Model order reduction using a collocation scheme on chimera meshes: addressing the Kolmogorov N-width barrier

Michel Bergmann, Michele Giuliano Carlino, Angelo Iollo

MODEL ORDER REDUCTION USING A COLLOCATION SCHEME ON CHIMERA MESHES: ADDRESSING THE KOLMOGOROV N-WIDTH BARRIER*

MICHEL BERGMANN†, MICHELE GIULIANO CARLINO†‡, AND ANGELO IOLLO†

Abstract. In this paper, we propose a novel collocation-based Model Order Reduction (cMOR) strategy for solving parametric advection-diffusion PDEs on moving Chimera grids. Unlike traditional projection-based MOR, cMOR solves the High-Dimensional Model on a small subset of collocation points and extends the solution to the entire domain using a global reduced basis. By leveraging the ADER method on unsteady Chimera meshes, cMOR addresses the computational challenges posed by convection-dominated problems, particularly the Kolmogorov N-width barrier. Our results demonstrate the efficiency of cMOR in reducing the computational cost while maintaining accuracy.

Key words. Model Order Reduction (MOR), Collocation-based MOR (cMOR), Parametric PDEs, Chimera grids, ADER, Kolmogorov N-width, Moving meshes.

MSC codes. 65M08, 65M50, 65M55, 65D32.

1. Introduction.

Projection-based Model Order Reduction (pMOR) is a numerical integration technique designed to reduce the computational cost of solving problems arising from the discretization of parametric Partial Differential Equations (PDEs). It achieves this by representing the solution within a smaller, typically linear, subspace [28, 5] *a priori* chosen to capture the dominant behavior of the system. After incorporating this ansatz into the high-dimensional numerical model, the corresponding residuals are projected onto a subspace [35, 21, 22, 2, 25, 24] that may eventually differ from the one used for representing the solution [20]. This process yields a Reduced-Order Model (ROM) that significantly reduces the number of Degrees of Freedom (DOFs) while preserving the essential characteristics of the original system. Consequently, the ROM enables faster multi-query simulations or analyses compared to the original model.

During the offline (learning) stage, the linear solution subspace, spanned by the so-called empirical Reduced Basis (RB), is determined using the Proper Orthogonal Decomposition (POD) technique [28]. POD is applied to a database of snapshots from a High-Dimensional Model (HDM) corresponding to specific parameter instances. In the subsequent online step, the reduced model is used to solve for new parameter instances, yielding out-of-sample solutions.

The actual online cost reduction of pMOR depends on both the dimension N of the reduced space and the number $M \gg N$ of DOFs in the original HDM. If the HDM is nonlinear, the projection of the residuals involves $\mathcal{O}(M)$ operations, except for particular cases where nonlinearities are polynomial. To address this computational bottleneck, several methods have been developed, including Energy-Conserving Sampling and Weighting (ECSW), the Empirical Interpolation Method (EIM), Empirical Quadratures and more broadly, mesh sampling techniques (e.g., see [17, 18, 14, 20, 41, 36] and their references). These methods, collectively known as

*Submitted to the editors September 19, 2024.

†Université de Bordeaux, IMB, UMR 5251, 351 Cours de la Libération, F-33400 Talance, France, and Equipe-Projet Memphis, Inria de l'Université de Bordeaux, 200 Av. de la Vieille Tour, F-33400 Talance, France (michel.bergmann@inria.fr, angelo.iollo@inria.fr).

‡DAAA, ONERA, Institut Polytechnique de Paris, 8 Rue des Vertugadins, F-92190 Meudon, France (michele.carlino@onera.fr)

hyper-reduction techniques, aim to select a subset of points defining a reduced mesh from a larger mesh for constructing the ROM. The selected points, known as empirical quadrature points, are significantly fewer than the full set of canonical quadrature points required to assemble the algebraic components of the reduced model. Hence, this selection substantially reduces the global computational cost of the construction process.

pMOR proves particularly valuable when dealing with high-dimensional and computationally expensive models, which frequently occur in the design and analysis of complex engineering systems. Such applications involve numerous parameters and often require real-time or near real-time performance. Examples include structural mechanics, fluid dynamics, and control systems (e.g., [29, 31, 8, 1, 42]).

Despite the success of pMOR in many applications, challenges remain for certain types of problems. Specifically, projecting the residuals of an HDM can be delicate. For example, Galerkin-type projections for models dominated by advection phenomena may result in an unstable time integration or ill-conditioned problems. In such cases, Petrov-Galerkin-type residual minimization approaches are necessary [9, 17, 18, 19]. However, even these approaches do not guarantee optimal results. The approximation obtained can significantly differ from the optimal projection in the reduced space and is highly dependent on the norms used.

In this article, we propose an alternative reduced-order strategy with respect to pMOR. This approach, called collocation Model Order Reduction (cMOR), differs from traditional pMOR in that it does not obtain the solution in the reduced space through projection. Instead, cMOR finds the solution at specific collocation points, which are determined using hyper-reduction techniques. Like pMOR, cMOR includes an offline learning step to define the RB from a set of snapshots. In the online phase, the solution is computed locally by integrating the HDM over this empirical subset of points. The local solution is then extended to the entire domain by the RB. The use of cMOR offers several advantages. It inherits the simplicity of the underlying HDM, making it straightforward to implement. In this sense, cMOR is a consistent approach that can be easily integrated into existing HDM codes.

The HDM underlying the cMOR investigated in this paper is the ADER method applied to moving chimera grids for advection-diffusion or pure advection problems [4, 3, 10]. This advanced method avoids complex time-dependent transmission conditions between grid blocks and achieves second-order accuracy in both space and time [3, 4]. It also supports the use of multiple overlapping foreground meshes with independent movements. In the context of ROMs, this method provides an ideal framework for PDE solutions that depend on geometrical parameters, possibly on domains that evolve over time.

Chimera grids are composed of overlapping mesh blocks that form an overset grid for discretizing PDEs. Typically, a background mesh is combined with foreground patches that fit the domain boundaries. This method simplifies mesh adaptation for boundary layers, unsteady problems with changing geometry (e.g., fluid-structure interaction), and multiply connected domains. After generating the mesh patches, they are combined to create overlapping zones for exchanging information between grids. A compact transmission condition, using only the first layer of cells around each point, is often used to minimize communication between grids.

Since the reduced bases can be defined onto their respective, potentially unsteady, grid blocks, the challenges associated with the $\mathcal{O}\left(N^{-1/2}\right)$ rate of decrease of the Kolmogorov width for convection-dominated problems [34] can be effectively addressed. The key advantage of our approach lies in the possibility of defining a local RB that

is aligned with the unsteady features of the solution.

The idea of tracking the coherent structures or the salient features of the solution is not new. In [26], this was achieved using a Lagrangian map. Similar approaches have since been discussed in [33, 32] as well as [6, 16, 27, 15]. However, a key difficulty with these approaches is ensuring the bijectivity of the mappings to preserve the geometry of the domain, a task that is not trivial in general domains [15, 27, 39]. The present approach, which combines the ADER scheme with cMOR, bypasses this difficulty.

The paper is organized as follows. In Sec. 2.1 the family of PDEs considered is introduced. The moving Chimera mesh and its construction features are presented in Sec. 2.2. The ADER approach on moving Chimera grids is explained in Sec. 3. Section 4 is devoted to cMORs based on the ADER method. Numerical results are detailed Sec. 5. Conclusions and future developments are given in Sec. 6.

2. Preliminary concepts.

2.1. Problem formulation and Kolmogorov N-width. Let $\Omega(t) \subset \mathbb{R}^2$ be a computational domain that evolves over time. The time-dependence $t \in [0, t_{\text{fin}}]$ is explicitly noted[1]. The problem we address is formulated as follows: *Find* $u : \Omega(t) \times [0, t_{fin}] \to \mathbb{R}$ *such that*

$$\partial_t u + \nabla \cdot \mathbf{F}(u, \nabla u) = f, \quad \mathbf{x} \in \Omega(t), \quad t \in [0, t_{\text{fin}}], \tag{2.1}$$

where appropriate initial and boundary conditions are provided. The advection-diffusion term $\mathbf{F}(\bullet, \nabla\bullet)$ and the forcing term $f(\mathbf{x}, t)$ are given. Specifically, the advection-diffusion term is defined as

$$\mathbf{F}(u, \nabla u) = \mathbf{a}u - \nu \nabla u, \tag{2.2}$$

where $\mathbf{a} \in \mathbb{R}^2$ is the advection field, and $\nu > 0$ is the viscous term. If $\mathbf{a}$ and ν do not depend on the solution u, the problem in (2.1) is linear.

Now, let problem (2.1) be parameterized by $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P$. The parameter $\boldsymbol{\mu}$ can represent, for instance, an advection or diffusion parameter in (2.2), or a geometric parameter that characterizes the computational domain. In such cases, the domain is parameterized as $\Omega(t, \boldsymbol{\mu})$. The original problem (2.1) can then be reformulated in residual form as follows: *Find* $u(\boldsymbol{\mu}) : \Omega(t, \boldsymbol{\mu}) \times [0, t_{fin}] \to \mathbb{R}$ *such that*

$$\mathcal{R}(u(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0, \quad \mathbf{x} \in \Omega(t, \boldsymbol{\mu}), \quad \boldsymbol{\mu} \in \mathcal{P}. \tag{2.3}$$

Problem (2.3) naturally induces a solution map

$$\phi : \mathcal{P} \to V, \tag{2.4}$$

which associates a specific instance of the parameter $\boldsymbol{\mu} \in \mathcal{P}$ with the solution $u(\boldsymbol{\mu}) = \phi(\boldsymbol{\mu}) \in V$, where V is a suitable Hilbert space endowed with the norm $\|\cdot\|_V$. The image $\text{im}(\phi) := \phi(\mathcal{P})$ of the solution map (2.4) describes a manifold in V, where each point is a solution $u(\boldsymbol{\mu})$ corresponding to a specific parameter $\boldsymbol{\mu} \in \mathcal{P}$. Model-order reduction of (2.3) seeks a finite-dimensional subspace $V_N \subset V$, with $\dim V_N = N$, such that the worst-case approximation error for an element of $\text{im}(\phi)$

$$d_{V_N}\left(\text{im}(\phi)\right) := \sup_{v \in \text{im}(\phi)} \inf_{v_N \in V_N} \|v - v_N\|_V \tag{2.5}$$

[1]Throughout this paper, the evolving domain over the entire time interval, $\bigcup_{t\in[0,t_{\text{fin}}]} \Omega(t)$ is denoted by $\Omega(t) \times [0, t_{\text{fin}}]$, by abuse of notation, for ease of reading.

is minimized. This minimum is theoretically characterized by the Kolmogorov N-width, given by

$$d_N\left(\mathrm{im}(\phi)\right) := \inf_{\substack{W\subseteq V \\ \dim W\leq N}} d_W\left(\mathrm{im}(\phi)\right) = \inf_{\substack{W\subseteq V \\ \dim W\leq N}} \sup_{v\in\mathrm{im}(\phi)} \inf_{w\in W} \|v-w\|_V, \tag{2.6}$$

where W is a finite-dimensional subspace of V. In practice, the Kolmogorov N-width $d_N(\mathrm{im}(\phi))$ in (2.6) helps to understand how effectively functions in V can be approximated by functions in W, especially as N increases. In particular, it measures the best approximation that can be achieved among all possible choices of approximating elements linearly spanned by a finite-dimensional subspace of V.

The existence of a subspace V_N that satisfies the equality $d_{V_N}(\mathrm{im}(\phi)) = d_N(\mathrm{im}(\phi))$ is always guaranteed [34]. However, evaluating this subspace is non-trivial, and in some contexts, its exact computation may be practically infeasible. A heuristic estimate for (2.6) can be obtained from the rate of decay of the eigenvalues returned by the Proper Orthogonal Decomposition (POD) on a set of snapshots for problem (2.3). Specifically, if the eigenvalues decay rapidly, it suggests that the low-dimensional space V_N, which approximates the manifold described by $\mathrm{im}(\phi)$, is adequate. This means that most of the energy of the low-dimensional approximation is captured by the first N modes, which are then used to define the reduced space V_N. Conversely, if the decay is slow, more modes N are required to capture the essential features of the dynamical system.

2.2. The chimera mesh. The overset grid, or Chimera mesh, consists of multiple overlapping mesh blocks that discretize the computational domain $\Omega(t) \subset \mathbb{R}^2$ (see Fig. 1 for an example). These blocks overlap in a sub-region known as the *overlapping zone*, and each block can have a different topology. In this paper, all blocks consist of quadrilateral cells. As outlined in [30], the process starts with constructing a Cartesian background mesh $\mathcal{T}_{\mathrm{bg}}(t)$, followed by the introduction of one or more foreground meshes $\mathcal{T}_{\mathrm{fg}}(t)$ within the computational domain. This creates overlapping regions where data is exchanged between meshes, forming *hole zones* in the background mesh, where no data is stored (see Fig. 1, left). The dependence of the domain and its partitions on time $t \in [0, t_{\mathrm{fin}}]$ allows for the potential evolution of the computational geometry. Typically, a specified number n_ℓ of cell layers is imposed in both background and foreground meshes for these overlapping zones, with holes defined as background cells fully covered by the foreground mesh but not part of the overlapping layers.

