting $\mathcal{Q}(\theta|\theta^{\text{old}})$ with a prior for θ in the M-step. In the remainder, we exploit both extensions of Algorithm 2.

Algorithm 2: General expectation-maximization (EM) procedure.

Data: $\mathbf{Y} \in \mathbb{R}^{d \times Q}$ observations; θ_0 initial estimate of the parameters, probabilistic model of (24)

Result: θ^*

```

1 Initialization: set  $\theta^* \leftarrow \theta_0$ .
2 for  $k = 0, \dots$ , until convergence do
3   E-step: Determine  $P_{i,j} = \Pr(Z = i | Y = y_j, \theta^*)$  for  $i = 1, \dots, N, j = 1, \dots, Q$ ;
4   M-step: Solve  $\theta^* \leftarrow \arg \max_{\theta \in \Theta} \mathcal{Q}(\theta|\theta^{\text{old}})$ 
```

3.5.2 Probabilistic model

The application of the general EM algorithm critically depends on the choice of the probabilistic models for the observations. Following [41], we consider the mixture model for the location of the target points

$$Y = \sum_{i=1}^N \delta(Z, i) Y^{(i)} + \delta(Z, N+1) Y^{(N+1)}, \quad (26)$$

where $\delta(i, j) = 1$ if $i = j$ and 0 otherwise, $Y^{(i)} \sim \mathcal{N}(m_i, \sigma^2 \mathbb{1})$ for $i = 1, \dots, N$, $Y^{(N+1)} \sim \text{Uniform}(\mathcal{D})$, and $Z \sim \text{Multinomial}(\{1, \dots, N+1\})$ are independent random variables. Given $w \in [0, 1]$, we set

$$\Pr(Z = i) = \begin{cases} \frac{1-w}{N} & i = 1, \dots, N, \\ w & i = N+1; \end{cases} \quad |\mathcal{D}| = Q. \quad (27)$$

In order to ensure that the GMM centroids move coherently (cf. [41]) and lie in Ω , we propose the deformation model:

$$m_i = F[v](\xi_i), \quad i = 1, \dots, N; \quad (28)$$

where $\{\xi_i\}_{i=1}^N$ is the reference point cloud, and $F[v]$ is the flow of the velocity $v \in V_h$. Note that the deformation model (28) differs from the one of [41] and is designed to enforce the bijectivity of the mapping in Ω . In the remainder, we assume that $w, \mathcal{D}$ are user-defined constants, while the velocity v and the variance σ^2 are the design parameters, i.e., $\theta = (v, \sigma^2)$.

As discussed in the next section, the probabilistic model introduced above enables the application of (a variant of) Algorithm 2. Below, we comment on our modeling choices.

- Eq. (26) reads as a GMM with a perturbation given by the uniform random variable $Y^{(N+1)}$: the latter is intended to account for noise and outliers in the dataset. **The parameter w is designed to account for numerical noise and/or mesh artifacts that might ultimately lead to overfitting; in the numerical experiments, we set $w = 10^{-6}$ for the two-dimensional test cases and $w = 10^{-8}$ for the three-dimensional test case. Finally, we do not explicitly construct the domain $\mathcal{D}$: we simply assume that all the observed datapoints belong to $\mathcal{D}$.**
- The choice of the deformation model (28) ensures that the GMM centroids lie in $\bar{\Omega}$; however, the mixture model (26) does not satisfy the constraint $\Pr(Y \in \bar{\Omega}|\theta) = 1$. Even if we did not experience any convergence issue for the experiments of section 4, we envision that this shortcoming of our model might lead to convergence issues for the EM iterative procedure. **This formulation implies that the registration procedure could, in certain settings, converge to a non-physical local minimum; for instance, points from the**

suction side of an airfoil might erroneously establish correspondences with points from the pressure side by “jumping” across the interior of the geometry. The development of more sophisticated probabilistic models for point-set registration in bounded domains is the subject of ongoing research.

- In several applications, we might be interested in exploiting several sensors at the same time. In these scenarios, we should deal with multiple point clouds to register at the same time. Below, we discuss how to update the probabilistic model *a posteriori* to cope with multiple point clouds.

3.5.3 EM procedure for point-set registration

Exploiting the previous hypotheses, we can derive an explicit expression for $P_{i,j} = \Pr(Z = i | Y = y_j, \theta)$. If we recall the definition of the conditional probability and the definition of the marginal probability, we find

$$P_{i,j} = \Pr(Z = i | Y = y_j, \theta) = \frac{\Pr(Z = i, Y = y_j | \theta)}{\Pr(Y = y_j | \theta)} = \frac{\Pr(Z = i, Y = y_j | \theta)}{\sum_k \Pr(Z = k, Y = y_j | \theta)}.$$

Next, we notice that, by construction, if $Z = i$, we have $Y = Y^{(i)}$ (cf. (26)); therefore, $\Pr(Z = i, Y = y_j | \theta) = \Pr(Z = i, Y^{(i)} = y_j | \theta)$ and the latter is equal to $\Pr(Z = i) \Pr(Y^{(i)} = y_j | \theta)$ since Z and $Y^{(1)}, \dots, Y^{(N)}$ are independent. Therefore,

$$P_{i,j} = \frac{\Pr(Z = i) \Pr(Y^{(i)} = y_j | \theta)}{\sum_k \Pr(Z = k) \Pr(Y^{(k)} = y_j | \theta)}.$$

Finally, if we recall the probability laws for $\mathbf{Z}$ and $Y^{(i)}$ (cf. (26) and (27)), we obtain

$$P_{i,j} = \frac{\exp\left(-\frac{\|y_j - m_i\|_2^2}{2\sigma^2}\right)}{\sum_{k=1}^N \exp\left(-\frac{\|y_j - m_k\|_2^2}{2\sigma^2}\right) + c}, \quad \text{where } c = \frac{w}{1-w} \frac{N}{Q} (2\pi\sigma^2)^{d/2}, \quad (29)$$

for $j = 1, \dots, N$ and $i = 1, \dots, N, j = 1, \dots, Q$.

If multiple point clouds are used, we should ensure that $\Pr(Z = i, Y = y_j | \theta) = 0$ if the indices i, j correspond to different clouds to ensure that each coherent structure is independently tracked: we should hence modify (29) as follows

$$P_{i,j} = \frac{\exp\left(-\frac{\|y_j - m_i\|_2^2}{2\sigma^2}\right) \bar{P}_{i,j}}{\sum_{k=1}^N \exp\left(-\frac{\|y_j - m_k\|_2^2}{2\sigma^2}\right) \bar{P}_{k,j} + c}, \quad (30)$$

where $\bar{P}_{i,j} = 0$ if i, j belong to different clouds and 1 otherwise.

Similarly, exploiting the same argument as in (29), we find

$$\log(\Pr(Y = y_j, Z = i | \theta)) = \begin{cases} -\frac{d}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \|y_j - m_i\|_2^2 + \log\left(\frac{1-w}{N(2\pi)^{d/2}}\right) & i = 1, \dots, N \\ \log(w) - 2 \log Q & i = N+1. \end{cases}$$

and then

$$-\mathcal{Q}(\theta | \theta^{\text{old}}) := -\sum_{i=1}^N \sum_{j=1}^Q P_{i,j} \log(\Pr(Y = y_j, Z = i | \theta)) = \left(\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^Q P_{i,j} \|y_j - m_i\|_2^2 \right) + \frac{dQ_p}{2} \log(\sigma^2) + C,$$

where $Q_p = \sum_{i=1}^N \sum_{j=1}^Q P_{i,j}$, and C is a constant that is independent of θ and can thus be omitted. If we express the GMM centroids $\{m_i\}_{i=1}^N$ using the deformed corresponding elements of the reference point cloud through (28), we finally obtain

$$-\mathcal{Q}(v, \sigma^2 | \theta^{\text{old}}) = \left(\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{j=1}^Q P_{i,j}(\theta^{\text{old}}) \|y_j - F[v](\xi_i)\|_2^2 \right) + \frac{dQ_p}{2} \log(\sigma^2) + C, \quad (31)$$

Following [38] and [41], we propose to split the M-step in two parts: first, we compute the velocity field v^* based on the previous estimate of σ^2 ; second, we compute the variance σ^2 . Furthermore, we augment the log-likelihood with an additional term that ensures the smoothness of the velocity field and the (approximate) satisfaction of the constraint $v \cdot \mathbf{n}|_{\partial\Omega} = 0$. In more detail, we consider the optimization problem to find the velocity v^* :

$$\max_{v \in V_h} \mathcal{Q}(v, \sigma^{2,\text{old}} | \sigma^{\text{old}}) - \frac{\lambda}{\sigma^{2,\text{old}}} \mathbf{f}_{\text{pen}}(v) \quad (32)$$

where $\lambda > 0$ is a given constant and f_{pen} is the penalty function of section 3.4. If we omit the terms that do not depend on v , we obtain the minimization statement

$$\min_{v \in V_h} f^{\text{obj}}(v) := \frac{1}{2} \sum_{i=1}^Q \sum_{j=1}^N P_{i,j}(\theta^{\text{old}}) \|y_j - F[v](\xi_i)\|_2^2 + \frac{\lambda}{2} f_{\text{pen}}(v). \quad (33)$$

Note that the maximization step (33) corresponds to the variational problem introduced in (2). Given the new estimate v^* , we then set $\sigma^{2,*}$ as

$$\sigma^{2,*} = \arg \min_{\sigma^2 \in \mathbb{R}_+} -Q(v^*, \sigma^2 | \sigma^{\text{old}}) \Rightarrow \sigma^{2,*} = \frac{\sum_{i=1}^N \sum_{j=1}^Q P_{i,j} \|y_j - F[v^*](\xi_i)\|_2^2}{d \sum_{i=1}^N \sum_{j=1}^Q P_{i,j}}. \quad (34)$$

Algorithm 3 summarizes the overall procedure. In this work, we consider the termination condition

$$R_1 := \frac{1}{N} f^{\text{tg}}(v^*) < \epsilon_1, \quad R_2 := \frac{\|\mathbf{K}^{-1} \nabla f^{\text{obj}}(v^*)\|_2}{\|\mathbf{K}^{-1} \nabla f^{\text{obj}}(v^0)\|_2} < \epsilon_2, \quad R_3 := |\sigma^{2,*} - \sigma^{2,\text{old}}| < \epsilon_3, \quad (35)$$

where $\epsilon_1, \epsilon_2, \epsilon_3$ are user-defined parameters. Note that R_1 measures the misfit error, R_2 is a weighted norm of the objective gradient, finally R_3 measures the increment of the estimate of the variance σ^2 .

