Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 284 (2015) 1147–1180

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Reduced Basis Isogeometric Methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils

Andrea Manzoni, Filippo Salmoiraghi, Luca Heltai*

SISSA-International School for Advanced Studies, Via Bonomea 265, 34136 Trieste, Italy

Available online 17 December 2014

Highlights

- Optimization and design problems require many costly evaluations of system of PDEs.
- Reduced basis methods allow the reliable and real time solution of PDEs.
- Iso-Geometric Analysis is used for complex geometrical parametrizations.
- We present the first coupling of RB methods with IGA for aero and hydro dynamics.
- We validate the coupling with potential flows around NACA airfoils.

Abstract

We present a Reduced Basis (RB) method based on Isogeometric Analysis (IGA) for the rapid and reliable evaluation of PDE systems characterized by complex geometrical features. At the current state of the art, this is the first case of coupling between RB and IGA methods. The construction of the RB method relies on an Isogeometric Boundary Element Method (IGA-BEM) as the high-fidelity technique, allowing a direct interface with Computer Aided Design (CAD) tools. A suitable Empirical Interpolation Method (EIM) ensures an efficient offline/online decomposition between the construction and the evaluation of the RB method. We consider the real-time simulation of potential flows past airfoils, parametrized with respect to the angle of attack and the NACA number identifying their shape, and we provide a validation of our methodology with respect to experimental data and reference numerical codes, showing in both cases a very good agreement.

Keywords: Isogeometric analysis; Reduced order models; Reduced basis method; Proper orthogonal decomposition; Boundary element method; Computational fluid dynamics

1. Overview and motivations

Optimization, control and design problems in aero and hydro dynamics require several evaluations of systems described by partial differential equations (PDEs) and characterized by complex geometrical features. Each realization usually entails some expensive steps: (i) the generation of a volumetric mesh starting from CAD based surfaces, (ii) a high fidelity flow simulation and (iii) the evaluation of the objective function which drives the optimization

* Corresponding author. Tel.: +39 040 3787449; fax: +39 040 3787528.
E-mail addresses: andrea.manzoni@sissa.it (A. Manzoni), filippo.salmoiraghi@sissa.it (F. Salmoiraghi), luca.heltai@sissa.it (L. Heltai).

http://dx.doi.org/10.1016/j.cma.2014.11.037

and possibly a geometrical transformation (of the CAD structure and/or the computational mesh) in case of optimal design problems. Such an optimization process may require up to $\mathcal{O}(10^3)$ system evaluations, yielding a very strong computational effort. To face such a problem in a very efficient way, we develop a Reduced Basis method based on Isogeometric Analysis (RB-IGA) for the real-time and reliable solution of flow problems defined over complex geometries. By combining these two techniques, which have become very well-spread tools of scientific computing during the last decade, we implement a new, integrated framework which alleviates the computational burden which usually characterizes optimization, control and design problems in aero and hydro dynamics.

Reduced basis methods – and more generally speaking Reduced-Order Models (ROMs) – are nowadays a well-known paradigm to enhance computational efficiency in several engineering fields [1]. Among many contributions in computational mechanics and fluid dynamics, we point out the chance to get real-time visualization and output evaluation [2,3] and the efficient solution of optimal control [4] and optimal design problems [5]. Several works regarding optimal design have also pointed out the need of a simultaneous reduction of both computational and geometrical complexity [6,7,5]. Complex shape deformations by means of low-dimensional parametrizations are rather difficult to obtain, especially if one wishes to remain flexible with respect to the obtainable configurations. The description of complex shapes is usually obtained with several design parameters, yielding an intrinsic degree of complexity, especially when one is interested in the efficient solution of PDEs in such a multi-scenario context.

To face such a numerical challenge, a proper combination of RB methods and IGA results in a very attractive and versatile tool. RB methods enable to solve parametrized PDEs in a very short amount of time (at the limit, in a real-time fashion), involving very few degrees of freedom thanks to suitable low-dimensional, problem-dependent approximation spaces. IGA provides instead, in this context, a very general tool to develop complex geometrical parametrizations – recovering a parametrized form to the PDEs at hand – and a very efficient high-fidelity solver (involving the same set of basis functions for both the CAD geometry and the PDE solution representation [8]) over which we develop our RB method.

As a benchmark to investigate our coupled RB-IGA method, we consider the solution of potential flows past NACA airfoils, parametrized with respect to the angle of attack and the NACA number identifying their shape. This problem can find several motivations in many applied contexts, ranging from wing profiles to marine propellers and wind farms.

For the high-fidelity approximation of the problem we use a *Galerkin isogeometric boundary element method*, where the potential flow problem is modeled through a suitable boundary integral formulation. This allows to discretize and solve the problem only on the boundary of the domain at hand, leading to a further reduction of the computational complexity, allowing a direct interface with CAD tools and the capability to describe shapes (like the ones related with the NACA family) in a very general and low-dimensional way, depending on very few design parameters.

The proposed RB method exploits a Galerkin projection over a low-dimensional subspace made by well-chosen high-fidelity solutions, according to either (i) a proper orthogonal decomposition, or (ii) a greedy algorithm based on a suitable error indicator. We compare the computational features of both these strategies, by validating our results against experimental data. Thanks to a suitable formulation of PDE operators relying on IGA parametrizations, and the application of an Empirical Interpolation Method (EIM) [9], we can express each parametrized operator as a finite sum of products involving parameter-dependent functions and parameter-independent vectors/matrices. This plays a key role in view of a complete Offline/Online decomposition, so that the cost entailed by each evaluation of the RB approximation (for any new parameter instance) is completely independent of the dimension of the high-fidelity (here, the IGA-BEM) approximation.

We have implemented a Galerkin isogeometric boundary element method for potential flows, addressing the correct treatment of lifting effects and wake for Galerkin IGA-BEM, and for the first time:

i. we have coupled IGA and RB method in a versatile and integrated framework, capable to manage PDE systems characterized by complex geometrical features;
ii. we have provided a validation of computational results – obtained both with the high-fidelity IGA-BEM and the RB methods – with respect to experimental data and reference numerical codes (Xfoil and other IGA-BEM tools), showing in both cases a very good agreement.

We highlight two last (indeed, very relevant) points. RB approximation of parametrized potential flows has already been presented in [10], but without addressing the correct treatment of lifting effects and wake. In this respect, the boundary integral formulation we present in Section 2 is the key ingredient in order to tackle both wake and Kutta conditions and to recover correct physical solutions. The case of affinely parametrized airfoils has been addressed in [11] and, more recently, in [12] for the solution of certified parametric optimization problems; a more general Free-Form Deformation (FFD) technique has been presented in [6]. Although more flexible, this latter technique does not allow to produce airfoil shapes in the NACA family, due to the perturbation of a lattice of control points which are not directly related to the airfoil boundary. The RB-IGA coupling presented in this work allows to (i) recover solutions of physical meaning and (ii) treat each airfoil by means of a shape parametrization involving its NACA number explicitly.

The IGA framework offers very interesting features from the perspective of geometrical parametrization, thanks to its direct interface with CAD data structures, and it is certainly not limited to the simplified setting we present in this work. Arbitrarily complex geometrical features can be described with very few control points which could, in principle, be used directly as parameters for the RB approach. The example we present in this work should be seen as a proof of concept to explore the capabilities of the RB-IGA coupling. We start this exploration by considering potential flows, and we use as high-fidelity solver an IGA-BEM approach, but the techniques we describe are not limited to these two cases.

Using Finite Element (FE) IGA as the high fidelity solver would allow the treatment of more realistic flow problems, as done, for instance, in [13] using standard FE analysis. A foreseeable difficulty with the FE-IGA approach for general problems, however, would be the automatic volumetric IGA grid generation, which is still an open problem for arbitrary CAD geometries. The fact that potential flows admit a boundary integral representation, and therefore do not require volumetric meshes, makes the RB-IGA-BEM coupling a good starting point for this study.