This discretization method is particularly efficient when the domain boundary $\partial\Omega(t)$ includes internal boundaries $\Gamma_{\mathrm{int}}(t)$ and an external boundary $\Gamma_{\mathrm{ext}}(t)$. Let $\Omega_{\mathrm{int}}(t)$ be the region bounded by the internal boundary $\Gamma_{\mathrm{int}}(t)$. For any cell $\Omega_i(t)$ in a given partition, its cell center $\mathbf{x}_i(t)$ is defined as:

$$\mathbf{x}_i(t) = \frac{1}{|\Omega_i(t)|}\int_{\Omega_i(t)} \mathbf{x}\, d\mathbf{x}, \tag{2.7}$$

where $|\Omega_i(t)|$ represents the area of the cell.

The subdomains corresponding to the background and foreground partitions are defined as:

$$\Omega_{\mathrm{bg}}(t) = \bigcup_{\Omega_i(t)\in\mathcal{T}_{\mathrm{bg}}(t)} \Omega_i(t), \quad \text{and} \quad \Omega_{\mathrm{fg}}(t) = \bigcup_{\Omega_i(t)\in\mathcal{T}_{\mathrm{fg}}(t)} \Omega_i(t). \tag{2.8}$$

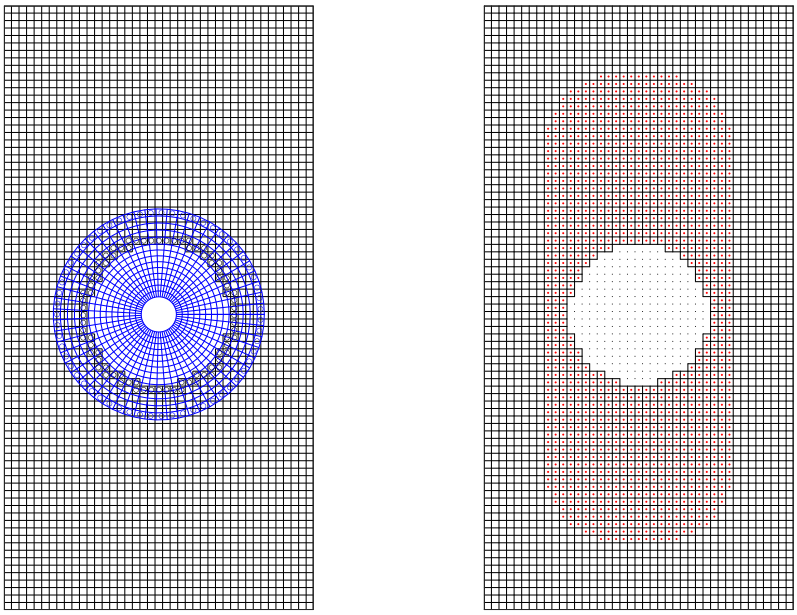

Fig. 1: Example of Chimera discretization for a computational domain $\Omega(t)$ at a given time $t \in [0, t_{\text{fin}}]$. On the left, the domain is discretized using a background Cartesian grid (black) and a polar mesh (blue) for the internal circular boundary. Fringe cells are indicated by circles (black for the background, blue for the foreground). On the right, only the background mesh is shown, with hole cells outlined by coarse dotted lines. Assuming the internal cylinder moves vertically during the simulation, cells marked with red points define the dynamic partition $\mathcal{T}_{\text{bg}}^{\text{D}}$, where the hole remains confined. The remaining cells form the static partition $\mathcal{T}_{\text{bg}}^{\text{S}}$.

Finally, in our two-dimensional setting, the characteristic size of a cell $\Omega_i(t)$ is given by $\ell_i(t) = \sqrt{|\Omega_i(t)|}$. We define C_i as the circle centered at $\mathbf{x}_i(t)$ with a radius of $2\ell_i$.

According to a traditional classification [38], cells are categorized as: i) Internal: cells where all differential operators involved in the PDE are discretized according to the chosen numerical scheme; ii) Fringe: cells that facilitate communication between different mesh blocks, located at the external boundaries of foreground partitions and the boundaries of holes in the background mesh; iii) Hole: inactive cells where no local discretization of differential operators occurs. Figure 1 illustrates internal and fringe cells on the left and hole cells on the right.

For the purposes of this paper, an additional cell classification is proposed for the background partition $\mathcal{T}_{\text{bg}}(t)$. At a specific time t, let $\mathcal{T}_{\text{bg}}^{\text{hole}}(t)$ denote the sub-partition containing all cells that make up the hole. The remaining cells are classified into either the sub-partition $\mathcal{T}_{\text{bg}}^{\text{ol}}(t)$, consisting of cells whose centers are fully covered by the foreground subdomain $\Omega_{\text{fg}}(t)$, or the sub-partition $\mathcal{T}_{\text{bg}}^{\text{nol}}(t)$, which contains cells not covered by $\Omega_{\text{fg}}(t)$ (as shown in Fig. 1, right). This results in the following unique

partition:

$$\mathcal{T}_{\mathrm{bg}}(t) = \mathcal{T}_{\mathrm{bg}}^{\mathrm{hole}}(t) \cup \mathcal{T}_{\mathrm{bg}}^{\mathrm{ol}}(t) \cup \mathcal{T}_{\mathrm{bg}}^{\mathrm{nol}}(t) \quad \forall t \in [0, t_{\mathrm{fin}}], \tag{2.9}$$

with $\mathcal{T}_{\mathrm{bg}}^{\star}$ being mutually disjoint for any $\star \in \{\mathrm{hole}, \mathrm{ol}, \mathrm{nol}\}$. This classification is specific to each time step. By considering the union of these sub-partitions over the entire simulation period, we define static and dynamic partitions as follows:

$$\mathcal{T}_{\mathrm{bg}}^{\mathrm{S}} = \bigcup_{t \in [0, t_{\mathrm{fin}}]} \mathcal{T}_{\mathrm{bg}}^{\mathrm{nol}}(t), \quad \text{and} \quad \mathcal{T}_{\mathrm{bg}}^{\mathrm{D}} = \bigcup_{t \in [0, t_{\mathrm{fin}}]} \mathcal{T}_{\mathrm{bg}}^{\mathrm{ol}}(t). \tag{2.10}$$

Cells that are never covered by the foreground subdomain remain internal and active (i.e., they cannot become hole cells during the simulation). These cells are referred to as static because they remain in their original classification throughout the simulation. The same applies to the foreground partition, so we define the static subset as $\mathfrak{S} = \{\mathcal{T}_{\mathrm{bg}}^{\mathrm{S}}, \mathcal{T}_{\mathrm{fg}}\}$. Conversely, cells in $\mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$ define a subregion where the hole is confined throughout the simulation. Cells in this partition can potentially change their classification from internal to fringe or hole, thus they are termed dynamic. As a result, $\mathcal{T}_{\mathrm{bg}}^{\mathrm{hole}}(t) \subset \mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$ for any $t \in [0, t_{\mathrm{fin}}]$. The stencil $\mathcal{S}_i(t)$ centered on $\Omega_i(t)$ is constructed to be compact. For more details, refer to [3, 11].

3. The finite volume ADER on moving chimera meshes. Let the time interval $[0, t_{\mathrm{fin}}]$ be divided into N_t sub-intervals $[t^n, t^{n+1})$ for $n = 0, \dots, N_t - 1$, with $t^0 = 0$ and $t^{N_t} = t_{\mathrm{fin}}$. The time step $\Delta t = t^{n+1} - t^n$ satisfies the CFL condition [3]. The value of a generic time-dependent variable $\omega(t)$ at time t^n is denoted as ω^n. In this section we omit the dependency of the solution on the specific instance $\boldsymbol{\mu} \in \mathcal{P}$ for clarity.

The cell average

$$u_i^n = \frac{1}{|\Omega_i^n|} \int_{\Omega_i^n} u(\mathbf{x}, t^n) \, \mathrm{d}\mathbf{x},$$

is assumed to be constant throughout the cell and defined at the cell center $\mathbf{x}_i^n$.

To achieve second-order convergence, the cell average u_i^n must be properly reconstructed. Let $E_i^n = \Omega_i^n \cup \bigcup_{\Omega_j^n \in \mathcal{S}_i^n}$ be the subset in Ω^n defined by the active cell Ω_i^n of characteristic size ℓ_i^n and its neighboring cells stored in the stencil $\mathcal{S}_i^n$. The discrete solution is reconstructed over the domain using a quadratic polynomial basis. Specifically, we use a local quadratic Taylor polynomial basis $\mathcal{P}_2(E_i^n)$, defined as

$$\mathcal{P}_2(E_i^n) := \operatorname{span}\left\{1, \frac{x - x_i}{\ell_i}, \frac{y - y_i}{\ell_i}, \frac{(x - x_i)(y - y_i)}{\ell_i^2}, \frac{(x - x_i)^2}{2\ell_i^2}, \frac{(y - y_i)^2}{2\ell_i^2}\right\}.$$

Let $\Pi_{E_i^n} u$ be the local reconstruction of u^n using the basis defined above. The solution in E_i^n is known only at the cell centers $\mathbf{x}_j^n$ within the stencil $\mathcal{S}_i^n$. Thus, the polynomial coefficients of the reconstruction are determined by imposing

$$u_i^n(\mathbf{x}_j^n) = \Pi_{E_i^n} u(\mathbf{x}_j^n)$$

for each cell in the stencil. This requires the stencil to include at least five cells. Generally, for the mesh type used in this paper, the stencil cardinality exceeds five elements. Consequently, the reconstruction is exact for the pivot cell Ω_i^n, while for the other cells the reconstruction is obtained in a least-squares sense [3, 4, 11].

3.1. Foreground mesh motion. Any point coordinate $\mathbf{X}$ of the foreground mesh follows the Cauchy problem:

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{V}, \tag{3.1}$$

where $\mathbf{V}$ is the deformation velocity of the foreground mesh. In this paper, $\mathbf{V}$ depends on the unknown geometric parameter $\boldsymbol{\mu}$, time, and potentially space, though the latter generalization does not affect the numerical scheme description. If $\mathbf{V}$ is independent of space, the mesh translates; otherwise, it deforms. The mesh motion (3.1) is defined by an appropriate initial condition.

Let $\mathcal{C}_i^n = \Omega_i(t) \times [t^n, t^{n+1}]$ represent the space-time finite volume describing the evolution of cell Ω_i from time t^n to t^{n+1}. Let $\hat{\mathcal{C}} = [0,1]^3$ be a reference space-time volume. To solve both the mesh motion (3.1) and problem (2.1) using a finite volume approach, we find a map $\mathcal{X}_i : \hat{\mathcal{C}} \to \mathcal{C}_i^n$ using an isoparametric or Lagrangian method. For this purpose, the reference coordinates in the unit cube $\hat{\mathcal{C}}$ are denoted by $\boldsymbol{\xi} = (\xi, \eta, \tau)$. The map $\mathcal{X}_i$ is then formally expressed as:

$$\mathcal{X}_i : \begin{cases} x = x(\xi, \eta, \tau) \\ y = y(\xi, \eta, \tau) \\ t = t^n + \Delta t\, \tau \end{cases}. \tag{3.2}$$

Note that the time variable t depends only on the reference time τ.

