

RESEARCH ARTICLE

WILEY

Efficient geometrical parametrization for finite-volume-based reduced order methods

Giovanni Stabile¹ | Matteo Zancanaro | Gianluigi Rozza²

Mathematics Area, mathLab, SISSA,
International School for Advanced
Studies, Trieste, Italy

Correspondence

Giovanni Stabile, Mathematics Area,
mathLab, SISSA, International School for
Advanced Studies, 34136 Trieste, Italy.
Email: gstable@sissa.it

Funding information

H2020 European Research Council,
Consolidator Grant project Advanced
Reduced Order; Istituto Nazionale di Alta
Matematica “Francesco Severi”,
INdAM-GNCS; Ministero dell’Istruzione,
dell’Università e della Ricerca, MIUR
FARE-X-AROMA-CFD

Summary

In this work, we present an approach for the efficient treatment of parametrized geometries in the context of proper orthogonal decomposition (POD)-Galerkin reduced order methods based on finite-volume full order approximations. On the contrary to what is normally done in the framework of finite-element reduced order methods, different geometries are not mapped to a common reference domain: the method relies on basis functions defined on an average deformed configuration and makes use of the discrete empirical interpolation method to handle together nonaffinity of the parametrization and nonlinearities. In the first numerical example, different mesh motion strategies, based on a Laplacian smoothing technique and on a radial basis function approach, are analyzed and compared on a heat transfer problem. Particular attention is devoted to the role of the nonorthogonal correction. In the second numerical example, the methodology is tested on a geometrically parametrized incompressible Navier-Stokes problem. In this case, the reduced order model is constructed following the same segregated approach used at the full order level.

KEYWORDS

finite-volume method, geometrical parametrization, incompressible flows, proper orthogonal decomposition-Galerkin, reduced order methods

1 | INTRODUCTION AND MOTIVATION

In several cases, there is the need to repeatedly solve partial differential equations (PDEs) on different parametric domains. Such situations occur, for example, in the case of geometric design optimization where different parametric domains are in need of being tested in order to determine the configuration which is maximizing or minimizing a certain quantity of interest.

Finite volume (FV) approximations, due to the fact that they enforce conservation at local level, are particularly suited to be employed for the discretization of systems of PDEs generated by conservation laws. Conservation laws model a large range of different physical problems such as fluid dynamics, heat transfer, solid mechanics, and so on. In particular, the FV method is particularly used and convenient to model hyperbolic conservation laws that are common in fluid dynamics problems.

For this reason, the FV method is widespread in many different engineering fields such as mechanical, aerospace, civil, nuclear, and so on, and nonengineering ones such as meteorology, environmental marine sciences, medicine, and so on. As also for other full order discretization techniques (finite elements, finite

differences, spectral elements) in several cases (eg, uncertainty quantification, inverse problems, optimization, real-time control, etc.), the numerical simulation of the governing equations using standard techniques becomes not affordable. This fact is particularly evident for the case of geometrical optimization¹ on which this work is focused. A viable way to reduce the computational burden is given by the reduced order modeling methodology; for a comprehensive review of the reduced order modeling methodology the interested reader may see References 2–4.

Reduced order models (ROMs) and in particular projection-based ROMs, which are the focus of this work, have been applied to a large variety of different mathematical problems based on linear elliptic equations,⁵ linear parabolic equations,⁶ and also nonlinear problems.^{7,8}

Some of the ingredients and the issues encountered to generate a projection-based ROM are common to all the full order discretization technique but some aspects require particular attention if we change full order discretization. The usage of the reduced basis (RB) method for shape optimization in a finite-element setting has been widely studied and the state of the art counts already several scientific contributions.^{9–12}

Summarizing pioneering and more recent works dealing with the coupling of the FV methods with projection-based reduced order methods, it is worth mentioning References 13 and 14 that present one of the first contributions dealing with FV and the RB method and Reference 15 dealing with the empirical interpolation method in a FV setting. For more recent contributions, it is worth mentioning References 16 and 17 where pressure stabilization techniques normally employed in a finite element setting have been adapted to an FV one and Reference 18 that proposes a new structure preserving strategy specifically tailored for a FV setting, and References 19–21 that extend FV-POD-Galerkin ROMs to a turbulence setting and to thermal mixing problems. Focusing on works that are specifically dealing with geometrically parametrized problems in an FV setting, we mention References 22 and 23 that focus the attention on inviscid Euler equations, Reference 24 focusing on turbulent compressible Navier-Stokes equations, and Reference 25 that deals with PDE-constrained optimization problems.

The main focus of this work is the development of computational strategies for the geometrical parametrization of reduced order methods starting from an FV full order discretization. As highlighted in Section 3.2, this issue, respect to a finite element setting, requires particular attention. The strategy usually employed to deal with geometrical parametrization with ROMs is to write all the equations in a reference domain making use of a map $\mathcal{M}(\boldsymbol{\mu}) : \Omega(\boldsymbol{\mu}) \rightarrow \tilde{\Omega}$ which transforms the equations from the physical domain $\Omega(\boldsymbol{\mu})$ to the reference one $\tilde{\Omega}$. This approach has been adopted successfully in several cases starting from finite element discretization. The affinity of the differential operators can be reconstructed by domain decomposition and piecewise affine reference mappings^{5,26,27} or by the usage of empirical interpolation techniques.^{28–30} The first approach cannot be easily transferred to an FV setting (see Section 3.2), while the second approach has been used in some of the previously mentioned works dealing with geometrical parametrization in a FV environment. In the present work, all the possible domain configurations are solved in the physical domain and the RB functions are computed by means of an inner product computed on an average deformed configuration similarly to what done in Reference 22.

The online-offline splitting is guaranteed by the usage of a discrete empirical interpolation procedure at both matrix and vector levels. The resolution of the equations is carried out on the physical domain where the mesh is deformed by the usage of an arbitrary lagrangian Eulerian framework.³¹

The manuscript is organized as follows: Section 2 briefly introduces the parametrized mathematical problem, and Section 3 presents the full order FV formulation and its main features with a particular attention on the different mesh motion strategies. In Section 4, we describe the methodologies used to construct the ROM with emphasis on the empirical interpolation techniques used to ensure an efficient offline-online splitting. In Section 5, we test the proposed methodologies on a geometrically parametrized heat transfer problem and the attention here is focused on the different mesh motion strategy. In Section 6, the methodology is tested instead on a steady incompressible Navier-Stokes setting. Finally, in Section 7, some conclusions and perspectives for future works are given.

2 | THE PARAMETRIZED MATHEMATICAL PROBLEM

The interest of this work is on generic stationary linear or nonlinear PDEs describing conservation laws whose domain is parametrized by a parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^k$, where \mathcal{P} is a k -dimensional parameter space. Given a generic scalar

or vectorial quantity of interests $u(x, \mu)$, the abstract problem reads:

$$\begin{cases} \mathcal{N}(x, u(\mu)) = 0 & \text{in } \Omega(\mu), \\ u(x, \mu) = g_D(x, \mu) & \text{on } \Gamma_D(\mu), \\ u_n(x, \mu) = g_N(x, \mu) & \text{on } \Gamma_N(\mu), \end{cases} \quad (1)$$

where $\mathcal{N}(x, \mu)$ is a generic linear or nonlinear differential operator, $\Omega(\mu) \subset \mathbb{R}^d$ is a bounded parametrized domain, $\Gamma_D(\mu)$ and $\Gamma_N(\mu)$ are its parametrized Dirichlet and Neumann boundaries, respectively, with the relative boundary conditions $g_D(x, \mu)$ and $g_N(x, \mu)$, while the subscript \square_n indicates the normal derivative. RB methods are based on the assumption that the response of such parametrized PDE is approximated by a reduced number of dominant modes, and therefore, it is possible to project the original equations onto a low-dimensional subspace in order to reduced the computational cost. In this work, this low-dimensional space is obtained using a proper orthogonal decomposition (POD) approach. More details on this aspect are reported in Section 4.

3 | THE FV DISCRETIZATION AND THE GEOMETRICAL PARAMETRIZATION

In this section, we recall the main features of the full order discretization method used to discretize the mathematical problem and the techniques used to deal with the parametrization of the geometry. We stress the attention on key aspects of this full order discretization technique that will play a crucial role in order to ensure an accurate and efficient model reduction strategy for parametrized geometries.

The abstract problem of Equation (1) is discretized using an FV approach. The procedure used here is recalled for a simple heat transfer problem but the procedure is general and can be easily extended also to other type of equations. On the contrary respect to projection-based (finite element method, spectral element method) full order discretization techniques, which start from a weak formulation of the governing equations, the starting point of the FV method is the integral form of the equations written in conservative form. For a simple steady heat transfer problem on a parametrized domain, which is described by the Poisson equation, the formulation reads:

$$\begin{cases} \int_{\tilde{\Omega}(\mu)} \text{div}(\alpha_\theta \nabla \theta) d\omega = \int_{\tilde{\Omega}(\mu)} f d\omega & \forall \tilde{\Omega}(\mu) \in \Omega(\mu), \\ \theta(x, \mu) = \theta_D(x, \mu) & \text{on } \Gamma_D(\mu), \\ \theta_n(x, \mu) = \theta_N(\mu, x) & \text{on } \Gamma_N(\mu), \end{cases} \quad (2)$$

where θ is the temperature field, α_θ is the thermal diffusivity, f is a generic forcing term, $\tilde{\Omega}(\mu)$ is an arbitrary control volume inside $\Omega(\mu)$, θ_D and θ_N are the Dirichlet and Neumann boundary conditions for the temperature field, while the other terms assume the same meaning of those reported in the abstract problem formulation (Equation (1)).

3.1 | The FV discretization

In order to solve the problem of Equation (2), the domain $\Omega(\mu)$ is subdivided into a tessellation $\mathcal{T}(\mu) = \{\Omega_i(\mu)\}_{i=1}^{N_h}$ composed by a set of N_h convex and nonoverlapping polygonals in two-dimensional (2D) or polyhedrons in three-dimensional (3D) (FVs) such that $\Omega(\mu) = \cup_{i=1}^{N_h} \Omega_i(\mu)$ and $\Omega_i(\mu) \cap \Omega_j(\mu) = \emptyset$ for $i \neq j$. Once the tessellation is set, the divergence term is transformed into a surface integral using Gauss's theorem and the integral terms of Equation (2) can be discretized as follows:

$$\begin{aligned} \sum_{i=1}^{N_h} \int_{\Omega_i(\mu)} \text{div}(\alpha_\theta \nabla \theta) d\omega &= \sum_{i=1}^{N_h} \int_{\partial\Omega_i(\mu)} \mathbf{n} \cdot (\alpha_\theta \nabla \theta) ds = \sum_{i=1}^{N_h} \sum_{f=1}^{N_f} \alpha_{\theta_{if}} \mathbf{S}_{if} \cdot (\nabla \theta)_{if}, \\ \int_{\tilde{\Omega}(\mu)} f d\omega &= \sum_{i=1}^{N_h} \int_{\Omega_i(\mu)} f d\omega = \sum_{i=1}^{N_h} f_i V_i. \end{aligned} \quad (3)$$

In the above expressions, the subscript \square_i indicates the value at the center of the icell, the subscript \square_{if} indicates the value of the variables at the center of the f -face relative to the i -cell, \mathbf{n} represents the unit normal vector respect

to the control volume surface, V are the cell volumes, and \mathbf{S} are the surface vectors. One of the key points of the FV method, especially for advection dominated problem, is the evaluation of the conservative variables at the center of the faces starting from their values at the center of the cells. Several numerical schemes are, in fact, possible such as central differencing, first-order upwind, second-order upwind, Rusanov, MUSCL, and so on. For the particular case, the term $\mathbf{S}_{if} \cdot (\nabla \theta)_{if}$, which represents the flux of the gradient of the temperature field passing through the face f , can be evaluated using both an explicit or implicit scheme with or without orthogonal correction. In case of orthogonal meshes, the flux relative to the gradient term can be estimated as:

$$\mathbf{S}_{if} \cdot (\nabla \theta)_{if} = \mathbf{S}_{if} \cdot \frac{\theta_i - \theta_j}{|\mathbf{d}|}, \quad (4)$$

where by orthogonal meshes we mean those cases where the surface vector \mathbf{S}_{if} (see Figure 1, relative to the face which is dividing two cells, is parallel to the distance vector \mathbf{d} which is connecting the cell centers. Unfortunately, orthogonal meshes, especially considering complex geometries, are not common and in many cases even not realizable. In the present case, even starting from a fully orthogonal grid, after the mesh motion, we will end up with a nonorthogonal grid. For this reason, in order to have accurate results, Equation (4) needs to take into account also a nonorthogonal correction. In the present work, following what presented in Reference 32, when a nonorthogonal correction is considered, the product $\mathbf{S}_{if} \cdot (\nabla \theta)_{if}$ is split into two parts:

$$\mathbf{S}_{if} \cdot (\nabla \theta)_{if} = |\Delta_{if}| \cdot \frac{\theta_i - \theta_j}{|\mathbf{d}|} + \mathbf{k}_{if} \cdot (\nabla \theta)_{if}, \quad (5)$$

where $\mathbf{S}_{if} = \Delta_{if} + \mathbf{k}_{if}$ is decomposed into two vectors with Δ_{if} parallel to \mathbf{d} while \mathbf{k}_{if} can be selected using different approaches such as minimum correction approach, orthogonal correction approach, over-relaxed approach, and so on. We refer to Reference 32 for more details. Therefore, the total contribution is split into the first orthogonal contribution and into a non orthogonal one. The term $(\nabla \theta)_{if}$ that represents the value of the gradient at the center of the face is calculated explicitly via interpolation after the calculation of the value of the gradient at the centers of the neighboring cells $(\nabla \theta)_i$ and $(\nabla \theta)_j$. Then the value at the center of the face $(\nabla \theta)_{if}$ is evaluated using an interpolation scheme:

$$(\nabla \theta)_{if} = w_x (\nabla \theta)_i + (1 - w_x) (\nabla \theta)_j, \quad (6)$$

where w_x is a proper weight that depends on the employed interpolation scheme. Moreover, in some cases, in order to ensure that the discretized diffusion term preserves its bounded behavior also after the correction, the nonorthogonal contribution can be limited. Therefore, when using a correction for the computation of the flux of the gradient, one needs to resolve the discrete equations using an iterative scheme because the term $(\nabla \theta)_{if}$ is based on the previous converged value of the conservative variable θ . This additional term acts like a source term in the discretized equations. It is beyond the scope of this work to go into the details concerning the different methodologies used to deal with nonorthogonal meshes but it was important to briefly recall them since they play an important role also at the reduced order level. As we will see in the numerical examples, different nonorthogonal correction approaches will lead to different dimensions of the computational stencil that we need to store during the discrete empirical interpolation method (D-EIM) procedure and will also lead to different values of the error respect to the full order model. In the numerical examples, we will examine two different cases, one without nonorthogonal correction and one with a bounded orthogonal correction.

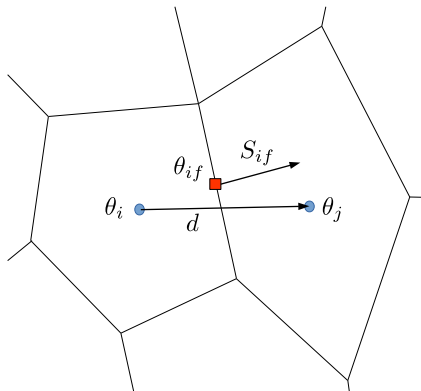


FIGURE 1 Example of a nonorthogonal mesh for a two-dimensional problem [Colour figure can be viewed at wileyonlinelibrary.com]

Once that the discretization of all the terms inside the continuous formulation is described, it is finally possible to recast Equation (3) in matrix form and to transform it into the following system of linear equations:

$$\mathbf{A}(\boldsymbol{\mu})\boldsymbol{\theta}(\boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}), \quad (7)$$

where $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is a matrix which represents the discretized Laplace operator, $\boldsymbol{\theta} \in \mathbb{R}^{N_h}$ is a vector of unknown values, $\mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is a source term vector which accounts for physical source terms, for implicit nonorthogonal corrections and for boundary conditions.