Algorithm 3: Expectation-maximization (EM) procedure for registration in bounded domains.

Data: $\mathbf{Y} \in \mathbb{R}^{d \times Q}$ observations; v_0, σ_0^2 initial estimate of the parameters, w (cf. (27)), λ (cf. (34)),

$\epsilon_1, \epsilon_2, \epsilon_3$ (cf. (35)).

Result: v^*

- 1 *Initialization:* set $v^* \leftarrow v_0, \sigma^{2,*} \leftarrow \sigma_0^2$.
 - 2 **for** $k = 0, \dots$, until convergence **do**
 - 3 **E-step:** Determine $P_{i,j}$ using (29) (or (30)) for $i = 1, \dots, N, j = 1, \dots, Q$;
 - 4 **M-step (I):** Estimate the solution v^* to (33);
 - 5 **M-step (II):** Set $\sigma^{2,*}$ using (34);
 - 6 Check the termination condition (35).
-

The initialization of the velocity v_0 and the initial variance σ_0^2 in Algorithm 2 is critical for the performance of the method. In all our experiments, we consider $v_0 = 0$; even if the optimization problem is non-convex, we did not experience any convergence issue. As regards the initial variance, we notice that the choice of a large (if compared to the distance between the point clouds) initial variance σ_0^2 facilitates global alignment. In our experiments, we set $\sigma_0^2 = 1$. Finally, the weighting parameter λ acts as a numerical relaxation coefficient for stability, with minimal observed impact on final alignment quality: in our experiments, we set $\lambda = 10^{-4}$.

Remark 3.1. As in [38], we augmented the log-likelihood with a regularization term that encodes prior information on the parameters. However, as opposed to [38], the regularization $\lambda f_{\text{pen}}(v)$ in (32) is divided by $\sigma^{2,\text{old}}$: this choice is justified by the fact that the penalty became less effective as $\sigma^{2,\text{old}} \rightarrow 0$. In conclusion, as opposed to [38], we cannot rigorously interpret the regularization $f_{\text{pen}}(v)$ as a prior for the parameters.

3.5.4 Solution to the optimization problem

The estimate of v^* (cf. M-step (I), Algorithm 3) requires the solution to a highly nonlinear non-convex minimization problem. Towards this end, we resort to a left-preconditioned gradient-descent technique:

$$\mathbf{v}^{*,(q+1)} = \mathbf{v}^{*,(q)} - \eta^{(q)} \mathbf{K}^{-1} \nabla f^{\text{obj}}(\mathbf{v}^{*,(q)}), \quad q = 1, 2, \dots, \quad (36)$$

where $\mathbf{K}$ is the symmetric positive definite matrix introduced in (20) and $\eta^{(q)}$ is determined using line search. The choice of $\mathbf{K}^{-1}$ as left preconditioner is consistent with the variational interpretation of the velocity field (cf. [59]): we refer to [26, section 3.1.1] for a thorough discussion on the subject.

In the numerical experiments, we considered two distinct line search methods: backtracking line search and critical point line search (see, e.g., [23, Appendix A]) as implemented in the PETSC functions `SNESLINESEARCHBT` and `SNESLINESEARCHCP`. The two methods exhibit comparable performance in the numerical experiments of section 4.

Since the the objective of (33) is updated at each outer loop iteration of the EM procedure, we empirically found that an inexact minimization based on limited updates (e.g., a few gradient steps) is preferable in practice. We remark that our empirical finding is consistent with the extensive literature on EM-like procedures [50].

4 Numerical results

4.1 Transonic inviscid flow past a NACA0012 airfoil

4.1.1 Model problem

We consider the transonic inviscid flow past a NACA0012 airfoil for varying inflow Mach number $\mu = M_\infty \in [0.8, 0.85]$ and angle of attack $\text{AoA} = 0.4^\circ$. We introduce the vector of conserved variables $U = (\rho, \rho v_1, \rho v_2, \rho E)$ where ρ is the flow density, v is the flow velocity, E is the total specific energy; we also introduce the pressure p that is related to the state variables through the equation of state $p = (\gamma - 1) (\rho E - \frac{1}{2} \rho \|v\|_2^2)$, where $\gamma = 7/5$ is the ratio of specific heats. The computational domain $\Omega = \Omega_{\text{out}} \setminus \Omega_{\text{naca}}$ is depicted in Figure 1(a): the airfoil Ω_{naca} has unit chord and its leading edge is located at the origin $(0, 0)$; Ω_{out} is the union of a half circle of radius 100 centered at the trailing edge $(1, 0)$ and the rectangle $(1, 100) \times (-100, 100)$. CFD calculations are performed using the DG solver Aghora [49] on a structured mesh with $N_e = 14800$ quadrilateral elements (cf. Figure 1(b)). Figures 1(c) and 1(d) show the behavior of the density profile in the proximity of the airfoil for $M_\infty = 0.8$ and $M_\infty = 0.85$, respectively: we notice that the solution exhibits two normal shocks — one on the upper (suction) side and one on the lower (pressure) side of the airfoil — whose shape and location strongly depend on the value of the parameter.

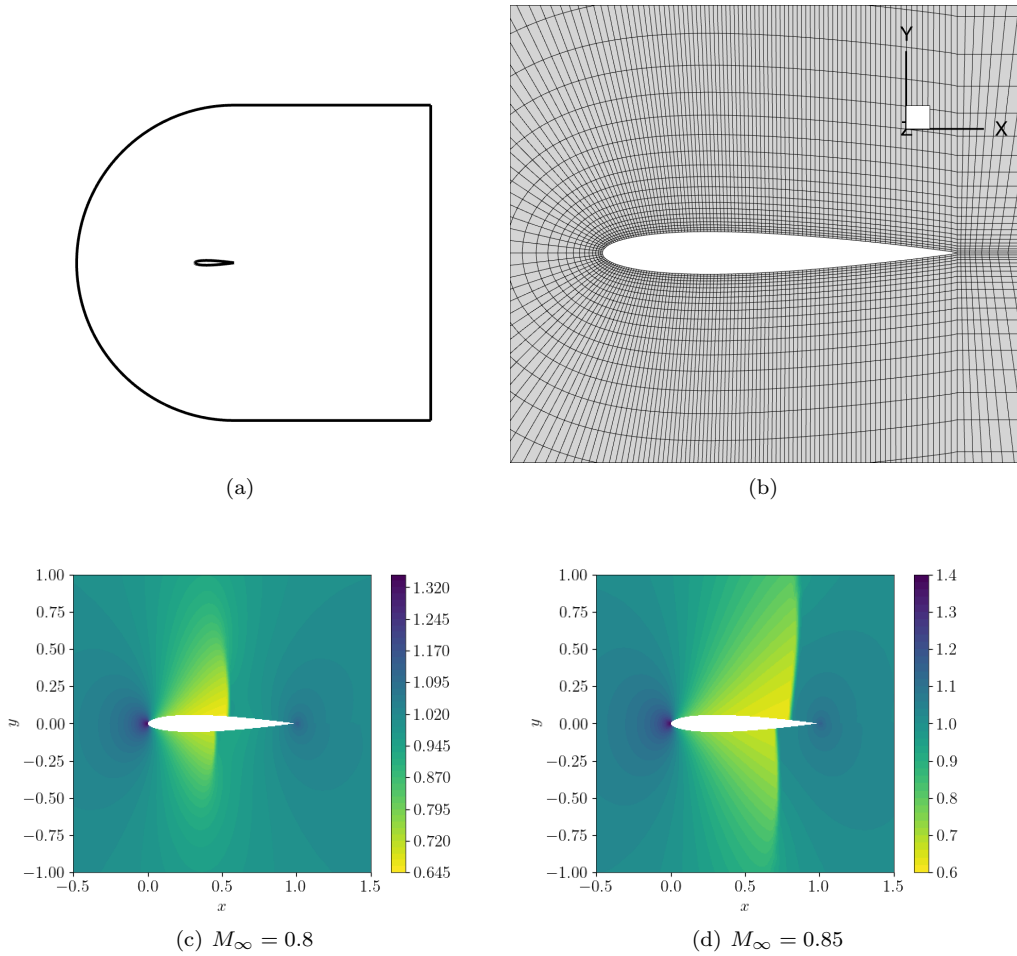


Figure 1: transonic flow past a NACA airfoil. (a) geometric configuration. (b) computational mesh. (c)-(d) density profiles for two values of the free-stream Mach number and angle of attack 0.4° .

4.1.2 Setup

We apply the nonlinear interpolation strategy of section 2 in combination with the registration method of section 3 to the two snapshots $U_0 = U(\mu_0 = 0.8)$ and $U_1 = U(\mu_1 = 0.85)$. Towards this end, we apply Algorithm 3 with initial velocity $v_0 = 0$, we consider the cut-off frequency $\kappa_0 = 4.5$, initial variance $\sigma_0^2 = 1$, and the hyper-parameters $w = 10^{-6}$, $\lambda = 10^{-4}$; we consider the termination condition $R_3 < 10^{-6}$ (that is, we set $\epsilon_1 = \epsilon_2 = 0$ and $\epsilon_3 = 0$ in (35)), and we set $s = 2$ in the definition of the penalty matrix $\mathbf{K} = (\mathbf{B}\mathbf{M}^{-1})^{s-1} \mathbf{B}$ in (20). To reduce the geometric error induced by registration, we imposed strongly the boundary condition $v = 0$ at the

trailing edge of the foil, where the normal to the boundary is discontinuous. At each M-step, we perform two iterations of the gradient descent technique (36) and we resort to a direct solver based on Cholesky factorization to solve the $s = 2$ linear systems of type $\mathbf{B}\mathbf{x} = \mathbf{b}$ — since $\mathbf{B}$ is fixed, Cholesky factorization can be applied once for all at the beginning of the procedure. In view of the assessment, we consider several structured FE grids of increasing size and polynomial degree $p = 1$ or $p = 2$; furthermore, we resort to the explicit second-order Runge Kutta method (18) for several choices of Δt to integrate the ODE (3).