Following the approach originally proposed in [23] for general advection-diffusion-reaction PDEs on static meshes and extended in [3, 4] for overset grids, we adopt a nodal formulation based on a tensor product of Gauss-Legendre quadrature points over the reference space-time unit cube $\hat{\mathcal{C}}$. This formulation allows us to construct an L^2-orthogonal Lagrangian basis for projecting the solution of (3.1). To achieve second-order accuracy, the nodes along one direction in the unit interval $[0,1]$ are $\{(5 \pm \sqrt{15})/10, 1/2\}$. Throughout the reference volume $\hat{\mathcal{C}}$, there are 27 Gauss-Legendre nodes $\hat{\boldsymbol{\xi}}_l$ defining 27 Lagrange polynomials $\theta_l : \hat{\mathcal{C}} \to \mathbb{R}$ satisfying:

$$\theta_l(\hat{\boldsymbol{\xi}}_m) = \delta_{lm} \quad \text{and} \quad \int_{\hat{\mathcal{C}}} \theta_l \theta_m \, \mathrm{d}\boldsymbol{\xi} = \delta_{lm} \|\theta_l\|^2_{L^2(\hat{\mathcal{C}})} \tag{3.3}$$

for $l, m = 1, \ldots, 27$, with δ_{lm} being the Kronecker symbol. We introduce the following functional spaces:

$$\Theta = \left\{ \theta \in L^2(\hat{\mathcal{C}}) : (0,1)^2 \ni (\xi, \eta) \mapsto \theta \in H^1((0,1)) \right\}, \tag{3.4}$$

$$Q = \left\{ q \in H^1(\hat{\mathcal{C}}) : q(\xi, \eta, 0) = (\Pi_{E_i^n} q \circ \mathcal{X}_i|_{\tau=0})(\xi, \eta) \text{ and } J_i^{-1} \begin{bmatrix} \partial_\xi q \\ \partial_\eta q \\ \partial_\tau q \end{bmatrix} \in L^2(\hat{\mathcal{C}}; \mathbb{R}^3) \right\}.$$

Here, H^1 denotes the Sobolev Space of L^2-integrable functions whose first-order derivatives are also L^2-integrable [37], and J_i is the Jacobian tensor of the map $\mathcal{X}_i$. In these spaces, we define the following scalar products:

$$\begin{aligned} \langle \theta, q \rangle &:= \int_{\hat{\mathcal{C}}} \theta q \, \mathrm{d}\boldsymbol{\xi}, \\ [\theta, q]_\tau &:= \int_0^1 \int_0^1 \theta(\xi, \eta, \tau) q(\xi, \eta, \tau) \, \mathrm{d}\xi \mathrm{d}\eta \quad \forall \tau \in [0,1], \end{aligned} \tag{3.5}$$

respectively. Note that the introduced Lagrangian polynomial basis defines a finite-dimensional subspace of both functional spaces in (3.4).

With this functional setting, we can locally solve the foreground mesh motion (3.1) within the space-time volume $\mathcal{C}_i^n$ for any cell $\Omega_i(t)$. The problem is closed by setting the solution $\mathbf{X}^n$ at the current time equal to $\mathbf{X}(t^n)$ from the previous time step in $\mathcal{C}_i^{n-1}$. Since $\mathbf{X} = \mathbf{X}(t) = \mathbf{X}(t(\tau))$ via the map (3.2), the motion equation becomes:

$$\frac{\partial \mathbf{X}}{\partial \tau} = \Delta t \mathbf{V} \quad \text{in } \hat{\mathcal{C}}, \tag{3.6}$$

where the space and velocity components ($\mathbf{X}$ and $\mathbf{V}$, respectively) retain the same symbols for both physical $(\mathbf{x}, t)$ and reference $\boldsymbol{\xi}$ variables by convention. The goal is to project equation (3.6) onto space Q using any test function θ from Θ. The unknown space and velocity are approximated using the introduced Lagrangian basis as:

$$\mathbf{X} = \sum_{l=1}^{27} \theta_l \hat{\mathbf{X}}_l \quad \text{and} \quad \mathbf{V} = \sum_{l=1}^{27} \theta_l \hat{\mathbf{V}}_l, \tag{3.7}$$

where $\hat{\mathbf{X}}_l = \mathbf{X}|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}_l}$ and $\hat{\mathbf{V}}_l = \mathbf{V}|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}_l}$ are the Lagrangian polynomial coefficients, with $(\hat{\mathbf{X}}_l)_{l=1}^{27}$ being the unknowns. After integrating by parts along the τ-direction, equation (3.6) reduces to:

$$\sum_{l=1}^{27} ([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\mathbf{X}}_l = \Delta t \left(\sum_{l=1}^{27} \langle \theta_k, \theta_l \rangle \hat{\mathbf{V}}_l \right) + \left(\sum_{l=1}^{27} [\theta_k, \theta_l]_1 \hat{\mathbf{X}}_l^n \right), \tag{3.8}$$

for any $k = 1, \dots, 27$. If the deformation velocity $\mathbf{V}$ depends on space $\mathbf{X}$, problem (3.8) is solved using a fixed point algorithm at the current iteration $n+1$ in the space-time volume $\mathcal{C}_i^{n+1}$. The unknowns are $\hat{\mathbf{X}}_l^{n+1}$ for $l = 1, \dots, 27$.

3.2. Prediction step. When the foreground mesh undergoes displacement during the time interval $[t^n, t^{n+1}]$, the mapping function $\mathcal{X}$ is fully determined. This mapping allows the original problem to be solved locally within a specific control space-time volume, denoted as $\mathcal{C}_i^n$. The problem can be stated as follows: *find* $q : \mathcal{C}_i^n \to \mathbb{R}$ *such that*

$$\begin{cases} \partial_t q + \nabla \cdot \mathbf{F}(q, \nabla q) = f & \text{in } \mathcal{C}_i^n \\ q|_{t=t^n} = \Pi_{E_i^n} u & \text{on } \Omega_i^n \end{cases}. \tag{3.9}$$

where the equation represents the local restriction of the original PDE to the control volume $\mathcal{C}_i^n$. The initial condition is provided by the interpolation of the solution u over the space-time element Ω_i^n, denoted as $\Pi_{E_i^n} u$. Importantly, this local problem (3.9) does not consider information from neighboring cells, making this step computationally parallelizable, with no interdependencies among different cells.

To discretize the problem (3.9) using the isogeometric approach, it is first transformed into reference coordinates $\boldsymbol{\xi} \in \hat{\mathcal{C}}$. Then, the problem is projected onto the functional space Q (introduced earlier) and approximated using a Lagrangian basis defined by a Gauss-Legendre nodal formulation.

The inverse Jacobian matrix J_i^{-1} of the mapping $\mathcal{X}_i$ is expressed as:

$$J_i^{-1} = \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}(\mathbf{x}, t)} = \begin{bmatrix} \partial_x \xi & \partial_y \xi & \partial_t \xi \\ \partial_x \eta & \partial_y \eta & \partial_t \eta \\ 0 & 0 & 1/\Delta t \end{bmatrix} = \begin{bmatrix} J_{s,i}^{-1} & \partial_t \boldsymbol{\Xi} \\ \mathbf{0}_2^{\mathrm{T}} & 1/\Delta t \end{bmatrix},$$

where $J_{s,i}^{-1}$ is a submatrix representing spatial derivatives, $\partial_t \boldsymbol{\Xi}$ is the time derivative vector, and $\mathbf{0}_2$ is a two-dimensional zero vector. This notation simplifies the representation of the local problem in reference coordinates $\hat{\mathcal{C}}$, leading to the reformulated problem:

$$\partial_\tau q + \Delta t \mathcal{F}^{\mathrm{t}}\left(\hat{\nabla} q\right) + \Delta t J_{s,i}^{-\mathrm{T}} \hat{\nabla} \cdot \boldsymbol{\mathcal{F}}^{\mathrm{st}}\left(q, \hat{\nabla} q\right) = \Delta t f, \tag{3.10}$$

with

$$\hat{\nabla} = \begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}, \quad \mathcal{F}^{\mathrm{t}}\left(\hat{\nabla} q\right) = \hat{\nabla} q \cdot \partial_t \boldsymbol{\Xi}, \quad \boldsymbol{\mathcal{F}}^{\mathrm{st}}\left(q, \hat{\nabla} q\right) = \mathbf{F}\left(q, J_{s,i}^{-\mathrm{T}} \hat{\nabla} q\right).$$

Here, q and the source term f are represented in both physical $(\mathbf{x}, t)$ and reference $(\boldsymbol{\xi})$ coordinates.

The weak formulation of the reference coordinate problem (3.10) is obtained by integrating by parts along the τ-direction and approximating the trial and test functions in the functional spaces Q and Θ, respectively:

$$\begin{aligned}([\theta_k,\theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l\rangle)\hat{q}_l + \Delta t \langle \theta_k, \theta_l\rangle \hat{\mathcal{F}}_l^{\mathrm{t}} + \Delta t \langle \theta_k, (\partial_x \xi\, \partial_\xi + \partial_x \eta\, \partial_\eta)\theta_l\rangle \hat{\mathcal{F}}_{\xi,l}^{\mathrm{st}} \\ + \Delta t \langle \theta_k, (\partial_y \xi\, \partial_\xi + \partial_y \eta\, \partial_\eta)\theta_l\rangle \hat{\mathcal{F}}_{\eta,l}^{\mathrm{st}} = \Delta t \langle \theta_k, f\rangle + [\theta_k, \Pi_{E_i^n} u \circ \mathcal{X}_i]_0,\end{aligned}$$

for $k = 1, \ldots, 27$, where $\hat{q}_l$ represents the unknown values at reference coordinates $\hat{\boldsymbol{\xi}}_l$, and the fluxes $\hat{\mathcal{F}}_l^{\mathrm{t}}$ and $(\hat{\mathcal{F}}_{\xi,l}^{\mathrm{st}}, \hat{\mathcal{F}}_{\eta,l}^{\mathrm{st}})$ are evaluated at $\hat{\boldsymbol{\xi}}_l$. Given that these fluxes depend on the unknowns $\hat{q}_l$, a fixed-point iteration method is used to solve the weak formulation.

3.3. Correction step. The solution of problem (2.1) obtained so far is purely local, meaning it does not incorporate information from neighboring cells. Therefore, a correction step is performed using a finite space-time volume $\mathcal{C}_i^n$. We define the space-time first-order derivative operator as $\nabla_{\mathbf{x},t} = [\nabla^{\mathrm{T}}, \partial_t]^{\mathrm{T}}$. Additionally, let $\mathbf{U} = [\mathbf{F}^{\mathrm{T}}(u, \nabla u), u]^{\mathrm{T}}$ represent a vector that separates the original equation into components that are derived only with respect to space and those derived only with respect to time. Thus, the PDE (2.1) can be rewritten as

$$\nabla_{\mathbf{x},t} \cdot \mathbf{U} = f \quad \text{in } \Omega(t) \times [0, t_{\mathrm{fin}}]. \tag{3.11}$$

By integrating equation (3.11) over the volume $\mathcal{C}_i^n$ and applying the divergence theorem, we obtain:

$$\oint_{\partial \mathcal{C}_i^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t}\, \mathrm{d}\Gamma = \int_{\mathcal{C}_i^n} f\, \mathrm{d}\mathcal{C}, \tag{3.12}$$

where $\mathbf{n}_{\mathbf{x},t} = [\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}, n_t]^{\mathrm{T}} = [n_x, n_y, n_t]^{\mathrm{T}}$ is the unit outer normal vector to the boundary $\partial \mathcal{C}_i^n$, which is given by

$$\partial \mathcal{C}_i^n = \Omega_i^n \cup \Omega_i^{n+1} \cup \bigcup_{j=1}^{4} \Gamma_{ij}^n, \tag{3.13}$$

with Γ_{ij}^n representing the space-time manifold that describes the evolution of edge $e_{ij}^n \subset \partial \Omega_i^n$ at time t^n to edge $e_{ij}^{n+1} \subset \partial \Omega_i^{n+1}$ at time t^{n+1}. Using equation (3.13), the left-hand side can be rewritten as:

$$|\Omega_i^{n+1}|U_i^{n+1} - |\Omega_i^n|U_i^n + \sum_{j=1}^{4} \int_{\Gamma_{ij}^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t} \, \mathrm{d}\Gamma = \int_{\mathcal{C}_i^n} f \, \mathrm{d}\mathcal{C}, \tag{3.14}$$

where

$$U_i^\alpha := \frac{1}{|\Omega_i^\alpha|} \int_{\Omega_i^\alpha} u(x, y, t^\alpha) \, \mathrm{d}\mathbf{x} \quad \text{for } \alpha = n, n+1, \tag{3.15}$$

denotes the cell-averaged solution u. In equation (3.14), U_i^{n+1} represents the unknown timestep $n+1$, U_i^n is the known solution from the previous ADER step, and the last term on the left-hand side is the space-time flux. The formulation in equation (3.14) remains exact, as no approximations have been made so far. In order to discretize the solution, this last term must be approximated. For this purpose, a Rusanov numerical flux is employed:

$$[\mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t}]_{\Gamma_{ij}^n} \simeq \Psi(q_j^+, q_j^-) = \frac{1}{2} \left[\left(\mathbf{U}_j^+ + \mathbf{U}_j^- \right) \cdot \mathbf{n}_{\Gamma_{ij}^n} - s \left(q_j^+ - q_j^- \right) \right], \tag{3.16}$$

where $\mathbf{n}_{\Gamma_{ij}^n}$ represents the evaluation of $\mathbf{n}_{\mathbf{x},t}$ along Γ_{ij}^n, and s is a stabilization term. In expression (3.16), the flux is approximated using the local solution q obtained earlier in the prediction step. The superscripts '+' and '−' indicate the evaluation of a quantity along the manifold Γ_{ij}^n outside and inside the space-time cell $\mathcal{C}_i^n$, respectively. Specifically, quantities outside the volume belong to the neighboring cell that shares the manifold Γ_{ij}^n with $\mathcal{C}_i^n$.