Moreover, we remark that RB methods have been historically built upon finite element discretizations. Only few applications have exploited a boundary element method as high-fidelity approximation; see, e.g., [14–16] for problems in electromagnetism. Nevertheless, in the mentioned works the parametric dependence is always related to physical features. Again, we provide in this work, for the first time ever, a RB method built over a boundary element approximation featuring also a complex geometric parameter dependence, thanks to several key features provided by isogeometric analysis.

The structure of the paper is as follows. In Section 2 we discuss the continuous formulation of the problem, which we formalize as a parametrized boundary integral equation. Its discretization is described in Section 3, which introduces the high-fidelity IGA-BEM method used to construct the reduced-order approximation, described in Section 4. We show some numerical results in Section 5 and we draw some conclusions in Section 6. For a deeper insight about the less relevant, but still important, details concerning the construction of our framework, see [17].

2. Modeling potential flows about NACA airfoils

Viscous Newtonian fluids are often modeled by Navier–Stokes equations. Numerical solution of these equations can be very expensive, and their use at the beginning of the design stages might be an overkill. For these cases, a good starting point consists in neglecting viscous and transient effects, reducing the equation of motions to the static incompressible Euler equations, coupled with Bernoulli equations, for the unknown velocity $\boldsymbol{V}$ and pressure p.

2.1. *Problem formulation*

Once we neglect viscosity, the equations of motions can capture inviscid features, such as lift, but not viscous effects, like turbulence or boundary layers. If we further assume that the fluid motion is irrotational, so that $\psi \equiv \nabla \times \boldsymbol{V} = 0$, then there exists a scalar potential function Φ such that $\boldsymbol{V} = \nabla\Phi$ and the equations of motions simply become

$$\begin{aligned} \frac{1}{2}|\nabla\Phi|^2 + \frac{p}{\rho} &= 0 \quad \text{in } \Omega \subseteq \mathbb{R}^2 \\ -\Delta\Phi &= 0 \quad \text{in } \Omega, \end{aligned} \tag{1}$$

where the potential Φ and the pressure p are uncoupled variables. It is customary to split the potential into a uniform, known, flow potential (which is also harmonic) and an unknown *perturbation potential* ϕ, that is

$$\Phi = \phi_\infty + \phi = \boldsymbol{V}_\infty \cdot \boldsymbol{x} + \phi, \tag{2}$$

where $\boldsymbol{V}_\infty$ is the inflow velocity of the fluid, and $\boldsymbol{x}$ denotes the spatial variables.

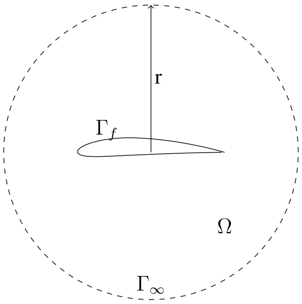

Fig. 1. Original domain Ω.

We denote with $\Omega \subset \mathbb{R}^2$ a planar region surrounding an airfoil delimited by the outer boundary Γ_∞ (which is assumed to be infinitely far from the airfoil) and by the airfoil boundary Γ_f (see Fig. 1), such that $\partial\Omega = \Gamma_\infty \cup \Gamma_f$.

Far from the airfoil the perturbation potential is set to be zero, i.e., $\lim_{|r|\to\infty} \phi(r) = 0$, and we treat this latter condition as a Dirichlet boundary condition on Γ_∞:

$$\phi = 0 \quad \text{on } \Gamma_\infty. \tag{3}$$

On the airfoil Γ_f the fluid is only allowed to slip and we impose the non penetration condition

$$\boldsymbol{V} \cdot \boldsymbol{n} = \nabla \Phi \cdot \boldsymbol{n} = \boldsymbol{V}_\infty \cdot \boldsymbol{n} + \nabla\phi \cdot \boldsymbol{n} = 0, \tag{4}$$

that is

$$\frac{\partial \phi}{\partial n} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n} \quad \text{on } \Gamma_f, \tag{5}$$

where $\boldsymbol{n}$ is the outer normal to Γ_f.

The problem to be tackled reads as follows: given $\boldsymbol{V}_\infty$, find ϕ such that

$$\begin{aligned} &-\Delta\phi = 0 \quad \text{in } \Omega \\ &\frac{1}{2}|\nabla\phi|^2 + \frac{p}{\rho} = 0 \quad \text{in } \Omega \\ &\frac{\partial \phi}{\partial n} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n} \quad \text{on } \Gamma_f \\ &\phi = 0 \quad \text{on } \Gamma_\infty. \end{aligned} \tag{6}$$

Without further assumptions, the solution to Problem (6) has always null circulation. Kutta–Joukowski theorem then states that no lift can be produced by the airfoil, and the stagnation points of the flow will be in non-physical locations [18]. From the mathematical standpoint, it is necessary to relax the irrotationality condition in order to allow for some circulation and induce a lift in the airfoil. This is usually done on the basis of experimental observations: when sharp trailing edges are present in the geometry, a thin turbulent layer detaches from the trailing edge (TE) of the airfoil, forming a *wake* where the irrotationality assumption is no longer satisfied.

Potential theory is still useful if we introduce a cut Γ_w in the domain (see Fig. 2). This operation makes the domain Ω simply connected and allows the potential ϕ to be a discontinuous function when crossing Γ_w. From the physical standpoint, we can model the flow by considering the airfoil as a smooth surface Γ_f with a sharp trailing edge TE, and by assuming that the vorticity is concentrated on an infinitely thin wake Γ_w (a vortex sheet) detaching from the trailing edge. Vorticity is released into the fluid in the form of a jump in the potential ϕ, making the flow almost everywhere irrotational, except on the wake.

Let us define Γ_{w^+} and Γ_{w^-} as

$$\Gamma_{w^+} := \Gamma_w + \frac{\varepsilon}{2}\boldsymbol{n} \quad \text{and} \quad \Gamma_{w^-} := \Gamma_w - \frac{\varepsilon}{2}\boldsymbol{n}, \tag{7}$$

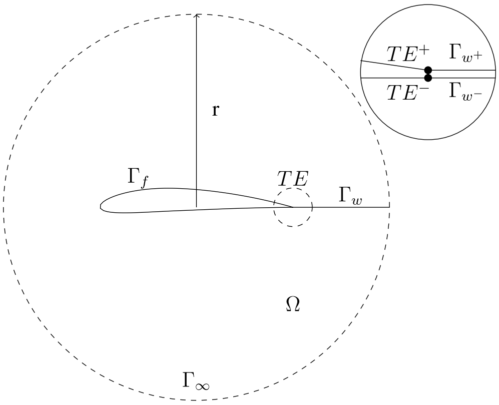

Fig. 2. Domain in presence of the wake Γ_w; here TE denotes the Trailing Edge of the airfoil.

such that

$$\lim_{\varepsilon\to 0} \Gamma_{w^+} = \lim_{\varepsilon\to 0} \Gamma_{w^-} = \Gamma_w, \tag{8}$$

where $\boldsymbol{n}$ denotes the unit vector normal to Γ_w. Consider then TE^+ and TE^- defined as

$$TE^+ := TE + \frac{\varepsilon}{2}\boldsymbol{n} \quad \text{and} \quad TE^- := TE - \frac{\varepsilon}{2}\boldsymbol{n}, \tag{9}$$

such that

$$\lim_{\varepsilon\to 0} TE^+ = \lim_{\varepsilon\to 0} TE^- = TE. \tag{10}$$

We allow the solution on the trailing edge and along the wake to be discontinuous, that is

$$\phi(TE^+) \neq \phi(TE^-) \qquad \phi(\Gamma_{w^+}) \neq \phi(\Gamma_{w^-}). \tag{11}$$

On the wake Γ_w we impose the potential jump to be equal to the one at the trailing edge:

$$[\phi] = \phi(TE^+) - \phi(TE^-) \quad \text{on } \Gamma_w, \tag{12}$$

where $[a]$ denotes the jump operator, defined as the difference between the quantity a on Γ_{w^+} and Γ_{w^-}. We refer to (12) as the *wake condition*, and we remark that on the wake we are only interested in the potential jump and not in the potential itself.