Recalling the definitions of section 3.2, we define the sampled point cloud as

$$Y_i = \{x \in \Omega_{\text{discr}} : \zeta_s(x, U_i^{\text{hf}}) > 0.7\}, \quad i = 0, 1.$$

Figure 2 shows the point clouds for the two snapshots. In the experiments, we consider Y_0^s as reference point cloud and Y_1^s as target point cloud. We also investigated the performance of the alternative sensor:

$$Y_i^{\text{iso}} = \{x \in \Omega_{\text{discr}} : \zeta_{\text{iso}}(x, U_i^{\text{hf}}) > 0.22 \text{ or } \zeta_{\text{iso}}(x, U_i^{\text{hf}}) < -0.1\}, \quad i = 0, 1.$$

where the function ζ_{iso} is defined in (16). In our experience, we found nearly-equivalent results to the ones obtained with the previous sensor: for this reason, we do not further discuss this alternative.

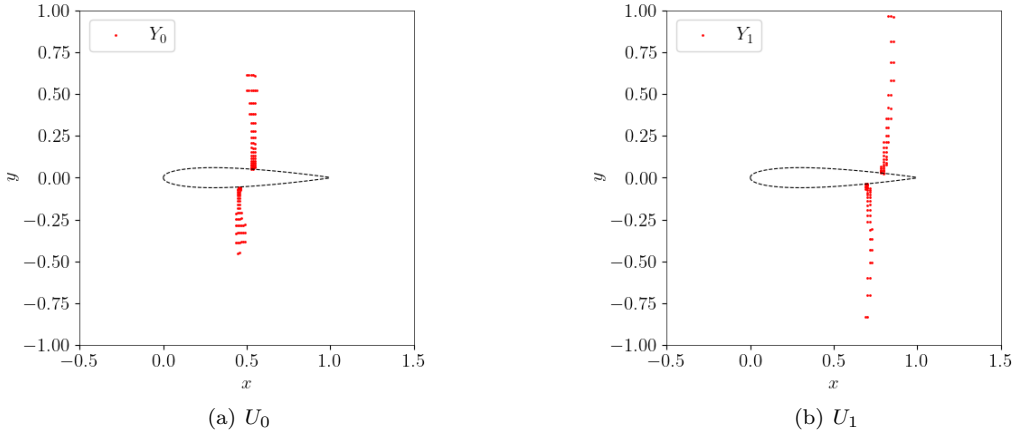


Figure 2: transonic flow past a NACA airfoil; selected points for the two snapshots, $U_i = U(\mu_i)$, $i = 0, 1$, with the sensor (15a) ($\text{tol} = 0.7$).

4.1.3 Results (I): registration problem

Figure 3 shows the evolution of the residuals R_1, R_2, R_3 defined in (35) with respect to the EM outer loop iteration count, for different FE meshes and two choices of the time step Δt (polynomial degree $p = 2$). As we increase the size of the spatial mesh, we notice an improvement in the alignment performance; on the other hand, we do not observe any significant change when we reduce the time step from $\Delta t = 0.2$ to $\Delta t = 0.05$.

In order to assess the geometric error, we denote by $\Gamma_{\text{fe}} := \{\xi_i^\Gamma\}_{i=1}^{N_\Gamma} \subset \partial\Omega_{\text{naca}}$ the nodes of the FE mesh used for registration on the airfoil and we define the geometric error as

$$d(X(\Gamma_{\text{fe}}, 1), \partial\Omega_{\text{naca}}) = \max_{i \in \{1, \dots, N_\Gamma\}} \min_{x \in \partial\Omega_{\text{naca}}} \|X(\xi_i^\Gamma, 1) - x\|_2. \quad (37)$$

In order to compute $\min_{x \in \partial\Omega_{\text{naca}}} \|X(\xi_i^\Gamma, 1) - x\|_2$ for a given index $i \in \{1, \dots, N_\Gamma\}$, we solve an optimization problem that exploits the analytic expression of the boundary of the airfoil. Figure 4 shows the behavior of (37) for several grid sizes and several time step sizes; for reference, we also report the geometric error of the original grid. We observe that for $N_e \gtrsim 10^4$ the geometric error is of the same order as for the original grid. Interestingly, the time step size and the polynomial order have moderate impact on the accuracy: the latter is expected as we do not rely on curved meshes.

4.1.4 Results (II): flow prediction

We apply the CDI method (11c) to predict the flow state for $M_\infty \in [0.8, 0.85]$. Figure 5 shows the truth pressure field as predicted by the high-fidelity (HF) DG solver; Figure 6 shows the CDI prediction based on the registration method outlined above; finally, Figure 7 shows the performance of a simple convex interpolation.

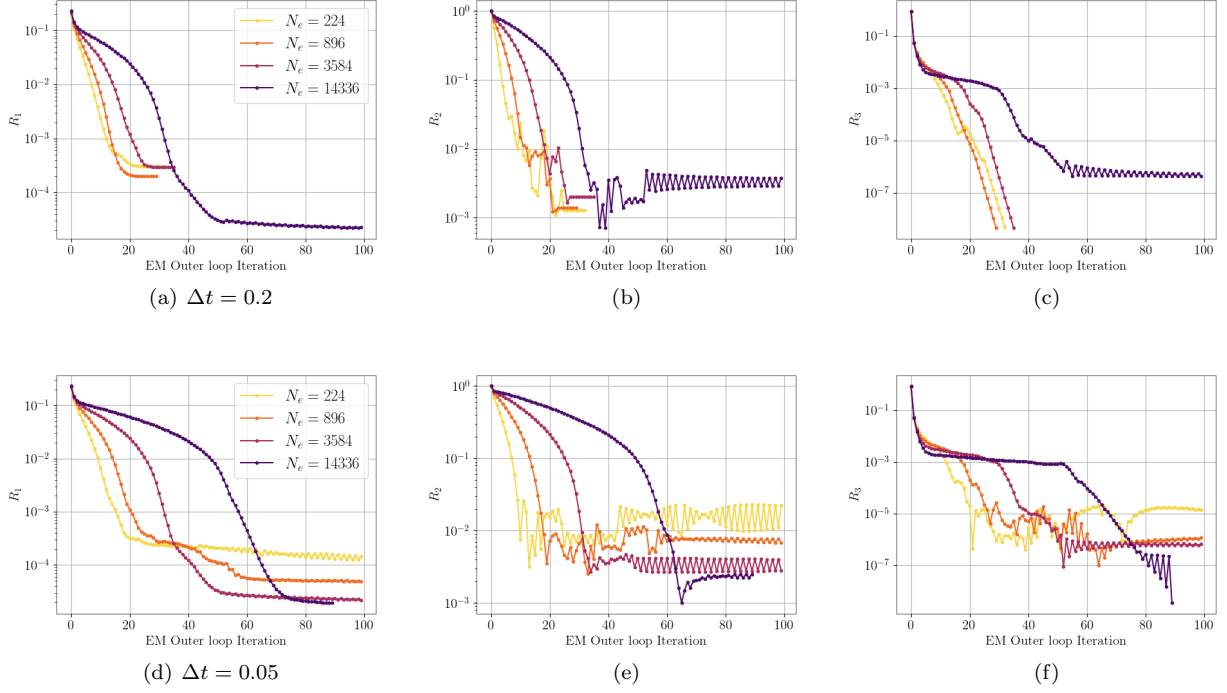


Figure 3: transonic flow past a NACA airfoil; evolution of the residuals R_1, R_2, R_3 in (35) with respect to the EM iteration count, for different FE meshes with N_e . (a)-(b)-(c) $\Delta t = 0.2$, linear elements ($p = 1$). (d)-(e)-(f) $\Delta t = 0.05$, quadratic elements ($p = 2$).

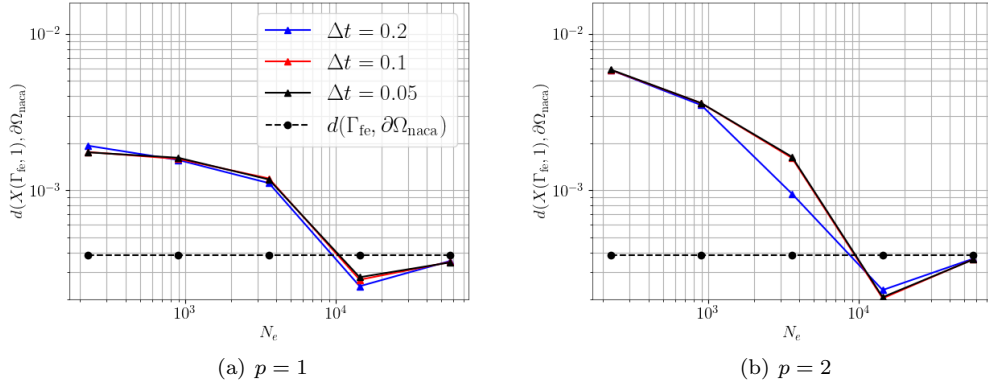


Figure 4: transonic flow past a NACA airfoil; evolution of the geometric error (37) with respect the FE mesh size for linear ($p = 1$) and quadratic ($p = 2$) polynomials.

We notice that CDI outperforms convex interpolation, for the same amount of training data (that is, two snapshots).

In order to offer a more quantitative comparison between the different methods, we report in Figure 8 the behavior of the pressure coefficient

$$C_p := \frac{p - p_\infty}{\frac{1}{2}\rho_\infty \|v_\infty\|_2^2}$$

on both the suction and the pressure sides of the airfoil for both implementations of the CDI, and we compare with convex interpolation. Furthermore, Figure 9 shows the relative error in the prediction of the density field

$$E_\rho := \frac{\|\rho - \hat{\rho}\|_{L^2(\Omega)}}{\|\rho - \rho_\infty\|_{L^2(\Omega)}},$$

where $\hat{\rho}$ denotes the CDI (or CI) estimate. We consider two different discretizations of the registration task: the results of Figure 9(a) correspond to a linear ($p = 1$) FE discretization and $\Delta t = 0.1$, while the results of

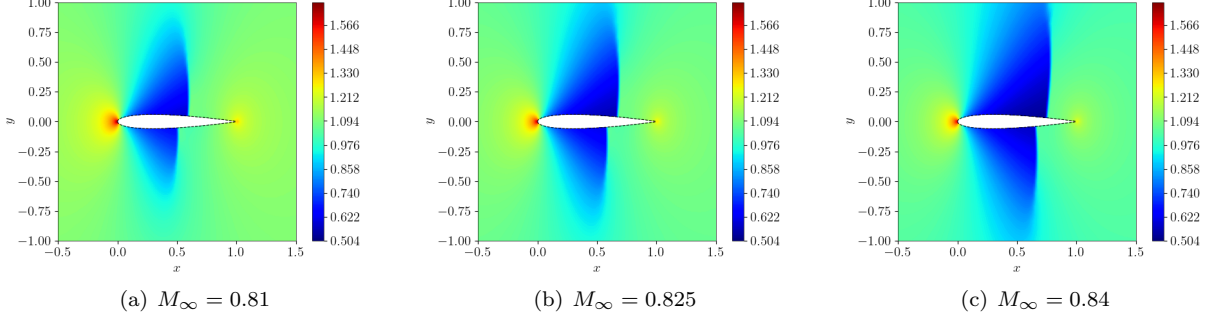


Figure 5: transonic flow past a NACA airfoil; HF estimates of the pressure field for three parameter values.