The definition of stabilization s for these types of problems can be found in [3], which draws inspiration from [40]. Based on a reformulation of the original problem (2.1) in relation to Cattaneo relaxation [12, 13], and incorporating an advective-diffusive component (which may be nonlinear) defined in equation (2.2), the stabilization coefficient s in equation (3.16) is given by:

$$s = \frac{1}{2} \left| \sigma + \sqrt{\left(a_x^2 + \frac{4\nu}{\varepsilon} \right) n_x^2 + 2 a_x a_y n_x n_y + \left(a_y^2 + \frac{4\nu}{\varepsilon} \right) n_y^2} \right|, \tag{3.17}$$

with $\sigma = \mathbf{a} \cdot \mathbf{n}_{\mathbf{x}} + 2n_t$ and the nonnegative relaxation parameter ε satisfying the property

$$\frac{C_2 \varepsilon}{h_\Omega^2} = \mathcal{O}(1), \tag{3.18}$$

where C_2 is a constant determined by the order of convergence (which is 2 in this case), and $h_\Omega = \min_i \ell_i$ is a global characteristic size of the mesh. The stabilization parameter defined in (3.17) ensures the method remains effective in the limit where the advection-diffusion problem (2.1) becomes purely hyperbolic, i.e., as the diffusion ν vanishes. In this limiting case, the coefficient (3.17) simplifies to:

$$s = \left| \mathbf{a} \cdot \mathbf{n}_{\mathbf{x}} + n_t \right|, \tag{3.19}$$

which is the classical stabilization coefficient for hyperbolic problems with an evolving frame, as occurs when the mesh evolves. For a more in-depth treatment of this topic, the interested reader is referred to [3]. For formal proofs of all results, it is recommended to consult [40] or [10].

3.4. Chimera mesh evolution. The movement of the foreground mesh also induces changes in the background mesh within the hole region. Let $\Omega_i(t)$ be a generic cell in the background partition. If $\Omega_i(t)$ belongs to the non-overlapping partition $\mathcal{T}_{\text{bg}}^{\text{nol}}$, it remains active at all times, and ADER steps are performed as previously described (with $\mathbf{V} \equiv \mathbf{0}$, since the mesh is stationary). This scenario is straightforward. However, if $\Omega_i(t)$ belongs to the overlapping partition $\mathcal{T}_{\text{bg}}^{\text{ol}}$, its state can evolve in one of three ways between time t^n and t^{n+1}: i) it remains active; ii) it transitions from active to hole; ii) it transitions from hole to active.

The first scenario is handled identically to a cell in $\mathcal{T}_{\text{bg}}^{\text{nol}}$. In the second scenario, only the ADER prediction step is performed, enabling the computation of the Rusanov flux for neighboring cells that are not within the hole. Afterward, the cell (along with any data stored within it) is removed.

In the third scenario, where a cell transitions from hole to active, a second-order polynomial reconstruction of the solution is performed at time t^n. This reconstruction uses information from the stencil of the closest cell in the foreground mesh. Once the reconstruction is complete, the usual ADER steps are applied.

This procedure has been thoroughly tested in various contexts and with different numerical approaches on evolving chimera meshes, as detailed in [3, 4, 11, 10].

4. The collocated method. Let $S \subset \Omega(t)$ be a computational subdomain composed entirely of static cells, defined as:

$$S = \bigcup_{\Omega_i \in T} \Omega_i, \quad \text{with } T \in \mathfrak{S} = \{\mathcal{T}_{\text{bg}}^{\text{S}}, \mathcal{T}_{\text{fg}}\}, \tag{4.1}$$

where $\mathcal{T}_{\text{bg}}^{\text{S}}$ is the static partition of the background mesh, and $\mathcal{T}_{\text{fg}}(t)$ is the foreground partition. The number of cells in S is denoted by M. In both partitions, the cell classification remains constant throughout the simulation. Thus, a cell that starts as internal or fringe (e.g., in the foreground) will stay internal or fringe for the entire simulation and will never become a hole. Within subdomain S, the collocated reduced approach is applied, while in the dynamic subpartition $\mathcal{T}_{\text{bg}}^{\text{D}}$ (active cells in the background), the HDM, as described in Section 3, is used.

During the offline stage, the HDM generates a database of N_s snapshots, which vary over time $t \in [0, t_{\text{fin}}]$ and parameter instances $\boldsymbol{\mu} \in \mathcal{P}$. These snapshots, denoted as $u_b = u(\mathbf{x}, t_{k(b)}; \boldsymbol{\mu}_{j(b)})$, represent the solution only after the correction step and are functions of space $\mathbf{x} \in S$ at specific times $t_{k(b)}$ and parameter values $\boldsymbol{\mu}_{j(b)}$.

The reduced basis functions φ_i (for $i = 1, \ldots, N$) are computed using the POD method to optimally represent the snapshots. The reduced subspace $V_N = \text{Span}\{\varphi_i\}_{i=1}^N$ is the rank-N subspace that minimizes the difference between the snapshots and their projections onto V_N. The projection operator $\mathcal{P}_{V_N}$ is defined as:

$$L^2(S) \ni g \mapsto \mathcal{P}_{V_N} g := \sum_{i=1}^{N} \left(\int_S g \varphi_i \, \mathrm{d}\mathbf{x} \right) \varphi_i \in V_N. \tag{4.2}$$

Here, the basis functions φ_i depend only on the spatial variable $\mathbf{x} \in S$, while the projection coefficients depend on time and parameters $(t, \boldsymbol{\mu}) \in [0, t_{\text{fin}}] \times \mathcal{P}$ [7].

The POD basis functions φ_i are obtained by solving the optimization problem:

$$\begin{aligned} \min_{\varphi_1, \ldots, \varphi_N} \quad & \sum_{b=1}^{N_s} \int_S \left(u_b - \mathcal{P}_{V_N} u_b \right)^2 \, \mathrm{d}\mathbf{x} \\ \text{subject to} \quad & \int_S \varphi_{i_1} \varphi_{i_2} \, \mathrm{d}\mathbf{x} = \delta_{i_1, i_2} \quad \forall i_1, i_2 = 1, \ldots, N, \end{aligned} \tag{4.3}$$

where the orthogonality constraint ensures that the basis functions are orthonormal.

To solve the corresponding dicrete POD problem, snapshots and basis functions are stored in matrices $U \in \mathbb{R}^{M \times N_s}$ and $\Phi \in \mathbb{R}^{M \times N}$, respectively:

$$U = \begin{bmatrix} u_1(\mathbf{x}_1) & \cdots & u_{N_s}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ u_1(\mathbf{x}_M) & \cdots & u_{N_s}(\mathbf{x}_M) \end{bmatrix} \text{ and } \Phi = \begin{bmatrix} \varphi_1(\mathbf{x}_1) & \cdots & \varphi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_M) & \cdots & \varphi_N(\mathbf{x}_M) \end{bmatrix}.$$

Here, $\mathbf{x}_i$ denotes the center of mass of cell Ω_i. The weights for quadrature are stored in a symmetric positive definite matrix $W \in \mathbb{R}^{M \times M}$, taken as $W = \operatorname{diag}(|\Omega_i|)$ to simplify the midpoint quadrature formula.

The POD problem (4.3) can then be reformulated as:

$$\begin{aligned} &\min_{\Phi} && \|U - \Phi\Phi^{\mathrm{T}} W U\|^2_{\mathrm{F}_W} \\ &\text{subject to} && \Phi^{\mathrm{T}} W \Phi = I_{N,N}, \end{aligned} \tag{4.4}$$

where the weighted Frobenius norm is defined as $\|A\|^2_{\mathrm{F}_W} := \operatorname{tr}(A^{\mathrm{T}} W A)$.

By defining $\hat{U} = (W^{\frac{1}{2}})^{\mathrm{T}} U$ using the Cholesky decomposition of W, the N-truncated Singular Value Decomposition (SVD) of $\hat{U}$ is $\hat{U} \simeq \hat{U}_N \hat{\Sigma}_N \hat{V}_N^{\mathrm{T}}$. According to the Eckart-Young theorem, the solution to (4.4) is:

$$\Phi = (W^{\frac{1}{2}})^{-\mathrm{T}} \hat{U}_N. \tag{4.5}$$

4.1. Projection onto the reduced space by hyper-reduction. The task of projecting a high-fidelity solution u onto a reduced space V_N involves computing integrals over a domain S. This projection is typically represented as:

$$\int_S u\varphi_i \,\mathrm{d}\mathbf{x} \approx \sum_{j=1}^{M} \omega_j u(\mathbf{x}_j)\varphi_i(\mathbf{x}_j) \quad \forall i = 1, \ldots, N,$$

where ω_j are the quadrature weights corresponding to points $\mathbf{x}_j$, and M is the total number of quadrature points.

To reduce the computational cost, the goal is to approximate these integrals using a smaller subset of quadrature points, $L \ll M$, with corresponding empirical weights $\tilde{\omega}_\kappa$. The integral then becomes:

$$\int_S u\varphi_i \,\mathrm{d}\mathbf{x} \approx \sum_{\kappa=1}^{L} \tilde{\omega}_\kappa u(\mathbf{x}_{m(\kappa)})\varphi_i(\mathbf{x}_{m(\kappa)}) \quad \forall i = 1, \ldots, N,$$

where $m(\kappa)$ maps each index κ to the appropriate quadrature point.

The empirical quadrature weights and points are determined to best approximate the original projection integrals for each $i = 1, \ldots, N$. To achieve this, define:

$$G_i = \begin{bmatrix} u_1|_{\mathbf{x}_1}\varphi_i(\mathbf{x}_1) & \ldots & u_1|_{\mathbf{x}_M}\varphi_i(\mathbf{x}_M) \\ \vdots & \ddots & \vdots \\ u_{N_s}|_{\mathbf{x}_1}\varphi_i(\mathbf{x}_1) & \ldots & u_{N_s}|_{\mathbf{x}_M}\varphi_i(\mathbf{x}_M) \end{bmatrix} \in \mathbb{R}^{N_s \times M},$$

and

$$\mathbf{d}_i = \begin{bmatrix} \int_S u_1 \varphi_i \, \mathrm{d}\mathbf{x} \\ \vdots \\ \int_S u_{N_s} \varphi_i \, \mathrm{d}\mathbf{x} \end{bmatrix} \in \mathbb{R}^{N_s}.$$

The objective is to find non-negative empirical weights $\tilde{\boldsymbol{\omega}}$ that minimize the number of non-zero entries, thereby promoting sparsity, while still accurately approximating the projection integrals. This problem can be formulated as:

$$\tilde{\boldsymbol{\omega}} = \arg\min_{\mathbf{w} \in \Lambda} \|\mathbf{w}\|_0,$$

where $G \in \mathbb{R}^{NN_s \times M}$, $\mathbf{d} \in \mathbb{R}^{NN_s}$, and $\Lambda = \{\mathbf{w} \in \mathbb{R}^M_+ : \|G\mathbf{w} - \mathbf{d}\|_2 \leq \epsilon \|\mathbf{d}\|_2\}$.

However, since this problem is NP-hard, it is typically approximated by solving the Non-Negative Least Squares (NNLS) problem [14, 17, 20]:

$$\tilde{\boldsymbol{\omega}} = \arg\min_{\mathbf{w} \in \mathbb{R}^M_+} \|G\mathbf{w} - \mathbf{d}\|_2^2, \tag{4.6}$$

using the Lawson-Hanson algorithm. This algorithm encourages sparsity in the solution and ensures that the solution meets the error tolerance ϵ, i.e., $\|G\mathbf{w} - \mathbf{d}\|_2 \leq \epsilon \|\mathbf{d}\|_2$.

The resulting empirical weights $\tilde{\boldsymbol{\omega}}$ are then used to select a reduced set of quadrature points, corresponding to the non-zero weights. This approach enables efficient integration in the reduced-order model while maintaining accuracy comparable to the full quadrature, especially as the tolerance ϵ decreases, leading to a more refined set of quadrature points.

4.2. Collocation and hyper-reduced prolongation operator. The projection of the high-fidelity solution vector $\mathbf{u}$, defined over the discretized subdomain S, onto a reduced space can be expressed as:

$$\hat{\mathbf{u}} = \Phi \Phi^{\mathrm{T}} W \mathbf{u},$$

where $\Phi \in \mathbb{R}^{M \times N}$ is the matrix of reduced basis vectors, and $W \in \mathbb{R}^{M \times M}$ is the weight matrix corresponding to all quadrature points. This formulation assumes that the entire set of M quadrature points is used in the projection.

