V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

MODEL ORDER REDUCTION BY GEOMETRICAL PARAMETRIZATION FOR SHAPE OPTIMIZATION IN COMPUTATIONAL FLUID DYNAMICS[‡]

Gianluigi Rozza[*], Andrea Manzoni[*]

[*] CMCS - Modelling and Scientific Computing
MATHICSE - Mathematics Institute of Computational Science and Engineering
EPFL - Ecole Polytechnique Fédérale de Lausanne
Station 8, CH - 1015 Lausanne, Switzerland
e-mail: gianluigi.rozza@epfl.ch, andrea.manzoni@epfl.ch

Key words: Model Order Reduction; Reduced Basis Methods; Shape Parametrization; Flow Control; Shape Optimization; Stokes Equations.

Abstract. *Shape Optimization problems governed by partial differential equations result from many applications in computational fluid dynamics; they involve the repetitive evaluation of outputs expressed as functionals of the field variables and usually imply big computational efforts. For this reason looking for computational efficiency in numerical methods and algorithms is mandatory. The interplay between scientific computing and new reduction strategies is crucial in applications of great complexity. In order to achieve an efficient model order reduction, reduced basis methods built upon a high-fidelity "truth" finite element approximation – and combined with suitable geometrical parametrization techniques for efficient shape description – can be introduced, thus decreasing both the computational effort and the geometrical complexity.*

Starting from an excursus on classical approaches – such as local boundary variation and shape boundary parametrization – we focus on more efficient parametrization techniques which are well suited for a combination with a reduced basis approach, such as the one based on affine mapping (even automatic), nonaffine mapping (coupled with a suitable empirical interpolation technique for better numerical performances) and free-form deformations. We thus describe (and compare) the principal features of these parametrization techniques by showing some applications dealing with shape optimization of parametrized configurations in viscous flows, and discussing computational advantages and efficiency obtained by geometrical and computational model order reduction.

[‡]This work has been supported in part by the Swiss National Science Foundation (Project 200021-122136). We are grateful to the family of Prof. F. Saleri for granting the use of the `MLife` library as a basis for FE simulations. We acknowledge Prof. A.T. Patera and his group for the `rbMIT` software used for RB computations. We thank Prof. A. Quarteroni and T. Lassila for their feedbacks and C. Gunther for data provided.

1 INTRODUCTION

In many engineering contexts, the main focus when investigating fluid flow processes is to control and optimize by influencing some parameters such that some outputs of interest are minimized or maximized. The reduction of drag forces acting on airfoils or the wall shear stress minimization in a cardiovascular bypass anastomosis are just two examples from very different fields, which can be solved using the same tools and strategies.[1,2,3]

Optimal flow control problems can thus be formulated as the minimization of a given cost functional (or output) controlling some physical parameters (such as model coefficients, source terms or boundary values) or, alternatively, some geometrical quantities. We refer to this second case such as shape optimization problems.[4] Such problems involve the study of a system of partial differential equations (PDEs) and the evaluation of an output depending on the field variables, combining flow simulation, mathematical optimization and shape variation.[5] Since optimization procedures require repetitive evaluations of the output and the flow field, computational costs can become unacceptably high and strategies to reduce numerical efforts and model order are being developed.
We thus intend to focus on different shape parametrization techniques in order to manage with the geometrical complexity. Moreover, we are interested in using reduced basis (RB) methods, since they provide a great computational reduction in a multi-query context like shape optimization. To do this, we need to write our problem on a reference domain, and to freeze geometric variations in the coefficients of the equation by using a suitable parametric mapping. Since classical approaches such as *local boundary variation* and *shape boundary parametrization* are not well suited for being used together with RB methods, we need to introduce different and more advanced shape parametrization strategies.

Depending on the role played by design variables (or input parameters) we have different shape parametrization techniques. When input parameters represent geometrical properties (such as lengths, angles, diameters) we may divide the flow domain in different subdomains and build *affine or nonaffine mappings* on single subdomains. On the other hand, if input parameters are introduced as coefficients of a shape basis vector, they can represent a set of control points describing directly a shape boundary or giving some free-form perturbations of the metric of the whole domain. In both cases the parametrization involves naturally some given shape families, such as Bezier curves or B-splines; nevertheless, within the second approach (the so-called *free-form deformation techniques*), design parameters are not directly connected neither to geometrical properties nor to the shape boundary, and parametric mappings turn out to be flexible even if very low-dimensional.

The paper is organized as follows: after a short summary of the most important features related with shape optimization by flow control (Section 2), we will review in Section 3 two relevant parametrization techniques which historically have been proposed. Basic aspects of reduced basis methods are presented in Section 4, while new geometrical parametrization approaches – well suited to RB methods - are introduced in Section 5. Finally, some applications will be presented in Section 6 and conclusions follow in Section 7.

2 SHAPE OPTIMIZATION BY FLOW CONTROL

The goal of a shape optimization problem by flow control is minimizing a given cost functional by finding the optimal shape of the domain where the PDEs problem is defined. In an abstract setting depending (for the sake of simplicity) on flow scalar variables, given a set of admissible shapes $\mathcal{O}_{ad}$, we thus aim in finding the optimal shape $\hat{\Omega}_o \in O_{ad}$ s.t.

$$\hat{\Omega}_o = \arg \min_{\Omega_o \in \mathcal{O}_{ad}} \mathcal{J}(\Omega_o, y(\Omega_o)) \quad \text{with} \quad a(y(\Omega_o), \phi) = F(\phi) \;\; \forall \phi \in V, \tag{1}$$

where $y = y(\Omega_o) \in V$ is the flow (state) variable, V an Hilbert space, $a(\cdot, \cdot)$ a bilinear form on $V \times V$ and $F(\cdot)$ a linear form on V. Hence, a shape optimization framework consists of three main stages: the flow computation, the optimization algorithm and a suitable approach for representing and deforming efficiently the shape of underlying geometry. Such a problem usually involves high computational costs since *(i)* iterative procedures for optimization require the evaluation of field $y(\Omega_o)$ and cost functional $\mathcal{J}(\Omega_o, y(\Omega_o))$ for many possible configurations; *(ii)* PDEs can be expensive to solve (great number of unknowns, nonlinearities, fine physical details to be captured by very fine meshes) and *(iii)* discretization assembling procedures are expensive when geometry keeps changing.

For these reasons, both geometrical and computational reduction are mandatory; the use of RB methods, together with suitable shape parametrizations, allows to get these reductions. We target this work on geometrical reduction through efficient shape parametrizations; since first options historically introduced are not well indicated also for a more effective computational reduction, more general *ad hoc* procedures have been developed.

2.1 An abstract setting for shape optimization

For the analysis of the problem (1) we can use the framework based on the *Lagrangian functional*,[5,6] which treats a shape optimization problem as a constrained minimization problem. We can define the Lagrangian functional associated to (1) as

$$\mathcal{L}(\Omega, y, p) := \mathcal{J}(\Omega, y) + F(p) - a(y, p) \tag{2}$$

where $p \in V$ is the Lagrangian multiplier. The optimum $(\hat{\Omega}, \hat{y}(\hat{\Omega}), \hat{p}(\hat{\Omega})) \in \mathcal{O}_{ad} \times V \times V$, whether exists, is a stationary point of the Lagrangian functional $\mathcal{L}(\Omega, y, p)$ and fullfills[7] the following (first order) optimality conditions[a]:

$$\begin{array}{lll} \partial_p \mathcal{L}(\Omega, y(\Omega), p(\Omega))[\phi] = 0 & \forall\, \phi \in V & \textit{state equation} \\ \partial_y \mathcal{L}(\Omega, y(\Omega), p(\Omega))[\psi] = 0 & \forall\, \psi \in V & \textit{adjoint equation} \\ \partial_\Omega \mathcal{L}(\Omega, y(\Omega), p(\Omega))[\mathbf{V}] = 0 & \forall\, \mathbf{V} \in S & \textit{optimality condition.} \end{array} \tag{3}$$

The second equation of the system (3) is the *adjoint equation* (for which $p(\Omega) \in V$ is defined as the adjoint variable) whose weak form reads as follows:

[a] This abstract formulation is valid if V does not depend on Ω_o; the same result stands also for problems in which $V = V(\Omega_o)$ introducing suitable Lagrange multipliers for treating Dirichlet boundary conditions. See the following footnote b for the definition of the space S and of speed fields $\mathbf{V}$.

$$\text{find } \; p(\Omega) \in V(\Omega) \; : \; a'(y(\Omega))(p(\Omega), \psi) = \partial_y \mathcal{J}(\Omega, y(\Omega))[\psi] \qquad \forall \psi \in V(\Omega). \tag{4}$$

Finally, by differentiating the Lagrangian functional with respect to the shape[b], we obtain

$$\partial_\Omega \mathcal{L}(\Omega, y(\Omega), p(\Omega))[\mathbf{V}] = d\tilde{\mathcal{J}}(\Omega; \mathbf{V}),$$

from which it is possible to extract the gradient of the cost functional, needed e.g. for gradient-based numerical optimization procedures.