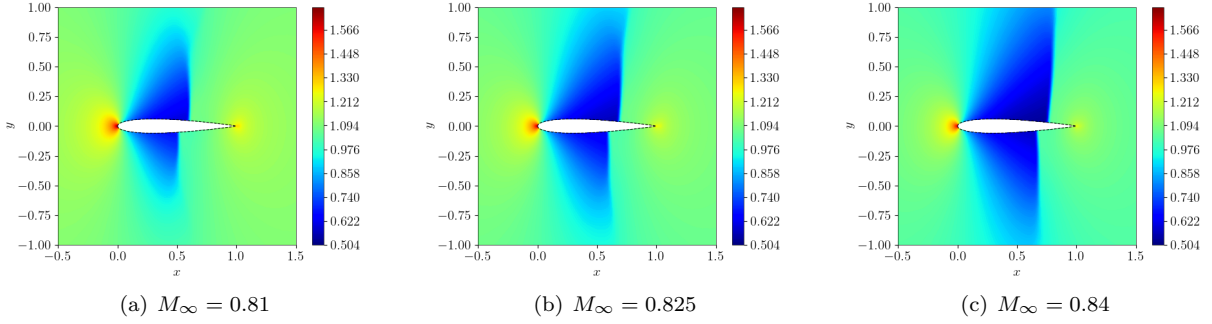


Figure 6: transonic flow past a NACA airfoil; CDI estimates of the pressure field for three parameter values with **shock-based** registration ($\Delta t = 0.1$ and $p = 1$).

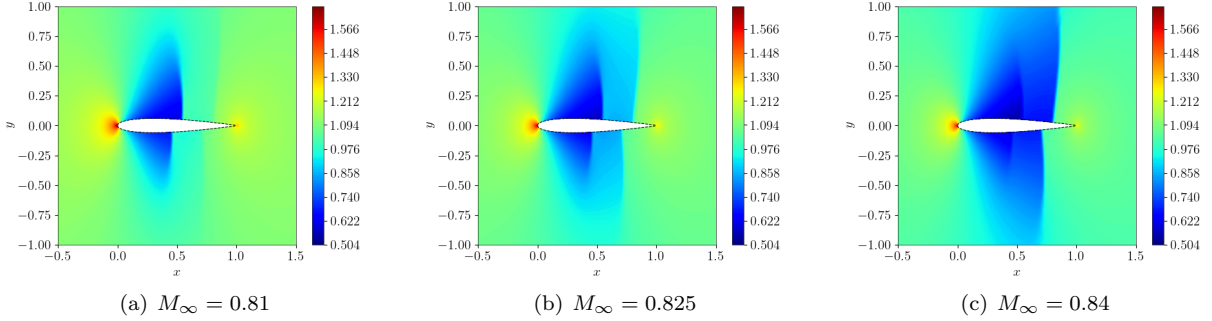


Figure 7: transonic flow past a NACA airfoil; CI estimates of the pressure field for three parameter values.

Figure 9(b) correspond to a quadratic ($p = 2$) FE discretization and $\Delta t = 0.05$. We observe that CDI clearly outperforms convex interpolation for this test case; we further notice that the polynomial degree of the FE discretization, as well as the time step Δt have a negligible impact on the performance.

4.1.5 Results (III): data augmentation

We exploit the CDI model to generate artificial snapshots in the range $\mu \in [0.8, 0.85]$. These snapshots are used to build a reduced basis that can be later used for model-based (projection-based, collocated) ROMs. In more detail, we generate $N_{\text{snap}} = 500$ artificial pressure field snapshot with the CDI model using different choices of the interpolation parameter s in (11c). We apply proper orthogonal decomposition to the snapshot set of N_{snap} synthetic snapshots to build a reduced basis $\{\varphi_1, \dots, \varphi_M\}$. We denote by $\mathcal{W}_m = \text{Span}(\varphi_1, \dots, \varphi_m)$ the truncated POD subspace for the pressure field of dimension m .

Figure 10(a) shows the behavior of the relative projection error

$$E_m^{\text{proj}} = \frac{\|p_\mu - \Pi_{\mathcal{W}_m} p_\mu\|_{L^2}}{\|p_\mu - p_\infty\|_{L^2}}, \quad (38)$$

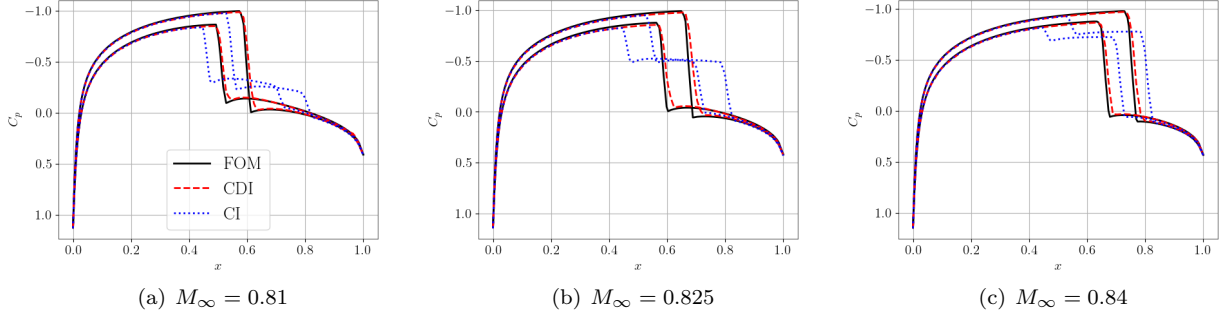


Figure 8: transonic flow past a NACA airfoil; prediction of the pressure coefficients C_p for several parameter values ($N_e = 14336$).

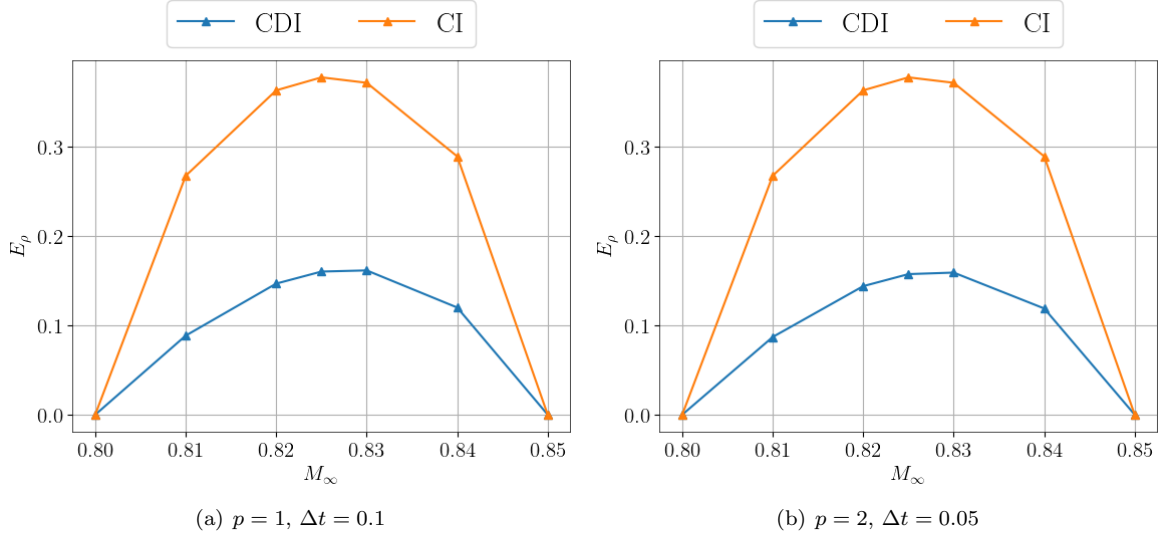


Figure 9: transonic flow past a NACA airfoil; relative prediction L^2 error for the density field ρ , for two different discretizations. p ($N_e = 14336$).

with respect to the size of the POD space, for the test parameter $\mu_{\text{test}} = 0.825$. Finally, Figure 10(b) shows the projected pressure field for $m = 26$. We observe that data augmentation enhances approximation properties of the reduced space: the projection error drops from $E^{\text{proj}} = 3 \cdot 10^{-1}$ for $\text{span}\{p_0, p_1\}$ to approximately $E^{\text{proj}} = 7 \cdot 10^{-2}$ for $m \gtrsim 20$.

4.2 Transonic inviscid flow past a NACA2314 airfoil

We extend our method to the inviscid flow past a asymmetric (NACA2314) airfoil. As in the previous test case, we study the flow behavior for varying inflow Mach number $M_\infty \in [0.8, 0.85]$ and angle of attack $\text{AoA} = 0.1^\circ$. We consider the shock sensor (15a) with the same hyper-parameters as in section 4.1: Figure 11 shows the selected points. Numerical simulations are performed with the same CFD code Aghora and we run the registration method in Algorithm 3 with the same hyper-parameters.

Figure 12 compares the C_p prediction results of the FOM, the CDI and the CI models for the NACA2314 test case. Furthermore, Figure 13 shows the behavior of the intrados and the extrados shock locations with respect to the inflow Mach number: CDI accurately predicts the intrados shock location, while the error on the extrados is slightly larger; this can be explained by the fact that the intrados shock location varies much more ($\approx 30\%$ vs $\approx 10\%$ of the chord length) and the dependence on the Mach number is nonlinear.