In contrast, a hyper-reduced version of this projection can be defined as:

$$\hat{\mathbf{u}}_\epsilon = \Phi \tilde{\Phi}^{\mathrm{T}} \tilde{W}_\epsilon \tilde{\mathbf{u}} = \mathcal{P}_S(\tilde{\mathbf{u}}), \tag{4.7}$$

where $\mathcal{P}_S(\cdot) = \Phi \tilde{\Phi}^{\mathrm{T}} \tilde{W}_\epsilon \cdot$ is a prolongation operator that extends the solution from the reduced set of quadrature points to the entire subdomain S. Here, $\tilde{W}_\epsilon \in \mathbb{R}^{L \times L}$ contains the nonzero entries of the empirical weights, $\tilde{\Phi} \in \mathbb{R}^{L \times N}$ is the reduced basis matrix evaluated only at the points corresponding to non-zero $\tilde{\omega}_\kappa$, and $\tilde{\mathbf{u}} \in \mathbb{R}^L$ is the solution vector at the reduced quadrature points.

The hyper-reduced representation $\hat{\mathbf{u}}_\epsilon$ depends on a tolerance parameter ϵ, which controls the accuracy of the approximation. As ϵ approaches zero, the hyper-reduced projection $\hat{\mathbf{u}}_\epsilon$ converges to the full projection $\hat{\mathbf{u}}$, with $\tilde{W}_\epsilon$ approaching W, $\tilde{\Phi}$ approaching Φ, and the dimension L approaching M.

Since $\hat{\mathbf{u}}_\epsilon$ is a prolongation of the solution from the selected empirical quadrature points to the entire subdomain S, the HDM can be solved exclusively at the quadrature points. Hence, the prolongation operator can be restricted to act solely on the local stencils of the quadrature points where the HDM is solved.

The algorithms (4.1) and (4.2) for the offline and online procedures respectively, detail the key steps of this hyper-reduced collocated approach. During the online phase, the prediction step over the selected empirical cells can be parallelized due to its locality. The correction step is performed exclusively at the quadrature points and is also embarrassingly parallel. As a result, the computational cost of the scheme is $\mathcal{O}(L)$, leading to a reduction in computational complexity by a factor of $\mathcal{O}(L/M)$, M is the total number of cells in the region S.

Remark 4.1. Unlike existing literature, where hyper-reduction is typically used to reduce the computational cost of assembling matrices for the discrete projection of the non-linear terms of the model, the focus here is on hyper-reducing the discrete solution projection operator itself. The objective is to identify a subset of cells within the computational domain where the HDM needs to be solved locally. Once the solution is obtained in these selected cells, it can then be extended to local stencils or propagated to the rest of the domain. In this respect, this is a collocation method.

Algorithm 4.1 Offline stage

`/* Sub-partitions identification. */`

Require: Chimera mesh movement coherently with the snapshots U.

1: Apply (2.10) over for identifying $\mathcal{T}_{\text{bg}}^{\text{S}}$ and $\mathcal{T}_{\text{bg}}^{\text{D}}$;
2: $\mathfrak{S} \leftarrow \mathfrak{S}\{\mathcal{T}_{\text{bg}}^{\text{S}}, \mathcal{T}_{\text{fg}}\}$; ▷ `Set of static sub-partitions.`
3: **return** $\mathfrak{S}$ and $\mathcal{T}_{\text{bg}}^{\text{D}}$.

`/* Reduced Basis creation */`

Require: Snapshots U; Reduced dimension N.

4: $\hat{U} \simeq \hat{U}_N \hat{\Sigma}_N \hat{V}_N^{\text{T}}$; ▷ N-`truncated SVD for` U.
5: $\Phi = (W^{\frac{1}{2}})^{-\text{T}} \hat{U}_N$; ▷ `Reduced basis`
6: **return** Φ.

`/* Hyper-reduction step */`

Require: Snapshots U; Reduced basis Φ; hyper-reduction tolerance ϵ; static sub-partition $\mathfrak{S}$

7: $\mathbb{X} \leftarrow \{\Omega_i\}_{\Omega_i \in \mathcal{T}_{\text{fg}}}$; ▷ `Set of cells needing ADER correction solution.`
8: $\mathbb{Y} \leftarrow \emptyset$; ▷ `Set of cells only needing ADER prediction solution.`
9: **for** $T \in \mathfrak{S}$ **do**
10: Solve NNLS problem (4.6) for $\tilde{\boldsymbol{\omega}}$ with tol. ϵ; ▷ `Get empirical weights`
11: Get the L indexes k of positive entries of $\tilde{\boldsymbol{\omega}}$;
12: Get the empirical cells corresponding to the L positive entries of $\tilde{\boldsymbol{\omega}}$;
13: Update $\mathbb{X}$ w.r.t. information of line 12;
14: Find all cells sharing an edge with cells found in line 12;
15: Update $\mathbb{Y}$ w.r.t. information of line 14;
16: $\tilde{W}_\epsilon \leftarrow \text{diag}(\tilde{\boldsymbol{\omega}}|_{\tilde{\omega}>0})$ ▷ `Hyper-reduced weight matrix.`
17: Hyper-reduction $\tilde{\Phi}$ of the reduced basis Φ according to line 12;
return $\tilde{W}_\epsilon$, $\tilde{\Phi}$, $\mathbb{X}$.

5. Numerical results.

Algorithm 4.2 Online stage: collocated ADER approach

Require: $\tilde{W}_\epsilon$, Φ, $\tilde{\Phi}$, $\mathbb{X}$, $\mathbb{Y}$, t_{fin}, u_0.
Compute Δt; $t \leftarrow 0$;
while $t \leq t_{\text{fin}}$ **do**
`/* Mesh motion */`
  for any active cell **do**
    Find local map $\mathcal{X}_i$;
  Move the foreground mesh and update the hole;
`/* ADER prediction step on empirical cells */`
  for all cells in $\mathbb{X} \cup \mathbb{Y}$ **do**
    Solve local problem (3.9) as explained in Sec. 3.2;
`/* ADER correction step on empirical cells */`
  for all cells in $\mathbb{X}$ **do**
    Solve local problem (3.14) as explained in Sec. 3.3;
`/* Extension of the solution all over` Ω^{n+1} `*/`
  Extend the solution via (4.7).
`/* Extension of the solution all over` Ω^{n+1} `*/`
  Manage the chimera as explained in Sec. 3.4;
  Compute Δt and update time t.

5.1. Premilinary definitions. For a given cMOR solution u, the following space-time error is considered

$$\text{Err}(u, u_{\text{ref}}) = \sqrt{\frac{\int_0^{t_{\text{fin}}} \int_{\Omega(t)} (u_{\text{ref}} - u)^2 \, \mathrm{d}\mathbf{x}\, \mathrm{d}t}{\int_0^{t_{\text{fin}}} \int_{\Omega(t)} u_{\text{ref}}^2 \, \mathrm{d}\mathbf{x}\, \mathrm{d}t}} \cdot 100\%, \tag{5.1}$$

where u_{ref} can be either the exact solution of the PDE, if it is available, or the HDM solution according to the method introduced in Sec. 3. The accuracy of the cMOR solution depends on the number N of employed POD modes and on the tolerance ϵ of the NNLS algorithm.

Let M be the number of all active cells in the Chimera mesh. This value is averaged over all time steps since it can differ from one time step to another due to the evolution of the background mesh hole. For the sub-partitions $\mathcal{T}_{\text{bg}}^{\text{S}}$, $\mathcal{T}_{\text{fg}}$ (static) and $\mathcal{T}_{\text{bg}}^{\text{D}}$ (dynamic), the cell number are M_{bg}^{S}, M_{fg} and M_{bg}^{D}, respectively. For the static sub-partitions, the hyper-reduced number of cells effectively involved in the computation of the local residual are denoted by $\tilde{M}_{\text{bg}}^{\text{S}}$ and $\tilde{M}_{\text{fg}}$ for $\mathcal{T}_{\text{bg}}^{\text{S}}$ and $\mathcal{T}_{\text{fg}}$, respectively.

For all test cases, the frequency in time for collecting the snapshots is $5\Delta t$ for $\boldsymbol{\mu} \in \mathcal{P}_h \subset \mathcal{P}$, where $\mathcal{P}_h$ is a discrete subset of $\mathcal{P}$. Two local different reduced basis are defined in the static sub-partitions $\mathcal{T}_{\text{bg}}^{\text{S}}$ and $\mathcal{T}_{\text{fg}}$. For both regions, the number of modes of the basis is equal and is denoted by N.

When a nonzero NNLS tolerance ϵ is employed, the cMOR solution is referred as hyper-reduced. When no NNLS tolerance is used, this means that the residual is locally solved onto all active cells of the overset configuration. Thus, this solution is projected onto the reduced basis by exploiting all the theoretical Gauss quadrature points. This solution is referred as cROM. In particular, the reduced solution provides the best cMOR error.

5.2. Convergence analysis.

5.2.1. Linear advection-diffusion. The evolving computational domain is $\Omega(t) = \Omega_\square - B(\mathbf{c}(t), r)$, where $\Omega_\square$ is the channel $[-\pi, \pi] \times [-2\pi, 2\pi]$ and $B(\mathbf{c}(t), r)$ is the translating cylinder of moving center $\mathbf{c}(t) = (c_x(t), c_y(t))$ and radius r defined as

$$B(\mathbf{c}(t), r) := \left\{ \mathbf{x} \in \mathbb{R}^2 : |\mathbf{x} - \mathbf{c}(t)|^2 \leq r^2 \right\}. \tag{5.2}$$

For the test case, the radius r is 0.5. At $t = 0$ the cylinder is centered at the origin of the axis. It can only move vertically, i.e., $c_x = 0$ for all times. In particular, the vertical displacement respects the motion equation $\dot{c}_x(t) = \mu$, with $\mu \in \mathcal{P} = [-2, 2]$. In the advection-diffusion equation (2.2) $\nu = 0.05$ for the diffusion and $\mathbf{a} = [1, 1]^{\mathrm{T}}$ for the advection. Qualitatively, the problem is dominated by advection. The exact solution for this case reads

$$u_{\mathrm{ex}}(x, y, t) = \exp\left[-x^2 - (y - c_y(t))^2 + r^2\right] + \cos(x - 0.5t)\sin(y - 0.5t). \tag{5.3}$$

Final time t_{fin} is 1.5. The forcing term as well as the initial and boundary conditions are set in order to recover the exact solution (5.3).

The background mesh for $\Omega_\square$ has cells of size 0.05. The foreground mesh is polar. Its smallest and largest cells have sizes $\sim 6 \times 10{-}4$ and 0.05, respectively. Its internal circumference has radius r; the external one has radius 0.72. This chimera configuration is represented in Fig. 1, left. The average number M of active cells is 4147. The active cells in background is $M_{\mathrm{bg}} = 3321$; in foreground $M_{\mathrm{fg}} = 826$. The database of snapshots is considered for all values of parameter μ in the discrete set $\mathcal{P}_h = \{-2 + 0.5k\}_{k=0}^{8}$. Since the cylinder can only move vertically, the dynamic sub-partition $\mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$ is composed of all cells confined in the region $[-2.3, 2.3] \times [-5.3, 5.3]$, as depicted in the white region of Fig. 2, left. In this region the solution is found by the HDM. The static sub-partitions $\mathcal{T}_{\mathrm{bg}}^{\mathrm{S}}$ and $\mathcal{T}_{\mathrm{bg}}$, correspond to the regions where the ROM is defined. Figure 2 shows the first mode in both these sub-regions.

For the analysis, the first 25 modes are considered, and NNLS tolerances ϵ are set at 1×10^{-6}, 1×10^{-7}, 1×10^{-8}, and 1×10^{-9}. Figure 3 presents bar plots showing the percentage of cells used for the collocation, i.e., the percentage of cells where the HDM is employed. The percentage increases in the foreground partition $\mathcal{T}_{\mathrm{fg}}$, where the solution varies more due to the movement of the cylinder. As expected, reducing the NNLS tolerance ϵ leads to an increase in the percentage of employed cells for a fixed number of modes. This is because the NNLS algorithm, with smaller tolerances, attempts to capture all the theoretical Gaussian quadrature points.

For the same tolerance values ϵ, both in-sample and out-of-sample simulations are conducted. The in-sample simulation corresponds to a parameter $\mu = 1$ sampled in the learning phase, while the out-of-sample simulation corresponds to $\mu = 1.7$ (not included in the database). Figure 4 displays the percentage relative error as a function of the number of modes and NNLS tolerance ϵ. As expected, decreasing ϵ reduces the error, bringing it closer to the error of the ROM without hyper-reduction at a fixed number of modes. Among all the scenarios considered in Fig. 4 accounting to different values of modes and NNLS tolerance, the most demanding involves 25 modes and $\epsilon = 1 \times 10^{-9}$. In this case, the error is of about 10^{-3}%, while using fewer than 10% of the cells in $\mathcal{T}_{\mathrm{bg}}^{\mathrm{S}}$ and slightly more than 60% in the foreground. In $\mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$ we have 40% of the cells of the background (equivalent to 150 cells) that employ the HDM. Overall, the total number of cells where the HDM is solved is about 50%. The number of grid points where the HDM is applied strongly depends on the size of the region $\mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$. In this test case, the minimum number of cells on which the HDM will still be required is 1350, which corresponds to the number of cells belonging to $\mathcal{T}_{\mathrm{bg}}^{\mathrm{D}}$.