2.2 Numerical strategies for shape optimization

From a numerical point of view, discretization techniques (e.g. finite elements or finite volumes) are employed for approximating the solution of state and adjoint equations in (3), together with a numerical optimization algorithm. Gradient based procedures, quasi Newton methods (like sequential quadratic programming) or other nonlinear programming techniques can be used for this goal. In order to manage with shape deformations and numerical computations of shape sensitivities and/or derivatives, some shape parametrization has to be introduced; in principle, any method which realizes a geometry representation is applicable (e.g. simply grid locations). However, the greater is the number of design variables, the higher will be the required computational costs: thus, a trade-off between low-dimensionality and flexibility is a key issue in defining a shape parametrization. Let us introduce a parameter vectors $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_P) \in \mathcal{D} \subset \mathbb{R}^P$ and a generical parametrized "original" configuration of the domain $\Omega_o(\boldsymbol{\mu})$; depending on the role played by parameters $\mu_1, \ldots, \mu_P$, we will have different parametrization techniques; in each case, problem (1) is translated into the following parametrized optimization problem:

$$\text{find } \hat{\boldsymbol{\mu}} \in \mathcal{D} = \arg\min_{\boldsymbol{\mu} \in \mathcal{D}} J(\boldsymbol{\mu}, y(\boldsymbol{\mu})) \quad \text{ with } \quad a(y(\boldsymbol{\mu}), \phi; \boldsymbol{\mu}) = F(\phi; \boldsymbol{\mu}) \;\; \forall \phi \in V, \tag{5}$$

where $J(\boldsymbol{\mu}, \cdot)$, $a(\cdot, \cdot; \boldsymbol{\mu})$ and $F(\cdot; \boldsymbol{\mu})$ are the corresponding parametrized cost functional, bilinear and linear forms, respectively (see Section 5.3). Thus, an optimization result may depend on the order/complexity of parametrization, on the number of design variables and their relationship with geometry, and on shape deformations by parameters variation. Moreover, we have different behaviors in terms of geometry smoothness, local shape control and grid deformation. Different parametrization approaches will be compared with respect to these features in Section 6. Let us now briefly discuss some classical parametrization techniques deeply used in shape optimization.

[b] The *speed method*[8] is a general technique for setting an equivalence between domain transformations and speed fields. Given a shape $\Omega_o \subset \mathbb{R}^2$ to be optimized by acting on its boundary $\Gamma_c^o \subseteq \partial\Omega_o$, we can introduce a speed field $\mathbf{V} : [0, \tau] \times \mathbb{R}^2 \to \mathbb{R}^2$ such that $\mathbf{V}(\cdot, \mathbf{x}) \in \mathcal{C}^{0,1}([0, \tau]; \mathbb{R}^2) \; \forall \mathbf{x} \in \mathbb{R}^2$ and that $\mathbf{V}(t, \cdot) = \mathbf{0}$ on $\partial\Omega \setminus \Gamma_c^o$ for all $t \in [0, \tau]$; S is the space of speeds satisfying these properties. Under the action of the speed $\mathbf{V}$, the domain Ω_o can be mapped into a new perturbed domain $\Omega_o^t(\mathbf{V}) := \mathcal{T}_\mathbf{V}(\Omega_o; t)$, for $t \geq 0$, where $\mathbf{x} \to \mathcal{T}_\mathbf{V}(\mathbf{x}; t) \equiv \mathbf{x}_t(t)$ is given as the solution of $\dot{\mathbf{x}}_t(t) = \mathbf{V}(t, \mathbf{x}_t(t)), t \in [0, \tau]$, with $\mathbf{x}_t(0) = \mathbf{x}$. Thus, considering $\tilde{\mathcal{J}}(\Omega_o) = \mathcal{J}(\Omega_o, y(\Omega_o))$, its *shape derivative*, at Ω_o in direction $\mathbf{V}$, is given by $d\tilde{\mathcal{J}}(\Omega_o; \mathbf{V}) = j'(0)$, where $j(t) = \tilde{\mathcal{J}}(\Omega_o^t)$; furthermore, if the map $\mathbf{V} \mapsto d\tilde{\mathcal{J}}(\Omega_o; \mathbf{V})$ is linear and continuous, it is denoted $\nabla\tilde{\mathcal{J}}(\Omega_o)$ and referred to as the *(shape) gradient* of $\tilde{\mathcal{J}}$.

3 CLASSICAL PARAMETRIZATION APPROACHES

Since shape representation is highly problem-dependent, various methods have been proposed; following Samareh,[9] we distinguish among discrete approaches, polynomial or spline parametrizations, domain element approaches and free-form deformations[c]. After a brief historical survey, we focus on suitable techniques within RB framework.

3.1 Local boundary variation by mesh nodes displacement

One of the most common strategy for shape deformation uses the coordinates of the boundary points as design variables. In this case, during the optimization steps, the shape is deformed through a local boundary variation (LBV),[7] by defining a deformation field acting on the boundary nodes. In this way, starting from an initial shape $\Omega_o^{(0)}$, the new shape $\Omega_o^{(k+1)}, k > 0$ is obtained by $\Omega_o^{(k)}$ through the transformation $\mathcal{T}_{\mathbf{V}^{(k)}}(\cdot)$ depending on the *speed vector* $\mathbf{V}^{(k)}$ related to the gradient $\nabla\mathcal{J}(\Omega_o^{(k)})$; the simplest transformation is an identity perturbation $\mathcal{T}_{\mathbf{V}}(\cdot;t) = (I + t\mathbf{V})(\cdot)$, leading to the following updating relation:

$$\Omega_o^{(k+1)} = \Omega_o^{(k)} + \tau^{(k)}\mathbf{V}^{(k)}, \qquad k > 0.$$

In this way, the whole computational mesh is deformed, and no remeshing is *a priori* requested. For complex shapes or flows, we need very fine meshes, leading to very large number of design variables and thus to high computational costs. A local shape control is allowed, even if geometrical smoothness is not assured; moreover, also the selection of step-size $\tau^{(k)}$ may be very difficult, impacting deeply on the stability of the algorithm.

3.2 Analytical shape boundary parametrization

Another classical approach largely adopted in shape optimization is based on a polynomial (or spline) shape parametrization (PSP), which helps in reducing sensibly the number of design variables. In this case, shape is described by a control point set through which we combine some basis shapes chosen in a given family, such as Bezier curves, B-splines or Nurbs.[6,10,11] Within this approach, shape is described by a curve $\phi : [0,1] \to \mathbb{R}^2$ s.t.

$$s \to \boldsymbol{\phi}(s) = \sum_{i=1}^{P} \boldsymbol{\mu}_i \varphi_i^n(s),$$

where P is the number of control points $\{\boldsymbol{\mu}_i = (\mu_i^1, \mu_i^2)\}_{i=1}^P$ and the $\{\varphi_i^n(s)\}_{i=1}^P$ are e.g. Bernstein polynomials (for Bezier representation of simple curves) of degree n or B-spline[d] basis functions of degree n. In the first case $\varphi_i^n(s) = \binom{n}{i} s^i (1-s)^{n-i}$, and control points are strictly related to the curve position, since the curve is contained into the convex hull of the Bezier control polygon. Within a polynomial parametrization, a descent algorithm can be applied directly to the control points $\boldsymbol{\mu}_i$, giving $\boldsymbol{\mu}_i^{(k+1)} = \boldsymbol{\mu}_i^{(k)} - \boldsymbol{\tau}^k \partial\mathcal{J}/\partial\boldsymbol{\mu}_i(\boldsymbol{\mu}_1^{(k)}, \ldots, \boldsymbol{\mu}_P^{(k)})$,

[c] Many other approaches - such as CAD based parametrizations - are also possible.

[d] Instead of using high-degree Bezier curves for the representation of more complex geometries, we can use composite (low-degree) Bezier curves, referred to as B-splines.

$i = 1, \dots, P$, where the sensitivities $\{\partial \mathcal{J}/\partial \boldsymbol{\mu}_i\}_{i=1}^P$ can be easily computed[6] from the expression of the shape gradient $\nabla \mathcal{J}(\Omega)$. With respect to LBV, PSP allows to reduce the geometrical complexity by some orders of magnitude; nevertheless, a large number of parameters may be required to represent complex shapes, i.e. the representation complexity is directly related with design complexity. Moreover, a remeshing is requested at each iteration, making this technique particularly expensive for large and complex flow simulations, where FE assembling and storage can lead to very high computational costs.

4 COMPUTATIONAL REDUCTION BY REDUCED BASIS METHODS

Our approach to shape optimization takes advantage of *reduced basis* (RB) methods for rapid and reliable prediction of engineering outputs associated with parametric PDEs.[12] This method is premised upon a classical finite element (FE) method "truth" approximation space $X^{\mathcal{N}}$ of (typically very large) dimension $\mathcal{N}$ and is based on the use of "snapshot" FE solutions of the PDEs (for certain values of the parameters) as global approximation basis functions previously computed and stored.

The RB framework requires a parameter independent domain Ω as the snapshots we use for building RB spaces have to be defined relative to the same spatial configuration. We thus consider Ω as reference domain related to the parameter-dependent "original" domain of interest $\Omega_o(\boldsymbol{\mu})$ through a parametric mapping $T(\cdot; \boldsymbol{\mu})$, s.t. $\Omega_o(\boldsymbol{\mu}) = T(\Omega; \boldsymbol{\mu})$. The standard Galerkin FE approximation of the state problem in (5) is to find $y^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ s.t. $a(y^{\mathcal{N}}(\boldsymbol{\mu}), w; \boldsymbol{\mu}) = F(w; \boldsymbol{\mu})$, $\forall w \in X^{\mathcal{N}}$. The RB method gives an efficient way to compute an approximation $y_N^{\mathcal{N}}(\boldsymbol{\mu})$ of $y^{\mathcal{N}}(\boldsymbol{\mu})$ by using a Galerkin projection on a reduced subspace made up of well-chosen FE solutions, i.e. corresponding to a specific choice $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ of parameter values. Indicating by $X_N^{\mathcal{N}} = \text{span}\{y_{\mathcal{N}}(\boldsymbol{\mu}^n), n = 1, \dots, N\}$, RB formulation is to find $y_N^{\mathcal{N}} \in X_N^{\mathcal{N}}$ s.t. $a(y_N^{\mathcal{N}}, w; \boldsymbol{\mu}) = F(w; \boldsymbol{\mu})$, $\forall w \in X_N^{\mathcal{N}}$.