4.3 Viscous flow past an ONERA M6 wing

4.3.1 Model problem

We consider the transonic viscous turbulent flow past an Onera M6 wing [52] for varying angle of attack $\mu = \text{AoA} \in [3.06^\circ, 5.1^\circ]$. We set $M_\infty = 0.84$ and we consider the Reynolds number $\text{Re} = 14.6 \cdot 10^6$. We rely on

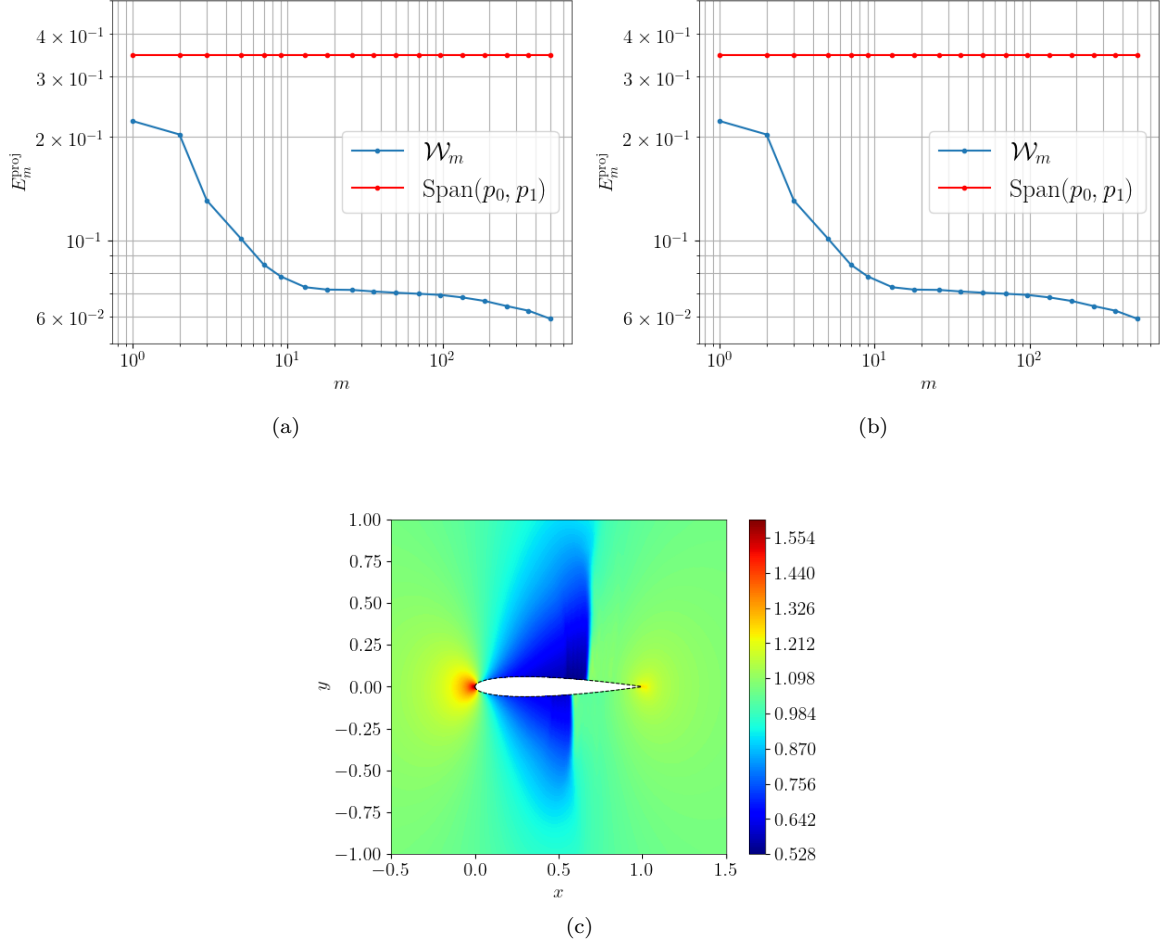


Figure 10: (a) L^2 projection error of FOM snapshot p_μ with $\mu = 0.825$ onto subspace $\mathcal{W}_m$, (b) Scalar field $\Pi_{\mathcal{W}_m} p_\mu$ with $m = 10$.

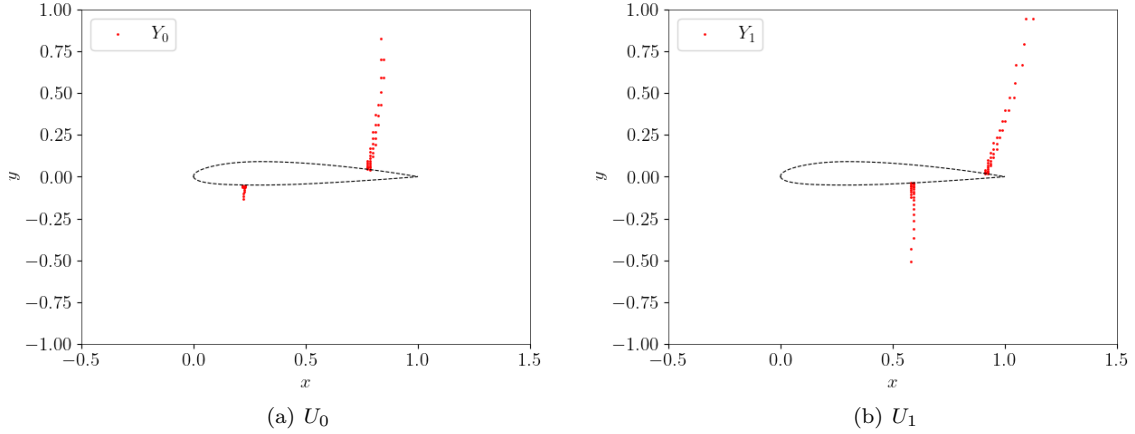


Figure 11: transonic flow past a NACA2314 airfoil; selected points for the two snapshots, $U_i = U(\mu_i)$, $i = 0, 1$, with the sensor (15a) ($\text{tol} = 0.7$).

the compressible Reynolds-averaged Navier Stokes (RANS) equations with Spalart-Allmaras turbulence model [4]; the vector of state variables U is given by $U = \text{vec}(\rho, \rho v_1, \rho v_2, \rho v_3, \rho E, \rho \nu_T)$ where ρ, v, E are the same as in the previous test case, while ν_T is the turbulent viscosity. CFD calculations are performed using the DG solver CODA [54]. We consider two meshes with $N_e = 1,351,648$ and $N_e = 10,813,180$ elements: we refer to the two meshes as “coarse” and “fine”, respectively. In both cases, prisms elements are employed in the boundary layer region near the wing, while tetrahedral elements are employed in the far field, which is set at 12 wing

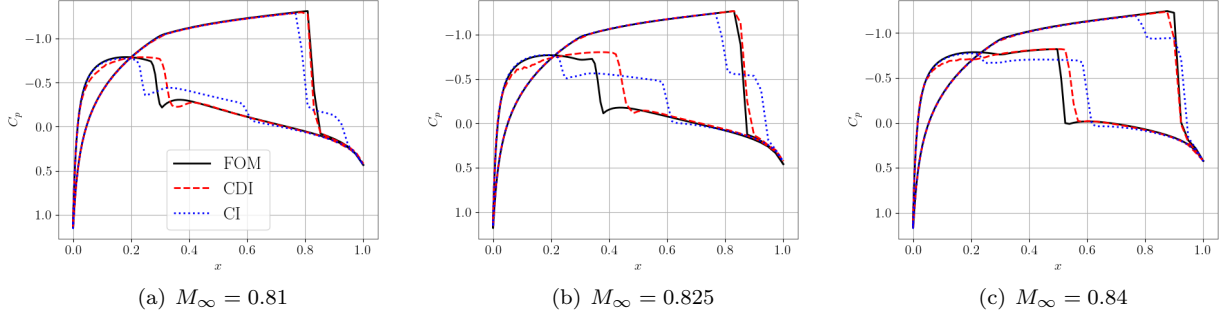


Figure 12: transonic flow past a asymmetric NACA2314 airfoil; prediction of the pressure coefficients C_p for several parameter values ($N_e = 14336$).

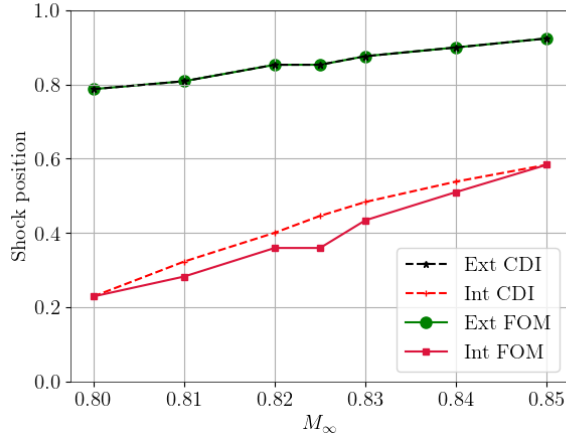


Figure 13: Displacement of shock position with farfield Mach number

root chords. Table 1 provides the estimates of the lift coefficient for two values of the angle of attack and the two meshes: we observe that the lift coefficients are converged with less than 1% error between the two CFD meshes.

We resort to a $p = 1$ discretization for the computation of the HF snapshots; we rely on the ROE up-winding method with positivity preserving limiter to handle the convection term. We consider a pseudo-transient continuation method based on the implicit backward Euler time-stepping scheme to solve the nonlinear system of equations; we initialize the CFL condition to one and then we consider a [switched evolution relaxation \(SER\)](#) ramp with exponent of 0.4 to increase the time step adaptively [17]. At each pseudo-time iteration, we exploit the Newton method to solve the nonlinear system of equations: we consider a matrix-free GMRES approach and we resort to a finite difference method to approximate the Jacobian-vector products without explicitly assemble of the Jacobian matrix. Furthermore, we use a restarted GMRES method with a Jacobi/ILU preconditioner stack to ensure efficient convergence. Finally, for $p = 0$, we initialize the DG solver using the free-stream solution; for $p = 1$, we use the $p = 0$ solution. We set the convergence level to 10^{-10} for $p = 0$ computation, and to 10^{-8} for $p = 1$ computation.

	Coarse	Fine
$AoA = 3.06^\circ$	0.26743	0.26976
$AoA = 4.08^\circ$	0.35867	0.36003

Table 1: Lift coefficient C_l obtained for the two CFD meshes.

Figure 14(a) shows the computational mesh; Figures 14(b) and (c) show the density profiles for two parameter values. As extensively discussed in the literature (see, e.g., [31]), this test features a λ -configuration (λ -shock) of isobars on the upper surface of the wing; the location and the shape of the supersonic regions is highly sensitive to the inflow conditions (Mach number and angle of attack).