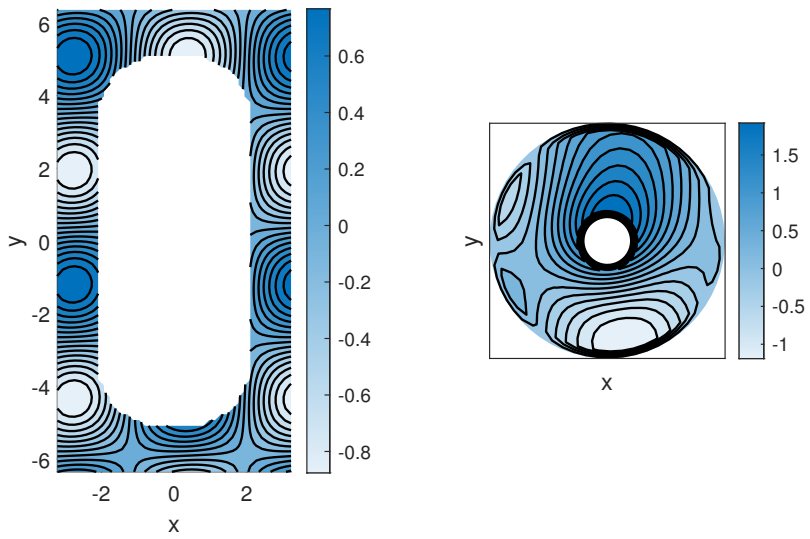

Fig. 2: First mode on the static partition $\mathcal{T}_{\text{bg}}^{\text{S}}$ of background mesh (left) and the first mode on the foreground partition $\mathcal{T}_{\text{fg}}$ (right) for the linear case. The void zone (in white) in background corresponds to the dynamic sub-partition $\mathcal{T}_{\text{bg}}^{\text{D}}$. In that region the solution is found by the HDM.

5.2.2. Nonlinear advection-diffusion. As evident from the previous test case, the larger the movement, the larger the number of cells included in the dynamic sub-partition $\mathcal{T}_{\text{bg}}^{\text{D}}$, where the solution is recovered by the HDM. For this reason, this test case considers a smaller movement as compared to the dimension of the background partition to evaluate the performance of the approach when hyper-reduction involves the quasi-totality of active cells.

In this case, the computational domain is defined as $\Omega(t) = \Omega_{\square} - B(\mathbf{c}(t), r)$, where $\Omega_{\square} = [-2\pi, 2\pi]^2$ represents the square domain and $B(\mathbf{c}(t), r)$ denotes the cylinder with radius $r = 0.5$, as defined in (5.2). Initially, the cylinder is centered at $(0, 0)$ and, as in the previous scenario, it moves only vertically according to the motion equation $\dot{c}_y = \mu$, with $\mu \in \mathcal{P} = [0.1, 2]$. For equation (2.1), the parameters are set as $\nu = 0.05$ and $\mathbf{a} = \tanh(u)[0, 1]^{\text{T}}$, making the advection component nonlinear. The solution is initially zero everywhere at $t = 0$.

Homogeneous Dirichlet boundary conditions, $u \equiv 0$, are imposed on the vertical sides of $\Omega_{\square}$, while homogeneous Neumann conditions enforce a zero normal derivative on the horizontal sides. On the remaining boundary $\partial B(\mathbf{c}(t), r)$, a Dirichlet boundary condition $u = \mu$ is applied. Consequently, the parameter μ governs both the displacement of the internal boundary and the corresponding boundary condition. Additionally, the following force term is applied: $f = \sin(x)\sin(y)$.

For this test case, the exact solution is not known *a priori*; therefore, the reference solution is obtained using the ADER approach. The simulation runs until the final time $t_{\text{fin}} = 0.5$.

The background mesh is Cartesian with cells of size 0.05. The foreground mesh is the same employed for the previous test case. The average number of active cells

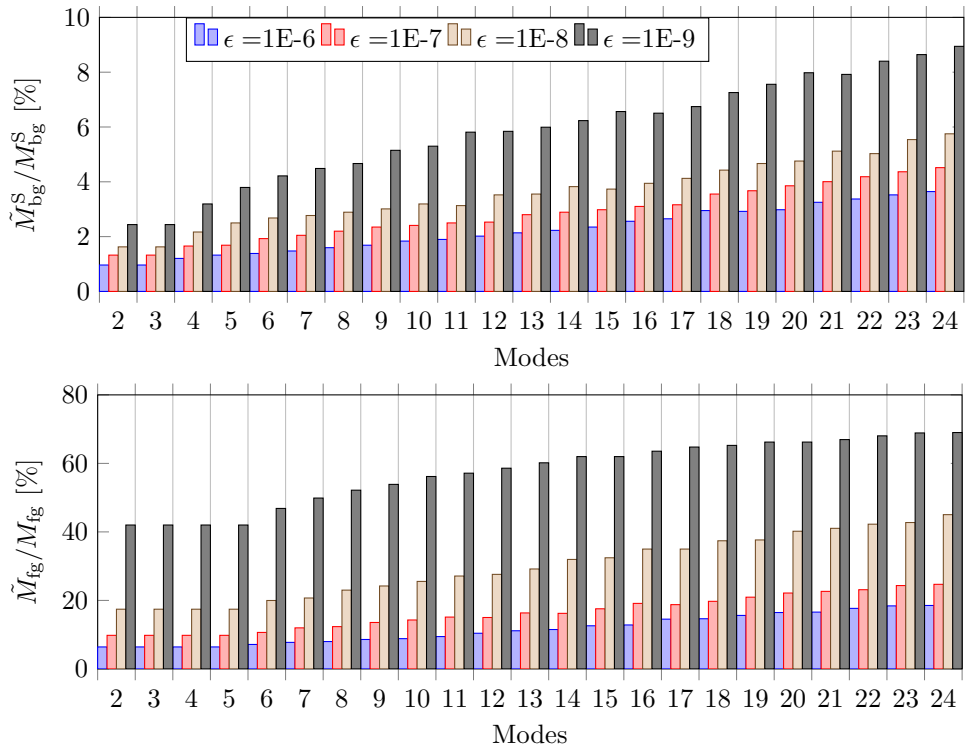

Fig. 3: Barplot of number of used cells for locally solving the residual over the total number of cells available in the sub-partition for the linear test case. Top and bottom plots refer to $\mathcal{T}^{\mathrm{S}}_{\mathrm{bg}}$ and $\mathcal{T}_{\mathrm{fg}}$, respectively. The bars are in percentage and differ w.r.t. the used NNLS tolerance ϵ.

is $M = 7387$. Of these cells, $M_{\mathrm{bg}} = 6561$ are in the background and $M_{\mathrm{fg}} = 826$ in the foreground. The database of snapshots is composed of HDM solution with parameter $\mu \in \mathcal{P}_h = \{0.1, 0.5, 1, 1.5, 2\}$. Due to the limited movement of the cylinder, the dynamic sub-partition $\mathcal{T}^{\mathrm{D}}_{\mathrm{bg}}$ is smaller than the previous test case and confined to the region $[-2.1, 2.1] \times [2.1, 3.1]$ (see the void zone in white in Fig. 5). It is composed of $M^{\mathrm{D}}_{\mathrm{bg}} = 491$ cells, corresponding to the 7.5% of available cells in background. The remaining static sub-partitions correspond to the regions where the reduced solution is computed. Figure 5 shows the first modes on $\mathcal{T}^{\mathrm{S}}_{\mathrm{bg}}$ and $\mathcal{T}_{\mathrm{fg}}$.

The first 25 modes are considered, with NNLS tolerances ϵ set at 1×10^{-6}, 1×10^{-7}, 1×10^{-8}, and 1×10^{-9}. Figure 6 presents the percentage of involved cells among all available cells for $\mathcal{T}^{\mathrm{S}}_{\mathrm{bg}}$ (top) and $\mathcal{T}_{\mathrm{fg}}$ (bottom). As expected, for a fixed number of modes, these percentages increase as the tolerance decreases.

As in the previous analysis, both an in-sample case (for $\mu = 1$) and an out-of-sample case (for $\mu = 1.7$) are computed, varying the number of modes and the NNLS tolerance. Figure 7 shows the percentage relative error with respect to the HDM solution. As anticipated, the error decreases with lower NNLS tolerances and approaches the minimum error achieved by the reduced solution. For the in-sample case, the error curve for $\epsilon = 1 \times 10^{-9}$ closely aligns with the curve of the reduced

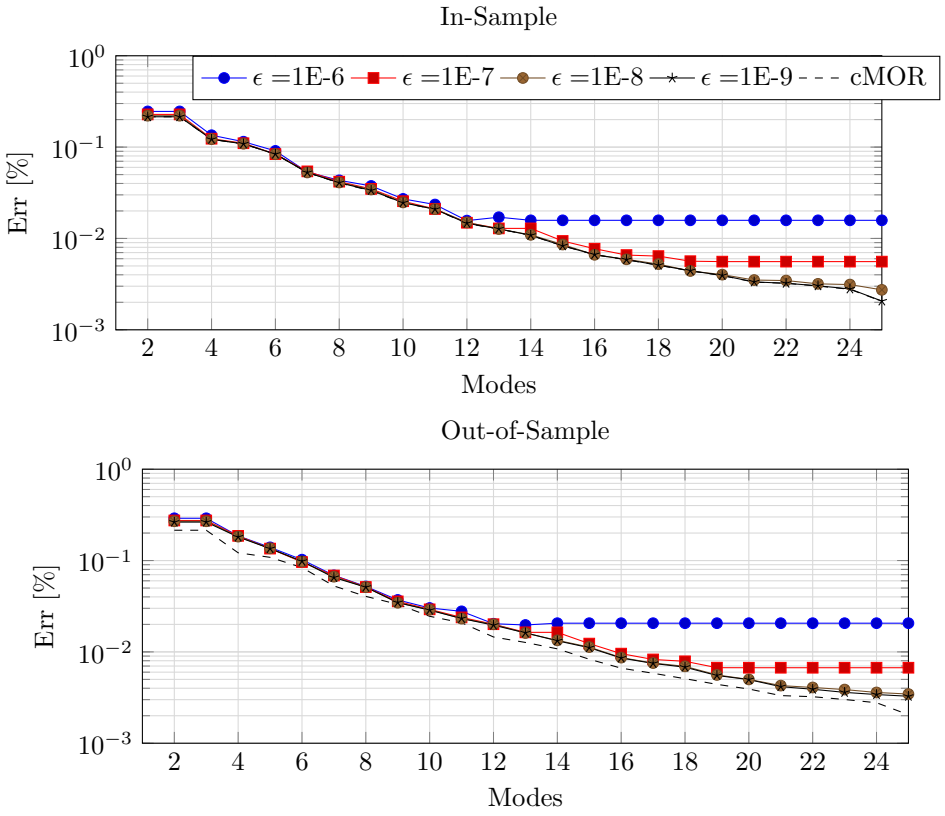

Fig. 4: Percentage relative error for in-sample (top) and out-of-sample (bottom) simulations of the linear test case. For both simulations, the different continuous curves refer to a specific value of NNLS tolerance ϵ. The dashed curve refers to a cMOR (not hyper-reduced) solution, i.e. $\epsilon = 0$ and the collocation involves all active cells. This solution provides the best cMOR error.

solution without hyper-reduction. For the out-of-sample case, these curves remain within the same order of magnitude.

Among all scenarios varying the number of modes and NNLS tolerances, in the most demanding case the HDM is used only in 7.5% (equivalent to 492 cells) of the active cells in the background. In other regions, just 5% of the background cells and approximately 25% of the foreground cells are effectively employed. Despite the cMOR approach involving only 14% of all active cells, the precision loss in both space and time is on the order of 1×10^{-3}% compared to the HDM solution.