Thanks to the (very) reduced dimension $N \ll \mathcal{N}$ of linear systems obtained from RB approximation, we can provide both reliable results and rapid response[12] in the real-time and multi-query contexts. Reliability is ensured by rigorous a posteriori estimations for the error in the RB approximation w.r.t. truth FE discretization; rapid response is ensured by an Offline–Online computational strategy that minimizes marginal cost and a rapidly convergent global RB space assembling. To achieve this second goal, we need to rely on the assumption of affine parametric dependence in $a(\cdot, \cdot; \boldsymbol{\mu})$ and $F(\cdot; \boldsymbol{\mu})$, given by:

$$a(y, w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(y, w), \qquad F(w; \boldsymbol{\mu}) = \sum_{q=1}^{Q_F} \Theta_F^q(\boldsymbol{\mu}) F^q(\boldsymbol{\mu}). \tag{6}$$

Hence, in an expensive Offline stage we prepare a very small RB "database", while in the Online stage, for each new $\boldsymbol{\mu} \in \mathcal{D}$, we rapidly evaluate both the field and the output (with error bounds) whose computational complexity is independent of FE dimension $\mathcal{N}$. This is essential in a parametrized optimization problem, where a great number of such evaluations are required. In cases where there is a nonaffine parametric dependence, an affine approximation is introduced through an empirical interpolation method (EIM).[14]

5 GEOMETRICAL REDUCTION BY SHAPE PARAMETRIZATION

Classical techniques introduced in Section 3 are not well suited for the use within RB framework, since a parametric mapping between a reference domain (where PDEs problems have to be solved) and the original domain $\Omega_o(\boldsymbol{\mu})$ is needed, rather than a boundary representation. At most, it may be possible to obtain such a mapping by using a polynomial parametrization, even if of great complexity and often computationally unaffordable. We thus have to introduce alternative shape parametrization techniques, leading naturally to the definition of a parametric mapping $T(\cdot;\boldsymbol{\mu})$; two alternative approaches are possible, according the role of the design variables, which can parametrize *(i)* geometrical properties or *(ii)* shape deformations. In the first case, a domain decomposition is leading to the definition of *local mappings on different subdomains*; in the second case, a global mapping can be built by using the so-called *free-form deformation* techniques.

5.1 Affine and nonaffine mappings built "by hand"

Shape parametrization based on geometrical properties is the simplest option we can consider. In general, we need to define a domain decomposition and a different mapping on each subdomain, since a unique mapping is not sufficient to describe the entire geometry. Each mapping can be either affine or nonaffine; moreover, they can be built "by hand" or automatically, by using for example the software `rbMIT`.[13] Let us consider in this section the former, while the latter will be discussed in Section 5.2. In order to build a parametric mapping related to geometrical properties, we introduce a domain decomposition of $\Omega_o(\boldsymbol{\mu})$,

$$\Omega_o(\boldsymbol{\mu}) = \bigcup_{k=1}^{K_{\text{dom}}} \Omega_o^k(\boldsymbol{\mu}), \tag{7}$$

consisting of mutually nonoverlapping open subdomains $\Omega_o^k(\boldsymbol{\mu})$, s.t. $\Omega_o^k(\boldsymbol{\mu}) \cap \Omega_o^{k'}(\boldsymbol{\mu}) = \emptyset$, $1 \leq k < k' \leq K_{\text{dom}}$. If related to geometrical properties used as input parameters (e.g. lengths, thicknesses, diameters or angles) the definition of parametric mappings can be done in a quite intuitive fashion[e]. Our reference domain is then simply defined for a reference parameter value $\boldsymbol{\mu}_{\text{ref}} \in \mathcal{D}$ as $\Omega \equiv \Omega_o(\boldsymbol{\mu}_{\text{ref}})$. In the following we will identify $\Omega^k = \Omega_o^k(\boldsymbol{\mu}_{\text{ref}})$, $1 \leq k \leq K_{\text{dom}}$, and denote the "$K_{\text{dom}}$" domain decomposition of Ω as our "RB triangulation"; it will play an important role in the generation of our affine representation (6). Both in affine and nonaffine cases, original and reference subdomains must be linked via a mapping $T(\cdot;\boldsymbol{\mu}) : \Omega^k \to \Omega_o^k(\boldsymbol{\mu})$, $1 \leq k \leq K_{\text{dom}}$, such that:

$$\Omega_o^k(\boldsymbol{\mu}) = T^k(\Omega^k;\boldsymbol{\mu}), \quad 1 \leq k \leq K_{\text{dom}}; \tag{8}$$

these mappings must be individually bijective and collectively continuous, which means they have to fulfill the following interface condition:

$$T^k(\boldsymbol{x};\boldsymbol{\mu}) = T^{k'}(\boldsymbol{x};\boldsymbol{\mu}), \quad \forall \boldsymbol{x} \in \Omega^k \cap \Omega^{k'}, \quad 1 \leq k < k' \leq K_{\text{dom}}. \tag{9}$$

[e]These regions can represent different material properties, but they can also be used for algorithmic purposes to ensure well-behaved mappings.

In the affine case, for the k^{th} subdomain ($1 \leq k \leq K_{\text{dom}}$) the concrete affine transformation is then given, for any $\boldsymbol{\mu} \in \mathcal{D}$ and for any $\boldsymbol{x} \in \Omega^k$, by

$$T_i^{\text{aff},k}(\boldsymbol{x}, \boldsymbol{\mu}) = C_i^{\text{aff},k}(\boldsymbol{\mu}) + \sum_{j=1}^{d} G_{ij}^{\text{aff},k}(\boldsymbol{\mu}) x_j, \quad 1 \leq i \leq d, \tag{10}$$

for given translation vectors $\mathbf{C}^{\text{aff},k} : \mathcal{D} \to \mathbb{R}^d$ and linear transformation matrices $\mathbf{G}^{\text{aff},k} : \mathcal{D} \to \mathbb{R}^{d \times d}$. The linear transformation matrices can effect rotation, scaling and/or shear and have to be invertible. The associated Jacobians can be defined as $J^{\text{aff},k}(\boldsymbol{\mu}) = |\det(\mathbf{G}^{\text{aff},k}(\boldsymbol{\mu}))|$, $1 \leq k \leq K_{\text{dom}}$; for invertible mappings they are strictly positive.
If the transformation does not take the form (10), transformation is said to be nonaffine. In this case, for the k^{th} subdomain ($1 \leq k \leq K_{\text{dom}}$), for any $\boldsymbol{\mu} \in \mathcal{D}$ and for any $\boldsymbol{x} \in \Omega^k$, $\boldsymbol{x}_o \in \Omega_o^k(\boldsymbol{\mu})$, the nonaffine transformation is given by a generic expression

$$\boldsymbol{x}_{o\,i} = T_i^{\text{naff},k}(\boldsymbol{x}, \boldsymbol{\mu}), \quad 1 \leq i \leq d, \tag{11}$$

and the Jacobians $J^{\text{naff},k}(\boldsymbol{x}, \boldsymbol{\mu}) = |\det(\mathbf{G}^{\text{naff},k}(\boldsymbol{x}, \boldsymbol{\mu}))|$, $1 \leq k \leq K_{\text{dom}}$, are strictly positive, being $\mathbf{G}^{\text{naff},k} : \Omega^k \times \mathcal{D} \to \mathbb{R}^{d \times d}$ the Jacobian matrices of the mappings $T^{\text{naff},k}(\cdot, \boldsymbol{\mu})$. The interface condition (9) allows us to interprete the set of local mappings as a global bijective piecewise affine transformation $T(\cdot; \boldsymbol{\mu}) : \Omega \to \Omega_o(\boldsymbol{\mu})$, given for any $\boldsymbol{\mu} \in \mathcal{D}$ by

$$T(\boldsymbol{x}, \boldsymbol{\mu}) = T^{\text{aff} \vee \text{naff},k}(\boldsymbol{x}; \boldsymbol{\mu}), \quad k = \min_{k' \in \{1,\ldots,K_{\text{dom}}\} | \boldsymbol{x} \in \Omega^{k'}} k'. \tag{12}$$

5.2 Automatic affine mappings by `rbMIT`

The software `rbMIT`[13] allows to build efficient affine mappings in an automatic fashion based on a domain decomposition made up by some "building blocks" introduced in this section; for clarity, we concentrate on a single subdomain[f].

5.2.1 Automatic affine mappings for a single subdomain

Since in the two-dimensional case ($d = 2$) straight lines are mapped into straight lines and parallelism is preserved, a parallelogramm is mapped into a parallelogram and hence a triangle into a triangle; moreover, affine transformations map ellipses into ellipses. These features will be exploited for the development of an automatic domain decomposition technique suitable for the RB context. Thus, affine mappings contains $d(d+1) = 6$ degrees of freedom (the mapping coefficients), and it is therefore sufficient, for any given $\boldsymbol{\mu} \in \mathcal{D}$, to consider the relationship between three non-colinear pre-image points in Ω, $(\boldsymbol{z}^1, \boldsymbol{z}^2, \boldsymbol{z}^3)$ and three parametrized image nodes in $\Omega_o(\boldsymbol{\mu})$, $(\boldsymbol{z}_o^1(\boldsymbol{\mu}), \boldsymbol{z}_o^2(\boldsymbol{\mu}), \boldsymbol{z}_o^3(\boldsymbol{\mu}))$. Note that every point consists of two components (z_1^i, z_2^i), $1 \leq i \leq 3$, resp. $(z_{o\,1}^i, z_{o\,2}^i)$, $1 \leq i \leq 3$, and therefore the application of (10) to these points gives a system of six independent equations to determine[g] the six mapping coefficients:

[f] We shall suppress the subdomain superscript for clarity of exposition; the matrices $\mathbf{C}^{\text{aff}}(\boldsymbol{\mu}) \in \mathbb{R}^d$ and $\mathbf{G}^{\text{aff}}(\boldsymbol{\mu}) \in \mathbb{R}^{d \times d}$ in (10) are now called "mapping coefficients".

[g] The assumption that the affine transformation is bijective thereby ensures that image nodes are perforce also non-colinear (if pre-image nodes are non-colinear) and hence equations are linear independent.

$$z_{o\,i}^m(\boldsymbol{\mu}) = C_i^{\text{aff}}(\boldsymbol{\mu}) + \sum_{j=1}^{2} G_{ij}^{\text{aff}}(\boldsymbol{\mu}) z_j^m, \quad 1 \le i \le 2, \quad 1 \le m \le 3; \tag{13}$$

Our RB triangulation shall be built on (standard) triangles, elliptical triangles and general "curvy" triangles, which are discussed in detail below and are the building blocks in the `rbMIT` software[13] used for the RB computations in this work.