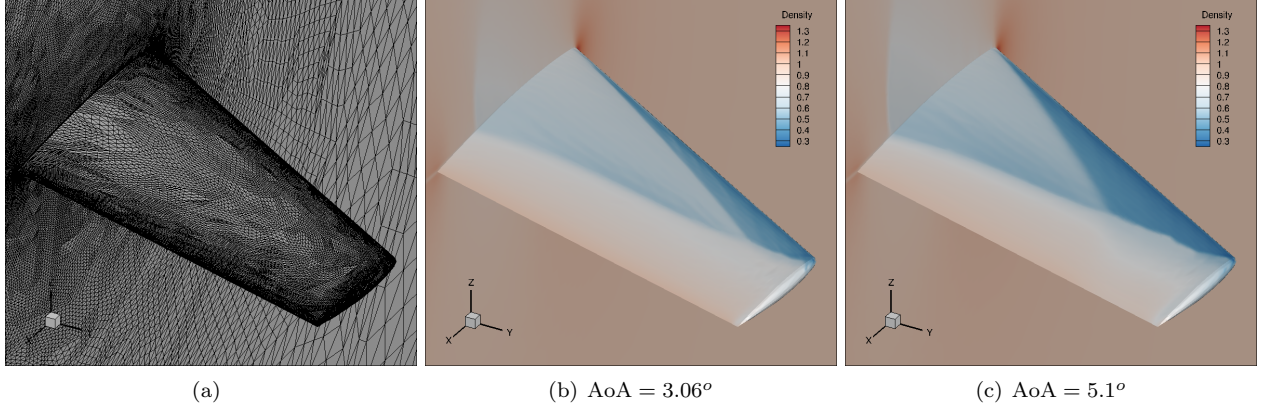


Figure 14: Viscous flow past an ONERA M6 wing. (a) computational mesh (“fine”). (b)-(c) density profiles for two values of the angle of attack.

4.3.2 Setup

Unless differently specified, the results hereafter are obtained on the fine mesh. We extract the source and the target point clouds from the CFD mesh. In more detail, We compute the shock sensor presented in (15a) on the CFD mesh nodes as follows:

$$Y_i^s = \{x \in \Omega_{\text{discr}} : \zeta_s(x, U_i^{\text{hf}}) > 0.6\}, \quad i = 0, 1; \quad (39)$$

For the selected snapshots, we retrieve two point sets of size $N_0 = 4143$ and $N_1 = 3673$ to represent the geometry of the lambda shock. To visualize the point clouds in Figure 15, we show vertical cuts of the indicator function $\chi_i(x) := \mathbb{1}_{Y_i^s}(x)$ for three different choices of x_3 .

As for the previous test case, we perform two steps of the gradient descent technique (33) at each outer loop iteration of Algorithm 3. We use a critical point line search method with an initial search interval of $[0, 10^{-3}]$, and maximum step $\eta_{\max} = 10^3$ to ensure smooth convergence of the EM procedure. We resort to the conjugate gradient method with ILU preconditioner to solve (36). We initialize Algorithm 3 with the velocity $v_0 = 0$; we set the initial covariance of the GMM to $\sigma_0^2 = 1$; we use $w = 10^{-8}$ for the noise of the probability model; furthermore, we consider the regularization parameter $\lambda = 10^{-4}$ in (33) and we set $s = 3$ in (20); we consider the cut-off frequency $\kappa_0 = 1$.

4.3.3 Results (I): registration problem

Figures 15 illustrate the effectiveness of the registration procedure to properly deform the point clouds. Each row corresponds to a different vertical cut ($x_3 = 0.3$, $x_3 = 0.7$ and $x_3 = 1.7$) of the indicator function $\chi_i(x) := \mathbb{1}_{Y_i^s}(x)$ associated with (39); the left column shows the indicator functions of the reference and the target point clouds on each cut; the middle and the right columns compare the indicator functions associated with the deformed point clouds for two choices of the temporal and spatial discretization ($p = 1$, $\Delta t = 0.1$ and $p = 2$, $\Delta t = 0.05$). We observe that our procedure is effective to deform the point clouds and is insensitive to the choice of the discretization.

Figure 16 shows the evolution of the residuals R_1, R_2, R_3 in (35) with respect to the EM iteration count, for two different discretizations. We remark that we plot the residuals R_1 and R_2 against the inner loop iterations, which include the iterations of the M-step. We notice a modest improvement in the alignment performance (cf. R_1) as we increase the polynomial order. Interestingly, we note that the line search is more stable with the finer discretization: this can be explained by observing that the effectiveness of the line search procedure is linked to the accuracy of the gradient estimate, which increases as we reduce the time step.

Figure 17 shows the predicted and the ground truth shock positions for the intermediate parameter $AoA = 4.08^\circ$. We observe that both forward (computed by integrating v) and backward (computed using $-v$) mappings displace the source and target points to the same location: this is crucial to ensure consistent estimations, as we expect that the shock topology is the same across the parameter range. Furthermore, we find that both displaced point clouds properly match the activation surface of the Ducros shock sensor for the intermediate (out-of-sample) parameter value.

4.3.4 Results (II): flow prediction and data augmentation

We exploit the solution to the registration algorithm to perform the nonlinear interpolation (11a) of the flow state. We here consider the results obtained for the coarser discretization ($p = 1$ and $\Delta t = 0.2$). Figure 18

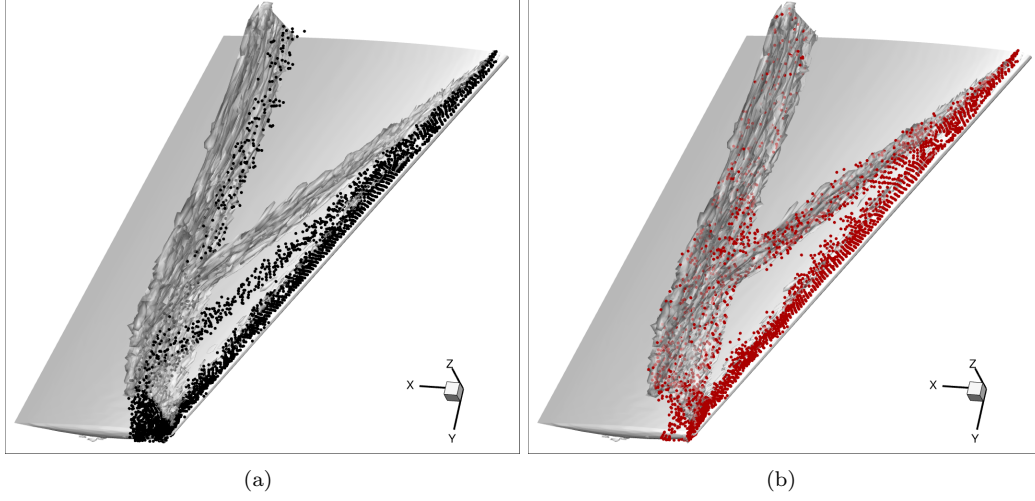


Figure 15: Registration alignment results on the fine mesh. Comparison with surface of activation of shock sensor. (a) Source point in black. (b) Mapped source points after registration in red.

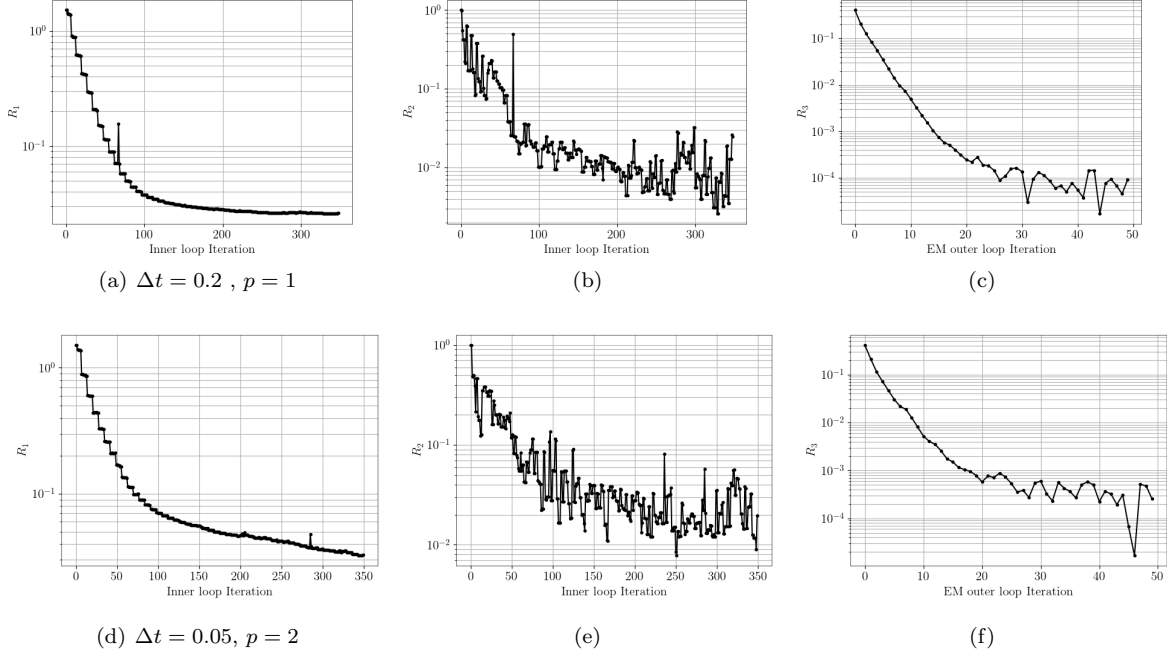


Figure 16: viscous flow past an ONERA M6 wing; evolution of the residuals R_1, R_2, R_3 in (35) with respect to the EM iteration count, for two different discretizations. (a)-(b)-(c) $\Delta t = 0.2$, linear elements ($p = 1$). (d)-(e)-(f) $\Delta t = 0.05$, quadratic elements ($p = 2$).

shows the CDI predictions of the density field, while Figure 19 shows the corresponding CI predictions: we observe that CDI coherently transports the shock over the wing, while CI only mixes the two snapshots. This observation is confirmed by the results of Figure 20, which depicts a volume cut at $x_2 = 8.8$: the CDI estimate features two distinct peaks of the density profile that are consistent with the FOM solution.