The physical variable is the velocity (not the potential), and we require its normal component to be continuous across the wake, which is equivalent to imposing a homogeneous Neumann boundary condition for the potential jump:

$$[\boldsymbol{V}\cdot\boldsymbol{n}] = [\boldsymbol{V}_\infty\cdot\boldsymbol{n} + \nabla\phi\cdot\boldsymbol{n}] = \left[\frac{\partial\phi}{\partial n}\right] = 0 \quad \text{on } \Gamma_w. \tag{13}$$

Excluding the pressure (which can be recovered a posteriori thanks to the decoupling of (1)) the equations of motion which include the wake read: given $\boldsymbol{V}_\infty$, find ϕ such that

$$\begin{cases} -\Delta\phi = 0 & \text{in } \Omega \\ [\phi] - \phi(TE^+) + \phi(TE^-) = 0 & \text{on } \Gamma_w \\ \dfrac{\partial\phi}{\partial n} = -\boldsymbol{V}_\infty\cdot\boldsymbol{n} & \text{on } \Gamma_f \\ \left[\dfrac{\partial\phi}{\partial n}\right] = 0 & \text{on } \Gamma_w \\ \phi = 0 & \text{on } \Gamma_\infty. \end{cases} \tag{14}$$

In problem (14), the potential jump across the wake is not fixed, and an infinite number of solutions exist, differing only by the circulation around the airfoil (see, e.g., [19]). A possible way to select only the physically plausible solution, is to introduce the well-known *Kutta condition* (see, e.g., [20] for a related discussion).

One possible form of the Kutta condition is to enforce a zero velocity condition at the trailing edge. Since this requirement is too strong in the numerical sense (and too far from the real flow observations, see, e.g., [18]), we impose a weaker condition by requiring that the velocity amplitude coincides on the upper and lower sides of the trailing edge:

$$V(TE^+) = V(TE^-), \tag{15}$$

and that its direction is consistent with the unperturbed flow velocity $\boldsymbol{V}_\infty$:

$$(\boldsymbol{\tau} \cdot (\boldsymbol{V}_\infty + \nabla_s \phi))^+ = -(\boldsymbol{\tau} \cdot (\boldsymbol{V}_\infty + \nabla_s \phi))^-, \tag{16}$$

where $\boldsymbol{\tau}$ denotes the unit tangential vector along the surface, pointing in the counter-clockwise direction, ∇_s is the surface gradient operator, $V = |\nabla \Phi|$ and the superscript $^+$ and $^-$ refer to TE^+ and TE^-, respectively. The definition of surface gradient applied to the potential ϕ is given by

$$\nabla_s \phi := \nabla \phi - (\boldsymbol{n} \cdot \nabla \phi)\boldsymbol{n}, \tag{17}$$

where $\boldsymbol{n}$ is the unit vector normal to the curve. Condition (15) is a constraint ensuring that the pressure jump on the trailing edge is zero. This constraint is also known as *dynamic* Kutta condition, which ensures uniqueness of the solution of problem (14) (see, e.g., [21]).

2.2. Boundary integral formulation on a reference domain

Let us consider the fundamental solution of the Laplace equation, or free space *Green's function*, i.e., the function $G : \mathbb{R}^2 \to \mathbb{R}$ such that

$$-\Delta G(\boldsymbol{x} - \boldsymbol{y}) = \delta(\boldsymbol{x} - \boldsymbol{y}), \tag{18}$$

which represents the potential $\Phi(\boldsymbol{x})$ given a pointwise source $\delta(\boldsymbol{x} - \boldsymbol{y})$ concentrated at $\boldsymbol{y}$. In the two dimensional case, G takes the following form:

$$G(\boldsymbol{x} - \boldsymbol{y}) = -\frac{1}{2\pi} \ln |\boldsymbol{x} - \boldsymbol{y}|, \tag{19}$$

so that the solution to the Laplace equation over the domain Ω with zero forcing term results as follows:

$$\phi(\boldsymbol{x}) = \int_{\partial\Omega} G(\boldsymbol{x} - \boldsymbol{y}) \frac{\partial \phi}{\partial n_y}(\boldsymbol{y}) \mathrm{d}\Gamma_y - \int_{\partial\Omega} \phi(\boldsymbol{y}) \frac{\partial G}{\partial n_y}(\boldsymbol{x} - \boldsymbol{y}) \mathrm{d}\Gamma_y, \tag{20}$$

where $\partial\Omega = \Gamma_f \cup \Gamma_{w^+} \cup \Gamma_{w^-} \cup \Gamma_\infty$. This boundary integral representation is valid for any $\boldsymbol{x} \in \Omega$. The two integrals

$$\int_{\partial\Omega} G \frac{\partial \phi}{\partial n} \mathrm{d}\Gamma \quad \text{and} \quad \int_{\partial\Omega} \phi \frac{\partial G}{\partial n} \mathrm{d}\Gamma \tag{21}$$

are usually referred to as *single layer potential* (SLP) and *double layer potential* (DLP), respectively. It can be shown that the SLP is a continuous function as the point $\boldsymbol{x}$ approaches and then crosses $\partial\Omega$. By contrast, as $\boldsymbol{x}$ crosses Γ, the DLP undergoes a jump discontinuity (see, e.g., [22] for a proof). For any $\boldsymbol{x}$ on Γ, Eq. (20) can be written as

$$\sigma(\boldsymbol{x})\phi(\boldsymbol{x}) = \int_{\partial\Omega} G(\boldsymbol{x} - \boldsymbol{y}) \frac{\partial \phi}{\partial n_y}(\boldsymbol{y}) \mathrm{d}\Gamma_y - ⨍_{\partial\Omega} \phi(\boldsymbol{y}) \frac{\partial G}{\partial n_y}(\boldsymbol{x} - \boldsymbol{y}) \mathrm{d}\Gamma_y, \tag{22}$$

where $⨍$ means that we are taking the principal value of the integral,[1]and the quantity σ is the fraction of the angle

[1] By principal value of the integral, we mean

$$⨍_{\partial\Omega} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x} - \boldsymbol{y})}{\partial n_y} dy := \lim_{\varepsilon \to 0} \int_{\Gamma \setminus B_\varepsilon} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x} - \boldsymbol{y})}{\partial n_y} dy, \tag{23}$$

where B_ε is the ball of radius ε centered in $\boldsymbol{x} \in \Gamma$.

at the point $\boldsymbol{x}$ by which the point itself sees the fluid domain Ω. If the boundary is Lipschitz, σ is almost everywhere equal to $\frac{1}{2}$, while on the (coinciding) boundaries $\Gamma_{w^+} = \Gamma_{w^-} = \Gamma_w$, σ is equal to one.

If we take into account the overlapping of the upper and lower boundaries of Γ_w, then the relevant unknowns on Γ_w become the *jumps* of the potential ϕ and of its normal derivative. Let us define

$$u := \begin{cases} \phi & \text{on } \Gamma_f \\ [\phi] & \text{on } \Gamma_w \end{cases} \quad \text{and} \quad w := \begin{cases} \dfrac{\partial \phi}{\partial n_y} & \text{on } \Gamma_f \\ \left[\dfrac{\partial \phi}{\partial n_y}\right] & \text{on } \Gamma_w, \end{cases} \tag{24}$$

respectively, where $[\phi]$ and $\left[\frac{\partial \phi}{\partial n_y}\right]$ are constant real functions on Γ_w. Exploiting the Neumann boundary conditions expressed in (5) and (13), we can write

$$w = \begin{cases} -\boldsymbol{V}_\infty \cdot \boldsymbol{n} & \text{on } \Gamma_f \\ 0 & \text{on } \Gamma_w. \end{cases} \tag{25}$$