5.2.2 Standard Triangles

In the case of a standard triangle subdomain the three vertices of the triangle in the reference domain shall serve as pre-image nodes while the three vertices of the triangle in the actual ($\boldsymbol{\mu}$-dependent) domain shall serve as image nodes. In this case, our three points uniquely define not only the transformation but also the reference domain and parametrized domains.[12] We can then readily establish the system of six linear equations to determine the six unknown mapping coefficients. In this way, we can construct an affine transformation from any reference triangle in $\mathbb{R}^2$ onto any desired triangle in $\mathbb{R}^2$.

5.2.3 Elliptical and Curvy Triangles

The class of elliptical triangles covers a much greater range of possible geometries and their formulation is also necessary for the more general case dealing with curvy triangles. We can distinguish two different kinds of elliptic triangles: "inwards" and "outwards" triangles. Both types are depicted in Figure 1. In both cases, the elliptical triangle $\Omega_o(\boldsymbol{\mu})$ is defined by the three vertices $\boldsymbol{z}_o^1(\boldsymbol{\mu})$, $\boldsymbol{z}_o^2(\boldsymbol{\mu})$, $\boldsymbol{z}_o^3(\boldsymbol{\mu})$, the two straight lines $\overline{\boldsymbol{z}_o^1(\boldsymbol{\mu})\boldsymbol{z}_o^2(\boldsymbol{\mu})}$ and $\overline{\boldsymbol{z}_o^1(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}$ as well as the elliptical arc $\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}^{\text{arc}}$.

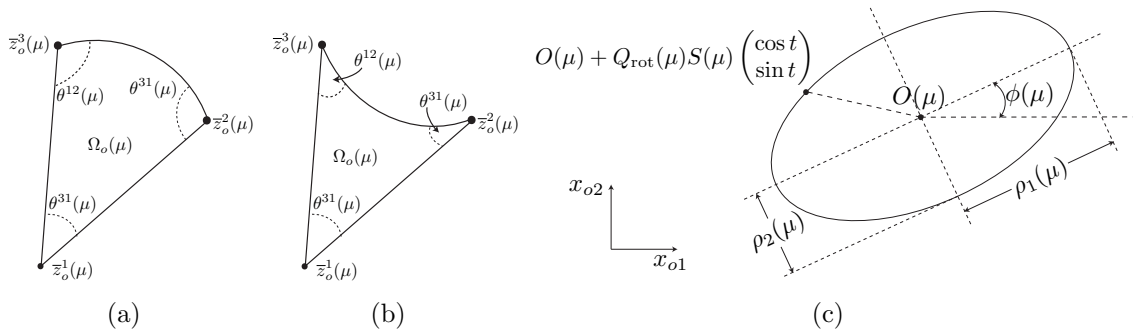

Figure 1: (a) "Inwards" elliptical triangle and (b) "outwards" elliptical triangle; (c) definition of a point on a prescribed parametrized ellipse.

We shall now precise the definition and description of the elliptical arc and explain the constraints that must be met by the location of the third point $\boldsymbol{z}_o^1(\boldsymbol{\mu})$ to ensure "proper" triangles and a continuous and well-defined global mapping in the multidomain context.

First, the description of the elliptical arc shall be derived from the definition of a parametrized ellipse (see Figure 1 (c)). The ellipse is described implicitly by

$$(\boldsymbol{x}_o - O(\boldsymbol{\mu}))^T Q_{\text{rot}}(\boldsymbol{\mu}) S^{-2}(\boldsymbol{\mu}) Q_{\text{rot}}(\boldsymbol{\mu})^T (\boldsymbol{x}_o - O(\boldsymbol{\mu})) = 1. \tag{14}$$

A particular point on this ellipse is then given by

$$\boldsymbol{x}_o \equiv \begin{pmatrix} x_{o1} \\ x_{o2} \end{pmatrix} = O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \tag{15}$$

for given $t \in \mathbb{R}$. As we can see in Figure 1, $O(\boldsymbol{\mu}) : D \to \mathbb{R}^2$ is the center of the ellipse, $\rho_1(\boldsymbol{\mu}) : D \to \mathbb{R}_+$ and $\rho_2 : D \to \mathbb{R}_+$ define the length of the semi-axes of the ellipse and $\phi(\boldsymbol{\mu}) : D \to \mathbb{R}$ is the angle of inclination. With these quantities, the scaling matrix $S(\boldsymbol{\mu})$ and the rotation matrix $Q_{\text{rot}}(\boldsymbol{\mu})$ can be defined:

$$S(\boldsymbol{\mu}) \equiv \begin{pmatrix} \rho_1(\boldsymbol{\mu}) & 0 \\ 0 & \rho_2(\boldsymbol{\mu}) \end{pmatrix}, \qquad Q_{\text{rot}}(\boldsymbol{\mu}) = \begin{pmatrix} \cos\phi(\boldsymbol{\mu}) & -\sin\phi(\boldsymbol{\mu}) \\ \sin\phi(\boldsymbol{\mu}) & \cos\phi(\boldsymbol{\mu}) \end{pmatrix}.$$

The description of the elliptical arc with these means is then as follows:

$$\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}^{\text{arc}} = \left\{ O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} \,\middle|\, t_2 \le t \le t_3 \right\}. \tag{16}$$

with $t_2 \in \mathbb{R}$ and $t_3 \in \mathbb{R}$ chosen such that the points $\boldsymbol{z}_o^2(\boldsymbol{\mu})$ and $\boldsymbol{z}_o^3(\boldsymbol{\mu})$ are given as the endpoints of the elliptical arc for $t = t_2$ and $t = t_3$:

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}) = O(\boldsymbol{\mu}) + Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad m = 2, 3. \tag{17}$$

In addition, we have to make sure that $0 \le t_3 - t_2 < \pi$. It remains to specify the location of the third point $\boldsymbol{z}_o^1(\boldsymbol{\mu})$. For elliptical triangles, this location is not arbitrary but has to be chosen in a way that ensures that the affine transformation generates the desired elliptical arc (16). First, this ensures a continuous global mapping; second, to obtain well-defined elliptical triangles (and thus a well defined domain), several internal angle conditions have to be met by the choice for $\boldsymbol{z}_o^1(\boldsymbol{\mu})$: $0 < \Theta^* < \pi,\ \forall\, \Theta^* \in \{\Theta^{12}, \Theta^{23}, \Theta^{31}\}$. The first requirement can be fulfilled by the expression of the three corner points as

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}) = O(\boldsymbol{\mu}) + \omega_m Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad 1 \le m \le 3, \tag{18}$$

for given $\omega_1 = \omega \in \mathbb{R}$, $\omega_2 = \omega_3 = 1$ and $t_1 \in [t_2, t_3]$. Pre-image points are thus given as

$$\boldsymbol{z}_o^m(\boldsymbol{\mu}_{\text{ref}}) = O(\boldsymbol{\mu}_{\text{ref}}) + \omega_m Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}}) S(\boldsymbol{\mu}_{\text{ref}}) \begin{pmatrix} \cos t_m \\ \sin t_m \end{pmatrix}, \quad 1 \le m \le 3. \tag{19}$$

From these representations we can identify our affine mapping as

$$\begin{aligned} \boldsymbol{z}_o^m(\boldsymbol{\mu}) = \mathbf{C}^{\text{aff}}(\boldsymbol{\mu}) + \mathbf{G}^{\text{aff}}(\boldsymbol{\mu}) \boldsymbol{z}^m = (O(\boldsymbol{\mu}) - Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) S(\boldsymbol{\mu}_{\text{ref}})^{-1} Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T O(\boldsymbol{\mu}_{\text{ref}})) \\ + (Q_{\text{rot}}(\boldsymbol{\mu}) S(\boldsymbol{\mu}) S(\boldsymbol{\mu}_{\text{ref}})^{-1} Q_{\text{rot}}(\boldsymbol{\mu}_{\text{ref}})^T) \boldsymbol{z}^m. \end{aligned}$$

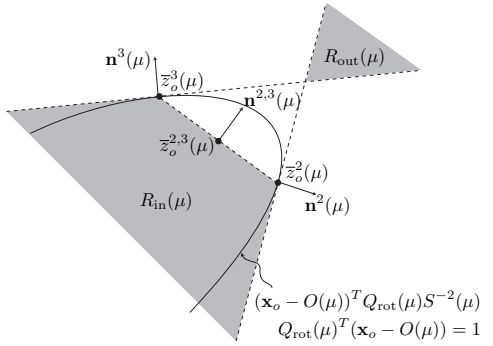

Figure 2: Regions in which $z_o^1(\boldsymbol{\mu})$ must reside in the inwards ($R_{\text{in}}(\boldsymbol{\mu})$) and the outwards case ($R_{\text{out}}(\boldsymbol{\mu})$).