We investigate the effectiveness of CDI and CI to initialize the DG solver for the RANS equations for three choices of the parameter value. Towards this end, we compare the convergence history in Figure 21 of the density residual for three out-of-sample parameters associated with three different initialization of the DG solver: (i) CDI, (ii) $p = 0$ DG solution, (iii) CI. **To reduce the computational burden, in this test, we employ the coarse CFD mesh.** We observe that CDI enables a $2 - 3\times$ reduction over CI of the total number of iterations that are required to meet the required tolerance. We justify this fact by observing that CDI is effective to convect the fronts where the solution is discontinuous: since Newton method is significantly more effective at adjusting amplitudes than at moving discontinuities, the ability of CDI to adequately predict the shock location greatly reduces the required number of iterations to reach convergence and contributes to robustify the nonlinear solver.

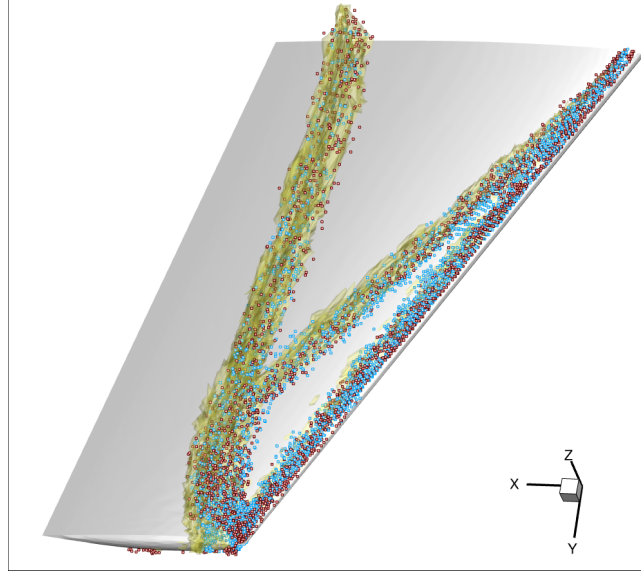


Figure 17: Prediction of the shock position for parameter $AoA = 4.08^\circ$. The yellow surface depicts the 0.7 isoline of the Ducros shock sensor relative to the FOM at $AoA = 4.08^\circ$. Red points identify the displaced target ($AoA = 5.10^\circ$) points that are mapped backward at $t = 0.5$. Blue points identify the displaced reference ($AoA = 3.06^\circ$) points that are mapped forward at $t = 0.5$.

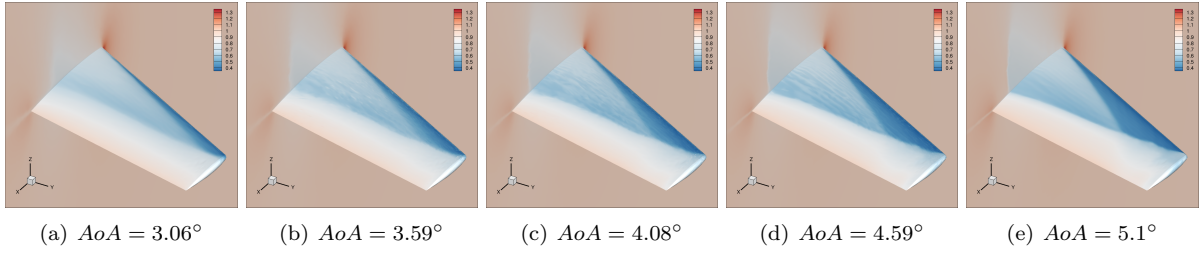


Figure 18: viscous flow past an ONERA M6 wing; CDI density prediction $\tilde{\rho}_{CDI}(\cdot, \mu)$ over range $AoA \in [3.06, 5.10]$.

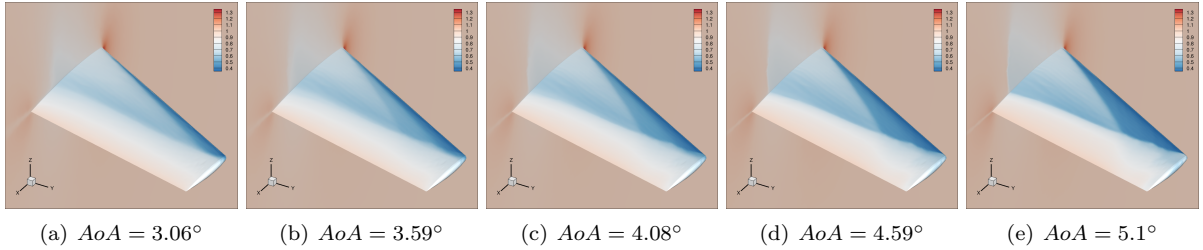


Figure 19: viscous flow past an ONERA M6 wing; CI density prediction $\tilde{\rho}_{CI}(\cdot, \mu)$ on range $AoA \in [3.06, 5.10]$.

Figure 22 shows the performance of CDI to generate meaningful artificial snapshots. We apply CDI to the density profile to generate 500 artificial snapshots; next, we define the $(m - 2)$ -dimensional POD space $\mathcal{Z}_{\text{lf}}$ associated with the projected snapshots $\{\hat{\rho}_{\text{CDI}}(s_i) - \Pi_{\mathcal{Z}_0} \hat{\rho}_{\text{CDI}}(s_i)\}_{i=1}^{500}$ where $\mathcal{Z}_0 = \text{span}\{\rho_0, \rho_1\}$; finally, we show the projection error (38) associated with the space $\mathcal{Z}_m = \mathcal{Z}_0 \oplus \mathcal{Z}_{\text{lf}}$ and we compare it with the error of $\mathcal{Z}_0$. We perform this test for the snapshots associated with the two different CFD meshes; on the other hand, the FE mesh that is employed for the discretization of the mapping and the hyper-parameters of Algorithm 3 are kept constant.

A different behavior is observed for the two discretizations. For the coarse mesh, data augmentation yields only a modest reduction of the $L^2(\Omega)$ error (from 0.106 to 0.102), whereas for the fine mesh the improvement is more pronounced (from 0.14 to 0.11). This can be attributed to the fact that (i) shocks are better resolved on the fine mesh and are therefore more reliably detected by the shock sensor, and (ii) the interpolation involved in the CDI procedure is more accurate on the fine mesh.

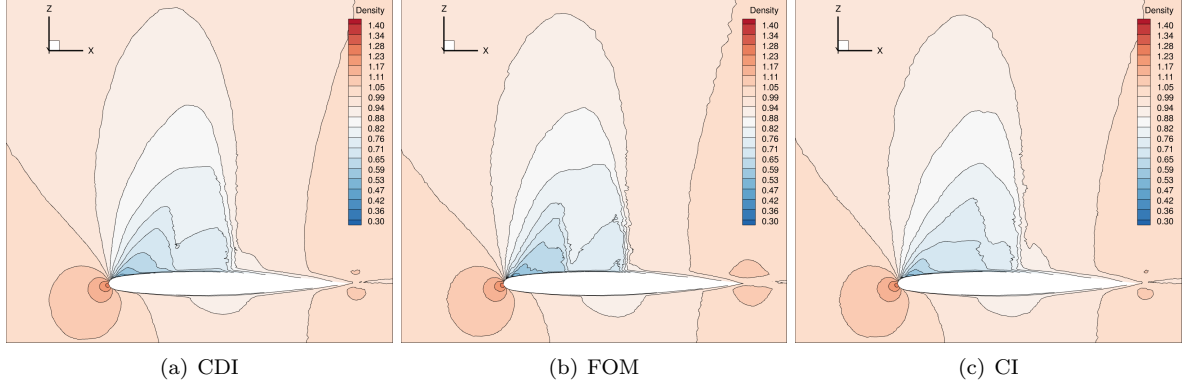


Figure 20: viscous flow past an ONERA M6 wing; cut of the full field at $Y = 8.8$ as predicted by CDI, FOM and CI.

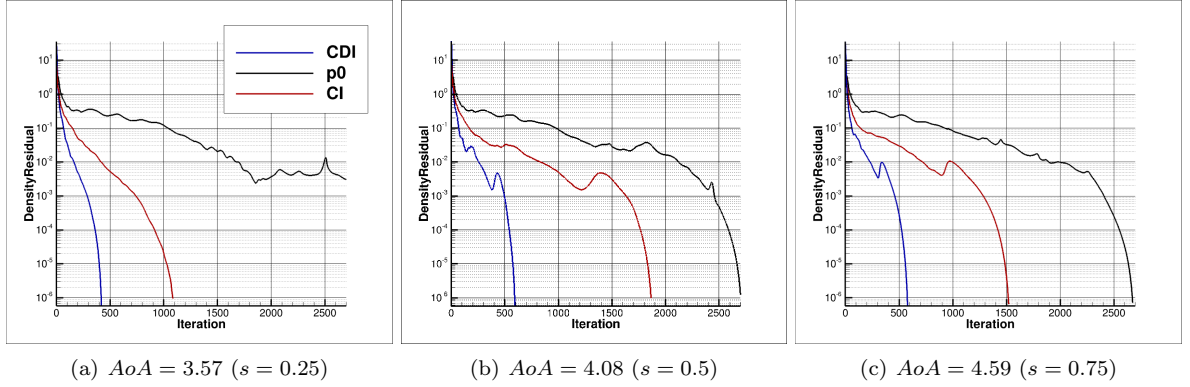


Figure 21: viscous flow past an ONERA M6 wing; convergence history of the density residuals for different initializations of the DG FOM.

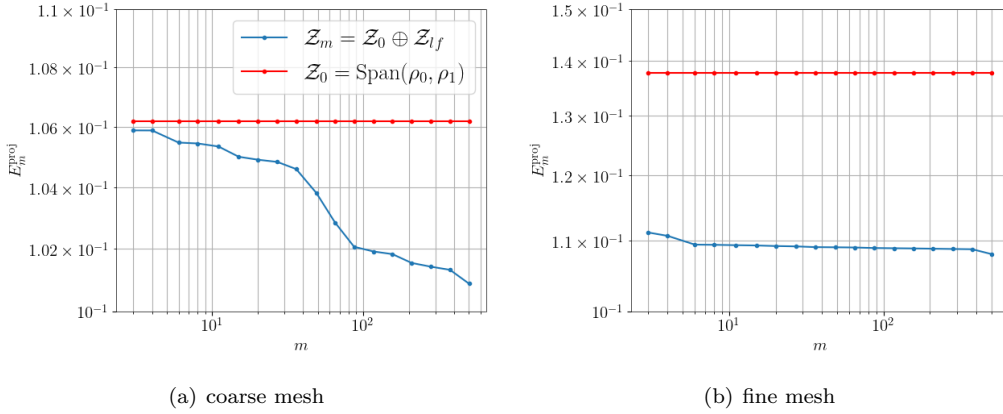


Figure 22: Data augmentation based on CDI for two different CFD meshes.