3.2 | Geometrical parametrization

Once the FV discretization is defined, one important aspect is the selection of the strategy used to solve a geometrically parametrized problem. The usual strategy applied in the reduced order modeling community, and more in general for a finite element discretization, is to construct a map $\mathcal{M}(\boldsymbol{\mu}) : \Omega(\boldsymbol{\mu}) \rightarrow \hat{\Omega}$ which maps all the possible domain configurations to a reference one $\hat{\Omega}$. This is also the procedure employed in an FV setting, for a linear mapping, in Reference 33. Unfortunately, this operation is affordable only for simple cases and for certain types of schemes and creates the additional complexities:

- In an FV code, respect to a finite element one, also at the element level, integrals are not written in a reference domain setting. In finite elements, in order to perform the integration of the shape functions, all the elements are mapped to a reference domain. In the FV method, such operation is not performed and, therefore, to write integrals in a reference setting, making use of a mapping, would be much more intrusive.
- For a general nonlinear map, also in the finite element case, the equations written into a reference domain transform a pure diffusion equation into a diffusion-advection equation.³³
- The FV method, with respect to the finite element one, involves also surface integrals. Even though it is possible to map infinitesimally small distances, areas, and volumes from a real to a reference domain, and to write the continuous equations in a reference framework, at the discrete level, there are additional source terms created by nonorthogonality (we refer to Section 3.1 for this issue) and by nonlinear interpolation schemes (ie, upwinding schemes). The FV method involves the computation of the values at center of the faces starting from the values at the center of the cells. Unless one is using a central differencing scheme, where the value at the center of the faces depends only on its distance with respect to the center of the cells, to write this operation in a reference domain becomes not trivial.
- One of the great advantages of the FV consists into the possibility of using a mesh made of polyhedra with an arbitrary number of faces. Writing equations in a reference domain using polyhedra with arbitrary shape may be extremely challenging and would be realistically possible only for tetrahedral elements. This would strongly limit the versatility of the method.

For the above reasons, in the present work, we opted to work with the equations directly on the physical domain and, for each new value of the input parameters, to deform the mesh and to solve the full order problem on the real physical domain. One of the drawbacks of this approach is given by the fact that an additional problem related to the motion of the grid points must be solved. Moreover, all the different snapshots are defined in a different domain, and therefore, an adjustment is needed. These aspects will be examined in details in the next subsections.

The proposed approach is not the only one possible to avoid the usage of a reference domain approach. In References 34–36, the authors make use of immersed methods during the order stage to deal with the geometrical parametrization and to avoid the mapping to a reference domain.

3.3 | Mesh motion strategies

As mentioned in the previous section, in this work, we rely on an arbitrary Lagrangian-Eulerian (ALE) formulation³¹ for the treatment of all the possible different geometric configurations. Therefore, the shape of the computational domain changes according to the parameter values. Working with such an approach, we have to distinguish between the motion

of boundary points, which is a given data defined by the parameter values and by the geometrical parametrization, and the motion of the internal points that depends on the employed mesh motion strategy. Dealing with a full order FV discretization, we highlight here, as shown in the description of the full order discretization, that the accuracy of the solution is strongly dependent on the quality of the mesh. For this reason, in the selection of the mesh motion strategy, it is crucial to select an algorithm that produces a deformed mesh that has a nonorthogonality value as small as possible. In this work, we analyze two different strategies to solve the mesh motion problem and we highlight their features in view of the construction of a ROM. Here the two methods are briefly recalled and only the aspects relevant for reduced order modeling perspectives are detailed. The interested reader may refer to References 37 and 38 and to references therein for more details.

3.3.1 | Laplace equation with variable diffusivity

The first type of approach is based on the solution of a Laplace equation with a variable diffusivity:

$$\operatorname{div}(\gamma \nabla \mathbf{s}) = 0 \quad \text{in } \Omega, \quad (8)$$

$$\mathbf{s} = \mathbf{s}_D \quad \text{on } \Gamma_D, \quad (9)$$

where \mathbf{s} is the displacement field of the grid points and γ is a diffusivity field. For the diffusivity field, several options are possible such as a constant value, a value proportional to the inverse of the distance respect to moving boundaries, and so on. In this work, it is chosen to use a diffusivity field, which is equal to the inverse of the square of the distance respect to the moving boundaries r :

$$\gamma = r^{-2}. \quad (10)$$

This value of the diffusivity coefficient proved to produce a smooth mesh motion and, therefore, a good quality of the deformed mesh in cases with moderate mesh deformations.³⁷ The computation of the displacement of the grid points, at the full order level, requires the resolution of a linear sparse system that has the same dimension of the order problem multiplied by the number of physical dimensions of the problem $d = 2, 3$ (ie, $N_h \cdot d$). The problem is in fact discretized as:

$$\mathbf{A}_D \mathbf{s} = \mathbf{b}_D(\mu), \quad (11)$$

where $\mathbf{A}_D \in \mathbb{R}^{N_h \cdot d \times N_h \cdot d}$ is the discretized Laplace operator and the term $\mathbf{b}_D \in \mathbb{R}^{N_h \cdot d}$ that depends on the boundary conditions \mathbf{s}_D is the source term. Therefore, also during the online stage, when a new parameter value needs to be simulated, one has to solve a full order mesh motion problem. For general deformation mappings, it is usually not possible to recover an affine decomposition of the boundary condition \mathbf{s}_D and, therefore, also for the mesh motion problem, in order to obtain an efficient offline-online splitting, we have to rely on hyper-reduction techniques. More details on this aspect are reported in the numerical example sections.

3.3.2 | Radial basis functions

The second examined approach consists in a radial basis function (RBF) interpolation approach. In this approach, the displacements of all the mesh points not belonging to the moving boundaries are approximated by an RBF interpolant function:

$$\mathbf{s}(\mathbf{x}) = \sum_{i=1}^{N_b} \beta_i \xi(\|\mathbf{x} - \mathbf{x}_{b_i}\|) + q(\mathbf{x}), \quad (12)$$

where \mathbf{x}_{b_i} are the coordinates of points for which we know the boundary displacements, N_b is the number of points on the boundary, ξ is a given basis function, and $q(\mathbf{x})$ is a polynomial. The coefficients β_i and the polynomial $q(\mathbf{x})$ are obtained by the imposition of interpolation conditions

$$\mathbf{s}(\mathbf{x}_{b_i}) = \mathbf{d}_{b_i}, \quad (13)$$

where \mathbf{d}_{b_i} is the displacement value at the boundary points and by the additional requirement:

$$\sum_{i=1}^{N_b} \beta_i q(\mathbf{x}_{b_i}) = 0. \quad (14)$$

In the present case, we select basis functions for which it is possible to use linear polynomials $q(\mathbf{x})$. For more informations concerning the selection of the order of polynomials, respect to the type of basis functions, the reader may see Reference 39. Finally, the values of the coefficients β_i and the coefficients δ_i of the linear polynomial q can be obtained solving the linear problem:

$$\begin{bmatrix} \mathbf{d}_b \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{b,b} & \mathbf{P}_b \\ \mathbf{P}_b^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta} \end{bmatrix}, \quad (15)$$

where $\mathbf{M}_{b,b} \in \mathbb{R}^{N_b \times N_b}$ is a matrix containing the evaluation of the basis function $\xi_{b_i b_j} = \xi(\|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\|)$ and $\mathbf{P}_b \in \mathbb{R}^{N_b \times (d+1)}$ is a matrix where d is the spatial dimension. Each row of this matrix, which contains the coordinates of the boundary points, is given by $(\mathbf{P}_b)_{i\bullet} = [1 \ \mathbf{x}_{b_i}]$. Once the system of Equation (15) is solved, one can obtain the displacement of all the internal points using the RBF interpolation:

$$\mathbf{d}_{in_i} = \mathbf{s}(\mathbf{x}_{in_i}), \quad (16)$$

where \mathbf{x}_{in_i} are the coordinates of the internal grid points. The computation of the displacement of the grid points entails the resolution of a dense system of equations that has dimension $N_b + d + 1$. Usually, the number of boundary points N_b is much smaller respect to the number of grid points N_h . Moreover, as it will be shown in the numerical examples, usually it is sufficient to select only a subset of the boundary points, and therefore, the dimension of this system is further reduced. Making a comparison with the Laplace equation approach, the RBF methodology entails the resolution of a smaller but dense system of equations. In the numerical examples, it will also be done a comparison in terms of mesh qualities after the mesh deformation.

4 | THE POD-GALERKIN ROM

Equation 7 entails the resolution of a possibly large system of equations. The resolution of this system of equations, in a multiquery setting, may become unfeasible. In this work, in order to alleviate the computational burden, we rely on a projection-based ROM and in particular on a POD-Galerkin ROM. Projection-based ROMs exploit the fact that, in most of the cases, the solution manifold lies in a low-dimensional space and can be, therefore, approximated by a linear combination of a reduced number of properly selected global basis functions:

$$\theta(\boldsymbol{\mu}, \mathbf{x}) = \sum_{i=1}^{N_r} a_i^\theta(\boldsymbol{\mu}) \varphi_i^\theta(\mathbf{x}), \quad (17)$$

where $\varphi_i^\theta(\mathbf{x})$ are the parameter independent basis functions and $a_i^\theta(\boldsymbol{\mu})$ are the parameter dependent coefficients. There exist different approaches to construct the set of basis functions such as the greedy approach, the POD, the proper generalized decomposition, and so on.^{2-4,40-42} We decided here to rely on a POD approach. Given a parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, we select a finite dimensional training set $\mathcal{K}_{\text{train}} \subset \mathcal{P}$ and for each of the possible combinations of the parameter values, we solve a full order problem. The full order problem is then solved for each $\boldsymbol{\kappa}_{j_{\text{train}}} \in \mathcal{K}_{\text{train}} = \{\boldsymbol{\kappa}_{1_{\text{train}}}, \dots, \boldsymbol{\kappa}_{N_{\text{train}}}\}$ where $\boldsymbol{\kappa}_{j_{\text{train}}}$ is the j -sample belonging to the finite dimensional training set $\mathcal{K}_{\text{train}}$. Each sample $\boldsymbol{\kappa}_{j_{\text{train}}}$ corresponds to a certain full order solution θ_j and the ensemble of whole the full order solutions gives the snapshots matrix:

$$\mathbf{S}_\theta = [\theta(\boldsymbol{\kappa}_{1_{\text{train}}}), \dots, \theta(\boldsymbol{\kappa}_{N_{\text{train}}})] \in \mathbb{R}^{N_h \times N_{\text{train}}}. \quad (18)$$

Once the snapshots matrix is set, it is possible to apply POD in order to generate an RB space to be used for the projection of the governing equations. Given a general scalar or vectorial parametrized function $\mathbf{u}(\boldsymbol{\mu}) : \Omega \rightarrow \mathbb{R}^d$, with a certain number of realizations $\mathbf{u}_1, \dots, \mathbf{u}_{N_{\text{train}}}$, the POD problem consists in finding, for each value of the dimension of POD space

$N_{\text{POD}} = 1, \dots, N_{\text{train}}$, the scalar coefficients $a_1^1, \dots, a_{N_{\text{train}}}^{N_{\text{train}}}$ and functions $\varphi_1, \dots, \varphi_{N_{\text{train}}}$ that minimize the quantity:

$$E_{N_{\text{POD}}} = \sum_{i=1}^{N_{\text{train}}} \| \mathbf{u}_i - \sum_{k=1}^{N_{\text{POD}}} a_i^k \varphi_k \|^2_{L_2(\Omega)} \quad \forall N_{\text{POD}} = 1, \dots, N, \quad (19)$$

$$\text{with } \langle \varphi_i, \varphi_j \rangle_{L_2(\Omega)} = \delta_{ij} \quad \forall i, j = 1, \dots, N_{\text{train}}. \quad (20)$$

It is well known⁴³ that the minimization of Equation (19) reduces to the following eigenvalue problem:

$$\mathbf{C}^u \mathbf{Q}^u = \mathbf{Q}^u \boldsymbol{\lambda}^u, \quad (21)$$

$$C_{ij}^u = \langle \mathbf{u}_i, \mathbf{u}_j \rangle_{L_2(\Omega)} \quad \text{for } i, j = 1, \dots, N_{\text{train}}, \quad (22)$$

where $\mathbf{Q}^u \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$ is the matrix containing the eigenvectors, while $\boldsymbol{\lambda}^u \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$ is the matrix containing the eigenvalues. We remind here that the interest is into parametrized geometries and that we do not map all the possible configurations to a reference one. For this reason, in this case, there is an additional difficulty given by the fact that the different snapshots “live” in different domains. The method used here consists into the usage of the method of snapshots where the correlation matrix of Equation (21) is constructed using an inner product with a modified mass matrix \mathbf{M}' which is referred to an ensemble average of all the different mesh configurations:

$$C_{ij}^\theta = \theta_i^T \mathbf{M}' \theta_j. \quad (23)$$

In the case of snapshots defined all in the same domain, the above expression reduces to the standard L_2 inner product. Operating with this procedure, during the offline stage, we need to additionally compute the mass matrix associated with the ensemble average of the parameter values inside the training set:

$$\mathbf{M}'(\bar{\kappa}) \quad \text{with} \quad \bar{\kappa} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \kappa_i. \quad (24)$$

Once the modified correlation matrix is assembled and the eigenvalue problem is solved, one can compute the basis functions in a standard way:

$$\varphi_i^\theta = \frac{1}{N_{\text{train}} \lambda_i^\theta} \sum_{j=1}^{N_{\text{train}}} \theta_j Q_{ij}^\theta. \quad (25)$$

Based on the eigenvalue decay of the POD eigenvalues, we can select only the first N_r basis functions and assemble the matrix:

$$\mathbf{L} = [\varphi_1^\theta, \dots, \varphi_{N_r}^\theta] \in \mathbb{R}^{N_h \times N_r}. \quad (26)$$

Once the bases are set, it is possible to approximate the solution vector with:

$$\boldsymbol{\theta} \approx \boldsymbol{\theta}_{rb} = \sum_{i=1}^{N_r} a_i^\theta(\boldsymbol{\mu}) \varphi_i^\theta(\mathbf{x}), \quad (27)$$

and finally, the original problem can be projected onto the POD space giving rise to the following reduced problem:

$$\mathbf{L}^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{L} \mathbf{a}^\theta = \mathbf{L}^T \mathbf{f}(\boldsymbol{\mu}), \quad (28)$$

that can be rewritten as

$$\mathbf{A}^r(\boldsymbol{\mu}) \mathbf{a}^\theta = \mathbf{f}^r(\boldsymbol{\mu}), \quad (29)$$

where $\mathbf{A}^r \in \mathbb{R}^{N_r \times N_r}$ is the discrete reduced parametrized differential operator, $\mathbf{a}^\theta \in \mathbb{R}^{N_r}$ is the unknown vector of reduced coefficients, and $\mathbf{f}^r(\boldsymbol{\mu}) \in \mathbb{R}^{N_r}$ is the reduced source term. Before solving the reduced problem for a new value of the parameter, one has to solve the mesh motion problem, to assemble the discretized operator $\mathbf{A}(\boldsymbol{\mu})$ and the source term $\mathbf{f}(\boldsymbol{\mu})$ and finally to perform the projection. The resolution of the mesh problem has the same dimension of the full order problem, as the assembly of the discretized operator. In order to ensure an efficient offline-online splitting, it is crucial to perform full order operations only during the offline stage. The approach employed here to ensure an efficient offline-online splitting is to rely on the discrete variant of the empirical interpolation method (D-EIM).^{15,44,45} Other approaches are also possible and among them we list the gappy-POD⁴⁶ or the GNAT method.^{47,48} Since the mass matrix of Equation (23) is computed on an ensemble average of all the deformed meshes, this deformed configuration can be considered as a “reference” geometry that is used for the computation of the RB functions. Therefore, the present methodology shares some common features with a standard reference-domain approach.*

4.1 | Discrete empirical interpolation method

As illustrated in the previous section, given the fact that it is possible to efficiently solve the mesh motion problem, also during the online stage (more details on this issue are reported in Section 5.1.1), Equation (28) still needs the assembly of a full order differential operator. In fact, for a new value of the parameter, it is not possible to recover an affine decomposition of the differential operator $\mathbf{A}(\boldsymbol{\mu})$ and of the source term vector $\mathbf{f}(\boldsymbol{\mu})$. The approach used here consists into an approximate affine expansion of the differential operator $\mathbf{A}(\boldsymbol{\mu})$ and of the source term vector $\mathbf{f}(\boldsymbol{\mu})$ as:

$$\mathbf{A}(\boldsymbol{\mu}) = \sum_{k=1}^{N_A} b_k^A(\boldsymbol{\mu}) \chi_k^A, \quad \mathbf{f}(\boldsymbol{\mu}) = \sum_{k=1}^{N_f} c_k^f(\boldsymbol{\mu}) \chi_k^f, \quad (30)$$

where $b_k^A(\boldsymbol{\mu})$, $c_k^f(\boldsymbol{\mu})$ and $\chi_k^A \in \mathbb{R}^{N_h \times N_h}$, $\chi_k^f \in \mathbb{R}^{N_h}$ are parameter-dependent coefficients and parameter-independent basis functions for the discretized differential operator \mathbf{A} and for the source term vector \mathbf{f} , respectively.