The second requirement - the internal angle conditions - is illustrated in Figure 2. In the inwards case, a necessary and sufficient condition to ensure the conditions $0 < \Theta^* < \pi$, $\forall \Theta^* \in \{\Theta^{12}, \Theta^{23}, \Theta^{31}\}$ is given for an inwards elliptical triangle by $\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in R_{\text{in}}(\boldsymbol{\mu})$, where

$$\begin{aligned} R_{in}(\boldsymbol{\mu}) = \{\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in \mathbb{R}^2 | (\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^2(\boldsymbol{\mu}))^T \boldsymbol{n}^2(\boldsymbol{\mu}) < 0, \\ (\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^3(\boldsymbol{\mu})))^T \boldsymbol{n}^3(\boldsymbol{\mu}) < 0, (\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^{2,3}(\boldsymbol{\mu}))^T \boldsymbol{n}^{2,3}(\boldsymbol{\mu}) < 0\}, \end{aligned} \tag{20}$$

and for the outwards elliptical triangle by $\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in R_{\text{out}}(\boldsymbol{\mu})$, where

$$R_{out}(\boldsymbol{\mu}) = \{\boldsymbol{z}_o^1(\boldsymbol{\mu}) \in \mathbb{R}^2 | (\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^2(\boldsymbol{\mu}))^T \boldsymbol{n}^2(\boldsymbol{\mu}) > 0, (\boldsymbol{z}_o^1(\boldsymbol{\mu}) - \boldsymbol{z}_o^3(\boldsymbol{\mu}))^T \boldsymbol{n}^3(\boldsymbol{\mu}) > 0\}. \tag{21}$$

Here $\boldsymbol{n}^2(\boldsymbol{\mu})$ and $\boldsymbol{n}^3(\boldsymbol{\mu})$ are the outwards-facing normals to the ellipse at $\boldsymbol{z}^2(\boldsymbol{\mu})$ and $\boldsymbol{z}^3(\boldsymbol{\mu})$ respectively, $\boldsymbol{z}_o^{2,3}(\boldsymbol{\mu}) = \frac{1}{2}(\boldsymbol{z}_o^2(\boldsymbol{\mu}) + \boldsymbol{z}_o^3(\boldsymbol{\mu}))$ and $\boldsymbol{n}^{2,3}(\boldsymbol{\mu})$ is the "outwards-facing" normal to the line segment $\overline{\boldsymbol{z}_o^2(\boldsymbol{\mu})\boldsymbol{z}_o^3(\boldsymbol{\mu})}$ at $\boldsymbol{z}_o^{2,3}(\boldsymbol{\mu})$. An important feature of the elliptical triangles is that they are consistent under refinement: so, if we split an elliptical triangle for which the internal angle conditions (20) and (21) are fulfilled, the resulting two elliptical triangles also satisfy the internal angle conditions. To enlarge the possible range of geometries even more, the elliptical triangles are extended to "curvy" triangles. This is done by replacing $(\cos t, \sin t)^T$ in (15) with a general parametrization $(g_1(t), g_2(t))^T$.

5.2.4 Piecewise-Affine Mappings for Multiple Subdomains

To treat more complex geometries, it is necessary to allow our domain to be built of several (standard, elliptical or curvy) triangles. We are then not restricted to a single affine mapping, but we deal with a piecewise affine mapping based on this domain decomposition. We can thus consider geometrical domains for which the boundary can be represented either by straight edges or by elliptical triangles as presented in Section 6.

The multi-domain mapping process is then performed in three steps. First, the RB triangulation is generated on the reference domain Ω together with the associated reference regions. In the second step, the necessary parameter-dependent affine mappings for each subdomain are constructed, as described in the previous section. In the last step we have to translate the parametric mappings obtained for each subdomain into PDE coefficients.

5.3 Parametrized formulation of PDEs problems

When the geometric map (10) is built, it is possible to write the PDEs problem under the affine form (6). This is straightforward for an affine geometrical mapping, while in the case of a nonaffine mapping we need to rely on the empirical interpolation method (EIM) – which is an interpolation method for parametric functions based on adaptively chosen interpolation points and global shape functions[14] – in order to approximate the nonaffine components of the transformation tensors. Let us consider the affine mapping case and an abstract problem like the one in (5), with a parameter-dependent domain $\Omega_o(\boldsymbol{\mu})$; this problem can be written in general form as: given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate $y_o(\boldsymbol{\mu}) \in X_o(\boldsymbol{\mu})$ s.t.

$$a_o(y_o(\boldsymbol{\mu}), \phi; \boldsymbol{\mu}) = f_o(\psi), \quad \forall \psi \in X_o(\boldsymbol{\mu}) \subseteq H^1(\Omega_o(\boldsymbol{\mu})).$$

A sufficient condition on the continuous and coercive bilinear form $a_o(\cdot, \cdot; \boldsymbol{\mu}) : X_o(\boldsymbol{\mu}) \times X_o(\boldsymbol{\mu}) \to \mathbb{R}$ that ensures an affine expansion is fulfilled if we have

$$a_o(w, v; \boldsymbol{\mu}) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega_o^k(\boldsymbol{\mu})} \begin{bmatrix} \frac{\partial w}{\partial x_{o1}} & \frac{\partial w}{\partial x_{o2}} & w \end{bmatrix} \mathbf{K}_{o,k}(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_{o1}} \\ \frac{\partial v}{\partial x_{o2}} \\ v \end{bmatrix}. \tag{22}$$

The matrices $\mathbf{K}_{o,k} : \mathcal{D} \to \mathbb{R}^{3\times 3}$, $1 \le k \le K_{\text{dom}}$, are in the symmetric case symmetric positive definite matrices. Similarly, we require that $f_o : X_o(\boldsymbol{\mu}) \to \mathbb{R}$ is written as

$$f_o(v) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega_o^k(\boldsymbol{\mu})} F_{o,k}(\boldsymbol{\mu}) v, \tag{23}$$

with $F_{o,k} : \mathcal{D} \to \mathbb{R}$, $1 \le k \le K_{\text{dom}}$. Identifying $y(\boldsymbol{\mu}) = y_o(\boldsymbol{\mu}) \circ T^{\text{aff}}(\cdot; \boldsymbol{\mu})$, and recalling that $\partial/\partial x_{oi} = \frac{\partial x_j}{\partial x_{oi}} \frac{\partial}{\partial x_j} = (G^{\text{aff},k}(\boldsymbol{\mu}))^{-1}_{1i} \partial/\partial x_1 + (G^{\text{aff},k}(\boldsymbol{\mu}))^{-1}_{2i} \partial/\partial x_2$, for $i = 1, 2$, and $d\Omega_o^k(\boldsymbol{\mu}) = J^{\text{aff},k}(\boldsymbol{\mu}) d\Omega$, it then follows that the transformed bilinear form a can be expressed as

$$a(w, v; \boldsymbol{\mu}) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega_k} \begin{bmatrix} \frac{\partial w}{\partial x_1} & \frac{\partial w}{\partial x_2} & w \end{bmatrix} \mathbf{K}^k(\boldsymbol{\mu}) \begin{bmatrix} \frac{\partial v}{\partial x_1} \\ \frac{\partial v}{\partial x_2} \\ v \end{bmatrix}, \tag{24}$$

where the $K^k : \mathcal{D} \to \mathbb{R}^{3\times 3}$ are given by $\mathbf{K}^k(\boldsymbol{\mu}) = J^{\text{aff},k}(\boldsymbol{\mu}) \mathbf{G}^k(\boldsymbol{\mu}) \mathbf{K}_{o,k}(\boldsymbol{\mu}) (\mathbf{G}^k(\boldsymbol{\mu}))^T$, for $1 \le k \le K_{\text{dom}}$; moreover, the $\mathbf{G}^k : \mathcal{D} \to \mathbb{R}^{3\times 3}$ are given by

$$\mathbf{G}^k(\boldsymbol{\mu}) = \begin{pmatrix} (\mathbf{G}^{\text{aff},k}(\boldsymbol{\mu}))^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}, \quad 1 \le k \le K_{\text{dom}}. \tag{25}$$

The transformed linear form can be expressed similarly as

$$f(v) = \sum_{k=1}^{K_{\text{dom}}} \int_{\Omega_k} F^k(\boldsymbol{\mu}) v, \tag{26}$$

where $F^k : \mathcal{D} \to \mathbb{R}$ is given by $F^k = J^{\text{aff},k}(\boldsymbol{\mu}) F_{o,k}(\boldsymbol{\mu})$, for $1 \le k \le K_{\text{dom}}$. In general, the $\mathbf{K}^k(\boldsymbol{\mu})$ and $F^k(\boldsymbol{\mu})$ will be different for each subdomain Ω^k. The affine formulation (24) can then be derived by simply expanding this expression (in terms of the subdomains Ω^k and the different entries of K_{ij}^k, $1 \le i, j \le 3$, $1 \le k \le K_{\text{dom}}$). This results in

$$a(w,v;\boldsymbol{\mu}) = K_{11}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_1} + K_{12}^1(\boldsymbol{\mu}) \int_{\Omega^1} \frac{\partial w}{\partial x_1}\frac{\partial v}{\partial x_2} + \cdots \tag{27}$$

The affine representation is now clear: for each term in (27) the (parameter-independent) integral represents $a^q(w,v)$, while the (parameter-dependent) prefactor represents $\Theta^q(\boldsymbol{\mu})$. The linear form f admits a similar treatment. The affine representation obtained by this process contains at most $Q_a = 3K_{\text{dom}}$ terms. In some special cases the number of nonzero terms in (27) is even reduced to $Q_a = 2K_{\text{dom}}$, like dealing with potential flows (without mixed derivatives in the Laplacian). In other situations, many terms can be economized if linear dependent entries are assembled together. Another possibility to reduce the number of terms Q_a is an intelligent choice of user-provided initial control points and edges for the RB triangulation, exploiting symmetry effects and isolate geometric variation.

5.4 Free-form deformation techniques

A more versatile but low-dimensional parametrization can be introduced by exploiting the so-called free-form deformation (FFD) techniques, in which the deformations of an initial design are parametrized, rather than the geometry itself. Originally introduced in the late 70s,[15] FFD techniques have been deeply used in computer graphics; only in last years they have been employed in optimal design problems,[16,17] also with RB methods.[18,19] With respect to other classical shape parametrizations, FFD helps in avoiding both problems of complex shapes and remeshing, thus realizing a computational reduction too.