5 Conclusions

We have presented a registration procedure for parametric model order reduction in two and three-dimensional bounded domains. Our method combines vector flows with an expectation-maximization algorithm and can be interpreted as a CPD approach adapted to bounded geometries. Even if the procedure is general (i.e., independent of the underlying PDE model), in this work we specifically target nonlinear interpolation of aerodynamic fields. [We remark that registration-based techniques have also been applied to nonlinear mechanics \(see, e.g., \[2\]\): we hence believe that our method has significant potential beyond fluid mechanics.](#)

We focused on the deployment of CDI to handle parameter-dependent discontinuities. The numerical results for transonic inviscid and viscous flows demonstrated that CDI effectively reconstructs flow features such as

shocks, and significantly outperforms standard linear interpolation techniques. However, the results for the ONERA M6 wing illustrate that the accuracy of CDI is highly-dependent on shock resolution and the appropriateness of the sensor: this observation motivates further research on the development of effective sensors for shock-dominated flows of aeronautic interest. We also noted that, in the current implementation, the registration cost remains significantly lower than that of a full 3D RANS simulation: for this reason, the computational overhead associated with the solution to the registration problem is modest, as the offline costs are dominated by the HF solves.

Several additional perspectives arise from this work. First, we plan to rigorously incorporate the tools developed in this work into the framework of projection-based model reduction: in this respect, we plan to extend the strategies of [14] and [10] to a broader class of parametric problems. To address this challenge, we will apply our technique to snapshot datasets with more than two elements and to cope with high-dimensional parameterizations. Since each alignment is an independent task, we envision that the registration process is highly parallelizable, though memory allocation for large datasets remains a practical consideration. Second, we intend to explore the use of non-stationary velocity field to align the coherent structures. This could allow to better learn the dynamics of the coherent structures with varying parameters, by using more training snapshots for instance. Third, we plan to investigate the use of second-order methods for registration [36] to improve the computational efficiency of the optimization procedure.

References

- [1] R. A. Adams and J. J. Fournier. *Sobolev spaces*, volume 140. Elsevier, second edition, 2003.
- [2] K. Agathos, S. P. Bordas, and E. Chatzi. Parametrized reduced order modeling for cracked solids. *International Journal for Numerical Methods in Engineering*, 121(20):4537–4565, 2020.
- [3] F. Alauzet. A changing-topology moving mesh technique for large displacements. *Engineering with Computers*, 30(2):175–200, 2014.
- [4] S. R. Allmaras and F. T. Johnson. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *Seventh international conference on computational fluid dynamics (IC-CFD7)*, volume 1902, pages 1–11. Big Island, HI, 2012.
- [5] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS project version 1.5. *Archive of numerical software*, 3(100), 2015.
- [6] P. F. Antonietti, G. Manzini, I. Mazzi, S. Scacchi, and M. Verani. The conforming virtual element method for polyharmonic and elastodynamics problems: a review. *The virtual element method and its applications*, pages 411–451, 2022.
- [7] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2015.
- [8] A. Banyaga. *The structure of classical diffeomorphism groups*, volume 400. Springer Science & Business Media, 2013.
- [9] J. Barnett, C. Farhat, and Y. Maday. Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility. *Journal of Computational Physics*, 492:112420, 2023.
- [10] N. Barral, T. Taddei, and I. Tifouti. Registration-based model reduction of parameterized pdes with spatio-parameter adaptivity. *Journal of Computational Physics*, 499:112727, 2024.
- [11] M. F. Beg, M. I. Miller, A. Trounev, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision*, 61:139–157, 2005.
- [12] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.
- [13] M. Bergmann, M. G. Carlino, and A. Iollo. Model order reduction using a collocation scheme on Chimera meshes: Addressing the Kolmogorov-width barrier. *SIAM Journal on Scientific Computing*, 47(4):A2272–A2298, 2025.
- [14] F. Bernard, A. Iollo, and S. Riffaud. Reduced-order model for the BGK equation based on POD and optimal transport. *Journal of Computational Physics*, 373:545–570, 2018.

- [15] C. M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [16] L. G. Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.
- [17] H. M. Bücker, B. Pollul, and A. Rasch. On cfl evolution strategies for implicit upwind methods in linearized euler equations. *International journal for numerical methods in fluids*, 59(1):1–18, 2009.
- [18] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.
- [19] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. Deformable templates using large deformation kinematics. *IEEE transactions on image processing*, 5(10):1435–1447, 1996.
- [20] S. Cucchiara, A. Iollo, T. Taddei, and H. Telib. Model order reduction by convex displacement interpolation. *Journal of Computational Physics*, 514:113230, 2024.
- [21] M. De Buhan, C. Dapogny, P. Frey, and C. Nardoni. An optimization method for elastic shape matching. *Comptes Rendus. Mathématique*, 354(8):783–787, 2016.
- [22] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.
- [23] J. E. Dennis Jr and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.
- [24] K. Han, S. Sun, X. Yan, C. You, H. Tang, J. Naushad, H. Ma, D. Kong, and X. Xie. Diffeomorphic image registration with neural velocity field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1869–1879, 2023.
- [25] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- [26] A. Iollo, J. Labatut, P. Mounoud, and T. Taddei. Mathematical aspects of registration methods in bounded domains. *arXiv preprint arXiv:2601.03010*, 2026.
- [27] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2):022923, 2014.
- [28] A. Iollo and T. Taddei. Mapping of coherent structures in parameterized flows by learning optimal transportation with Gaussian models. *Journal of Computational Physics*, 471:111671, 2022.
- [29] A. Kabalan, F. Casenave, F. Bordeu, V. Ehrlacher, and A. Ern. Elasticity-based morphing technique and application to reduced-order modeling. *Applied Mathematical Modelling*, 141:115929, 2025.
- [30] P. Krah, A. Marmin, B. Zorawski, J. Reiss, and K. Schneider. A robust shifted proper orthogonal decomposition: Proximal methods for decomposing flows with multiple transports. *SIAM Journal on Scientific Computing*, 47(2):A633–A656, 2025.
- [31] A. Kuzmin. On the lambda-shock formation on ONERA M6 wing. *International Journal of Applied Engineering Research*, 9(20):7029–7038, 2014.
- [32] B. P. Lamichhane. A mixed finite element method for the biharmonic problem using biorthogonal or quasi-biorthogonal systems. *Journal of Scientific Computing*, 46(3):379–396, 2011.
- [33] J. M. Lee. *Introduction to smooth manifolds*. Springer, 2003.
- [34] E. Luke, E. Collins, and E. Blades. A fast mesh deformation method using explicit interpolation. *Journal of Computational Physics*, 231(2):586–601, 2012.
- [35] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian. Robust feature matching for remote sensing image registration via locally linear transforming. *IEEE Transactions on Geoscience and Remote Sensing*, 53(12):6469–6481, 2015.
- [36] A. Mang and G. Biros. Constrained h^1 -regularization schemes for diffeomorphic image registration. *SIAM journal on imaging sciences*, 9(3):1154–1194, 2016.
- [37] A. T. McRae, C. J. Cotter, and C. J. Budd. Optimal-transport-based mesh adaptivity on the plane and sphere using finite elements. *SIAM Journal on Scientific Computing*, 40(2):A1121–A1148, 2018.

- [38] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.
- [39] M. A. Mirhoseini and M. J. Zahr. Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking. *Journal of Computational Physics*, 473:111739, 2023.
- [40] R. Mojgani and M. Balajewicz. Low-rank registration based manifolds for convection-dominated PDEs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 399–407, 2021.
- [41] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.
- [42] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.
- [43] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathematique*, 351(23-24):901–906, 2013.
- [44] M. Ohlberger and S. Rave. Reduced basis methods: Success, limitations and future challenges. In *Proceedings of the Conference Algoritmy*, 2016. 1–12.
- [45] G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [46] A. Quarteroni. *Numerical models for differential problems*. Springer, 2009.
- [47] A. H. Razavi and M. Yano. Registration-based nonlinear model reduction of parametrized aerodynamics problems with applications to transonic Euler and RANS flows. *Journal of Computational Physics*, 521:113576, 2025.
- [48] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018.
- [49] F. Renac, M. de la Llave Plata, E. Martin, J.-B. Chapelier, and V. Couaillier. Aghora: a high-order DG solver for turbulent flow simulations. In *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach: Results of a Collaborative Research Project Funded by the European Union, 2010-2014*, pages 315–335. Springer, 2015.
- [50] R. Salakhutdinov and S. T. Roweis. Adaptive overrelaxed bound optimization methods. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 664–671, 2003.
- [51] F. Santambrogio. *Optimal transport for applied mathematicians*, volume 87. Springer, 2015.
- [52] V. Schmitt and F. Charpin. Pressure distributions on the ONERA M6-wing at transonic Mach numbers, experimental data base for computer program assessment. *Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138*, 1979.
- [53] G. R. Shubin, A. Stephens, H. Glaz, A. Wardlaw, and L. Hackerman. Steady shock tracking, Newton’s method, and the supersonic blunt body problem. *SIAM Journal on Scientific and Statistical Computing*, 3(2):127–144, 1982.
- [54] P. Stefanin Volpiani, J.-B. Chapelier, A. Schwöppe, J. Jägersküpper, and S. Champagneux. Aircraft simulations using the new CFD software from ONERA, DLR, and Airbus. *Journal of Aircraft*, 61(3):857–869, 2024.
- [55] T. Taddei. A registration method for model order reduction: data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 2020.
- [56] T. Taddei. Compositional maps for registration in complex geometries. *SIAM Journal on Scientific Computing*, 47(1):B220–B251, 2025.
- [57] T. Taddei and L. Zhang. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs. *ESAIM: Mathematical Modelling and Numerical Analysis*, 55(1):99–130, 2021.
- [58] I. Tifouti, N. Barral, and T. Taddei. Registration-based model reduction with local reduced order bases. In *AIAA SCITECH 2025 Forum*, page 1571, 2025.

- [59] L. Younes. *Shapes and diffeomorphisms*, volume 171. Springer, second edition, 2010.
- [60] M. J. Zahr and P.-O. Persson. An optimization-based approach for high-order accurate discretization of conservation laws with discontinuous solutions. *Journal of Computational Physics*, 365:105–134, 2018.