For the computation of the basis functions, χ_k^A and χ_k^f different approaches are possible, here we used a matrix version of the snapshot POD method. The details of the computational procedure are reported in Algorithm 1. The starting point of the algorithm are the matrices $\mathbf{S}_A = \{\mathbf{A}(\boldsymbol{\kappa}_{i_{\text{train}}})\}_{i=1}^{N_{\text{train}}} \in \mathbb{R}^{N_h \times N_h \times N_{\text{train}}}$ and vectors $\mathbf{S}_f = \{\mathbf{f}(\boldsymbol{\kappa}_{i_{\text{train}}})\}_{i=1}^{N_{\text{train}}} \in \mathbb{R}^{N_h \times N_{\text{train}}}$. These need to be stored during the offline stage together with the full order model solutions. On the matrix and vector “snapshots,” we apply then POD with the method of snapshots. As inner product to compute the correlation matrices, we used the Frobenious inner product defined as $\langle \mathbf{A}, \mathbf{B} \rangle_F = \text{tr}(\mathbf{A}^T \mathbf{B})$.

Algorithm 1. THE POD algorithm for the D-EIM basis creation

Input: matrices $\mathbf{S}_A = \{\mathbf{A}(\boldsymbol{\kappa}_{i_{\text{train}}})\}_{i=1}^{N_{\text{train}}}$, vectors $\mathbf{S}_f = \{\mathbf{f}(\boldsymbol{\kappa}_{i_{\text{train}}})\}_{i=1}^{N_{\text{train}}}$, N_A , N_f

Output: bases $\{\chi_i^A\}_{i=1}^{N_A}$, $\{\chi_i^f\}_{i=1}^{N_f}$

- 1: Compute the correlation matrices $\mathbf{C}_{ij}^A = \langle \mathbf{A}(\boldsymbol{\kappa}_{i_{\text{train}}}), \mathbf{A}(\boldsymbol{\kappa}_{j_{\text{train}}}) \rangle_F$ and $\mathbf{C}_{ij}^f = \langle \mathbf{f}(\boldsymbol{\kappa}_{i_{\text{train}}}), \mathbf{f}(\boldsymbol{\kappa}_{j_{\text{train}}}) \rangle_F$
 - 2: Perform the eigenvalue decompositions $\mathbf{C}^A \mathbf{Q}^A = \mathbf{Q}^A \boldsymbol{\lambda}^A$ and $\mathbf{C}^f \mathbf{Q}^f = \mathbf{Q}^f \boldsymbol{\lambda}^f$
 - 3: compute $\chi_i^A = \frac{1}{N_{\text{train}} \lambda_i^A} \sum_{j=1}^{N_{\text{train}}} \mathbf{A}(\boldsymbol{\kappa}_{j_{\text{train}}}) \mathbf{Q}_{ij}^A$ for $i = 1 : N_A$
 - 4: compute $\chi_i^f = \frac{1}{N_{\text{train}} \lambda_i^f} \sum_{j=1}^{N_{\text{train}}} \mathbf{f}(\boldsymbol{\kappa}_{j_{\text{train}}}) \mathbf{Q}_{ij}^f$ for $i = 1 : N_f$
-

Once the basis functions are set, one needs to define a way to compute the coefficients vectors $\mathbf{b}^A(\boldsymbol{\mu}) \in \mathbb{R}^{N_A}$ and $\mathbf{c}^f(\boldsymbol{\mu}) \in \mathbb{R}^{N_f}$. To construct the above set of basis functions, we rely on the standard discrete empirical interpolation procedure as introduced in Reference 45 for parametrized functions and extended in Reference 49 to parametrized differential

*At the continuous level, there are not particular advantages with respect to a standard reference-domain formulation. As highlighted in Section 3, the main benefits are on the implementation side when the starting full order discretization is an FV one.

operators:

$$\mathbf{b}^A(\boldsymbol{\mu}) = \mathbf{B}_A^{-1} \mathbf{A}_I(\boldsymbol{\mu}), \quad (31)$$

$$\mathbf{A}_{I_l}(\boldsymbol{\mu}) = \sum_{k=1}^{N_h} \mathbf{P}_{A_{k \bullet i}}^T \mathbf{A}(\boldsymbol{\mu})_{\bullet k}, \quad (32)$$

$$\mathbf{c}^f(\boldsymbol{\mu}) = (\mathbf{P}_f^T \mathbf{U}_f) \mathbf{f}_I(\boldsymbol{\mu}), \quad (33)$$

$$\mathbf{f}_I(\boldsymbol{\mu}) = \mathbf{P}_f^T \mathbf{f}(\boldsymbol{\mu}), \quad (34)$$

where the terms $\mathbf{B}_A \in \mathbb{R}^{N_h \times N_h \times N_A}$, $\mathbf{P}_A \in \mathbb{R}^{N_h \times N_h \times N_A}$, $\mathbf{U}_f \in \mathbb{R}^{N_h \times N_f}$, and $\mathbf{P}_f \in \mathbb{R}^{N_h \times N_f}$ are obtained following Algorithm 2. In the above expression, we have introduced the \square_{\bullet} notation meaning that we let free the indices relative to the position of the \bullet symbol. For example, $\mathbf{P}_{A_{k \bullet i}}$ means we are considering the k -row of the i -layer of the \mathbf{P}_A 3D matrix. The \mathbf{P}_A and \mathbf{P}_f matrices assume the same meaning as those in Reference 45 with the difference that, working with parametrized differential operators, the \mathbf{P}_A term is a 3-D matrix. Each l -layer of the \mathbf{P}_A matrix is given by a matrix $\mathbf{E}_l^A \in \mathbb{R}^{N_h \times N_h}$ that is a matrix where the only nonzero element, which has unitary value and is located in the (x, y) position identified by the magic point with coordinates (I_{rl}^A, I_{cl}^A) obtained at the l -iteration of the D-EIM Algorithm 2. Each k -column of \mathbf{P}_f is given by the vector $\mathbf{E}_k^f \in \mathbb{R}^{N_h}$ which is a vector where the only nonzero element, which has also unitary value, is located at the position identified by the magic point I_k^f obtained at the k -iteration of the D-EIM Algorithm 2.

Algorithm 2. The D-EIM procedure

Input: $\{\chi_i^A\}_{i=1}^{N_A} \in \mathbb{R}^{N_h \times N_h \times N_A}$, $\{\chi_i^f\}_{i=1}^{N_f} \in \mathbb{R}^{N_h \times N_f}$

Output: $\mathbf{I}_A \in \mathbb{R}^{N_h \times 2}$, $\mathbf{I}_f \in \mathbb{R}^{N_h}$

- 1: $[|\rho_A|, I_{r1}^A, I_{c1}^A] = \max\{|\chi_1^A|\}$, $\mathbf{I}_1^A = [I_{r1}^A, I_{c1}^A]$ and $[|\rho^f|, I_1^f] = \max\{|\chi_1^f|\}$
 - 2: $\mathbf{E}_1^A = \mathbf{0}_{N_h \times N_h}$, $\mathbf{E}_1^A(I_{r1}^A, I_{c1}^A) = 1$, $\mathbf{E}_1^f = \mathbf{0}_{N_h, 1}$, $\mathbf{E}_1^f(I_1^f) = 1$
 - 3: $\mathbf{U}_A = [\chi_1^A]$, $\mathbf{P}_A = [\mathbf{E}_1^A]$, $\mathbf{I}_A = [\mathbf{I}_1^A]$ and $\mathbf{U}_f = [\chi_1^f]$, $\mathbf{P}_f = [\mathbf{E}_1^f]$, $\mathbf{I}_f = [I_1^f]$
 - 4: **for** $l = 2$ to N_A and $k = 2$ to N_f **do**
 - 5: Solve $(\mathbf{P}_A^T \mathbf{U}_A) \mathbf{c}_A = \mathbf{P}_A^T \chi_l^A$ for \mathbf{c}_A and Solve $(\mathbf{P}_f^T \mathbf{U}_f) \mathbf{c}_f = \mathbf{P}_f^T \chi_k^f$ for \mathbf{c}_f .
 - 6: $\mathbf{r}_A = \chi_l^A - \mathbf{U}_A \mathbf{c}_A$ and $\mathbf{r}_f = \chi_k^f - \mathbf{U}_f \mathbf{c}_f$
 - 7: $[|\rho^A|, I_{rl}^A, I_{cl}^A] = \max\{|\mathbf{r}_A|\}$, $\mathbf{I}_l^A = [I_{rl}^A, I_{cl}^A]$ and $[|\rho^f|, I_k^f] = \max\{|\mathbf{r}_f|\}$
 - 8: $\mathbf{E}_l^A = \mathbf{0}_{N_h \times N_h}$, $\mathbf{E}_l^A(I_{rl}^A, I_{cl}^A) = 1$, $\mathbf{E}_k^f = \mathbf{0}_{N_h, 1}$, $\mathbf{E}_k^f(I_k^f) = 1$
 - 9: $\mathbf{U}_A \leftarrow [\mathbf{U}_A, \chi_l^A]$, $\mathbf{P}_A \leftarrow [\mathbf{P}_A, \mathbf{E}_l^A]$, $\mathbf{I}_A \leftarrow \begin{bmatrix} \mathbf{I}_A \\ \mathbf{I}_l^A \end{bmatrix}$,
 - 10: $\mathbf{U}_f \leftarrow [\mathbf{U}_f, \chi_k^f]$, $\mathbf{P}_f \leftarrow [\mathbf{P}_f, \mathbf{E}_k^f]$, $\mathbf{I}_f \leftarrow \begin{bmatrix} \mathbf{I}_f \\ I_k^f \end{bmatrix}$
 - 11:
 - 12: **end for**
-

Finally, we can write the D-EIM approximation as:

$$\mathbf{A}(\boldsymbol{\mu}) \approx \sum_{k=1}^{N_A} \mathbf{D}_{A_{\bullet \bullet k}} \mathbf{A}_{I_k}(\boldsymbol{\mu}), \quad (35)$$

$$\mathbf{D}_{A_{\bullet \bullet i}} = \sum_{k=1}^{N_A} \mathbf{U}_{A_{\bullet \bullet k}} \mathbf{B}_{A_{ki}}^{-1}, \quad (36)$$

$$\mathbf{B}_{A_{ij}} = \sum_{k=1}^{N_h} \mathbf{P}_{A_{k,i}}^T \mathbf{U}_{A_{\bullet,kj}}, \quad (37)$$

$$\mathbf{f}(\boldsymbol{\mu}) \approx \mathbf{B}_f \mathbf{f}_f(\boldsymbol{\mu}), \quad (38)$$

$$\mathbf{B}_f = \mathbf{U}_f (\mathbf{P}_f^T \mathbf{U}_f)^{-1} \in \mathbb{R}^{N_h \times N_f}, \quad (39)$$

$$\mathbf{f}_f(\boldsymbol{\mu}) = \mathbf{P}_f^T \mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^{N_f}, \quad (40)$$

where the D-EIM approximation for the vector quantities is reported with the original notation presented in Reference 45, while the one for D-EIM approximation of the discretized differential operator $\mathbf{A}(\boldsymbol{\mu})$, since we are working with 3D sparse matrices, and we want to rely on sparse matrices linear algebra operations, the notation is slightly modified to account for 3D matrices. In the above formulations, we have introduced the additional variables $\mathbf{D}_A \in \mathbb{R}^{N_h \times N_h \times N_A}$.

In the end, the idea is to precompute during the offline stage the terms $\mathbf{A}_k^r \in \mathbb{R}^{N_r \times N_r}$ and $\mathbf{f}_k^r \in \mathbb{R}^{N_r}$ and to express reduced operator and vector as:

$$\mathbf{A}^r(\boldsymbol{\mu}) = \sum_{k=1}^{N_A} \mathbf{A}_k^r \mathbf{A}_{I_k}(\boldsymbol{\mu}), \quad \mathbf{f}^r(\boldsymbol{\mu}) = \sum_{k=1}^{N_f} \mathbf{f}_k^r \mathbf{f}_{I_k}(\boldsymbol{\mu}), \quad (41)$$

where:

$$\mathbf{A}_k^r = \mathbf{L}^T \mathbf{D}_{\bullet\bullet k} \mathbf{L}, \quad \mathbf{f}_k^r(\boldsymbol{\mu}) = \mathbf{L}^T \mathbf{B}_{\bullet k}. \quad (42)$$

The terms \mathbf{A}_k^r and \mathbf{f}_k^r are precomputed during the offline stage and $\mathbf{A}_I(\boldsymbol{\mu})$ and $\mathbf{f}_I(\boldsymbol{\mu})$ are vector of coefficients that must be computed during the online stage.

In the above formulations, it is important to note that the \mathbf{A}_I and \mathbf{f}_I vector of coefficients correspond to pointwise evaluations of the differential operator $\mathbf{A}(\boldsymbol{\mu})$ and of the source term vector \mathbf{f} in the locations corresponding to the magic points inside the matrix \mathbf{I}_A and the vector \mathbf{I}_f reported in Algorithm 2.

Therefore, the only additional complexity, during the online stage, corresponds to pointwise evaluation of the source term vector and of the discretized differential operator. In order to make the procedure efficient and independent with respect to mesh size, this pointwise evaluation must involve operations only on a subset of the entire domain and must be independent with respect to the total number of degrees of freedom of the full order problem. The FV method satisfies this property, in fact, the point evaluation of an operator can be computed also on a subset of the whole domain. Each subset includes the cells relative to the point evaluation and a certain number of neighboring layers of cells. The required number of neighboring layers of cells depends on the type of scheme employed for the computation of the operator. For example, a central differencing scheme, without orthogonal correction requires only one layer of cells, while for second order schemes or a central differencing scheme with orthogonal correction, at least two layers of cells are necessary. The process for the construction of the subsets is exemplified in Figure 2. In the figure are reported two examples of the required computational stencil for both a point on the diagonal of the discretized differential operator and a point outside of the diagonal of the discretized differential operator.