The two integrals appearing in (22) can be computed on $\Gamma := \Gamma_f \cup \Gamma_w \neq \partial\Omega := \Gamma_f \cup \Gamma_{w^+} \cup \Gamma_{w^-}$, and we can write:

$$\begin{aligned} \int_{\partial\Omega} G \frac{\partial \phi}{\partial n_y} &= \int_{\Gamma_f} G \frac{\partial \phi}{\partial n_y} + \int_{\Gamma_{w^+}} G \frac{\partial \phi}{\partial n_y} + \int_{\Gamma_{w^-}} G \frac{\partial \phi}{\partial n_y} \\ &= \int_{\Gamma_f} G \frac{\partial \phi}{\partial n_y} + \int_{\Gamma_w} G \left[\frac{\partial \phi}{\partial n_y}\right] = \int_{\Gamma} G w \end{aligned} \tag{26}$$

and

$$\begin{aligned} \fint_{\partial\Omega} \phi \frac{\partial G}{\partial n_y} &= \fint_{\Gamma_f} \phi \frac{\partial G}{\partial n_y} + \fint_{\Gamma_{w^+}} \phi \frac{\partial G}{\partial n_y} + \fint_{\Gamma_{w^-}} \phi \frac{\partial G}{\partial n_y} \\ &= \fint_{\Gamma_f} \phi \frac{\partial G}{\partial n_y} + \fint_{\Gamma_w} [\phi] \frac{\partial G}{\partial n_y} = \fint_{\Gamma} u \frac{\partial G}{\partial n_y}, \end{aligned} \tag{27}$$

respectively. If we exploit the boundary condition over Γ_f and Γ_w in (22), we can reformulate problem (14) as the following boundary integral equation (BIE): given w, find u such that

$$\begin{aligned} &\sigma(\boldsymbol{x})u(\boldsymbol{x}) + \fint_{\Gamma} u(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy = \int_{\Gamma} G(\boldsymbol{x}-\boldsymbol{y}) w(\boldsymbol{y}) dy && \text{on } \Gamma_f \\ &u(\boldsymbol{x}) - u(TE^+) + u(TE^-) = 0 && \text{on } \Gamma_w \end{aligned} \tag{28}$$

subject to the following constraint, representing the Kutta condition:

$$\boldsymbol{\tau}(TE^+) \cdot \nabla_s u|_{TE^+} + \boldsymbol{\tau}(TE^-) \cdot \nabla_s u|_{TE^-} = \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(TE^+) + \boldsymbol{\tau}(TE^-)) \quad \text{on } TE. \tag{29}$$

In particular, σ is almost everywhere equal to $\frac{1}{2}$ on the airfoil Γ_f.

We end up with an integral equation on the unknown potential and potential jump distribution on Γ_f and Γ_w respectively, closed by an algebraic equation which has to be satisfied on the trailing edge.

2.3. *Parametrized weak formulation*

Our final goal is the efficient description of potential flows about airfoils in different (physical and) geometrical scenarios described through the parameter vector

$$\boldsymbol{\mu} = \begin{Bmatrix} \text{Naca} \\ \alpha \end{Bmatrix} \in \mathcal{D} \subset \mathbb{R}^2. \tag{30}$$

Here α is the angle of attack and Naca is the four digit number used for the description of the airfoil shape (see Appendix A). For each value of the NACA number (when $\alpha = 0$), we obtain a parameter dependent geometry

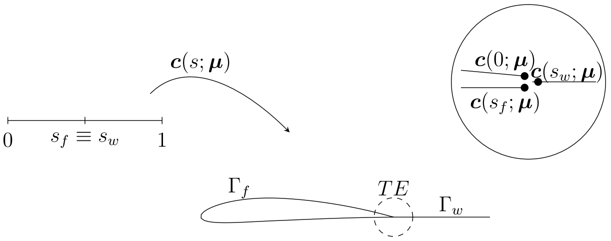

Fig. 3. Mapping of [0, 1] in Γ.

description, which is then rotated according to its angle of attack α. We selected the parameter space $\mathcal{D}$ by considering the fact that some 4-digits NACA profiles are not interesting from an engineering standpoint: we restrict the Naca range, namely [0000, 9999], to 7 not connected subregions within the ranges $[i208, i620]$, $i = 0, \dots, 6$. The inviscid approximation is only meaningful for small angles of attack, we therefore restrict the range of α to $[-20\,\text{deg}, 20\,\text{deg}]$. Provided that the geometry (as well as the boundary condition) are parameter dependent, the entire potential flows problem is parametrized, that is, both ϕ and our output of interest are parameter dependent. For the sake of notation, we omit the $\boldsymbol{\mu}$-dependence in the expression of the potential field $u = u(\boldsymbol{\mu})$ and of derived quantities.

In view of setting an isogeometric boundary element method as the high-fidelity approximation, it is more convenient to formulate the BIE (28) over the reference configuration [0, 1]. In order to map the boundary Γ in the interval [0, 1], let us introduce a change of coordinates, from the original domain Γ to a reference one. Let us denote by $\boldsymbol{c} : ((0,1) \setminus \{s_f\}) \times \mathcal{D} \to \Gamma \subset \mathbb{R}^2$ the map giving this transformation of coordinates (see Fig. 3), with $\boldsymbol{c}' > 0$ (hence invertible), such that $\forall \boldsymbol{\mu} \in \mathcal{D}$

$$\Gamma(\boldsymbol{\mu}) = \boldsymbol{c}(s;\boldsymbol{\mu}), \quad s \in (0,1) \setminus \{s_f\}. \tag{31}$$

In particular, we want to map the intervals $(0, s_f)$ and $(s_w, 1)$, with $s_f \equiv s_w$, onto Γ_f and Γ_w, respectively, so that

$$\begin{aligned} \Gamma_f(\boldsymbol{\mu}) &= \left\{\boldsymbol{c}(s;\boldsymbol{\mu}) | s \in (0, s_f), \boldsymbol{\mu} \in \mathcal{D}\right\} \\ \Gamma_w(\boldsymbol{\mu}) &= \{\boldsymbol{c}(s;\boldsymbol{\mu}) | s \in (s_w, 1), \boldsymbol{\mu} \in \mathcal{D}\}. \end{aligned} \tag{32}$$

Moreover, we want TE^+ and TE^- to be

$$TE^+ = \lim_{s\to 0^+} \boldsymbol{c}(s;\boldsymbol{\mu}), \qquad TE^- = \lim_{s\to s_f^-} \boldsymbol{c}(s;\boldsymbol{\mu}), \tag{33}$$

respectively. Notice that we parametrize the problem in a way that makes the position of trailing edge independent on the parameter $\boldsymbol{\mu}$. From now on, for the sake of readability, we will omit the limits and simply write $\boldsymbol{c}(0;\boldsymbol{\mu})$, $\boldsymbol{c}(s_f;\boldsymbol{\mu})$ and $\boldsymbol{c}(s_w;\boldsymbol{\mu})$.

In this way, by walking along the airfoil in counter clockwise direction from TE^+ to TE^-, $(0, s_f)$ is mapped onto the airfoil. The airfoil and the wake are geometrically coincident at the trailing edge:

$$\boldsymbol{c}(0;\boldsymbol{\mu}) \equiv \boldsymbol{c}(s_f;\boldsymbol{\mu}) \equiv \boldsymbol{c}(s_w;\boldsymbol{\mu}), \tag{34}$$

but we allow the solution u at the trailing edge to be discontinuous:

$$u(0) \neq u(s_f) \neq u(s_w). \tag{35}$$

If we apply the change of coordinates to the integrals appearing in (28), we obtain

$$\begin{aligned} \int_{\Gamma(\boldsymbol{\mu})} u(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy &= \int_0^1 u(q) \frac{\partial G}{\partial n_q} (\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu})) J(q;\boldsymbol{\mu}) dq \\ \int_{\Gamma(\boldsymbol{\mu})} G(\boldsymbol{x}-\boldsymbol{y}) w(\boldsymbol{y}) dy &= \int_0^1 G(\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu})) w(q) J(q;\boldsymbol{\mu}) dq, \end{aligned}$$

where $J = |\boldsymbol{c}'|$ is the determinant of the Jacobian of the transformation, and for the sake of simplicity we kept the same symbols in both the reference and the real configuration, i.e., $u(s) := u(\boldsymbol{c}(s;\boldsymbol{\mu}))$ and the parameter dependence is implicit.