A free-form deformation operates on a bivariate Bezier control area built around (and regardless of) the shape we want to optimize, manipulating a lattice of control points. In this way, design parameters are not directly connected neither to geometrical properties nor to the shape boundary. Moreover, both the space metrics and objects inside are deformed and, automatically, the computational finite element mesh too. In this way, FFD inherits from usual boundary parametrization techniques the possibility to handle with global deformations by acting on a small set of control points,[10] but providing an easier manipulation tool since no explicit shape parametrization of the object is required. Let us introduce the main features of FFD techniques. Given a fixed rectangular domain D

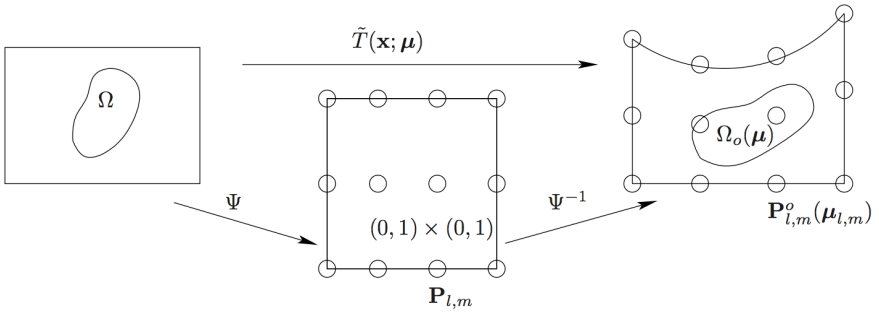

Figure 3: Schematic diagram of the FFD technique: unperturbed control points $\mathbf{P}_{l,m}$, perturbed control points $\mathbf{P}^o_{l,m}(\boldsymbol{\mu}_{l,m})$, mapping $\tilde{T}(\mathbf{x};\boldsymbol{\mu})$ and resulting deformation.

s.t. $\Omega \subset D$, we assume the existence of a differentiable and invertible map $\Psi : (x_1, x_2) \to (s, t)$ such that $\Psi(D) = (0, 1)^2$; by this *freezing* procedure, FFD can be defined in a simpler way in the coordinates (s, t) of the spline parameter space $(0, 1)^2$ (see Figure 3). We thus select an ordered mesh of $(L+1)\times(M+1)$ unperturbed control points $\mathbf{P}_{l,m} = [l/L, m/M]^T$, $l = 0, \dots, L$, $m = 0, \dots, M$, and modify the object by moving control points to a new position. The corresponding perturbed control points $\mathbf{P}^o_{l,m}(\boldsymbol{\mu}_{l,m}) = \mathbf{P}_{l,m} + \boldsymbol{\mu}_{l,m}$ are thus specified by a set of $(L+1)(M+1)$ parameter vectors $\boldsymbol{\mu}_{l,m} \in \mathbb{R}^2$, giving in all $2(L+1)(M+1)$ possible degrees of freedom. As often as not, only small subsets of these points are selected as design variables if we want to perform a sensible geometrical reduction; moreover, several rows or columns of control points may be fixed to obtain desired levels of continuity and to "anchor" certain parts of the domain. In general, among the control points $\mathbf{P}_{l,m}$, we indicate the effectively free scalar-valued parameters chosen as design variables as $\mu_1, \dots, \mu_P$ – each corresponding to the displacement of a control point in the s or t direction – and define the parametric map $\tilde{T}(\cdot, \boldsymbol{\mu}) : D \to \mathbb{R}^2$ by which the uploaded geometry is computed as follows:

$$\tilde{T}(\Psi(\mathbf{x}); \boldsymbol{\mu}) = \Psi^{-1}\left(\sum_{l=0}^{L}\sum_{m=0}^{M} b^{L,M}_{l,m}(\Psi(\mathbf{x}))\mathbf{P}^o_{l,m}(\boldsymbol{\mu}_{l,m})\right), \tag{28}$$

where $b^{L,M}_{l,m}(s, t) = b^L_l(s)b^M_m(t)$ are tensor products of unidimensional *Bernstein basis polynomials* $b^L_l(s) = \binom{L}{l}s^l(1-s)^{L-l}$ and $b^M_m(t) = \binom{M}{m}t^m(1-t)^{M-m}$ defined on the unit square $(s, t) \in [0, 1] \times [0, 1]$. Finally, we have $\Omega_o(\boldsymbol{\mu}) = T(\Omega; \boldsymbol{\mu})$, by using the restriction $T = \tilde{T}|_\Omega$.

By this procedure, we obtain a polinomial map defined on Ω. Nevertheless, the (now, global) tensor transformation $\mathbf{K} : \Omega\times\mathcal{D} \to \mathbb{R}^{3\times3}$ through which we obtain the transformed bilinear form (24) is given by $\mathbf{K}(\mathbf{x}, \boldsymbol{\mu}) = J^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu})\mathbf{G}(\mathbf{x}, \boldsymbol{\mu})\mathbf{K}_o(\boldsymbol{\mu})(\mathbf{G}(\mathbf{x}, \boldsymbol{\mu}))^T$, where

$$\mathbf{G}(\mathbf{x}, \boldsymbol{\mu}) = \begin{pmatrix} (\mathbf{G}^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu}))^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix}, \tag{29}$$

$\mathbf{G}^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu})$ is the Jacobian matrix[h] of $T(\cdot; \boldsymbol{\mu})$ and $J^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu}) = |\det(\mathbf{G}^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu}))|$. In the same way, transformation in (26) is given by $F(\mathbf{x}, \boldsymbol{\mu}) = J^{\text{naff}}(\mathbf{x}, \boldsymbol{\mu})\, F_o(\boldsymbol{\mu})$. Hence, in order to recover the affine representation – necessary to split the RB solution through an offline/online decomposition – we rely on empirical interpolation;[14] we thus approximate each components $K_{ij}(\mathbf{x}, \boldsymbol{\mu})$ and $F(\mathbf{x}, \boldsymbol{\mu})$ with an affine representation given by

$$\tilde{K}_{i,j}(\mathbf{x}, \boldsymbol{\mu}) = \sum_{n=1}^{N^a_{ij}} \tilde{\beta}^{i,j}_n(\boldsymbol{\mu})\tilde{\xi}^{i,j}_n(\mathbf{x}) + \varepsilon_{i,j}(\mathbf{x}; \boldsymbol{\mu}), \quad \tilde{F}(\mathbf{x}, \boldsymbol{\mu}) = \sum_{n=1}^{N^f} \tilde{\delta}_n(\boldsymbol{\mu})\tilde{\psi}_n(\mathbf{x}) + \varepsilon(\mathbf{x}; \boldsymbol{\mu}), \tag{30}$$

where all the $\tilde{\beta}^{i,j}_n$'s, $\tilde{\xi}^{i,j}_n$'s, $\tilde{\delta}_n$'s and $\tilde{\psi}_n$'s are efficiently computable scalar functions and the error terms are guaranteed to be under some tolerance, i.e. $\|\varepsilon_{i,j}(\cdot; \boldsymbol{\mu})\|_\infty \le \varepsilon^{EIM}_{tol}$, $\|\varepsilon(\cdot; \boldsymbol{\mu})\|_\infty \le \varepsilon^{EIM}_{tol}$, for all $\boldsymbol{\mu} \in \mathcal{D}$. We underline that the number and position of control points have a deep impact on FFD flexibility: it is crucial to maximize the influence of the control points by placing them close to the sensitive regions of the configuration.

[h] In the nonaffine case, the Jacobian matrix (and thus its determinant and all transformation tensors) depend both on the coordinates $\mathbf{x}$ and on the parameter $\boldsymbol{\mu}$.

6 NUMERICAL EXAMPLES

In this section we show some representative test cases to illustrate the functionality of the proposed shape parametrization techniques, coupled with a reduced basis method for the solution of the parametrized PDEs describing the related flows. As example of optimal design problem, we consider the minimization of drag forces acting on a body embedded in a viscous linear flow, formulated as:

$$\text{find } \hat{\Omega}_o \in \mathcal{O}_{ad}, \quad \hat{\Omega}_o = \arg\min_{\Omega_o \in \mathcal{O}_{ad}} \mathcal{J}(\Omega_o, \mathbf{v}(\Omega_o), p(\Omega_o)) = -\int_{\Gamma_B^o} (\mathbb{T}(\mathbf{v}, p)\mathbf{n}_o) \cdot \hat{\mathbf{v}}_\infty d\Gamma_o \quad (31)$$

where $(\mathbf{v}(\Omega_o), p(\Omega_o))$ are velocity and pressure[i] solutions of the following Stokes system:

$$\begin{array}{ll} -\nu\Delta\mathbf{v} + \nabla p = \mathbf{f} & \text{in } \Omega_o \\ \nabla \cdot \mathbf{v} = 0 & \text{in } \Omega_o \end{array} \quad \text{with} \quad \begin{array}{ll} \mathbf{v} = \mathbf{v}_\infty & \text{on } \Gamma_{in}^o \cup \Gamma_w^o \\ \mathbf{v} = \mathbf{0} & \text{on } \Gamma_B^o \\ -p \cdot \mathbf{n} + \nu \dfrac{\partial \mathbf{v}}{\partial \mathbf{n}} = \mathbf{0} & \text{on } \Gamma_{out}^o. \end{array} \quad (32)$$

Here $\mathbb{T}(\mathbf{v}, p) = -p\mathbb{I} + \nu(\nabla\mathbf{v} + \nabla^T\mathbf{v})/2$, while $\nu = \mu/\rho$ is the ratio between the dynamic viscosity μ and the fluid density ρ. The family $\mathcal{O}_{ad}$ of admissible shapes is here given by configurations in the form $\Omega_o = D \setminus B_o$, where D is a fixed rectangle and B_o is a Lipschitz domain of boundary $\Gamma_B^o = \partial B_o$ (see Figure 4). No-slip conditions are imposed on Γ_B^o, a given profile $\mathbf{v}_\infty$ (whose direction $\hat{\mathbf{v}}_\infty$ is horizontal) is imposed on inflow Γ_{in}^o and walls Γ_w^o, while a free-stress condition is imposed on the outflow Γ_{out}^o. The parametrized version of (32) is given by the following problem, obtained by mapping its weak formulation back onto the reference domain Ω: find $(\mathbf{v}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in X(\Omega) \times Q(\Omega)$ s.t.