5 | A PARAMETRIZED HEAT TRANSFER PROBLEM

In this section, we present the proposed methodology on a parametrized heat transfer problem. Particular attention is posed onto the selection of the best mesh motion algorithm, especially in view of model reduction purposes. The domain is parametrized by the parameter vector $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) \in [\boldsymbol{\mu}_{1,\min}, \boldsymbol{\mu}_{1,\max}] \times [\boldsymbol{\mu}_{2,\min}, \boldsymbol{\mu}_{2,\max}] \subset \mathbb{R}^2$. The parameter scheme is depicted on the left of Figure 3. In the same figure, it is reported also the chosen tessellation in its undeformed configuration, which accounts for 2700 quadrilateral cells. In the same figure, it is also depicted the mesh configuration using both the Laplacian smoothing and the RBF approach. The numerical tests are conducted using the in-house open-source

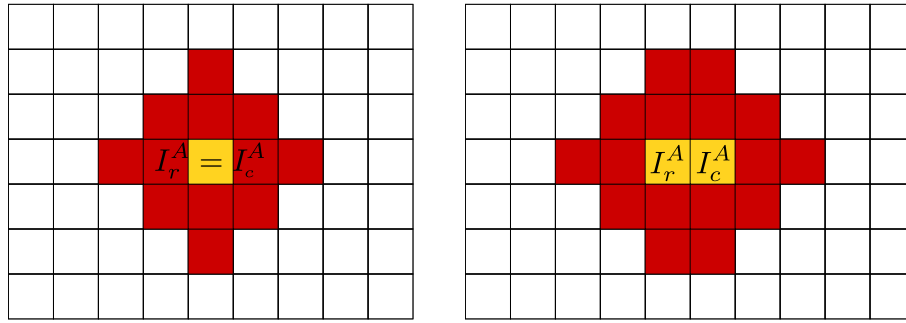


FIGURE 2 Example of the computational stencil needed for the evaluation of the D-EIM coefficients during the online stage. On the left, there is an example of a point on the diagonal of the discretized differential operator, while on the right, there is an example of a point outside the diagonal of the differential operator. In the image, two layers of cells are considered. The yellow cells depict the ones identified during the D-EIM procedure by the indices I_r^A and I_c^A , while the red ones depict the additional cells required for the differential operator evaluation [Colour figure can be viewed at wileyonlinelibrary.com]

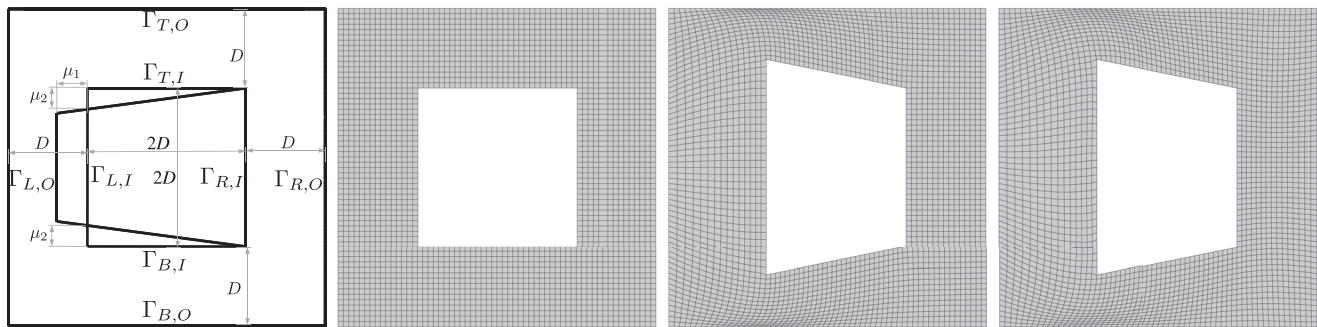


FIGURE 3 From left to right, we can observe the domain together with the dimensions ($D = 1$) and the parameters scheme, the tessellation in its undeformed configuration, the tessellation deformed using a Laplacian smoothing approach, and the tessellation deformed using a radial basis function approach

TABLE 1 Imposed values of the Dirichlet and Neumann boundary conditions for the different boundaries of the domain

| | $\Gamma_{L,O}$ | $\Gamma_{T,O}$ | $\Gamma_{R,O}$ | $\Gamma_{B,O}$ | $\Gamma_{L,I}$ | $\Gamma_{T,I}$ | $\Gamma_{R,I}$ | $\Gamma_{B,I}$ |
|------|-----------------|----------------|------------------|----------------|----------------|----------------|----------------|----------------|
| B.C. | $\theta_n = 10$ | $\theta = 10$ | $\theta_n = -10$ | $\theta = 10$ | $\theta = -10$ | $\theta = 0$ | $\theta = 0$ | $\theta = 0$ |

library *ITHACA-FV* (In real Time Highly Advanced Computational Applications for FVs)⁵⁰ which is a computational library based on the FV solver OpenFOAM 6.0.⁵¹ The computational domain is given by a square block with a hole inside. The boundary $\Gamma = \{\Gamma_{T,O}, \Gamma_{R,O}, \Gamma_{B,O}, \Gamma_{L,O}, \Gamma_{T,I}, \Gamma_{R,I}, \Gamma_{B,I}, \Gamma_{L,I}\}$ is subdivided into eight different parts which consist of the top, right, bottom, and left sides of the outer and inner polygons. The boundary conditions are set according to Table 1.

Before going deeply into numerical solutions, let us report here the strong form of the steady-state heat transfer equations and describe briefly what is the expected behavior of the physical solution for the problem we took into account:

$$\begin{cases} \operatorname{div}(\alpha_\theta \nabla \theta) = f & \text{in } \Omega(\mu), \\ \theta(x, \mu) = \theta_D(x, \mu) & \text{on } \Gamma_D(\mu), \\ \theta_n(x, \mu) = \theta_N(\mu, x) & \text{on } \Gamma_N(\mu), \end{cases} \quad (43)$$

where α_θ is the diffusivity of the homogeneous isotropic medium and has been fixed equal to 1, while a null forcing term f has been applied to the problem.

This is the most classical choice when dealing with elliptic problems. The Laplace operator only acts like a smoothing-diffusing term. For this reason, no peaks or discontinuities are expected into the domain, boundary temperatures should be spread all over Ω since the forcing term is adding no contribution. According with the *maximum principle*, the maximum (and the minimum as well) of the solution should be located on the boundaries since $f = 0$.

5.1 | The mesh motion strategies

In the numerical example, we examine different mesh motion strategies starting from different full order schemes. As reported in the previous section, and as exemplified in Figure 4, the selection of the full order numerical scheme used to deal with nonorthogonality affects the number of layers of additional cells around the ones identified by the magic points during the D-EIM procedure. In the picture, we report, for a case with just 5 D-EIM modes, the location of the cells identified by the magic points (yellow cells), together with the additional cells of the computational stencils necessary for the pointwise evaluation of the differential operators during the online stage (red cells). The identified cells are relative to the magic points identified for the source term vector \mathbf{f} .

Before analyzing the efficiency and the approximation properties of the ROM, we analyze the performances of the mesh motion strategies at the full order level. As extensively illustrated in the previous section, when using ALE approaches to deal with parametrized geometries, the accuracy of the results depends strongly on the quality of the deformed mesh. In a FV context, one of the most important indicators to certify the quality of a mesh consists into the nonorthogonality factor. We analyze the parametrized geometry in Figure 3 and we deform the mesh according to the discrete set of parameter samples $\mathcal{K}_{\text{train}}^1 = \{\kappa_{i_{\text{train}}}\}_{i=1}^{N_{\text{train}}} \in [-0.32, 0.32] \times [-0.32, 0.32]$. We set $N_{\text{train}} = 100$ and parameter samples are selected randomly inside the parameter space. The results of such test are reported qualitatively and quantitatively in the images and in the table in Figure 5. As can be seen from the image and the table, the Laplacian smoothing approach, compared with the RBF strategy, produces a deformed mesh with higher values of the nonorthogonality. This fact is particularly evident in the part of the domain close to the upper and lower boundaries. Moreover, changing the dimension of the parameter space used to determine the training set to $\mathcal{K}_{\text{train}}^2 \in [-0.45, 0.45] \times [-0.45, 0.45]$, as shown in Figure 6, for some values of the samples, the mesh motion strategy using a Laplacian smoothing approach produces a mesh where some of the cells have negative volumes. In these cases of course, the simulation results become unreliable. The RBF mesh motion strategy requires the selection of 56 control points on the moving boundaries and requires, therefore, as illustrated in Section 3.3.2, the resolution of relatively small linear system (especially if we consider the dimension of the full order problem $N_h = 2700$). The selection of the control points has been done automatically fixing a coarsening ratio which defines the ratio between the total number of points on the boundaries and the number of control points. The location of the control points identified by this automatic procedure is depicted in Figure 5. The RBFs are given by Gaussian functions with a radius $d_{\text{RBF}} = 0.6$.

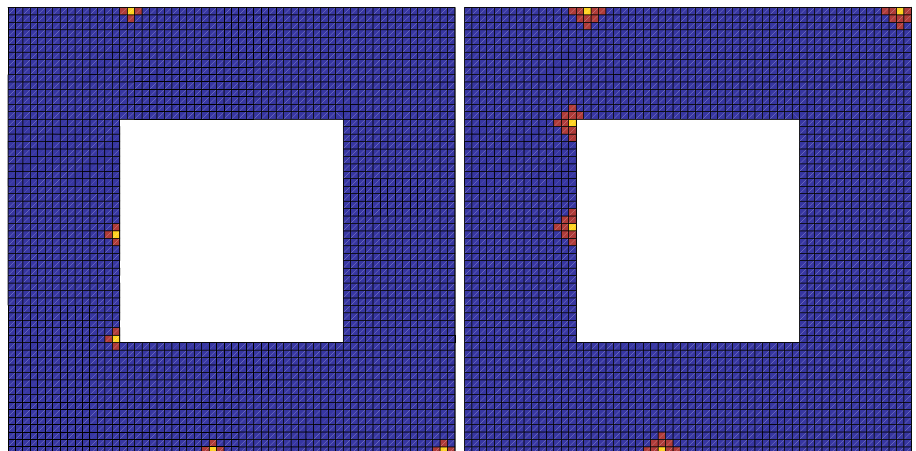


FIGURE 4 Comparison of the two mesh deformation strategies for large parametric geometrical variations. On the left, we observe the Laplacian smoothing approach that, as highlighted in the zoom, produces a deformed mesh that has cells with negative volumes (cells close to the left boundary). On the right, we have the RBF approach [Colour figure can be viewed at wileyonlinelibrary.com]

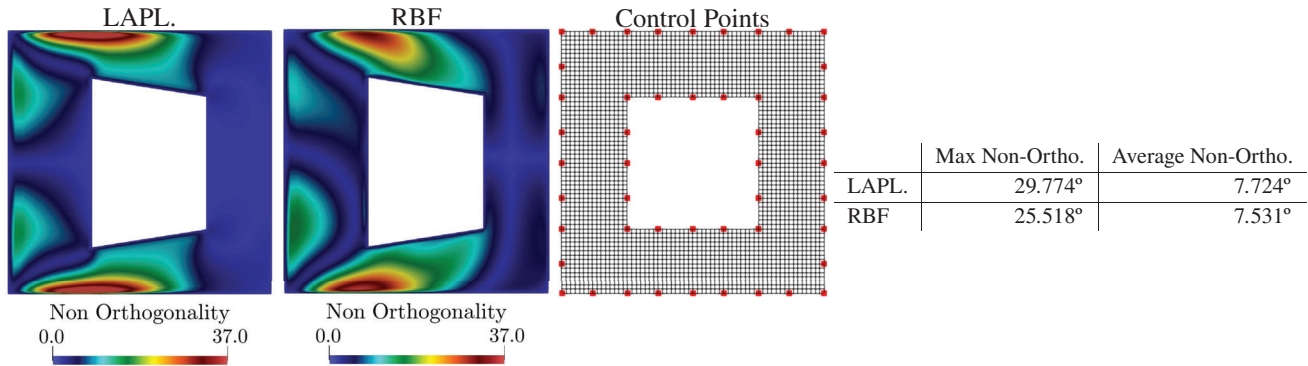


FIGURE 5 The first five magic points identified during the D-EIM procedure for the source term vector in the case of a linear scheme without orthogonal correction and in the case of a linear scheme with orthogonal correction

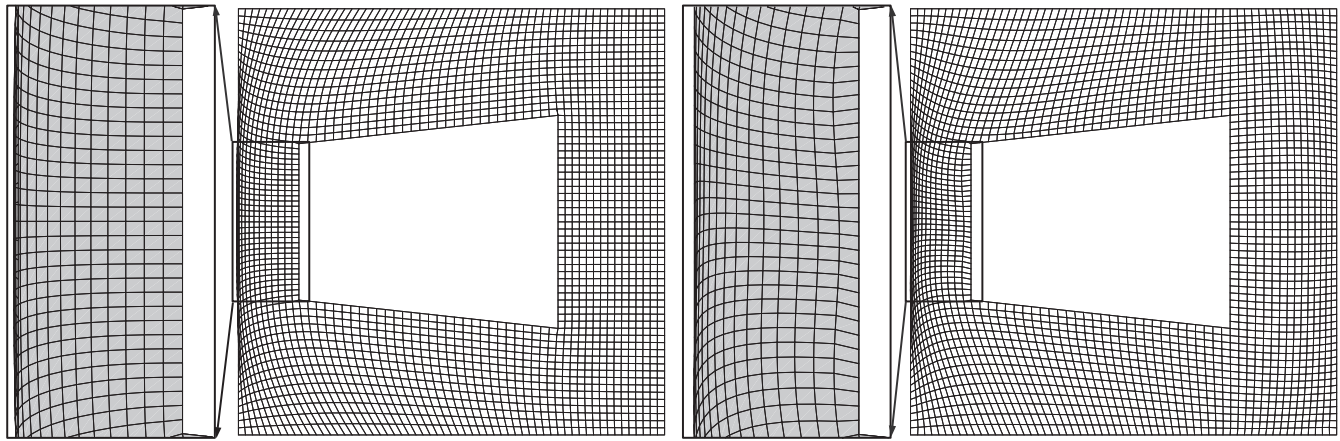


FIGURE 6 Nonorthogonality comparison between a Laplacian smoothing and an RBF approach. The two images on the left show, for one selected sample value $\kappa^* = (0.181184, 0.288162)$, the value of the nonorthogonality inside the domain. In the image on the right are reported the control points identified by the automatic coarsening procedure. In the table, it is reported the average value, over 100 sample points, for the maximum value of the nonorthogonality and for the average value of the nonorthogonality

5.1.1 | Reduction of the mesh motion problem in the case of Laplacian smoothing technique

The discretized operator \mathbf{A}_D of Equation (11) is always assembled in the same physical domain (ie, the undeformed one) and has a diffusivity value γ which is changing in space but that is always constant with respect to the parameter values. Therefore, it can be assembled only once during the offline stage and projected onto the RB space of the geometrical deformation field. For the source term $\mathbf{b}_D(\boldsymbol{\mu})$, in the general case, it is usually not possible to recover an affine decomposition and we have, therefore, to rely on an approximate affine expansion using the discrete empirical interpolation method (D-EIM):

$$\mathbf{b}_D(\boldsymbol{\mu}) \approx \sum_{i=1}^{N_{b_d}} a_i^D(\boldsymbol{\mu}) \chi_i^D. \quad (44)$$

Once this approximation is done, it is straightforward to obtain a ROM of the mesh motion problem. For this particular case, since the motion of the moving boundaries depends linearly with respect to the input parameters, two D-EIM modes for the source term \mathbf{b}_d were sufficient to obtain accurate results. The average value of the L_2 relative error between the full and the reduced order solutions over the entire testing set is equal to $1.98 \cdot 10^{-7}$. This result was obtained using only 2 POD modes for the mesh motion field and 2 D-EIM basis function for the source term vector.