We can now rewrite problem (28) in the following way: given $\boldsymbol{\mu} \in \mathcal{D}$, find u such that

$$\begin{aligned} &\sigma u(s) + (Nu)(s; \boldsymbol{\mu}) = (Dw)(s; \boldsymbol{\mu}) && \text{in } (0, s_f) \\ &u(s) - u(0) + u(s_f) = 0 && \text{in } (s_w, 1), \end{aligned} \tag{36}$$

subject to the constraint

$$\boldsymbol{\tau}(0; \boldsymbol{\mu}) \cdot \nabla_s u|_0 + \boldsymbol{\tau}(s_f; \boldsymbol{\mu}) \cdot \nabla_s u|_{s_f} = \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) \quad \text{on } \{0, s_f\}, \tag{37}$$

where the operators N and D are defined as follows:

$$(Nu)(s; \boldsymbol{\mu}) := \int_0^1 u(q) \frac{\partial G}{\partial n_q} (\boldsymbol{c}(s; \boldsymbol{\mu}) - \boldsymbol{c}(q; \boldsymbol{\mu})) J(q; \boldsymbol{\mu}) dq \tag{38}$$

$$(Dw)(s; \boldsymbol{\mu}) := \int_0^1 G(\boldsymbol{c}(s; \boldsymbol{\mu}) - \boldsymbol{c}(q; \boldsymbol{\mu})) w(q) J(q; \boldsymbol{\mu}) dq. \tag{39}$$

For the sake of the functional description of the weak formulation, let us introduce the Sobolev space $V := H^{\frac{1}{2}}([0, 1] \setminus \{s_f\})$ and its dual V'.

Moreover, let us condense Eq. (36) by introducing a global bilinear form. Given the indicator function χ_A defined as

$$\chi_A(s) = \begin{cases} 1 & \text{for } s \in A \\ 0 & \text{otherwise,} \end{cases} \tag{40}$$

we can write the variational formulation of (36) as follows: given $w \in V'$ and $\boldsymbol{\mu} \in \mathcal{D}$, find $u \in V$ such that

$$a(u, v; \boldsymbol{\mu}) = d(w, v; \boldsymbol{\mu}) \quad \forall v \in V \tag{41}$$

subject to the constraint that

$$V_{TE^+}(\boldsymbol{\mu}) - V_{TE^-}(\boldsymbol{\mu}) = 0 \quad \text{on } \{0, s_f\}, \tag{42}$$

where

$$a(u, v; \boldsymbol{\mu}) := \sigma m(u, v) + n(u, v; \boldsymbol{\mu}) + w(u, v). \tag{43}$$

For any given $\boldsymbol{\mu} \in \mathcal{D}$, the bilinear forms appearing in (43) and (41) are given by

$$\begin{aligned} m(u, v) &:= \int_0^1 v(s) u(s) \chi_{[0, s_f]}(s) \mathrm{d}s && \forall u, v \in V \\ n(u, v; \boldsymbol{\mu}) &:= \int_0^1 v(s) (Nu)(s; \boldsymbol{\mu}) \chi_{[0, s_f]}(s) \mathrm{d}s && \forall u, v \in V \\ w(u, v) &:= \int_0^1 v(s) \left[u(s) - u(0) + u(s_f)\right] \chi_{[s_f, 1]}(s) \mathrm{d}s && \forall u, v \in V \\ d(w, v; \boldsymbol{\mu}) &:= \int_0^1 v(s) (Dw)(s; \boldsymbol{\mu}) \chi_{[0, s_f]}(s) \mathrm{d}s && \forall v \in V,\ \forall w \in V'. \end{aligned}$$

In conclusion, problem (41)–(42) can be represented in the following saddle point form, which includes the Kutta condition as a constraint enforced through a Lagrange multiplier $\pi \in \mathbb{R}$: given $w \in V'$, find u and λ such that

$$\begin{cases} a(u, v; \boldsymbol{\mu}) + b(v, \lambda; \boldsymbol{\mu}) & = \quad d(w, v; \boldsymbol{\mu}) & \forall v \in V \\ b(u, \pi; \boldsymbol{\mu}) & = \quad \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) & \forall \pi \in \mathbb{R}, \end{cases} \tag{44}$$

where

$$b(u, \pi; \boldsymbol{\mu}) = \left[\boldsymbol{\tau}(0; \boldsymbol{\mu}) \cdot \nabla_s u|_0 + \boldsymbol{\tau}(s_f; \boldsymbol{\mu}) \cdot \nabla_s u|_{s_f}\right] \pi. \tag{45}$$

The high fidelity approximation of our model is based on the restriction of V and V' to an isogeometric finite dimensional space, which produces a discrete system of equations in place of the continuous formulation (44).

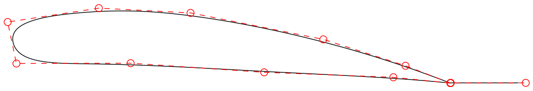

Fig. 4. Control polygon and B-spline curve for NACA 4412 airfoil geometry; coarse knot vector.

3. High-fidelity approximation: Galerkin isogeometric boundary element method (IGA-BEM)

In this section we provide a synthetic description of the high-fidelity IGA-BEM approximation, over which we build our RB methods. Even if several applications have been developed in the FE-IGA framework, relatively few works deal with an IGA-BEM for treating boundary integral equations. Some representative examples can be found, for instance, in [23–25] and [26] for two dimensional problems, in [27,28] for hydrodynamics, in [29] for acoustics and in [30] for shape optimization problems.

Let us introduce a set of *control points* $\{\boldsymbol{P}_i\}_{i=1}^{\mathcal{N}}$, $\boldsymbol{P}_i \in \mathbb{R}^2$, and a set of *B-splines basis functions* $\{\xi_i(s)\}_{i=1}^{\mathcal{N}}$, $\xi_i : [0,1] \to \mathbb{R}$, defined recursively as follows:

$$\xi_{i,0}(s) = \begin{cases} 1 & \theta_i \le s \le \theta_{i+1} \\ 0 & \text{otherwise,} \end{cases} \tag{46}$$

where

$$\xi_{i,k}(s) = \frac{s-\theta_i}{\theta_{i+k}-\theta i}\xi_{i,k-1}(s) + \frac{\theta_{i+k+1}-s}{\theta_{i+k+1}-\theta_{i+1}}\xi_{i+1,k-1}(s) \quad k = 1,\ldots,p. \tag{47}$$

Here $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_{n+p+1}\}^T$, $\theta_i \in \mathbb{R}$, is the *knot vector*, a non-decreasing set of coordinates in the s parameter space, whereas p is the polynomial order of the chosen basis functions. We can characterize the map $\boldsymbol{c}$ introduced in Section 2.3 as:

$$\boldsymbol{c}(s;\boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \xi_i(s)\boldsymbol{P}_i(\boldsymbol{\mu}), \quad \boldsymbol{P}_i \in \mathbb{R}^2. \tag{48}$$

Such a map, to which we refer as *IGA parametrization* from now on, depends on the parameter vector $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$ through the set of control points $\{\boldsymbol{P}_i\}_{i=1}^{\mathcal{N}}$, where $P \ll \mathcal{N}$; see Appendix A for further details about its construction.

This shape parametrization technique is quite different from other recent strategies, such as Free Form Deformation (FFD) or Radial Basis Function (RBF) [5], which typically require more parameters to represent shapes/deformations of comparable complexity. Moreover, in the IGA context the parameters are physically meaningful, shape variations are automatically propagated within the entire flow of the analysis, from the definition of the operators to the construction of algebraic structures. In particular, the IGA parametrization we consider is capable of recovering the whole family of NACA airfoils exactly, by prescribing only its NACA number. This enables to compare the results with experimental data, available for airfoils inside the NACA family (See, for example, Fig. 4 for a possible B-spline parametrisation of the NACA 4412 airfoil geometry).