$$\begin{cases} a(\mathbf{v}(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) + b(p(\boldsymbol{\mu}), \mathbf{w}; \boldsymbol{\mu}) = F(\mathbf{w}) & \forall\ \mathbf{w} \in X(\Omega) \equiv (H^1_{0,\Gamma_D}(\Omega))^2 \\ b(q, \mathbf{v}(\boldsymbol{\mu}); \boldsymbol{\mu}) = G(q) & \forall\ \ q \in Q(\Omega) \equiv L^2(\Omega) \end{cases} \quad (33)$$

with $R\mathbf{v}_g \in (H^1(\Omega))^2$ a lifting function s.t. $R\mathbf{v}_g|_{\Gamma_{in}\cup\Gamma_w} = \mathbf{v}_\infty$, $R\mathbf{v}_g|_{\Gamma_B} = \mathbf{0}$ and

$$a(\mathbf{v}, \mathbf{w}; \boldsymbol{\mu}) = \int_\Omega \frac{\partial \mathbf{v}}{\partial x_i} \nu_{ij}(\mathbf{x}; \boldsymbol{\mu}) \frac{\partial \mathbf{w}}{\partial x_j} d\Omega, \quad b(p, \mathbf{w}; \boldsymbol{\mu}) = -\int_\Omega p\chi_{ij}(\mathbf{x}; \boldsymbol{\mu}) \frac{\partial w_j}{\partial x_i} d\Omega, \quad (34)$$

$$F(\mathbf{w}) = \int_\Omega \mathbf{f} \cdot \mathbf{w}\ |\det(\mathbf{G}^{\text{naff}})| d\Omega - a(R\mathbf{v}_g, \mathbf{w}; \boldsymbol{\mu}), \quad G(q) = b(q, R\mathbf{v}_g; \boldsymbol{\mu}). \quad (35)$$

The transformation tensors are defined as $\boldsymbol{\nu}(\mathbf{x}; \boldsymbol{\mu}) = (\mathbf{G}^{\text{naff}})^{-1}\boldsymbol{\nu}^o(\mathbf{G}^{\text{naff}})^{-T}|\det(\mathbf{G}^{\text{naff}})|$ and $\boldsymbol{\chi}^k(\boldsymbol{\mu}) = (\mathbf{G}^{\text{naff}})^{-1}|\det(\mathbf{G}^{\text{naff}})|$, where $\boldsymbol{\nu}^o_{ij} = \nu\delta_{ij}$ (being δ_{ij} the Kronecker symbol) and, as before, $\mathbf{G}^{\text{naff}}(\cdot, \boldsymbol{\mu})$ is the Jacobian matrix of $T(\cdot; \boldsymbol{\mu})$. Problem (31) thus becomes:

$$\text{find } \hat{\boldsymbol{\mu}} \in \mathcal{D}, \quad \hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu} \in \mathcal{D}} J(\boldsymbol{\mu}, \mathbf{v}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) = \int_{\Gamma_B} \eta(\mathbf{v}(\boldsymbol{\mu}), p(\boldsymbol{\mu}))|\mathbf{G}^{\text{naff}}\ \mathbf{t}| d\Gamma$$

where $\eta(\boldsymbol{\mu}, \mathbf{v}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) = -([(\mathbb{T}(\mathbf{v}, p)\mathbf{n}_o) \cdot \hat{\mathbf{v}}_\infty] \circ T)(\mathbf{x}; \boldsymbol{\mu})$ is the drag force mapped back onto Ω and $\mathbf{t}$ is the unit tangent vector to $\partial\Omega$. In order to solve (6), we use the parametrization strategies discussed and non-linear programming algorithms – e.g. sequential quadratic programming – implemented in `Matlab`. After discussing some numerical results, the main features of proposed parametrization techniques will be summarized.

[i] Here the pressure is considered as a normalized pressure (w.r.t. density ρ).

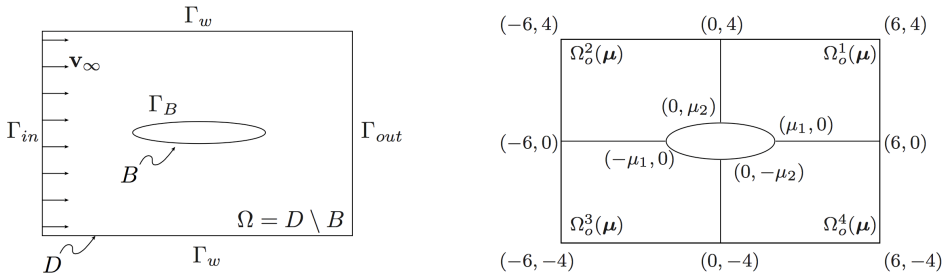

Figure 4: Schematic geometrical domain (left) and setup of test case 1 (right).

6.1 Test case 1: affine and non-affine mappings built "by hand"

The simplest representation of the body $B \subset D := [-6,6] \times [-4,4]$ is given by a parametrized ellipse with $P = 2$ parameters, where $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} := [0.5, 1.0] \times [0.25, 0.5]$ are its semiaxes, i.e. the semi-length and the semi-thickness of the body. By dividing the domain $\Omega_o(\boldsymbol{\mu})$ in four subdomains as depicted in Figure 4, we can build a map $T^{\text{aff},k}$ of the form (10) for each Ω^k, $1 \leq k \leq K_{dom} \equiv 4$, as follows (e.g. for $k = 1$):

$$C^{\text{aff},1}(\boldsymbol{\mu}) = \begin{pmatrix} 6(1 - \frac{\mu_1 - 6}{\mu_{1,ref} - 6}) \\ 4(1 - \frac{\mu_2 - 4}{\mu_{2,ref} - 4}) \end{pmatrix}, \qquad G^{\text{aff},1}(\boldsymbol{\mu}) = \begin{pmatrix} \frac{\mu_1 - 6}{\mu_{1,ref} - 6} & 0 \\ 0 & \frac{\mu_2 - 4}{\mu_{2,ref} - 4} \end{pmatrix},$$

where $\boldsymbol{\mu}_{ref} = (0.75, 0.375)$. By these affine mappings, we build the parametrized bilinear forms (34) and recover an affine development as in (6), consisting at most of $Q_a + Q_b = 3K_{dom} + 3K_{dom} = 24$ terms (being $\boldsymbol{\nu}^k$ and $\boldsymbol{\chi}^k$ symmetric tensors); in the same way, the affine development of linear forms in (35) are made at most of $Q_F = K_{dom} + 3K_{dom} = 16$ and $Q_G = Q_b = 12$ terms respectively. Here $\mathbf{v}_\infty = (1, 0)$, $\nu = 10^{-1}$ and $\mathbf{f} = (0, 0)$.

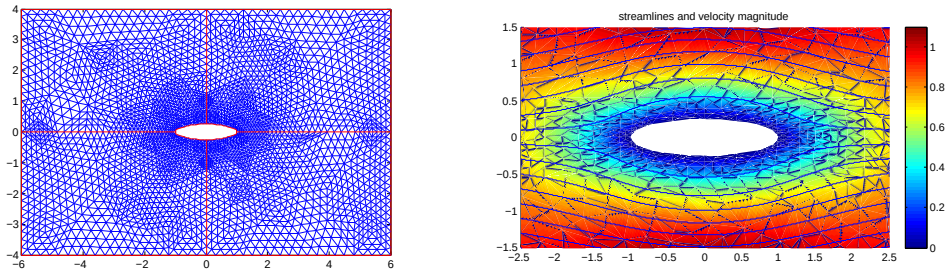

Figure 5: Test case 1: optimal shape for $\mu_1 = \mu_{1,\max}$: velocity (magnitude) and pressure fields.

Optimal shape and corresponding Stokes flow (e.g. obtained fixing $\mu_1 = \mu_{1,max}$) are shown in Figure 5; at each step, the Stokes flow and the drag functional have been evaluated through an RB approximation of problem (33) and output in (6) respectively. For more complex configurations, a shape parametrization built "by hand" may become very difficult and involve non-affine mappings; in this case, we prefer to build automatically affine mappings by exploiting `rbMIT` potentialities.

6.2 Test case 2: automatic affine mappings by `rbMIT`

Let us consider a rectangular domain $D = [-6, 6] \times [-4, 4]$ and a body B given by a symmetric (centered) airfoil profile of the NACA 4-digits family of unity length, whose thickness distribution is given by the following equation:

$$\pm x_{o2} = \frac{\mu_1}{0.2}(0.2969\sqrt{x_{o1}} - 0.1260x_{o1} - 0.3520x_{o1}^2 + 0.2832x_{o1}^3 - 0.1021x_{o1}^4),$$

where $\mu_1 \in [4, 24]$ is the maximum thickness. This parametrization can be transformed to the equivalent form (15) for a general curvy triangle as follows:

$$\mathbf{x}_o = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ 0 & \pm\mu_1/20 \end{pmatrix} \begin{pmatrix} 1 - t^2 \\ 0.2969t - 0.1260t^2 - 0.3520t^4 + 0.2832t^6 - 0.1021t^8 \end{pmatrix},$$

for $t \in [0, \sqrt{0.3}]$, and

$$\mathbf{x}_o = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & \pm\mu_1/20 \end{pmatrix} \begin{pmatrix} t^2 \\ 0.2969t - 0.1260t^2 - 0.3520t^4 + 0.2832t^6 - 0.1021t^8 \end{pmatrix},$$

for $t \in [\sqrt{0.3}, 1]$. In this way, we can deal with a complex geometry of real life interest without involving too many parameters (here $P = 1$). By using `rbMIT`, an affine mapping is automatically built on a RB triangulation made by $K_{dom} = 98$ subdomains, leading to an affine decomposition of $Q_a + Q_b = 65$ terms[j]; as for test case 1, $\mathbf{v}_\infty = (1, 0)$, $\nu = 10^{-1}$ and $\mathbf{f} = (0, 0)$; optimal shape and corresponding Stokes flow are shown in Figure 6.