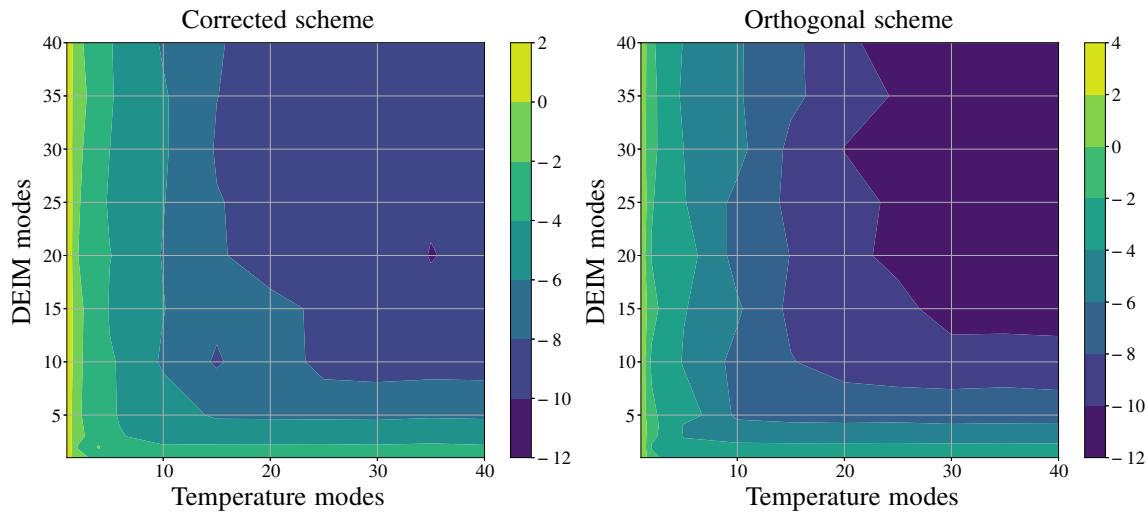


FIGURE 7 The average of the L^2 relative error norm of the temperature field for the Laplacian smoothing mesh motion strategy. The logarithm of the L^2 relative error norm is plotted against the number of modes used for the temperature field and the number of D-EIM modes used to approximate the discretized differential operator \mathbf{A} and the source term \mathbf{f} ($N_A = N_f$). The plots are reported with (left) and without (right) nonorthogonal correction [Colour figure can be viewed at wileyonlinelibrary.com]

5.2 | The reduced order model

Once we have determined the characteristic of the analyzed mesh motion strategies at the full order level, we examine the characteristics of a ROM generated using the different approaches. We will compare the results using two different schemes with and without orthogonal correction. The ROM is constructed using the discrete set $\mathcal{K}_{\text{train}}^1$ as defined in Section 5.1 and the results are tested on a different testing sets $\mathcal{K}_{\text{test}} = \{\kappa_{i_{\text{test}}}\}_{i=1}^{N_{\text{test}}} \in [-0.28, 0.28] \times [-0.28, 0.28]$ with $N_{\text{test}} = 100$. As mentioned in Section 2, in order to improve the accuracy of the results, it is possible to employ, at the full order level, a nonorthogonal correction strategy. In what follows we will indicate with θ_h the temperature field obtained using the full order model and with θ_{rb} the temperature field obtained using the ROM. The accuracy of the ROM is measured using the following expression to identify the approximation error:

$$\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{\|\theta_{h,i} - \theta_{rb,i}\|_{\Omega}}{\|\theta_{h,i}\|_{\Omega}}, \quad (45)$$

where N_{test} is the number of the parameter samples in the testing space and $\|\cdot\|_{\Omega}$ denotes the L_2 norm over the computational domain Ω . In Figure 7, we report a comparison of the approximation error for the Laplacian mesh motion strategy, with and without orthogonal correction, while changing the dimension of the RB space and the number of D-EIM modes used for the approximation of the discretized differential operator \mathbf{A} and the source term \mathbf{f} . In this case, we chose to use the same number of D-EIM modes $N_A = N_f$ for both \mathbf{A} and \mathbf{f} . In Figure 8, we report a similar plot but this time applying the RBF mesh motion procedure. In Figures 9 and 10, we report the eigenvalue decay relative to the POD procedure used to construct the modes for the temperature field and for the discretized differential operator \mathbf{A} and source term \mathbf{f} . To measure the efficiency of the different model reduction strategies, we measured also the computational speedup changing the number of D-EIM modes and the number of temperature modes. This comparison is depicted in Figure 11.

From the numerical example, we can draw several conclusions. We can note that the RBF mesh motion approach is generally performing slightly better respect to the Laplacian smoothing approach. Moreover, the orthogonal noncorrected scheme produces a ROM with better approximation properties respect to the corrected approach.[†] From Figure 11, one observes two different evidences. The first one is that the RBF approach produces a ROM with the better computational speedups and the same is true for the orthogonal approach. For what concern the orthogonal approach this fact is due

[†]The approximation error is measured using equation Equation (45) with respect to the FOM and not to the analytical solution. A corrected scheme usually produces numerical results with a smaller numerical error and, therefore, a comparison computed using the analytical solution might produced a different trend in the results.

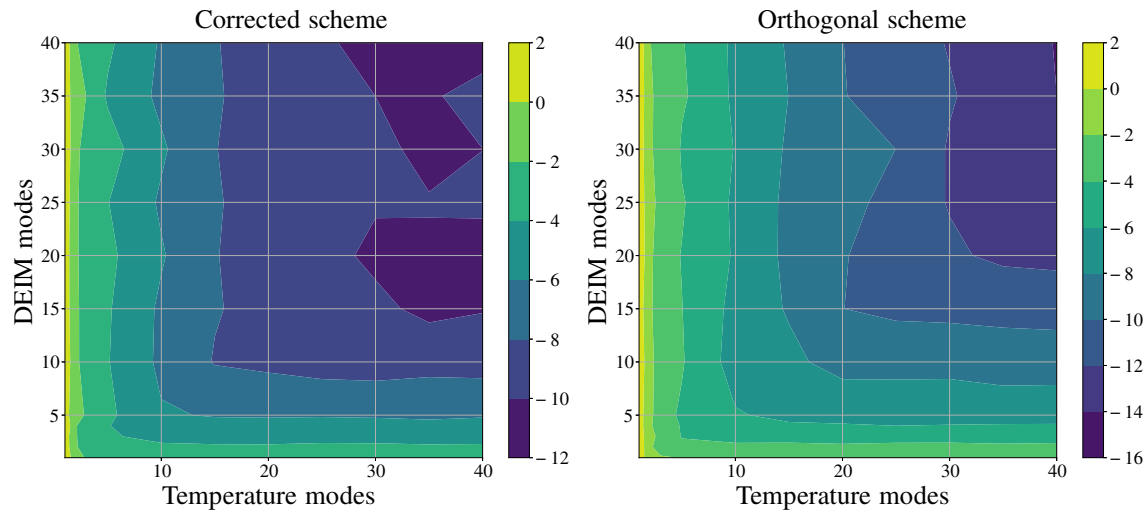


FIGURE 8 The average of the L^2 relative error norm of the temperature field for the RBF mesh motion strategy. The logarithm of the L^2 relative error norm is plotted against the number of modes used for the temperature field and the number of D-EIM modes used to approximate the discretized differential operator \mathbf{A} and the source term \mathbf{f} (with $N_A = N_f$). The plots are reported with (left) and without (right) non-orthogonal correction [Colour figure can be viewed at wileyonlinelibrary.com]

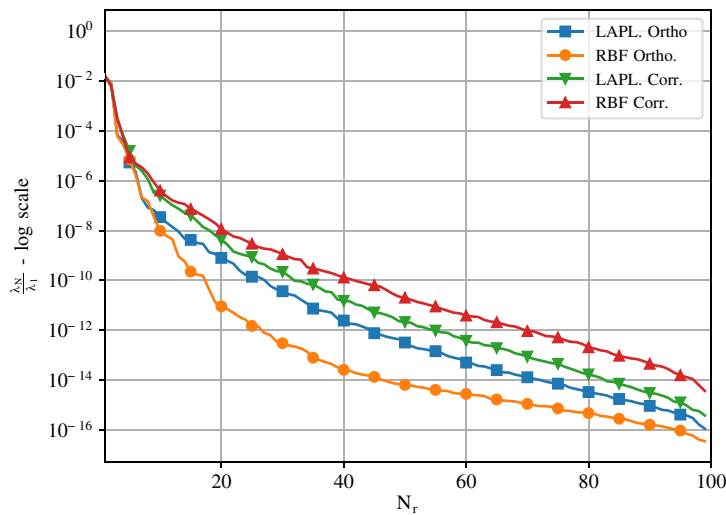


FIGURE 9 Plot reporting the eigenvalue decay relative to the POD procedure used to compute the modes for the temperature field [Colour figure can be viewed at wileyonlinelibrary.com]

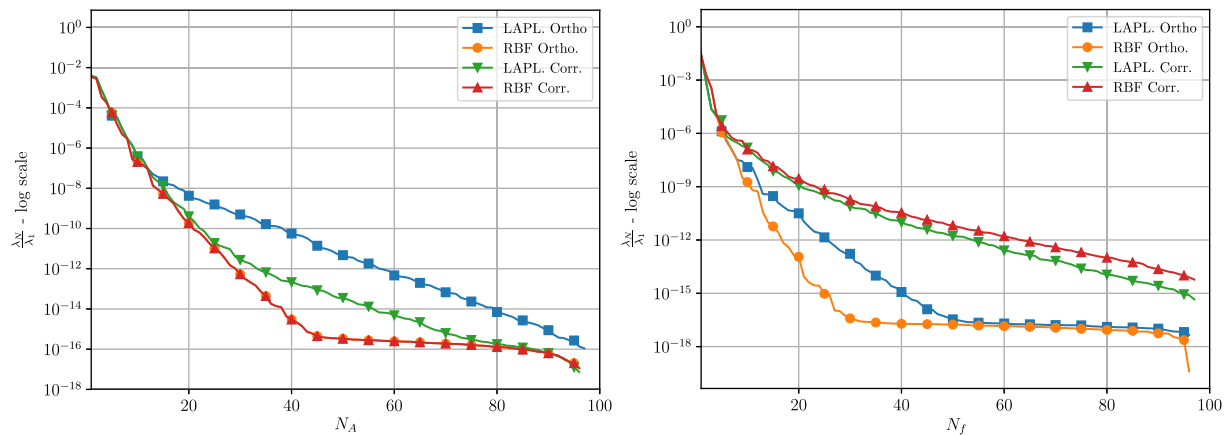


FIGURE 10 Plot reporting the eigenvalue decay relative to the POD procedure used to compute the D-EIM modes for the discretized differential operator \mathbf{A} (on the left) and the source term \mathbf{f} (on the right) [Colour figure can be viewed at wileyonlinelibrary.com]

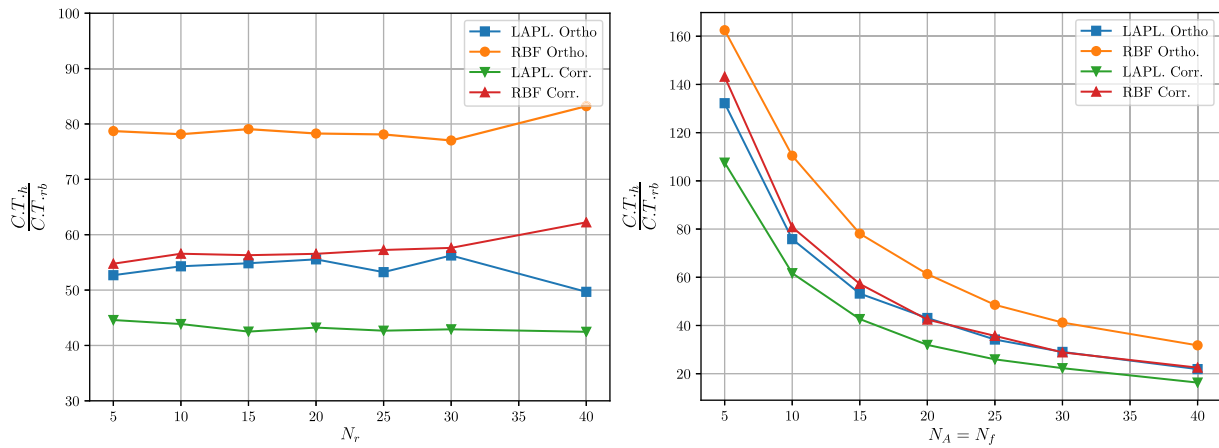


FIGURE 11 Comparison of the computational time, in terms of speedup, for the different strategies. On the left, we have a comparison keeping constant the number of D-EIM modes used to approximate the discretized differential operator \mathbf{A} and the source term \mathbf{f} ($N_A = N_f = 15$ and changing the number of temperature modes N_r). On the right, we have a comparison keeping constant the number of temperature modes and changing the number of D-EIM modes used to approximate the discretized differential operator \mathbf{A} and the source term \mathbf{f} [Colour figure can be viewed at wileyonlinelibrary.com]

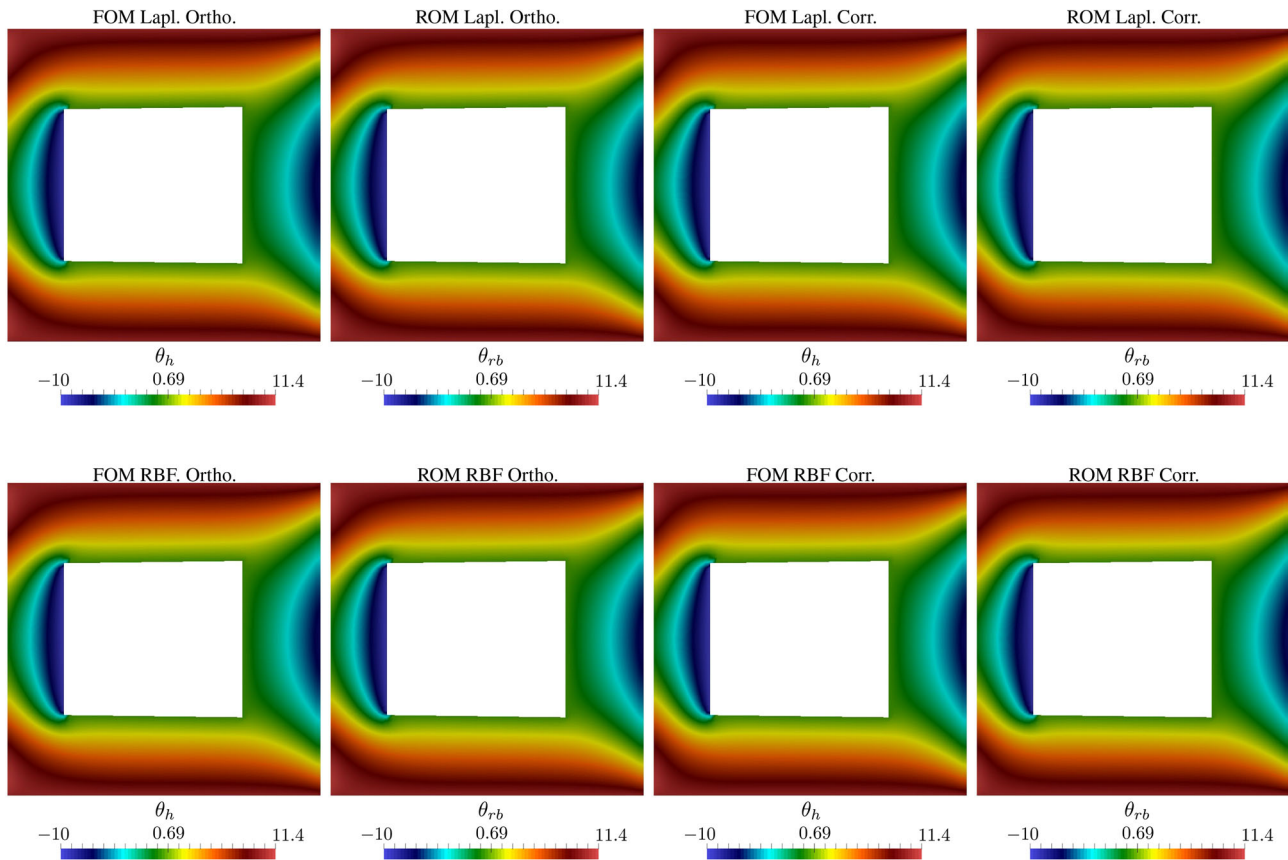


FIGURE 12 In the plot, we report the comparison between full order solution and reduced order solution using the Laplacian smoothing mesh motion approach (first row) and the RBF mesh motion approach (second row). Results are reported without (columns 1-2) and with (columns 3-4) orthogonal correction. All the results are for one selected sample point in the parameter space that was giving the worst results in terms of accuracy $\kappa^* = (-0.213777, 0.0335904)$ [Colour figure can be viewed at wileyonlinelibrary.com]

to the smaller computational stencil required by the methodology without correction. The second evidence is that an increase in the number of D-EIM modes produces a remarkable decrease in the computational speedup while the increase in the number of POD modes is negligible. In Figure 12 we report also a qualitative comparison between the full order model solutions and the reduced order ones with and without non-orthogonal correction.