5.3. Pure Advection of a Compactly Supported Bump. This subsection aims to evaluate the performance of the proposed cMOR using a chimera grid in comparison with a classical pMOR approach. To do this, we consider a purely advective and linear PDE. Specifically, the advection-diffusion term in (2.2) is defined by $\nu = 0$ and $\mathbf{a} = \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the unknown parameter. The computational domain is

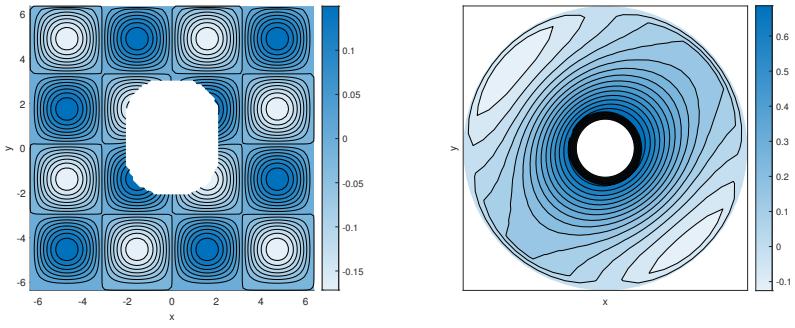

Fig. 5: First mode on the static partition $\mathcal{T}_{\text{bg}}^{\text{S}}$ of background mesh (left) and the first mode on the foreground partition $\mathcal{T}_{\text{fg}}$ (right) for the linear case. The void zone (in white) in background corresponds to the dynamic sub-partition $\mathcal{T}_{\text{bg}}^{\text{D}}$. In that region the solution is found by the HDM.

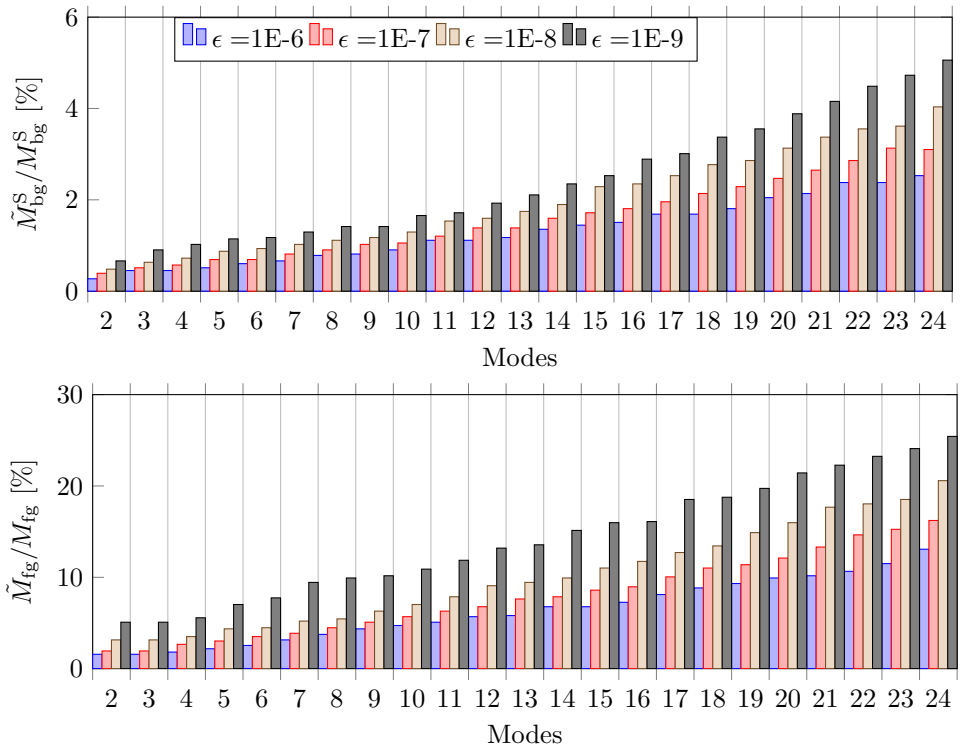

Fig. 6: Barplot of number of used cells for locally solving the residual over the total number of cells available in the sub-partition for the nonlinear test case. Top and bottom plots refer to $\mathcal{T}_{\text{bg}}^{\text{S}}$ and $\mathcal{T}_{\text{fg}}$, respectively. The bars are in percentage and differ w.r.t. the used NNLS tolerance ϵ.

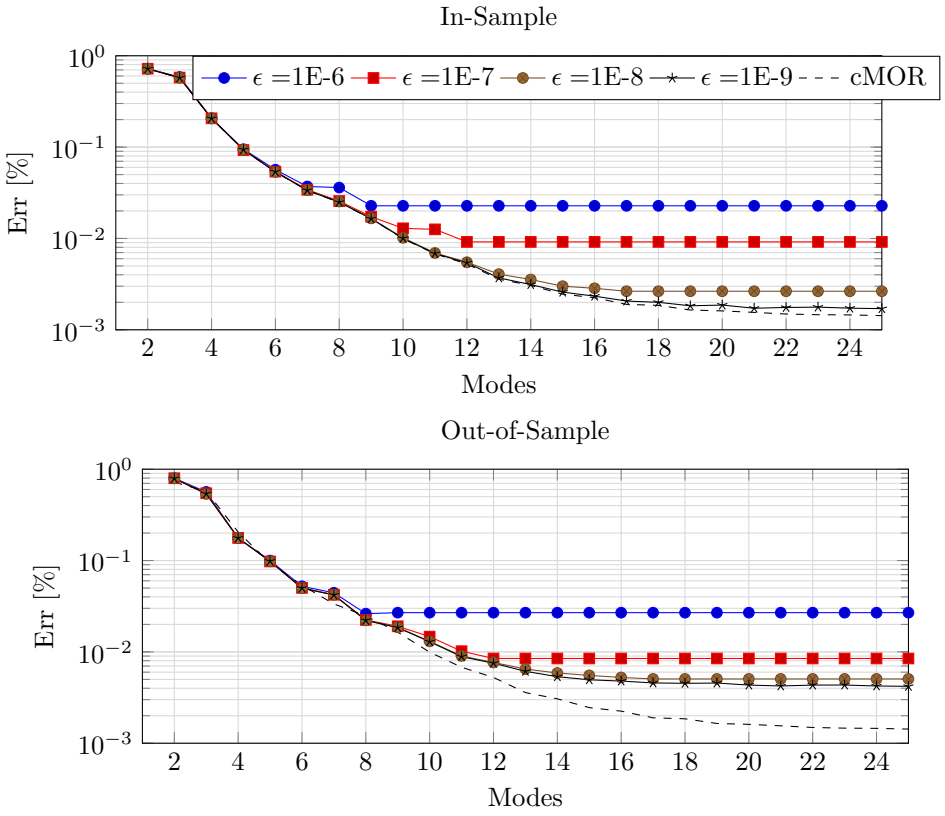

Fig. 7: Percentage relative error for in-sample (top) and out-of-sample (bottom) simulations of the linear test case. For both simulations, the different continuous curves refer to a specific value of NNLS tolerance ϵ. The dashed curve refers to a cMOR (not hyper-reduced) solution, i.e. $\epsilon = 0$ and the collocation involves all active cells. This solution provides the best cMOR error.

$\Omega = [-2\pi, 2\pi]^2$ with periodic boundary conditions. The initial condition is given by[2]

$$u(\mathbf{x}, t = 0) = u_0(\mathbf{x}) = \exp\left(-\frac{1}{|1 - \mathbf{x}|^2}\right) \chi_{\{|x|\leq 1\}}(\mathbf{x}). \tag{5.4}$$

The exact solution for this problem is $u(\mathbf{x}, t; \boldsymbol{\mu}) = u_0(\mathbf{x} - \boldsymbol{\mu} t)$, which represents the advection of the initial condition (5.4) with velocity $\boldsymbol{\mu}$.

We consider two discretization approaches for this test case. The first approach applies the proposed cMOR with a chimera configuration, where the foreground mesh is initially centered on the bump and moves along with it (i.e., with the same velocity). The foreground mesh is designed to completely enclose the support of the bump. Both the background and foreground meshes are Cartesian, with cell sizes of 0.05. Figure 8 illustrates a particular configuration of the bump along with the employed background and foreground meshes. In this setup, the NNLS tolerance ϵ is set to 1×10^{-8}.

[2]The indicator function $\chi_E(\mathbf{x})$ equals 1 if $\mathbf{x}$ is in $E \subseteq \mathbb{R}^2$, and 0 otherwise.

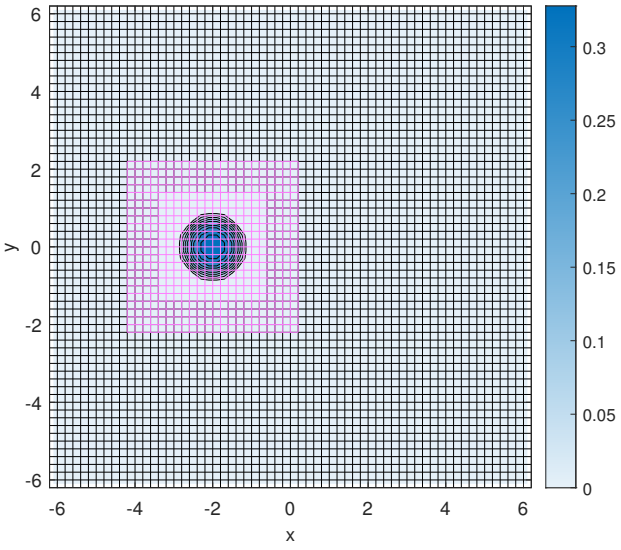

Fig. 8: Particular configuration of a pure advection of a bump. In color is the foreground mesh that follows the displacement of the bump.

The second approach uses a single mesh, identical to the background mesh of the first discretization. Here, the solution is first computed using the HDM and then projected onto the reduced basis obtained during the offline step.

In this test, the bump is assumed to move alternately along horizontal and vertical lines. Let μ_x and μ_y denote the x- and y-components of the parameter $\boldsymbol{\mu}$, which can take values in the set $[-1, 1]$. For both discretizations, the snapshot database is defined by instances of $\boldsymbol{\mu}$ in $\mathcal{P}_h = \{(0, \pm 1), (\pm 1, 0)\}$ up to the final time $t_{\text{fin}} = 1$. For both discretizations, an in-sample simulation with $\boldsymbol{\mu} = (0, 1)$ and an out-of-sample simulation with $\boldsymbol{\mu} = (0, 0.7)$ are considered.

In the cMOR approach, the relative errors in space and time are $\mathcal{O}(10^{-15})$, with just one mode. Moreover, as expected, no cells in the static background sub-partition are considered (since the solution must be zero in the background), and only one cell in the foreground is selected by the hyper-reduced approach to solve the entire solution. In contrast, in the single-mesh case using 25 modes and all active cells, the relative space-time error is on the order of $\mathcal{O}(1)$, even for the in-sample simulation.

This difference is also evident when examining the singular values of the two approaches (see Fig. 9), which provide a heuristic bound on the Kolmogorov N-width. A faster decay of singular values indicates a smaller Kolmogorov N-width. From an energetic perspective, only the first mode is significant for the cMOR approximation, corresponding to the bump itself, scaled to have a unitary L^2-norm, which is sufficient to capture the entire dynamics. Of course, the reason is that due to the moving foreground mesh, in its relative frame, the bump appears stationary. On the other hand, the decay of singular values for the one-block pMOR approach is much slower, and the orthogonal directions of the bump movement collected in the database are not adequately captured by the pMOR approach.

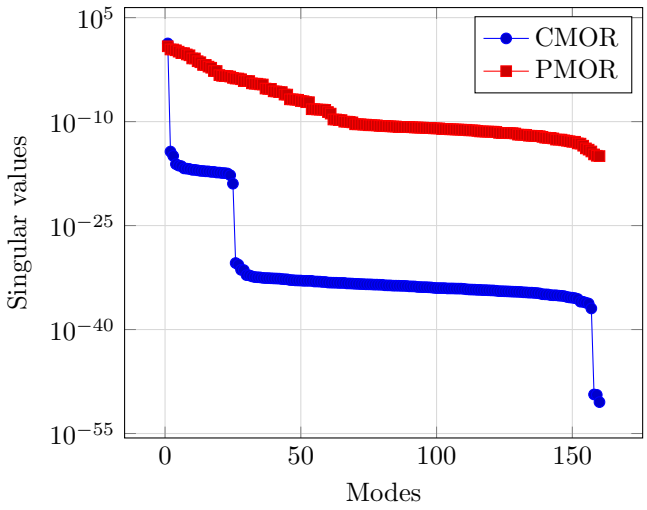

Fig. 9: Singular values for cMOR and pMOR discretizations for the pure advection of a bump.

5.4. Time-dependent geometrical parameter. This section is closed testing the novel cMOR method with time-dependent parameters $\mu(t)$. The analysis is conducted on both the linear and nonlinear test cases introduced in Sections 5.2.1 and 5.2.2, respectively. For these cases, the parameters follow the time-dependent laws:

$$\mu(t) = 2\cos(4t) \text{ (linear)} \quad \text{and} \quad \mu(t) = t^2 + 0.1 \text{ (nonlinear)}. \tag{5.5}$$

These time functions ensure that the parameters remain within their respective intervals of existence, $\mathcal{P}$, which are $[-2, 2]$ for the linear case and $[0.1, 2]$ for the nonlinear case. As a result, the same databases, corresponding POD basis, and the subset of employed cells for hyper-reduction can still be utilized.