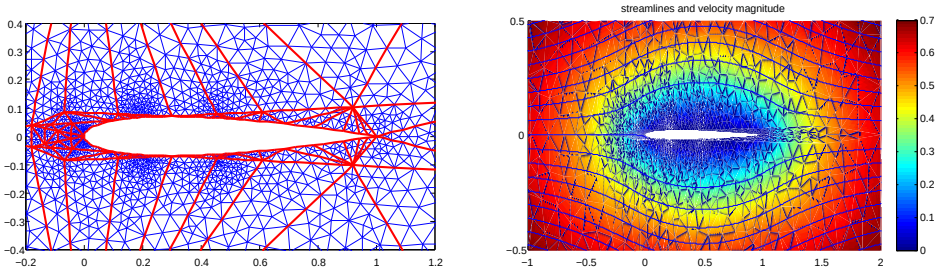

Figure 6: Test case 2: geometry decomposition and velocity field (magnitude and streamlines, right).

6.3 Test case 3: free-form deformation techniques

Let us consider a squared domain $D = [0, 1] \times [0, 1]$ and a body B given by a symmetric (centered) airfoil NACA profile as in test case 2, but of length $l = 0.2$. Let us introduce a grid of 6×6 control points on the domain $\Omega = D \times B$ of which the eight central ones (see Figure 7) are allowed to move in the vertical direction, giving $P = 8$ degrees of freedom, with $\mu_i \in [-0.25, 0.25]$, $i = 1, \ldots, 8$. Here $K_{dom} = 1$, $\nu = 10^{-1}$, $\mathbf{f} = (0, 0)$ and a parabolic flow is imposed at the inflow, i.e. $\mathbf{v}_\infty = (4x_2(1 - x_2), 0)$ on Γ_{in} and $\mathbf{v}_\infty = (0, 0)$ on Γ_w.

[j]Many terms are economized assembling together linear dependent entries and adding some initial points for a more "intelligent" RB triangulation.

Empirical interpolation on $\boldsymbol{\nu}$ and $\boldsymbol{\chi}$ components give affine expansions as in (30):

$$\tilde{\boldsymbol{\nu}}_{i,j}(\mathbf{x};\boldsymbol{\mu}) = \sum_{n=1}^{K_{ij}^a} \tilde{\beta}_n^{i,j}(\boldsymbol{\mu})\tilde{\xi}_n^{i,j}(\mathbf{x}), \qquad \tilde{\boldsymbol{\chi}}_{i,j}(\mathbf{x};\boldsymbol{\mu}) = \sum_{n=1}^{K_{ij}^b} \tilde{\gamma}_n^{i,j}(\boldsymbol{\mu})\tilde{\zeta}_n^{i,j}(\mathbf{x}),$$

leading to a problem dimension of $Q_a = \sum_{i,j=1}^{2} K_{ij}^a = 413$, $Q_b = \sum_{i,j=1}^{2} K_{ij}^b = 19$ for bilinear forms and $Q_F = Q_a$, $Q_G = Q_b$ for right-hand-sides. Optimal shape and corresponding Stokes flow are shown in Figure 7.

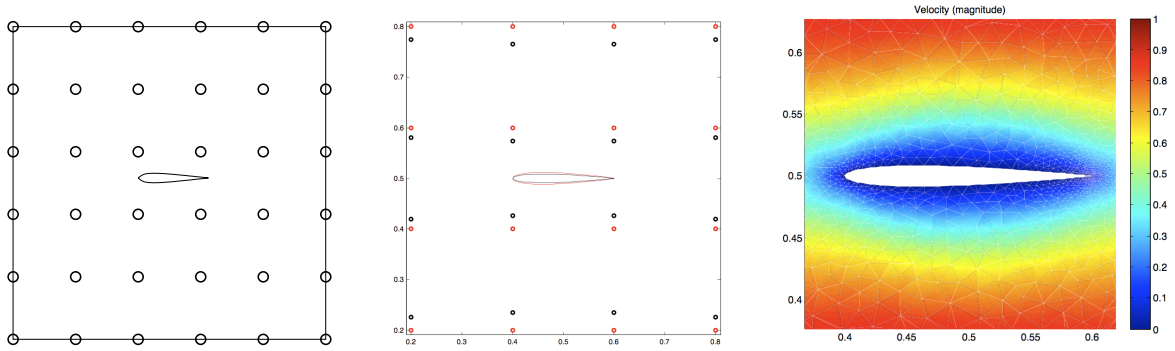

Figure 7: Test case 3: reference domain and FFD setting; shape deformation and control point displacement (unperturbed $\mathbf{P}_{l,m}$ and perturbed $\mathbf{P}_{l,m}^o$ control points are depicted in red and black respectively); detail of the velocity field around the optimal profile (magnitude).

6.4 A comparison of numerical results and proposed techniques

Approximation details for numerical test cases presented below are summarized in Table 1. For each test case, drag forces are minimized for profiles with smallest thickness among admissible values. Moreover, also in test case 3 optimal shape is a bi-convex symmetric profile, even if admissible configurations in this case are not *a priori* symmetric. By the proposed shape parametrization techniques we may increase computational complexity and flexibility without involving a growth in parametric complexity. A general summary with respect to some geometrical criteria is reported in Table 2. Affine mapping built "by hands" are well suited for simple descriptions related to basic geometrical properties used as input parameters (e.g. lengths, thicknesses, diameters or angles, etc.); instead, the `rbMIT` automatic procedure allows to deal with more complex configurations, but low-order parametrizations still lead to restricted families of admissible shapes.
FFD techniques are thus the most proper choice, being both versatile and low-dimensional. In this case, control point perturbations yield small and global deformations of the whole space and objects inside it – giving automatically a mesh deformation too – even if parametrized mappings are independent with respect to the shape parametrization and the computational mesh[k].

[k]Parametrization is mesh independent if for a given geometry and mesh we can construct a parametrization of arbitrary accuracy without remeshing.

Approximation data	Test case 1	Test case 2	Test case 3
Number of parameters P	2	1	8
Number of subdomains K_{dom}	4	98	1
Problem dimension $Q_a + Q_b$	24	65	432
Affine/non-affine problem	affine	affine	non-affine

Table 1: Approximation details for numerical test cases presented.

Criteria	LBV	PSP	Affine/non affine by hand	Automatic affine `rbMIT`	Free-Form (FFD)
design variable set	big	small	small	small	small
geometrical properties	no	yes/no	yes	yes/no	no
shape deformations	small	large	large	large	small
smooth perturbations	no	yes	yes	yes	yes
mesh deformation	yes	no	no	no	yes
mesh independence	no	no	no	no	yes
affine $\boldsymbol{\mu}$ dependence	no	no	yes/no	yes	no

Table 2: Summary of proposed shape parametrization techniques.

7 CONCLUSIONS

Three parametrization techniques for geometrical reduction in shape optimization have been presented, and an example of drag minimization on a profile in a Stokes flow has been discussed. Free-form deformations allow to realize a good compromise between versatility and complexity; moreover, reduced basis methods have been effectively coupled with these techniques for a further computational reduction in such a many-query context.

REFERENCES

[1] A. Jameson, J.C. Vassberg, Computational fluid dynamics for aerodynamic design: its current and future impact, In proceedings of the *39th AIAA Aerospace Sciences Meeting and Exibit*, Reno NV, Paper n°538 (2001).

[2] A. Quarteroni, G. Rozza, Optimal control and shape optimization of aorto-coronaric bypass anastomoses, *Math. Models Meth. Appl. Sci.*, **13**(12), 1801–1823 (2003).

[3] G. Rozza, Shape design by optimal flow control and reduced basis techniques: applications to bypass configurations in haemodynamics, *Ph.D. Thesis, EPFL* (2005).

[4] J. Haslinger and R.A.E. Mäkinen, Introduction to shape optimization: theory, approximation, and computation, *SIAM* (2003).

[5] M.D. Gunzburger, Perspectives in Flow Control and Optimization, *SIAM* (2003).

[6] J. Cea, Conception optimale ou identification de formes: calcul rapide de la dérivée directionelle de la fonction cout, *M2AN*, **20**(3), 371–402 (1986).

[7] G. Allaire, Conception optimale de structures, *Springer*, **Vol. 58** Mathématiques et Applications (2007).

[8] J. Sokolowski and J.-P. Zolésio, Introduction to shape optimization: shape sensitivity analysis, *Springer-Verlag* (1992).

[9] J.A. Samareh, A survey of shape parametrization techniques, Technical Report *NASA/CP*-1999-209136, NASA (1999).

[10] E.I. Amoiralis and I.K. Nikolos, Free-form deformation versus B-spline representation in inverse airfoil design, *J. Comput. Inf. Sci. Eng.* **8**, 024001 (2008).

[11] F. Bélahcène and J.A. Désidéri. Paramétrisation de Bézier adaptative pour l'optimisation de forme en Aérodynamique. *INRIA, Rapp. Rech. 4943* (2003).

[12] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Arch. Comput. Methods Engrg.*, **15**, 229–275 (2008).

[13] rbMIT Library. `http://augustine.mit.edu/methodology/methodology_rbmit_system.htm`. *MIT, Cambridge* (2007–2010). ©Massachusetts Institute of Technology.

[14] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *C. R. Math. Acad. Sci. Paris*, **339**(9), 667–672 (2004).

[15] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models, *Comput. Graph.*, **20**(4), 151–160 (1986).

[16] T. Lehnhäuser, M. Schäfer, A numerical approach for shape optimization of fluid flow domains. *Comput. Meth. Appl. Mech. Engrg.*, **194**, 5221–5241 (2005).

[17] M. Andreoli, A. Janka, and J.A. Désidéri, Free-form deformation parametrization for multilevel 3D shape optimization in aerodynamics, *INRIA, Rapp. Rech. 5019* (2003).

[18] T. Lassila and G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, *Comput. Meth. Appl. Mech. Engr.*, in press (2010).

[19] G. Rozza, T. Lassila, and A. Manzoni, Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map, In proceedings of the *International Conference on Spectral and High Order Methods*, ICOSAHOM 09, in press (2010).