6 | A PARAMETRIZED INCOMPRESSIBLE NAVIER-STOKES PROBLEM

In the previous section, we focused the attention on relatively simple heat transfer problems. The choice of such an easy case was justified by the fact that we wanted to reduce as much as impossible additional sources of problems that are generally linked with hyperbolic equations and advection dominated cases in order to provide a better understanding of the influence of the mesh motion strategy in view of model reduction purposes. In this section, we focus instead the attention on systems generated by the steady incompressible Navier-Stokes equations and on the difficulties that we have to face in order to extend the methodology introduced in the previous sections to a more complex scenario.

6.1 | Mathematical formulation

The focus is here on a geometrically parametrized problem that is approximated by the steady Navier-Stokes equations. Considering an Eulerian frame on a geometrically parametrized domain $\Omega(\mu) \subset \mathbb{R}^d$ with $d = 2, 3$, the problem consists in finding the vectorial velocity field $\mathbf{u}(\mu) : \Omega(\mu) \rightarrow \mathbb{R}^d$ and the scalar pressure field $p(\mu) : \Omega(\mu) \rightarrow \mathbb{R}$ such that:

$$\begin{cases} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot 2\nu \nabla^s \mathbf{u} = -\nabla p & \text{in } \Omega(\mu), \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega(\mu), \\ \mathbf{u}(x) = \mathbf{f}(x) & \text{on } \Gamma_{\text{In}}(\mu), \\ \mathbf{u}(x) = 0 & \text{on } \Gamma_0(\mu), \\ (\nu \nabla \mathbf{u} - p\mathbf{I})\mathbf{n} = 0 & \text{on } \Gamma_{\text{Out}}(\mu), \end{cases} \quad (46)$$

where $\Gamma(\mu) = \Gamma_{\text{In}}(\mu) \cup \Gamma_0(\mu) \cup \Gamma_{\text{Out}}(\mu)$ is the boundary of $\Omega(\mu)$ and is composed by three different parts $\Gamma_{\text{In}}(\mu)$, $\Gamma_{\text{Out}}(\mu)$, and $\Gamma_0(\mu)$ that indicate, respectively, inlet boundary, outlet boundary, and physical walls. The function $\mathbf{f}(x)$ represents the boundary conditions. The parameter dependency is given by a parameter vector $\mu \in \mathcal{P}$ with \mathcal{P} denoting the parameter space that describes the deformation of the physical domain. It is, moreover, assumed that the kinematic viscosity ν is constant in the spacial domain. For sake of brevity, the parameter dependency of the domain Ω will be omitted in the next formulations.

6.2 | The discrete formulation

The above equations are discretized at the full order level using a cell-centered FV approach and a SIMPLE algorithm to resolve the pressure velocity coupling. The equations are then written in integral form over each control volume Ω_i and the divergence terms are transformed into surface integral, thanks to the exploitation of Gauss's theorem and computed numerically as sum of fluxes over the surfaces of each cell Ω_i . In what follows we briefly report the discretization of each terms that appears inside the momentum and continuity equation. The nonlinear convective term is discretized as:

$$\int_{\Omega_i} \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) d\Omega = \int_{\partial\Omega_i} \mathbf{n} \cdot (\mathbf{u} \otimes \mathbf{u}) d\Gamma = \sum_f \mathbf{S}_f \cdot \mathbf{u}_f \otimes \mathbf{u}_f = \sum_f \mathbf{S}_f \cdot \mathbf{u}_f \otimes \mathbf{u}_f = \sum_f F_f \mathbf{u}_f, \quad (47)$$

where \mathbf{u}_f indicates the velocity at the center of the faces. Since the equation is solved using an iterative approach, the nonlinear term is linearized with the substitution of the term $\mathbf{S}_f \cdot \mathbf{u}_f = F_f$, which represents the mass flux over each face, with a previously calculated value of the mass flux that satisfies the continuity equation; for more details about this issue, we refer to Reference 32. The diffusive term is discretized as:

$$\int_{\Omega_i} \nabla \cdot 2\nu \nabla^s \mathbf{u} d\Omega = \int_{\partial\Omega_i} \mathbf{n} \cdot 2\nu \nabla^s \mathbf{u} d\Gamma = \int_{\partial\Omega_i} \mathbf{n} \cdot \nu \nabla \mathbf{u} d\Gamma = \nu \sum_f \mathbf{S}_f \cdot (\nabla \mathbf{u})_f, \quad (48)$$

where the first equality follows from the incompressibility constraint and the term $(\nabla \mathbf{u})_f$ indicates the gradient of the velocity field at the center of each face. This is calculated, starting from the values at the center of the neighbouring cells, using a finite difference scheme that includes a correction in the case of nonorthogonal meshes (see Equation (5)). For more details on available choices concerning the correction term, we refer to Reference 32.

The term originated from the gradient of pressure is discretized as:

$$\int_{\partial\Omega_e} \mathbf{n} p d\Gamma = \sum_f \mathbf{S}_f p_f, \quad (49)$$

while the term originated from the divergence of velocity is discretized as:

$$\int_{\partial\Omega} \mathbf{n} \cdot \mathbf{u} d\Gamma = \sum_{f=1}^{N_f} \mathbf{S}_f \cdot \mathbf{u}_f = \sum_{f=1}^{N_f} F_f. \quad (50)$$

6.3 | The SIMPLE algorithm

Now that the discretization of each term has been introduced we describe the solution strategy used to resolve the coupling between the momentum and continuity equation. The coupling between velocity and pressure is resolved using a SIMPLE strategy.⁵² The equations can be rewritten in matrix form as:

$$\begin{bmatrix} [\mathbf{A}_u] & [\nabla(\cdot)] \\ [\nabla \cdot (\cdot)] & [0] \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \quad (51)$$

where \mathbf{A}_u is the coefficient matrix coming from the momentum equation, $\mathbf{A}_u \mathbf{u} = \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (2\nu \nabla^s \mathbf{u})$. The above system matrix has a saddle point structure, which is usually not easy to solve using a coupled approach. For this reason, we rely on a segregated approach where the momentum equation is solved with a tentative pressure \tilde{p} and later corrected exploiting the divergence free constraint. The first equation could then, in principle, be solved using a direct solver; however, for efficiency reasons, it is decomposed into a diagonal, upper triangular, and lower triangular part $\mathbf{A}_u = \mathbf{D}_u + \mathbf{L}_u + \mathbf{U}_u$. The momentum equation is then rewritten considering this decomposition and using the previous converged value of the velocity to assemble the operator $\mathbf{H}(\mathbf{u}) = (\mathbf{L}_u + \mathbf{U}_u) \mathbf{u}_{k-1}$. Since now the system matrix is diagonal, it is easy to invert and a tentative value of the velocity can be computed as:

$$\mathbf{u} = \mathbf{D}_u^{-1} (\mathbf{H}(\mathbf{u}) - \nabla \tilde{p}). \quad (52)$$

Taking the divergence of the tentative velocity field and exploiting the divergence free constraint, one can derive a Poisson equation for pressure of the form:

$$\nabla \cdot (\mathbf{D}_u^{-1} \nabla p) = \nabla \cdot \mathbf{D}_u^{-1} (\mathbf{H}(\mathbf{u}_{k-1})). \quad (53)$$

Equations (52) and (53) are then solved using an iterative SIMPLE approach⁵² with an under-relaxation procedure to achieve convergence. The SIMPLE algorithm follows a segregated solution strategy where the momentum equation is firstly solved with a tentative pressure and an intermediate velocity field \mathbf{u}^* that is generally not divergence-free is computed. This intermediate velocity is plugged inside the continuity equation to construct a Poisson equation for pressure that is used to obtain a new pressure p that, if inserted inside the momentum equation, delivers a divergence free velocity field. This two steps are repeated until convergence.

6.4 | The reduced order model

Contrary to what was done in References 16 and 17 the ROM is constructed in such a way to be completely consistent with the procedure used at the full order level. In the previously mentioned works, the SIMPLE procedure used at the full order level was transformed into a saddle point problem at the reduced order level. This operation was possible because geometrical parametrization was not considered, and therefore, the reduced operators could be computed explicitly after

the offline stage using the RB functions. In this case, since the interest is into geometrical parametrization, and for a new value of the parameters, a new mesh is constructed, and the ROM is based on a reduced version of the SIMPLE algorithm presented in the previous subsection. During the online stage, the reduced operators are constructed on the deformed mesh using the methodology introduced in Section 4. For the construction of the RB spaces, we used a POD strategy, performed on an average deformed configuration, on the snapshots matrices of the velocity and pressure fields in order to obtain two separate set of RB functions:

$$\mathbf{L}_u = [\boldsymbol{\varphi}_1^u, \dots, \boldsymbol{\varphi}_{N_r^u}^u], \quad \mathbf{L}_p = [\boldsymbol{\varphi}_1^p, \dots, \boldsymbol{\varphi}_{N_r^p}^p]. \quad (54)$$

We assume then that the velocity and the pressure fields can be approximated by:

$$\mathbf{u} \approx \mathbf{u}_r = \sum_{i=1}^{N_r^u} a_i^u \boldsymbol{\varphi}_i^u, \quad p \approx p_r = \sum_{i=1}^{N_r^p} a_i^p \boldsymbol{\varphi}_i^p. \quad (55)$$

As shown in the previous section, the RB spaces are generated using a modified version of the mass matrix, which is defined as the ensemble average of all the mass matrices obtained during the training stage. In the case of a geometrically parametrized problem, Equations (54) and (55) can be rewritten as:

$$\mathbf{A}_u(\boldsymbol{\mu}_{k-1}, \boldsymbol{\mu}) \mathbf{u} = \mathbf{b}_u(\boldsymbol{\mu}, p), \quad \mathbf{A}_p(\boldsymbol{\mu}) p = \mathbf{b}_p(\boldsymbol{\mu}, \mathbf{u}). \quad (56)$$

The system matrices and the source terms of both the momentum and pressure equation are, therefore, parameter dependent. In a reduced setting, the SIMPLE algorithm can be rewritten following the procedure of Algorithm 3. For sake of clearness in the algorithm, the procedure is presented without additional hyperreduction but the offline and online parts can be properly decoupled relying on the D-EIM procedure shown for the heat transfer case. Therefore, during the online resolution, only pointwise evaluation of the discretized differential operators \mathbf{A}_u , \mathbf{A}_p and source terms \mathbf{b}_u , \mathbf{b}_p are required. The same concept is valid for the reconstruction part where it is required to reconstruct the solution \mathbf{u} and p only in some points of the domain identified by the magic points of the D-EIM algorithm. The spirit of the reduced algorithm introduced here is similar to the one introduced in Reference 53 with the main difference that, in the mentioned reference, the ROM is constructed only for the pressure equation and the velocity is reconstructed a posteriori from the pressure field. In the present case, we built instead a ROM which considers both the velocity and pressure fields at the same time and that completely mimic the full order SIMPLE algorithm.

Algorithm 3. The Reduced SIMPLE algorithm

Input: Tentative value of the velocity and pressure coefficient vectors \mathbf{a}_0^u and \mathbf{a}_0^p , $k = 1$, tolerance tol , $res = tol + 1$.

- 1: Reconstruct the first attempt full velocity and pressure fields: $\mathbf{u}_0 = \mathbf{L}_u \mathbf{a}_0^u$, $p_0 = \mathbf{L}_p \mathbf{a}_0^p$,
 - 2: **while** $res < tol$ **do**
 - 3: assemble reduced momentum equation $\mathbf{A}_u^r = \mathbf{L}_u^T \mathbf{A}_u(\boldsymbol{\mu}_{k-1}, \boldsymbol{\mu}) \mathbf{L}_u$; $\mathbf{b}_u^r = \mathbf{L}_u^T \mathbf{b}_u(\boldsymbol{\mu}, p_{k-1})$;
 - 4: compute reduced velocity residual $\rightarrow r_u = |\mathbf{A}_u^r \mathbf{a}_{k-1}^u - \mathbf{b}_u^r|$;
 - 5: solve $\mathbf{A}_u^r \mathbf{a}_k^u = \mathbf{b}_u^r$ and reconstruct $\mathbf{u}_k = \mathbf{L}_u \mathbf{a}_k^u$;
 - 6: assemble reduced pressure equation $\mathbf{A}_p^r = \mathbf{L}_p^T \mathbf{A}_p(\boldsymbol{\mu}) \mathbf{L}_p$; $\mathbf{b}_p^r = \mathbf{L}_p^T \mathbf{b}_p(\boldsymbol{\mu}, \mathbf{u}_k)$;
 - 7: compute reduced pressure residual $\rightarrow r_p = |\mathbf{A}_p^r \mathbf{a}_{k-1}^p - \mathbf{b}_p^r|$;
 - 8: solve $\mathbf{A}_p^r \mathbf{a}_k^p = \mathbf{b}_p^r$ and reconstruct $p_k = \mathbf{L}_p \mathbf{a}_k^p$;
 - 9: $res = \max(r_u, r_p)$, $k = k + 1$;
 - 10: **end while**
-

6.5 | A first numerical result on a geometrically parametrized incompressible flow problem

In this section, we present the numerical results for a geometrically parametrized incompressible Navier-Stokes problem. The case involves the parametrized angle of attack on a wing airfoil NACA 4412. In this case, since the focus is mainly the methodological development, we decided not to introduce additional complexities given by turbulence models and

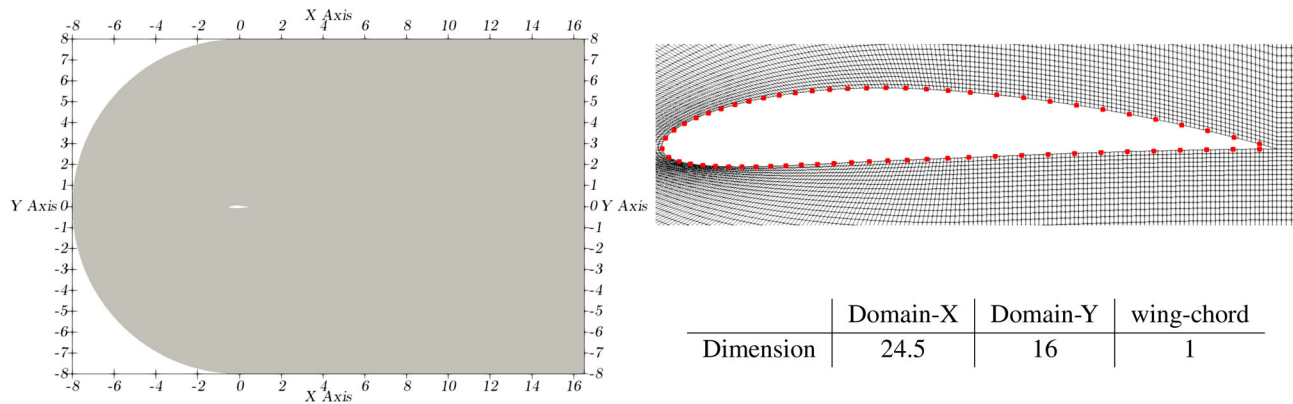


FIGURE 13 On the left, we report a sketch of the mesh for the geometrical parametrized wing problem, and on the right, we can observe the location of the 64 control points used to solve the RBF mesh motion problem together with the main dimensions of the domain [Colour figure can be viewed at [wileyonlinelibrary.com](#)]

FIGURE 14 The average of the L^2 relative error norm of pressure and velocity fields: $\frac{P_{FOM} - P_{ROM}}{P_{FOM}}$ and $\frac{(U_{FOM} - U_{\infty}) - (U_{ROM} - U_{\infty})}{U_{FOM} - U_{\infty}}$ where U_{∞} is the free stream velocity. The L^2 norm of the relative error for both pressure and velocity is plotted against the number of modes used for the reconstruction of the online solution in logarithmic y scale. In the same image, the reduced errors are compared with the projection errors obtained by the use of the same basis functions and by using the same norm [Colour figure can be viewed at [wileyonlinelibrary.com](#)]