The surface gradient (17) in the two dimensional isogeometric framework can be written as

$$\nabla_s u := J^{-1}(s)u'(s)\boldsymbol{\tau}(s) \tag{49}$$

where $J(s)$ is the determinant of the Jacobian of the transformation, $u'(s)$ is the first derivative of the potential and $\boldsymbol{\tau}(s)$ is the unit vector tangent to Γ. We refer to [31] for a detailed description of B-spline properties.

3.1. High-fidelity approximation

Boundary element methods (or *panel methods*) have been widely accepted as a useful tool for aerodynamic and hydrodynamic design since the pioneering work of Hess and Smith [32,33]. In this context, the boundary of the profile is divided into a set of portions (or *panels*), where the unknown can be directly the potential, the velocity or the vorticity, approximated as constant over each panel [34,35]. Higher-order panel methods – e.g. by employing

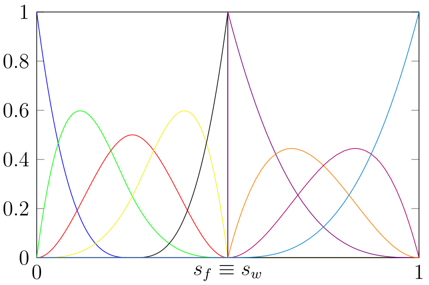

Fig. 5. The nine cubic B-spline basis functions for the open knot vector $\boldsymbol{\theta} = \{0, 0, 0, 0, 0.25, s_f, s_f, s_f, s_f, 1, 1, 1, 1\}$.

B-spline basis functions to represent both the geometry and the potential variable [24,27] – have been developed to improve the prediction of velocity and pressure near the trailing edge and at the tip of the lift-generating surface, where significant numerical integration errors might lead to the degradation of the solution accuracy; see also [36,37]. Since the derivatives of the basis functions can be obtained exactly, there is no need to rely on numerical differentiation to compute the velocity field from the potential: this is another reason that makes IGA a suitable tool for these problems.

The high-fidelity approximation of problem (44) can be obtained by projecting (41) onto the finite dimensional space $V_h \subset V$; in the same way, $V_h' \subset V'$ denotes a finite dimensional subspace of V'. The Galerkin-BEM formulation of the problem becomes: given $w_h \in V_h'$ and $\boldsymbol{\mu} \in \mathcal{D}$, find $u_h \in V_h$ such that

$$a(u_h, v_h; \boldsymbol{\mu}) = d(w_h, v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h, \tag{50}$$

where $u_h = u_h(\boldsymbol{\mu}) \in V_h$ denotes the high-fidelity BEM solution and $\mathcal{N} = \dim(V_h)$. The main idea behind isogeometric analysis is that the basis functions used to describe the geometry can also be employed as basis functions to express the (approximate) solution of the differential method. By following this approach, we can express the IGA-BEM solution of (50) as

$$u_h(s) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) u_i, \tag{51}$$

and we define the finite dimensional space $V_h \subset V$ as follows:

$$V_h = \text{span}\{\xi_i, i = 1, \dots, \mathcal{N}\}. \tag{52}$$

Equivalently, we can rewrite Eq. (50) as the following linear system:

$$\sum_{j=1}^{\mathcal{N}} a(\xi_j, \xi_i; \boldsymbol{\mu}) u_j = d(w_h, \xi_i; \boldsymbol{\mu}), \quad i = 1, \dots, \mathcal{N}. \tag{53}$$

Thanks to the properties of B-splines functions, by repeating $p+1$ times the knot s_f in the knot vector $\boldsymbol{\theta}$, we allow the solution to be discontinuous at the point s_f (see Fig. 5). In the same way, the approximate version of the Neumann boundary condition is given by

$$w_h(s) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) w_i. \tag{54}$$

Here u_i and w_i are respectively the control values of the potential u and of the Neumann datum w. Moreover, let us denote by N_f and N_w the number of basis functions for the airfoil and the wake, respectively. Let us recall that, thanks to (43), (50) can be expressed as

$$\sigma m(u_h, v_h) + n(u_h, v_h; \boldsymbol{\mu}) + w(u_h, v_h) = d(w_h, v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h. \tag{55}$$

Exploiting (51), (54), and the definition of the indicator function (40), the linear system (53) can be written in a more compact form as follows:

$$\widehat{\boldsymbol{A}}(\boldsymbol{\mu})\widehat{\boldsymbol{u}} = \widehat{\boldsymbol{f}}(\boldsymbol{\mu}), \tag{56}$$

where $\widehat{\boldsymbol{u}} = (u_1, \dots, u_{\mathcal{N}}) \in \mathbb{R}^{\mathcal{N}}$ is the vector of the degrees of freedom of the IGA-BEM solution,

$$\widehat{\boldsymbol{A}}(\boldsymbol{\mu}) = \begin{bmatrix} \sigma\boldsymbol{M} + \boldsymbol{N}(\boldsymbol{\mu}) \\ \boldsymbol{W} \end{bmatrix}, \qquad \widehat{\boldsymbol{f}}(\boldsymbol{\mu}) = \begin{Bmatrix} \boldsymbol{d}(\boldsymbol{\mu}) \\ 0 \end{Bmatrix} \tag{57}$$

and, for $i = 1, \dots, N_f$, $j = 1, \dots, \mathcal{N}$

$$\begin{aligned} M_{ij} &= m(\xi_j, \xi_i) = \int_0^1 \xi_i(s)\xi_j(s)\mathrm{d}s, \\ N_{ij}(\boldsymbol{\mu}) &= n(\xi_j, \xi_i; \boldsymbol{\mu}) = \int_0^1 \xi_i(s)(N\xi_j)(s; \boldsymbol{\mu})\mathrm{d}s, \end{aligned} \tag{58}$$

whereas, for $i = N_f + 1, \dots, \mathcal{N}$, $j = 1, \dots, \mathcal{N}$

$$W_{ij} = w(\xi_j, \xi_i) = \int_0^1 \xi_i(s)[\xi_j(s) - \xi_j(0) + \xi_j(s_f)]\mathrm{d}s \tag{59}$$

are the elements of the (rectangular) matrices $\boldsymbol{M}, \boldsymbol{N}(\boldsymbol{\mu})$ and $\boldsymbol{W}$, respectively. In the same way, for $i = 1, \dots, N_f$

$$d_i(\boldsymbol{\mu}) = d(w_h, \xi_i; \boldsymbol{\mu}) = \int_0^1 \xi_i(s) \left(D \sum_{j=1}^{\mathcal{N}} \xi_j(q) w_j \right)(s; \boldsymbol{\mu})\mathrm{d}s \tag{60}$$

are the components of the vector $\boldsymbol{d}(\boldsymbol{\mu})$. Here $\boldsymbol{M}$, $\boldsymbol{N}(\boldsymbol{\mu})$ and $\boldsymbol{W}$ represent the airfoil mass matrix, the Neumann matrix, and the wake condition matrix, respectively. The indicator function χ_A yields the splitting of matrix $\widehat{\boldsymbol{A}}$ in two (rectangular) submatrices.