Figure 10 shows the space-time errors as a function of the number of modes. For the cMOR solution, the NNLS tolerance is set to 1×10^{-8}. The error is compared against the best approximation error obtained by simply projecting the high-fidelity solution using all active cells over RB. In contrast to cases where the parameters are steady over time, the projection error stabilizes at around $10^{-2}\%$.

For the linear case, the cMOR error curve is nearly indistinguishable from the pMOR error curve, despite the cMOR approach utilizing less than 6% of active cells in the static background sub-partition and around 40% of cells in the foreground, even in the most demanding scenario (i.e., using 25 modes). Similarly, in the nonlinear test case, the cMOR and pMOR error curves are comparable. Here, the continuously evolving parameter affects both the geometry and the boundary conditions on the cylinder. When 25 modes are used, the error relative to the high-fidelity solution is less than 0.04%, with fewer than 4% of cells in $\mathcal{T}_{\text{bg}}^{\text{S}}$ and 20% of cells in the foreground $\mathcal{T}_{\text{fg}}$ effectively employed.

6. Conclusions and Future Perspectives. We introduced a novel strategy to reduce the computational costs associated with solving advection-diffusion PDEs. This strategy employs a collocation method where the HDM is solved on a small

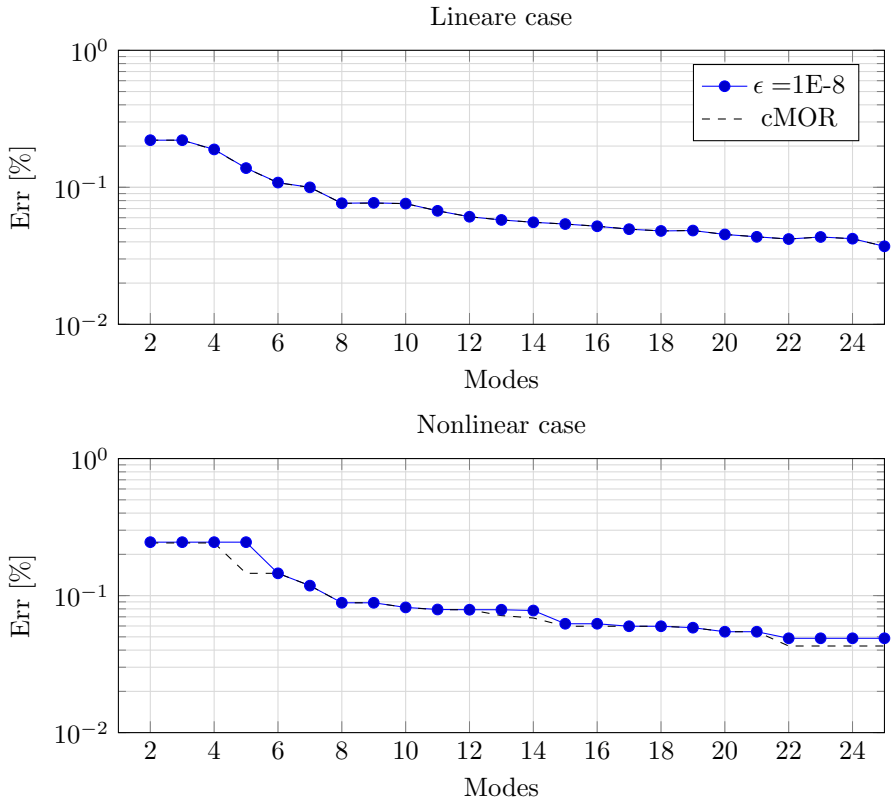

Fig. 10: Percentage relative error for linear (top) and nonlinear (bottom) simulations with time-dependent parameters. For both simulations, the continuous curve refers to a NNLS tolerance $\epsilon = 1 \times 10^{-8}$. The dashed curve refers to a cMOR (not hyper-reduced) solution, i.e. $\epsilon = 0$ and the collocation involves all active cells. This solution provides the best cMOR error.

subset of the original grid cells. The solution is then extended to the entire domain using a global reduced basis. In this way, cMOR functions as a consistent approach that effectively reduces the computational burden of advanced integration schemes, such as the ADER method, on unsteady Chimera meshes.

Our results demonstrated that combining cMOR with ADER on Chimera meshes effectively addresses the Kolmogorov barrier for advection-dominated phenomena. This is achieved by defining appropriate local reduced bases with respect to the moving Chimera meshes.

Ongoing research is focused on applying this approach to fluid-structure interaction problems in both incompressible and compressible regimes. Additionally, further theoretical work is being conducted to analyze the error introduced by the cMOR strategy and to explore its application to more traditional high-order finite-volume methods.

Acknowledgments. The authors acknowledge the support by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie

Actions, grant agreement 872442 (ARIA). This work has been co-funded by ONERA and Region Nouvelle-Aquitaine under contract number 225320 (Chaire PROVE).

REFERENCES

[1] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat, *A method for interpolating on manifolds structural dynamics reduced-order models*, International journal for numerical methods in engineering, 80 (2009), pp. 1241–1258.

[2] M. Bergmann, C.-H. Bruneau, and A. Iollo, *Enablers for robust pod models*, Journal of Computational Physics, 228 (2009), pp. 516–538.

[3] M. Bergmann, M. G. Carlino, and A. Iollo, *Second order ader scheme for unsteady advection-diffusion on moving overset grids with a compact transmission condition*, SIAM Journal on Scientific Computing, 44 (2022), pp. A524–A553.

[4] M. Bergmann, M. G. Carlino, A. Iollo, and H. Telib, *Ader scheme for incompressible navier-stokes equations on overset grids with a compact transmission condition*, Journal of Computational Physics, 467 (2022), p. 111414.

[5] G. Berkooz, P. Holmes, and J. L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual review of fluid mechanics, 25 (1993), pp. 539–575.

[6] 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 (2018), pp. 545–570.

[7] T. Bui-Thanh, M. Damodaran, and K. Willcox, *Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition*, AIAA journal, 42 (2004), pp. 1505–1516.

[8] K. Carlberg, C. Bou-Mosleh, and C. Farhat, *Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations*, International Journal for numerical methods in engineering, 86 (2011), pp. 155–181.

[9] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, *The gnat method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows*, Journal of Computational Physics, 242 (2013), pp. 623–647.

[10] M. G. Carlino, *ADER scheme on Overset Grids with Compact Transmission and Hyper-reduction: Application to Incompressible Navier-Stokes Equations*, PhD thesis, Université de Bordeaux, 2021.

[11] M. G. Carlino and W. Boscheri, *Arbitrary-lagrangian-eulerian finite volume imex schemes for the incompressible navier-stokes equations on evolving chimera meshes*, Journal of Computational Physics, 501 (2024), p. 112764.

[12] C. Cattaneo, *Sulla conduzione del calore*, Atti Sem. Mat. Fis. Univ. Modena, 3 (1948), pp. 83–101.

[13] C. Cattaneo, *A form of heat-conduction equations which eliminates the paradox of instantaneous propagation*, Comptes rendus, 247 (1958), p. 431.

[14] T. Chapman, P. Avery, P. Collins, and C. Farhat, *Accelerated mesh sampling for the hyper reduction of nonlinear computational models*, International Journal for Numerical Methods in Engineering, 109 (2017), pp. 1623–1654.

[15] S. Cucchiara, A. Iollo, T. Taddei, and H. Telib, *Model order reduction by convex displacement interpolation*, Journal of Computational Physics, 514 (2024), p. 113230.

[16] V. Ehrlacher, D. Lombardi, O. Mula, and F.-X. Vialard, *Nonlinear model reduction on metric spaces. application to one-dimensional conservative pdes in wasserstein spaces*, ESAIM: Mathematical Modelling and Numerical Analysis, 54 (2020), pp. 2159–2197.

[17] C. Farhat, P. Avery, T. Chapman, and J. Cortial, *Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency*, International Journal for Numerical Methods in Engineering, 98 (2014), pp. 625–662.

[18] C. Farhat, T. Chapman, and P. Avery, *Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models*, International journal for numerical methods in engineering, 102 (2015), pp. 1077–1110.

[19] P. Fenyes, *Multidisciplinary design and optimization of automotive structures-a parametric approach*, in 8th Symposium on Multidisciplinary Analysis and Optimization, 2000, p. 4706.

[20] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh, *Mesh sampling and weighting for the hyperreduction of nonlinear petrov–galerkin reduced-order models with local reduced-order bases*, International Journal for Numerical Methods in Engineering, 122 (2021), pp. 1846–1874.

[21] J. S. Hesthaven, C. Pagliantini, and G. Rozza, *Reduced basis methods for time-dependent problems*, Acta Numerica, 31 (2022), pp. 265–345.
[22] J. S. Hesthaven, G. Rozza, B. Stamm, J. S. Hesthaven, G. Rozza, and B. Stamm, *Parametrized differential equations*, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, (2016), pp. 15–25.
[23] A. Hidalgo and M. Dumbser, *Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations*, Journal of Scientific Computing, 48 (2011), pp. 173–189.
[24] A. Iollo, A. Dervieux, J.-A. Désidéri, and S. Lanteri, *Two stable pod-based approximations to the navier–stokes equations*, Computing and visualization in science, 3 (2000), pp. 61–66.
[25] A. Iollo, S. Lanteri, and J.-A. Désidéri, *Stability properties of pod–galerkin approximations for the compressible navier–stokes equations*, Theoretical and Computational Fluid Dynamics, 13 (2000), pp. 377–396.
[26] A. Iollo and D. Lombardi, *Advection modes by optimal mass transfer*, Physical Review E, 89 (2014), p. 022923.
[27] A. Iollo and T. Taddei, *Mapping of coherent structures in parameterized flows by learning optimal transportation with gaussian models*, Journal of Computational Physics, 471 (2022), p. 111671.
[28] J. L. Lumley, *The structure of inhomogeneous turbulent flows*, in Atmospheric Turbulence and Radio Wave Propagation, A. M. Yaglom and V. I. Tatarsky, eds., Berlin, Heidelberg, 1967, Springer, pp. 166–178, https://doi.org/10.1007/978-1-4612-1092-4_18.
[29] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. Pau, *A general, multipurpose interpolation procedure: the magic points*, (2007).
[30] R. L. Meakin, *Composite overset structured grids*, Handbook of Grid Generation, (1999), pp. 1–20.
[31] N. C. Nguyen and J. Peraire, *An efficient reduced-order modeling approach for non-linear parametrized partial differential equations*, International Journal for Numerical Methods in Engineering, 76 (2008), pp. 27–55.
[32] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday, *Overcoming slowly decaying kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems*, arXiv preprint arXiv:1911.06598, (2019).
[33] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz, *Localized discrete empirical interpolation method*, SIAM Journal on Scientific Computing, 36 (2014), pp. A168–A192.
[34] A. Pinkus, *N-widths in Approximation Theory*, vol. 7, Springer Science & Business Media, 2012.
[35] A. Quarteroni and G. Rozza, *Numerical solution of parametrized navier–stokes equations by reduced basis methods*, Numerical methods for partial differential equations: an international journal, 23 (2007), pp. 923–948.
[36] D. Ryckelynck, *A priori hyperreduction method: an adaptive approach*, Journal of Computational Physics, 202 (2005), pp. 346–366, https://doi.org/https://doi.org/10.1016/j.jcp.2004.07.015, https://www.sciencedirect.com/science/article/pii/S002199910400289X.
[37] S. Salsa, *Partial differential equations in action*, vol. 1, Springer, 2015.
[38] A. Sharma, S. Ananthan, J. Sitaraman, S. Thomas, and M. A. Sprague, *Overset meshes for incompressible flows: On preserving accuracy of underlying discretizations*, Journal of Computational Physics, 428 (2021), p. 109987.
[39] T. Taddei, *A registration method for model order reduction: data compression and geometry reduction*, SIAM Journal on Scientific Computing, 42 (2020), pp. A997–A1027.
[40] E. F. Toro and G. I. Montecinos, *Advection-diffusion-reaction equations: hyperbolization and high-order ader discretizations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2423–A2457.
[41] M. Yano and A. T. Patera, *An lp empirical quadrature procedure for reduced basis treatment of parametrized nonlinear pdes*, Computer Methods in Applied Mechanics and Engineering, 344 (2019), pp. 1104–1123.
[42] M. J. Zahr and C. Farhat, *Progressive construction of a parametric reduced-order model for pde-constrained optimization*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1111–1135.