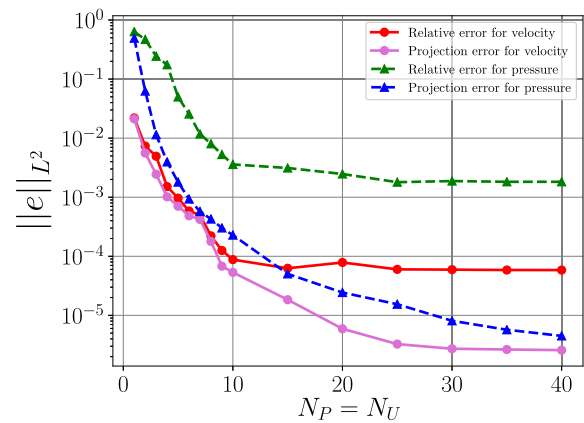
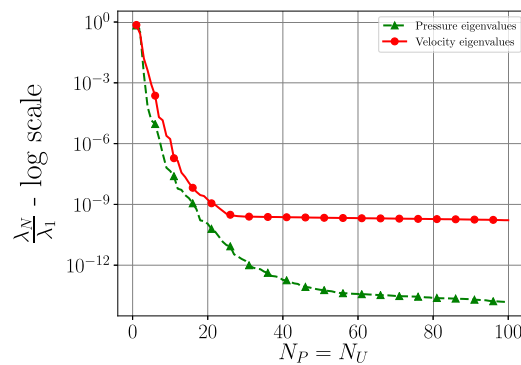


FIGURE 15 Plot reporting the eigenvalue decay relative to the POD procedure used to compute the modes for both pressure and velocity on the left, while on the right, the cumulates of the eigenvalues are reported [Colour figure can be viewed at [wileyonlinelibrary.com](#)]



| | Pressure | Velocity |
|--------|----------------|----------------|
| N = 1 | 0.674404466672 | 0.717187625760 |
| N = 2 | 0.997304641239 | 0.979072177457 |
| N = 5 | 0.999988310322 | 0.999728547289 |
| N = 7 | 0.999999521373 | 0.999981203705 |
| N = 10 | 0.999999956844 | 0.999999559123 |
| N = 20 | 0.99999999818 | 0.999999980819 |
| N = 40 | 0.99999999997 | 0.999999988151 |

we focused indeed on relatively small Reynolds number without turbulence modeling. The numerical test has been performed using only the RBF mesh motion strategy as introduced in the previous numerical example. In this case, the control points have been placed both on the moving boundaries and on the static patches. The locations on the airfoil are depicted in Figure 13. The kernel of the RBF interpolation is still given by Gaussian functions with a radius $d_{RBF} = 0.1$. In Figure 13, we report also the geometry of the problem with the dimensions of the domain. The domain is composed by a quadrilateral whose size is equal to 16.5 along the x direction, 16 along the y direction, and by a semicircle attached to the inlet side of the quadrilateral. The wing chord is equal to 1 and the foil is positioned on the center of the semicircle. The mesh counts 58 000 hexahedral cells. The geometrical parameter is given by the angle of attack that describes the rotation of the wing chord with respect to the inflow velocity. The deformed geometries are obtained rotating the wing around its barycenter. The training set $\mathcal{K}_{train} = \{\kappa_{i_{train}}\}_{i=1}^{N_{train}} \in [-10^\circ, 10^\circ]$ with $N_{train} = 100$ has been generated randomly inside the parameter space. The testing set $\mathcal{K}_{test} = \{\kappa_{i_{train}}\}_{i=1}^{N_{test}} \in [-9.5^\circ, 9.5^\circ]$ that has been used to verify the accuracy of the ROM

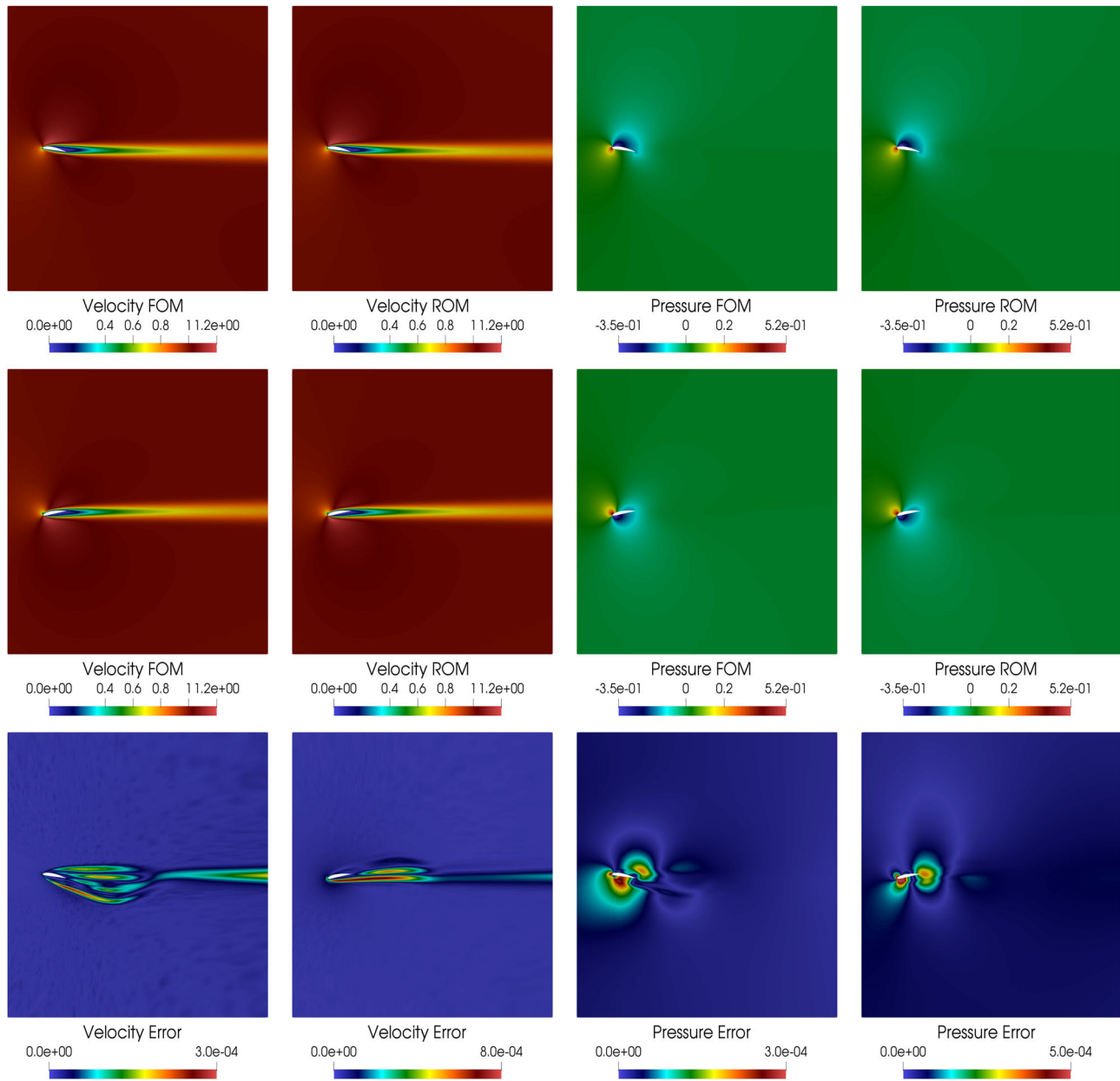


FIGURE 16 Plots of the full order and reduced order velocity and pressure fields and of the error between them for two selected extreme values inside the testing set ($\kappa_\alpha^1 = 9.15\circ$, $\kappa_\alpha^2 = -9.43\circ$). In the first and second row, from the left to the right, we report the FOM velocity field, the ROM velocity field, the FOM pressure field, and the ROM pressure field for the first and second testing points, respectively. In the third row, from the left to the right, we report the velocity error for the first testing point, the velocity error for the second testing point, the pressure error for the first testing point, and the pressure error for the second testing point, respectively [Colour figure can be viewed at wileyonlinelibrary.com]

counts $N_{\text{test}} = 50$ samples and has also been generated using a uniform random distribution. In the full order simulations, the convective term has been approximated using a second-order upwinding scheme, while the diffusive terms have been discretized with a linear scheme with nonorthogonal correction. The SIMPLE algorithm runs with under relaxation for both velocity and pressure using the relaxation factors $\alpha_u = 0.7$ and $\alpha_p = 0.3$. As mentioned in the previous section, the ROM has been constructed to be fully consistent with the SIMPLE procedure employed at the full order level. Therefore, the same under-relaxation factors are used also at the reduced order level. The velocity at the inlet is set constant and equal to 1 m/s. The kinematic viscosity is equal to $\nu = 3 \times 10^{-3}$ m²/s. In Figure 14, we show the convergence properties of the ROM changing the number of RB functions used to approximate both velocity and pressure. In Figure 15 it

FIGURE 17 Shape of the five different bump functions, where C is the chord of the airfoil [Colour figure can be viewed at wileyonlinelibrary.com]

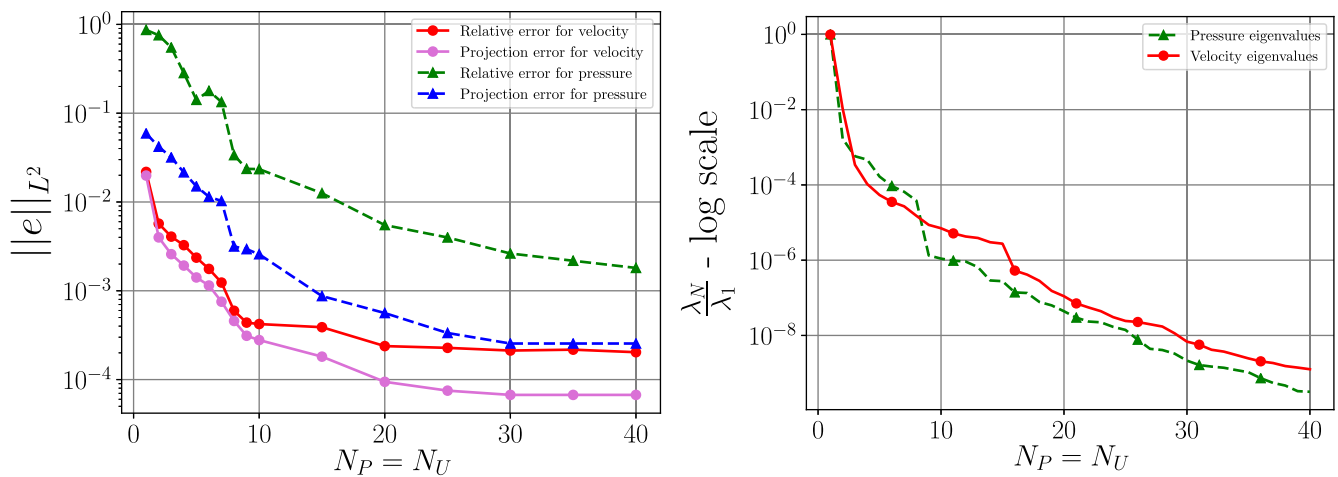
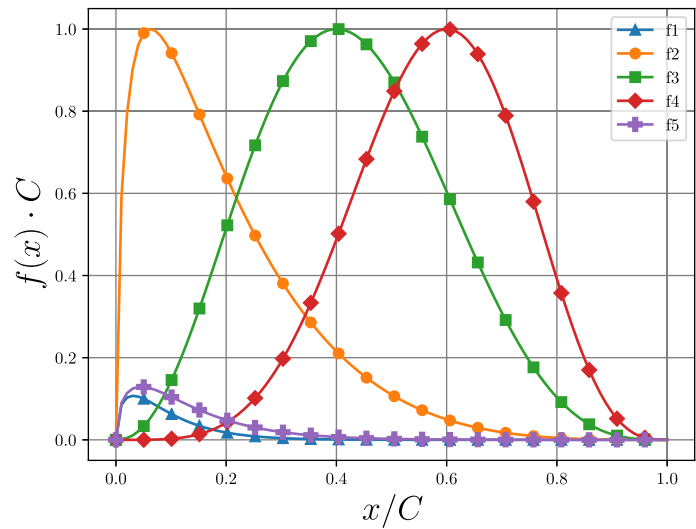


FIGURE 18 The average of the L^2 relative error norm of pressure and velocity fields has been reported on the left: $\frac{P_{\text{FOM}} - P_{\text{ROM}}}{P_{\text{FOM}}}$ and $\frac{(U_{\text{FOM}} - U_{\infty}) - (U_{\text{ROM}} - U_{\infty})}{U_{\text{FOM}} - U_{\infty}}$ where U_{∞} is the free stream velocity. The L^2 norm of the relative error for both pressure and velocity is plotted against the number of modes used for the reconstruction of the online solution in logarithmic y scale. In the same image, the reduced errors are compared with the projection errors obtained by the use of the same basis functions and by using the same norm. On the right, the eigenvalue decay relative to the POD procedure used to compute the modes for both pressure and velocity is shown [Colour figure can be viewed at wileyonlinelibrary.com]

is reported the eigenvalue decay of the POD procedure for both the velocity and the pressure field, while in Figure 16, we show a qualitative comparison between the full order and reduced order velocity and pressure fields for two selected values inside the testing set. The model produces accurate results for both velocity and pressure and more importantly it does not require any additional stabilization, which is typical of ROMs for incompressible flows.¹⁷ We believe that this fact is a consequence of the SIMPLE algorithm that has been used also at the reduced order level.

6.6 | A second numerical result on a geometrically parametrized incompressible flow problem

In this section, we show the numerical results obtained exploiting the segregated method explained in the previous sections on a shape deformation test case. In particular, by following what has been done in Reference 22, we want to tackle the shape deformation of a NACA 4412 airfoil by the superposition of some bump functions (Figure 17) to the geometry. The same case and strategy analyzed in the previous section are taken into consideration by only changing the geometry

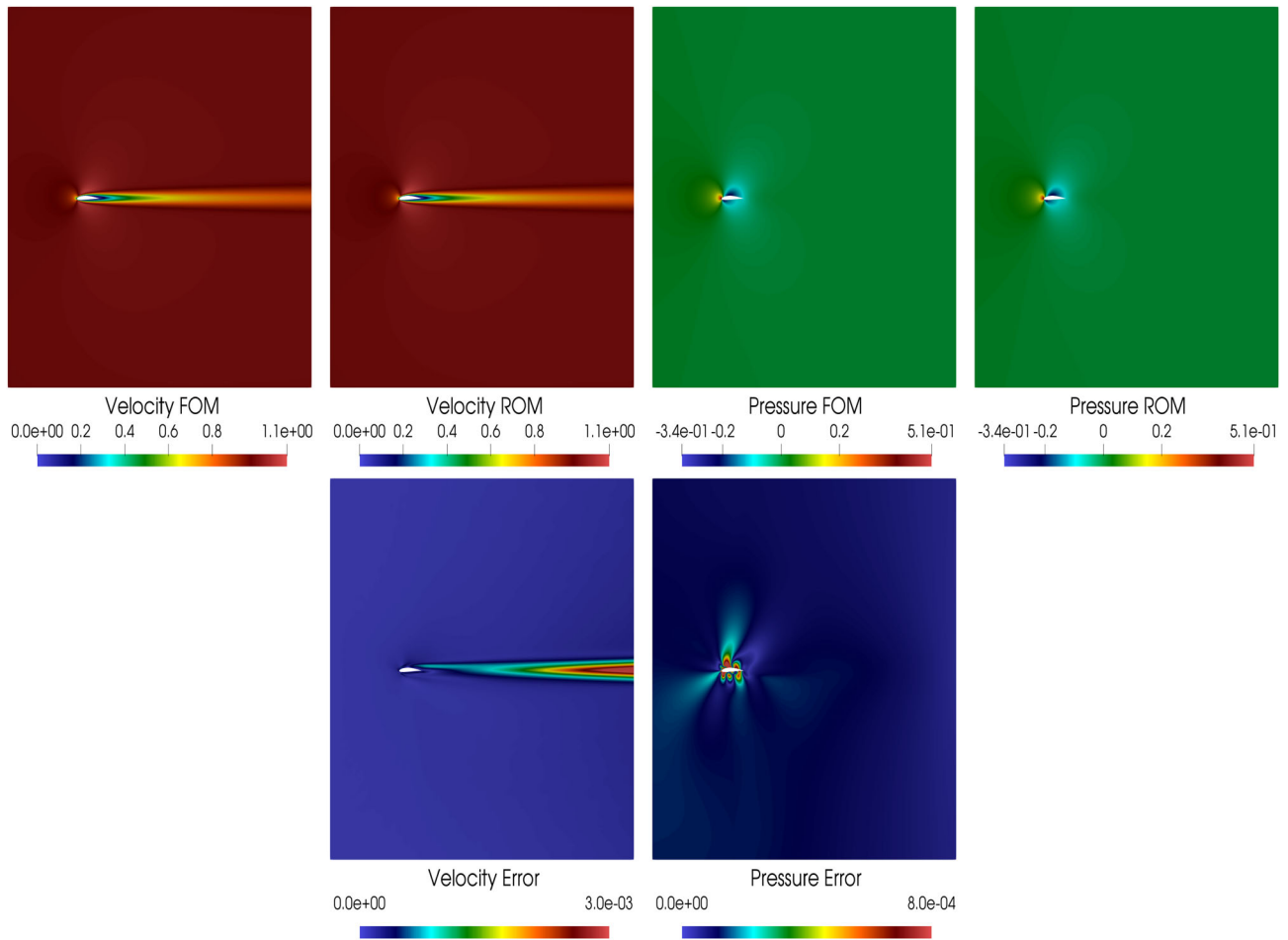


FIGURE 19 Plots of the full order and reduced order velocity and pressure fields and of the error between them for a set of shape parameters inside the testing set ($[1.68124, 0.01488, 0.98581, 0.27690, 0.91593, 0.30225, 0.93038, 1.18651, 0.00667, 1.05945] \times 10^{-2}$). In the first row, from the left to the right, we report the FOM velocity field, the ROM velocity field, the FOM pressure field, and the ROM pressure field, respectively. In the second row, from the left to the right, we report the velocity error and the pressure error, respectively. Online solutions have been obtained by the use of 30 basis functions [Colour figure can be viewed at wileyonlinelibrary.com]