As far as the (dynamic) Kutta condition (16) is concerned, we can express it as

$$J^{-1}(0; \boldsymbol{\mu})u'(0) + J^{-1}(s_f; \boldsymbol{\mu})u'(s_f) = -(\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) \cdot \boldsymbol{V}_\infty; \tag{61}$$

this condition is enforced as a constraint to the linear system (56), by setting

$$\boldsymbol{k}(\boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} J^{-1}(0; \boldsymbol{\mu})\xi_i'(0) + J^{-1}(s_f; \boldsymbol{\mu})\xi_i'(s_f). \tag{62}$$

Defining λ the corresponding Lagrange multiplier, system (56) can be written as

$$\begin{bmatrix} [\sigma\boldsymbol{M} + \boldsymbol{N}(\boldsymbol{\mu})] & \boldsymbol{k}^T(\boldsymbol{\mu}) \\ \boldsymbol{W} & \\ \boldsymbol{k}(\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{Bmatrix} \widehat{\boldsymbol{u}} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \widehat{\boldsymbol{f}}(\boldsymbol{\mu}) \\ -(\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) \cdot \boldsymbol{V}_\infty \end{Bmatrix}, \tag{63}$$

that is, in a more compact form, as

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}). \tag{64}$$

By solving a system like (64), we can obtain the high-fidelity approximation of boundary value problems in any prescribed, $\boldsymbol{\mu}$-dependent, exterior unbounded domain, requiring only the discretization of the boundary of the geometry. This makes the BEM discretization an ideal candidate for isogeometric analysis, since CAD geometries are usually provided as *surfaces* or *boundary representations*. At the same time, using a boundary integral discretization decreases dramatically the number of degrees of freedom of the system to be solved, although it leads to system matrices which are full, and whose terms require a careful treatment, since they arise from singular integrals (here we adopt the Telles algorithm [38], but other options are possible, see e.g. [39] or [40]). The fact that the system matrix is dense is usually considered a drawback of BEM; however, for isogeometric methods, it has a positive counterpart: high order and low order B-splines produce the same matrix bandwidth, making high order B-splines the very good candidates for isogeometric BEM approximations. By contrast, this is not the case in finite element methods, where high order B-splines would dramatically deteriorate the sparsity of the matrices. On the other hand, both IGA-BEM and FE-IGA suffer from the additional computational cost associated to evaluation of high order B-splines. When coupling IGA with RB methods, however, this additional cost is limited to the *offline* stage, offering the possibility to increase the accuracy of *online* computations at no additional cost.

3.2. *Affine parametric dependence and empirical interpolation*

Being able to express parametrized operators in a convenient form is crucial to speed up the evaluation of a reduced-order approximation. In our case, we can rewrite the matrix $\boldsymbol{A}(\boldsymbol{\mu})$ as

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{\mu}) &= \boldsymbol{A}^M + \boldsymbol{A}^w + \boldsymbol{A}^k(\boldsymbol{\mu}) + \boldsymbol{N}(\boldsymbol{\mu}) \\ &= \begin{bmatrix} \sigma\boldsymbol{M} & 0 \\ 0 & \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \boldsymbol{W} & \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \boldsymbol{k}^T(\boldsymbol{\mu}) \\ 0 & \\ \boldsymbol{k}(\boldsymbol{\mu}) & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{N}(\boldsymbol{\mu}) & 0 \\ 0 & \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{65}$$

where:

- $\boldsymbol{A}^M$ is the mass matrix (corrected with $\boldsymbol{\sigma}$ on the diagonal) with null elements in the rows and columns corresponding to the ones of the wake and Kutta condition;
- $\boldsymbol{A}^w$ is the wake matrix with null elements everywhere, except from the rows and columns corresponding to the wake;
- $\boldsymbol{A}^k(\boldsymbol{\mu})$ is the matrix with null elements everywhere except from the rows and columns corresponding to Kutta condition;
- $\boldsymbol{N}(\boldsymbol{\mu})$ is Neumann matrix with null elements in the rows and columns corresponding to the ones of the wake and Kutta condition.

A crucial assumption to ensure a very efficient offline/online splitting in the construction/evaluation of a RB method is the so-called *affine parametric dependence*. By this, we mean that we need to write system (64) in the form

$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})(\boldsymbol{N}^l + \boldsymbol{A}^{k,l}) + \boldsymbol{A}^M + \boldsymbol{A}^w\right)\boldsymbol{u}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \tag{66}$$

expressing the matrices $\boldsymbol{N}(\boldsymbol{\mu})$, $\boldsymbol{A}^k(\boldsymbol{\mu})$ and the vector $\boldsymbol{b}(\boldsymbol{\mu})$ as a linear combination of $\boldsymbol{\mu}$-dependent functions and $\boldsymbol{\mu}$-independent matrices (and vectors). We would obtain this expression naturally if the problem had an affine parametric dependence, but unfortunately this is not our case. The IGA parametrization entails a non affine parametric dependence, so that a further approximation stage is required in order to handle $\boldsymbol{\mu}$-dependent terms in a very efficient way. This procedure is carried out by the Empirical Interpolation Method (EIM), which features an Offline/Online decomposition and it has been largely applied in the context of RB methods to recover an approximated affine structure when dealing with non affine problems [41]. We remark that, in our framework, EIM itself enables to reduce the computational cost entailed by the solution of a high-fidelity BEM problem, resulting in a remarkable speed up of the assembling of BEM matrices, which is a very expensive stage of each BEM algorithm.

3.3. *Empirical interpolation method*

EIM is an interpolation technique for parameter-dependent functions based on adaptively chosen interpolation points and global shape functions [9]. Here we present two different ways to apply EIM to the problem at hand; further details about the general procedure are reported in Appendix C.

In the former case we apply EIM to get an affine approximation of the $\boldsymbol{\mu}$-dependent functions appearing in (57) and (60). In the latter case, we apply EIM directly to matrices and vectors related to the BEM parametrized problem. We denote the two options as *variational EIM* and *algebraic EIM*, respectively. This latter approach is indeed a simplified version of the so-called *matrix discrete EIM* technique described in [42–44] and recently analyzed in [45], which exploits the so-called discrete[2] EIM (DEIM).

3.3.1. Option 1: Variational EIM

Let us focus on the matrix $\boldsymbol{N}(\boldsymbol{\mu})$ and the right hand side $\boldsymbol{b}(\boldsymbol{\mu})$; the procedure is analogous for $\boldsymbol{A}^k(\boldsymbol{\mu})$. We can express $\boldsymbol{N}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ as follows:

$$\begin{aligned}
N_{ij}(\boldsymbol{\mu}) &= \begin{cases} \displaystyle\int_0^1 \xi_i(s)\,\Psi_j(s,\boldsymbol{\mu})\mathrm{d}s, & i=1,\dots,N_f,\ j=1,\dots,\mathcal{N} \\ 0, & i=N_f+1,\dots,\mathcal{N},\ j=1,\dots,\mathcal{N} \end{cases} \\
b_i(\boldsymbol{\mu}) &= \begin{cases} \displaystyle\int_0^1 \xi_i(s)\,\Theta(s,\boldsymbol{\mu})\mathrm{d}s, & i=1,\dots,N_f \\ 0, & i=N_f+1,\dots,\mathcal{N}, \end{cases}
\end{aligned} \tag{67}$$

where

$$\Psi_j(s;\boldsymbol{\mu}) = (N\xi_j)(s;\boldsymbol{\mu}), \qquad \Theta(s;\boldsymbol{\mu}) = \left(D\sum_{j=1}^{\mathcal{N}} \xi_j(q)h_j(\boldsymbol{\mu})\right)(s;\boldsymbol{\mu}). \tag{68}$$

In order to obtain an affine expansion of $\Psi_j(s,\boldsymbol{\mu})$, $j=1,\dots,\mathcal{N}$ and $\Theta(s,\boldsymbol{\mu})$, we apply the Empirical Interpolation Method to get an approximate expansion

$$\Psi_j(s;\boldsymbol{\mu}) \approx \sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\beta_j^l(s), \qquad \Theta(s;\boldsymbol{\mu}) \approx \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\beta^m(s), \tag{69}$$

respectively. Here $\{\gamma^l(\boldsymbol{\mu})\}_{l=1}^L, \{\gamma^m(\boldsymbol{\mu})\}_{m=1}^M : \mathcal{D} \to \mathbb{R}$ are two sets of L, respectively M $\boldsymbol{\mu}$-dependent functions, whereas $\beta_j^l, \beta^m : [0,1] \to \mathbb{R}$ are $\mathcal{N} \times L$, respectively M $\boldsymbol{\mu}$-independent functions. We point out that the IGA map depends on the basis functions exploited also in the high-fidelity discretization, so that each $\Psi_j(s;\boldsymbol{\mu})$, $j=1,\dots,\mathcal{N}$ has to be approximated as expressed in (69), entailing a large computational cost. Let us focus only on the non null elements of (66). We can write, for any $i=1,\dots,N_f$, $j=1,\dots,\mathcal{N}$

$$N_{ij}(\boldsymbol{\mu}) = \int_0^1 \xi_i(s)\,\Psi_j(s;\boldsymbol{\mu})\mathrm{d}s \approx \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})\int_0^1 \xi_i(s)\beta_j^l(s)\mathrm{d}s = \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})h_{ij}^l \tag{70}$$

where $h_{ij}^l = \int_0^1 \xi_i(s)\beta_j^l(s)\mathrm{d}s$. In this way, we can approximate the matrix $\boldsymbol{N}(\boldsymbol{\mu})$ by

$$\boldsymbol{N}(\boldsymbol{\mu}) \approx \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l, \qquad \boldsymbol{N}^l = \begin{bmatrix} h_{11}^l & \cdots & h_{1\mathcal{N}}^l \\ \vdots & \ddots & \vdots \\ h_{N_f1}^l & \cdots & h_{N_f\mathcal{N}}^l \\ 0 & \cdots & 0 \end{bmatrix}. \tag{71}$$