of the foil. All the five bump functions are added to the top of the airfoil and subtracted to the bottom so that intersections between the boundaries are avoided, thus the problem is parametrized by the use of 10 shape coefficients. The angle of attack of the reference foil is always set to zero while solving the problem. The training set $\mathcal{K}_{\text{train}} = \{\kappa_{i_{\text{train}}}\}_{i=1}^{N_{\text{train}}} \in [0, 0.02]^{10}$, where each row contains 10 different values and $N_{\text{train}} = 1000$, has been generated randomly inside the parameter space. The testing set $\mathcal{K}_{\text{test}} = \{\kappa_{i_{\text{train}}}\}_{i=1}^{N_{\text{test}}} \in [0, 0.018]^{10}$ that has been used to verify the accuracy of the ROM counts $N_{\text{test}} = 20$ samples and has also been generated using a uniform random distribution. In Figure 18, the trends of the errors and of the eigenvalues for both velocity and pressure are depicted. Figure 19 shows the comparison between full order model (FOM) and ROM solutions for both velocity and pressure for this last test case where one of the 20 online parameter sets has been employed to modify the shape of the airfoil. As one can notice the accordance between the two solutions is high and the accuracy of the method is satisfying.

7 | CONCLUSIONS AND FUTURE PERSPECTIVES

In this work, we presented a reduced order modeling strategy for geometrical parametrization starting from a full order FV discretization. The methodology makes use of an ALE approach to analyze all the possible parametrized configurations and of a modified inner product for the computation of the correlation matrices used during the POD procedure. During the online stage, in order to ensure an efficient offline-online decoupling, we make extensive use of the D-EIM at both matrix and vector level.

Different mesh motion strategies have been tested and compared. These are based on a Laplacian smoothing approach and on an RBF approach. Both methodologies have been analyzed with and without nonorthogonality correction and their perspectives in view of model reduction purposes have been deeply analyzed.

Both methodologies have been tested on a heat transfer problem and on an incompressible steady Navier-Stokes setting providing accurate results and great computational gains. The RBF approach proved to be more accurate in the presence of large parametric geometrical variations.

As future perspectives, it will be certainly interesting also to use the developed methodology to perform shape optimization in computational fluid dynamic problems at higher Reynolds number¹ and to couple the present approach to what developed in References 19 and 54 for turbulent flows.

Of interest is also to study the applicability of the method to deal with complex geometrical variations arising from free form deformation algorithms.

Moreover, the interest is also into studying fluid-structure interaction transient problem to deal with parametric interfaces.

ACKNOWLEDGEMENTS

We acknowledge the support provided by the European Research Council Consolidator Grant project Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics - GA 681447, H2020-ERC COG 2015 AROMA-CFD, as well as MIUR FARE-X-AROMA-CFD (Italian Ministry for Education University and Research) and INdAM-GNCS (Istituto Nazionale di Alta Matematica - Gruppo Nazionale di Calcolo Scientifico). The computations in this work have been performed with *ITHACA-FV*,⁵⁰ developed at SISSA mathLab, which is an implementation in OpenFOAM⁵¹ of several reduced order modeling techniques; we acknowledge developers and contributors to both libraries.

NOMENCLATURE

| | |
|--|--|
| ROM | reduced order model |
| POD | proper orthogonal decomposition |
| ALE | arbitrary Lagrangian Eulerian |
| D-EIM | discrete empirical interpolation methods |
| RBF | radial basis function |
| $\Omega(\boldsymbol{\mu})$ | parametrized domain |
| $\Gamma_D(\boldsymbol{\mu})$ | parametrized Dirichlet boundary |
| $\Gamma_N(\boldsymbol{\mu})$ | parametrized Neumann boundary |
| $\mathcal{N}(\mathbf{x}, u(\boldsymbol{\mu}))$ | generic parametrized differential operator |
| \square_i | variable at the center of the cell i |
| \square_{if} | variable at the center of the face f of cell i |
| \square_n | normal derivative |
| $\langle \cdot, \cdot \rangle_F$ | Frobenious inner product |
| \mathcal{T} | FV tessellation |
| θ | temperature field |
| θ_h | FOM solution for θ |
| θ_{rb} | ROM solution for θ |
| \mathbf{M}' | Modified mass matrix |
| $\ \cdot \ _{\Omega}$ | L_2 norm over the computational domain Ω |
| \mathbf{A} | discrete FOM Laplace operator |
| \mathbf{f} | discrete FOM source term |
| \mathbf{A}^r | discrete ROM Laplace operator |
| \mathbf{f}^r | discrete ROM source term |
| \mathbf{a}^θ | reduced coefficients for θ |
| N_h | number of DOFs for the FOM |
| N_A | number of D-EIM modes for \mathbf{A} |
| N_f | number of D-EIM modes for \mathbf{f} |
| N_r | number of DOFs for the ROM |
| N_{train} | number of training samples |
| N_{test} | number of testing samples |

| | |
|------------------------------|--|
| N_b | number of the RBF boundary points |
| $\mathcal{K}_{\text{train}}$ | discrete training set |
| $\mathcal{K}_{\text{test}}$ | discrete testing set |
| \mathcal{P} | parameter space |
| μ | parameter vector |
| $\kappa_{i_{\text{train}}}$ | sample point in the training set |
| $\kappa_{i_{\text{test}}}$ | sample point in the testing set |
| \mathbf{S}_θ | snapshots matrix for Temperature |
| \mathbf{S}_A | snapshots matrix for \mathbf{A} matrix |
| \mathbf{S}_f | snapshots matrix for \mathbf{f} vector |
| \mathbf{S}_M | snapshots matrix for \mathbf{M} matrix |
| φ_i^θ | basis function for θ |
| χ_k^A | matrix D-EIM basis for \mathbf{A} |
| χ_f^A | vector D-EIM basis for \mathbf{f} |
| \mathbf{b}^A | D-EIM coefficients for \mathbf{A} |
| \mathbf{c}^f | D-EIM coefficients for \mathbf{f} |

ORCID

Giovanni Stabile  <https://orcid.org/0000-0003-3434-8446>

Gianluigi Rozza  <https://orcid.org/0000-0002-0810-8812>

REFERENCES

- Lombardi M, Parolini N, Quarteroni A, Rozza G. Numerical Simulation of Sailing Boats: Dynamics, FSI, and Shape Optimization. *Variational Analysis and Aerospace Engineering: Mathematical Challenges for Aerospace Design*. Boston, MA: Springer; 2012:339-377.
- Quarteroni A, Manzoni A, Negri F. *Reduced Basis Methods for Partial Differential Equations*. Vol 92. Switzerland: Springer International Publishing; 2016.
- Hesthaven JS, Rozza G, Stamm B. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Vol 590. Berlin, Germany: Springer International Publishing; 2016.
- Benner P, Ohlberger M, Patera A, Rozza G, Urban K. *Model Reduction of Parametrized Systems*. MS&A series. Vol 17. Cham, Switzerland: Springer; 2017.
- Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch Comput Methods Eng*. 2008;15(3):229-275.
- Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: M2AN*. 2005;39(1):157-181.
- Veroy K, Prud'homme C, Patera AT. Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *Comptes Rendus Mathématique*. 2003;337(9):619-624.
- Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: M2AN*. 2007;41(3):575-605.
- Rozza G, Manzoni A. Model order reduction by geometrical parametrization for shape optimization in computational fluid dynamics. Paper presented at: Proceedings of the ECCOMAS CFD 2010, V European Conference on Computational Fluid Dynamics. 2010; MATHICSE Report 05; 2010.
- Rozza G, Huynh DBP, Manzoni A. Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numer Math*. 2013;125(1):115-152.
- Rozza G. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Commun Comput Phys*. 2011;9(01):1-48.
- Iapichino L, Quarteroni A, Rozza G. Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. *Comput Math Appl*. 2016;71(1):408-430.
- Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *Math Model Numer Anal*. 2008;42(2):277-302.
- Haasdonk B, Ohlberger M, Rozza G. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *ETNA, Electr Trans Numer Anal*. 2008;32:145-161.
- Drohmann M, Haasdonk B, Ohlberger M. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J Sci Comput*. 2012;34(2):A937-A969.
- Stabile G, Hijazi S, Mola A, Lorenzi S, Rozza G. POD-Galerkin reduced order methods for CFD using finite volume discretisation: Vortex shedding around a circular cylinder. *Commun Appl Ind Math*. 2017;8(1):210-236.

17. Stabile G, Rozza G. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Comput Fluids*. 2018;173:273–284.
18. Carlberg K, Choi Y, Sargsyan S. Conservative model reduction for finite-volume models. *J Comput Phys*. 2018;371:280–314.
19. Hijazi S, Stabile G, Mola A, Rozza G. Data-driven POD–Galerkin reduced order model for turbulent flows; 2019. Submitted, preprint arXiv:1907.09909.
20. Lorenzi S, Cammi A, Luzzi L, Rozza G. POD-Galerkin method for finite volume approximation of Navier–Stokes and RANS equations. *Comput Methods Appl Mech Eng*. 2016;311:151–179.
21. Georgaka S, Stabile G, Rozza G, Bluck MJ. Parametric POD-Galerkin model order reduction for unsteady-state heat transfer problems. *Commun Comput Phys*. 2019;27(1):1–32.
22. LeGresley P, Alonso J. Investigation of non-linear projection for POD based reduced order models for aerodynamics. Paper presented at: Proceedings of the 39th Aerospace Sciences Meeting and Exhibit American Institute of Aeronautics and Astronautics; 2001.
23. Zimmermann R, Görtz S. Non-linear reduced order models for steady aerodynamics. *Proc Comput Sci*. 2010;1(1):165–174.
24. Washabaugh KM, Zahr MJ, Farhat C. On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. Paper presented at: Proceedings of the 54th AIAA Aerospace Sciences Meeting American Institute of Aeronautics and Astronautics; 2016.
25. Zahr MJ, Farhat C. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *Int J Numer Methods Eng*. 2014;102(5):1111–1135.
26. Ballarin F, Manzoni A, Quarteroni A, Rozza G. Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *Int J Numer Methods Eng*. 2014;102(5):1136–1161.
27. Milani R, Quarteroni A, Rozza G. Reduced basis method for linear elasticity problems with many parameters. *Comput Methods Appl Mech Eng*. 2008;197(51–52):4812–4829.
28. Jäggi C, Iapichino L, Rozza G. An improvement on geometrical parameterizations by transfinite maps. *Comptes Rendus Mathématique*. 2014;352(3):263–268.
29. Lassila T, Rozza G. Parametric free-form shape design with PDE models and reduced basis method. *Comput Methods Appl Mech Eng*. 2010;199(23–24):1583–1592.
30. Rozza G. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Comput Vis Sci*. 2009;12(1):23–35.
31. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J Comput Phys*. 1997;135(2):203–216.
32. Jasak H. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows (PhD thesis). Imperial College, University of London; 1996.
33. Drohmann M, Haasdonk B, Ohlberger M. Reduced basis method for finite volume approximation of evolution equations on parametrized geometries. Paper presented at: Proceedings of ALGORITMY 2009; 2009:111–120.
34. Karatzas EN, Stabile G, Atallah N, Scovazzi G, Rozza G. A reduced order approach for the embedded shifted boundary fem and a heat exchange system on parametrized geometries. Paper presented at: Proceedings of the IUTAM Symposium on Model Order Reduction of Coupled Systems; May 22–25, 2018: 111–125; Stuttgart, Germany, Springer International Publishing; 2019.
35. Karatzas EN, Stabile G, Nouveau L, Scovazzi G, Rozza G. A reduced basis approach for PDEs on parametrized geometries based on the shifted boundary finite element method and application to a Stokes flow. *Comput Methods Appl Mech Eng*. 2019;347:568–587.
36. Karatzas EN, Ballarin F, Rozza G. Projection-based reduced order models for a cut finite element method in parametrized domains. *Comput Math Appl*. 2020;79(3):833–851.
37. Jasak H. Automatic mesh motion for the unstructured finite volume method. *Trans Famena*. 2006;30(2):1–20.
38. Boer A, Schoot MS, Bijl H. Mesh deformation based on radial basis function interpolation. *Comput Struct*. 2007;85(11–14):784–795.
39. Beckert A, Wendland H. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerosp Sci Technol*. 2001;5(2):125–134.
40. Chinesta F, Huerta A, Rozza G, Willcox K. Model reduction methods. *Encyclopedia of Computational Mechanics*. Hoboken, New Jersey: John Wiley & Sons; 2017:1–36.
41. Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng*. 2011;18(4):395.
42. Dumon A, Allery C, Ammar A. Proper general decomposition (PGD) for the resolution of Navier–Stokes equations. *J Comput Phys*. 2011;230(4):1387–1407.
43. Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J Numer Anal*. 2002;40(2):492–515.
44. Barrault M, Maday Y, Nguyen NC, Patera A. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*. 2004;339(9):667–672.
45. Chaturantabut S, Sorensen FC. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput*. 2010;32(5):2737–2764.
46. Willcox K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Comput Fluids*. 2006;35(2):208–226.
47. Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J Comput Phys*. 2013;242:623–647.
48. Carlberg K, Bou-Mosleh C, Farhat C. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Int J Numer Methods Eng*. 2010;86(2):155–181.

49. Negri F, Manzoni A, Amsallem D. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J Comput Phys*. 2015;303:431-454.
50. Stabile G, Rozza G. ITHACA-FV - In real Time Highly Advanced Computational Applications for Finite Volumes. <http://www.mathlab.sissa.it/ithaca-fv>. Accessed December 28, 2018.
51. OpenFOAM documentation website. <https://openfoam.org/>. Accessed December 28, 2018.
52. Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int J Heat Mass Transf*. 1972;15(10):1787-1806.
53. Isoz M. POD-DEIM based model order reduction for speed-up of flow parametric studies. *Ocean Eng*. 2019;186:106083.
54. Hijazi S, Ali S, Stabile G, Ballarin F, Rozza G. The effort of increasing reynolds number in projection-based reduced order methods: from laminar to turbulent flows. *Numerical Methods for Flows. FEF Special Volume, Springer LNCSE Series*. Vol 132. Switzerland: Springer-Verlag; 2019.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Stabile G, Zancanaro M, Rozza G. Efficient geometrical parametrization for finite-volume-based reduced order methods. *Int J Numer Methods Eng*. 2020;121:2655-2682. <https://doi.org/10.1002/nme.6324>