Here L is the number of EIM terms, $\mathcal{N}$ is the dimension of the problem at hand and N_f is the number of control values for the airfoil; L is selected so that each component of the matrix $\mathcal{N}(\boldsymbol{\mu})$ can be approximated up to a prescribed tolerance (see Appendix C). In the same way, for any $i=1,\dots,N_f$ we have

$$b_i(\boldsymbol{\mu}) = \int_0^1 \xi_i(s)\,\Theta(s;\boldsymbol{\mu})\mathrm{d}s \approx \sum_{m=1}^{M}\gamma^m(\boldsymbol{\mu})\int_0^1 \xi_i(s)\beta^m(s)\mathrm{d}s = \sum_{m=1}^{M}\gamma^m(\boldsymbol{\mu})h_i^m \tag{72}$$

[2] Both EIM and its discrete variant DEIM [46] start constructing a set of snapshots obtained by sampling the function to be approximated at different parameter values. Then, EIM computes a basis through a greedy algorithm, whereas DEIM performs a POD to extract a basis from this set of snapshots. Since for the case at hand we perform a greedy algorithm to select a suitable basis of matrices and vectors, we refer to our second option as *algebraic EIM* instead of DEIM—clearly, DEIM could in principle also be employed to perform this approximation, as shown e.g. in [45].

where $h_i^m = \int_0^1 \xi_i(s)\beta^m(s)\mathrm{d}s$. More compactly:

$$\boldsymbol{b}(\boldsymbol{\mu}) \approx \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \qquad \boldsymbol{b}^m = [h_1^m, \dots, h_{N_f}^m, 0]^T \tag{73}$$

and M is the number of EIM terms.

During the Offline stage, we compute the terms β^l and β^m, by assembling the structures $\boldsymbol{N}^l$ and $\boldsymbol{b}^m$ only once. During the Online stage, for any new value of $\boldsymbol{\mu}$, we solve two lower (and small) triangular systems (of dimension $L \times L$, $M \times M$, respectively) arising by imposing the interpolation constraints (see Appendix C), to obtain $\gamma^l(\boldsymbol{\mu})$ and $\gamma^m(\boldsymbol{\mu})$. Finally, we assemble the $\boldsymbol{\mu}$-dependent structures $\boldsymbol{N}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$, recovering the affine structure of the operators.

3.3.2. *Option 2: Algebraic EIM*

Since our goal is to provide an affine approximation of matrices and vectors, instead of simple functions, we have exploited the EIM algorithm in a new fashion, by applying it directly to matrices and vectors appearing in our BEM parametrized formulation, instead of on each of their components. As in the previous section, let us focus on the matrix $\boldsymbol{N}(\boldsymbol{\mu})$ and the right hand side $\boldsymbol{b}(\boldsymbol{\mu})$, for the sake of clarity. We can approximate these structures in the following form:

$$N_{ij}(\boldsymbol{\mu}) \approx \begin{cases} \displaystyle\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu}) h_{ij}^l, & i = 1, \dots, N_f,\ j = 1, \dots, \mathcal{N} \\ 0, & i = N_f + 1, \dots, \mathcal{N},\ j = 1, \dots, \mathcal{N} \end{cases}$$

$$b_i(\boldsymbol{\mu}) \approx \begin{cases} \displaystyle\sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) h_i^m, & i = 1, \dots, N_f \\ 0, & i = N_f + 1, \dots, \mathcal{N}, \end{cases} \tag{74}$$

that leads to the same formal structure of (71) and (73).

In this case, the algebraic EIM strategy returns directly the structures $\boldsymbol{N}^l$ and $\boldsymbol{b}^m$, instead of their components, by selecting (at each step) a couple (i, j) of indices; the set of selected indices form a *reduced mesh*, over which the operators are actually assembled. For the case at hand, the algebraic EIM entails less affine terms than the variational EIM (for the same prescribed tolerance), confirming the conclusions reported in [45]. We address the interested reader to [45] for further details related with this approach. Also in this case, once we have performed the EIM procedure, we can rewrite problem (64) in the affine form (66).

4. Reduced-order approximation

In this section we describe the main ingredients of the RB methods we employ to efficiently deal with the approximation of potential flows about parametrized airfoils. Starting from the parametrized formulation introduced in the previous sections, we address the construction of a reduced space and the projection stage required to compute the RB approximation. Finally, to make this picture more clear, we provide some details about the interplay among the techniques we have introduced and their Offline/Online decomposition.

4.1. *Reduced basis methods for parametrized PDEs*

The key idea of a RB method is to seek an approximate solution of (64), for any new value $\boldsymbol{\mu} \in \mathcal{D}$, as a linear combination of suitable, problem-dependent, basis functions, by means of a Galerkin projection onto a subspace $V_N \subset V_h$ of dimension $N \ll \mathcal{N}$:

$$V_N = \mathrm{span}\{u_h(\boldsymbol{\mu}^n),\ 1 \le n \le N\} = \mathrm{span}\{\zeta^n,\ 1 \le n \le N\}\,. \tag{75}$$

Each $u_h(\boldsymbol{\mu}^n)$ is computed by solving the IGA-BEM problem (64) for $\boldsymbol{\mu}^n$, $1 \le n \le N_{\max}$. Here $\{\zeta^n\}_{n=1}^N$ denotes (what shall ultimately be the *reduced basis*) an orthonormal basis of V_N. We take advantage of two popular methods for selecting the basis functions (and then build our ROMs): the *Proper Orthogonal Decomposition* (POD) and the *greedy-RB* methods, detailed in Section 4.2.

By expressing the RB solution as $u_N(\boldsymbol{\mu}) = \sum_{m=1}^N u_N^m(\boldsymbol{\mu})\zeta^m$, a Galerkin projection onto V_N yields the following RB problem:

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}) \quad \forall v_N \in V_N. \tag{76}$$

Equivalently, (76) can be written as

$$\sum_{m=1}^{N} a(\zeta^m, \zeta^n; \boldsymbol{\mu}) u_N^m(\boldsymbol{\mu}) = f(\zeta^n; \boldsymbol{\mu}), \quad n = 1, \dots, N. \tag{77}$$

From an algebraic standpoint, this latter equation expresses nothing but a linear system whose solution provides the RB approximation to problem (44) or (64), and it can be written in a more compact form as

$$\boldsymbol{A}_N(\boldsymbol{\mu})\boldsymbol{u}_N(\boldsymbol{\mu}) = \boldsymbol{b}_N(\boldsymbol{\mu}), \tag{78}$$

where $(\boldsymbol{A}_N(\boldsymbol{\mu}))_{nm} = a(\zeta^m, \zeta^n; \boldsymbol{\mu})$ and $(\boldsymbol{b}_N(\boldsymbol{\mu}))_n = f(\zeta^n; \boldsymbol{\mu})$ are the RB structures. We underline that (78) has the same form of the high-fidelity problem (64), but much smaller dimension, and that each basis function $\zeta^n \in V_h$ is given by

$$\zeta^n = \sum_{i=1}^{\mathcal{N}} \zeta_i^n \xi_i, \quad n = 1, \dots, N, \tag{79}$$

that is, it can be expressed as a linear combination of the BEM basis functions $\{\xi_i\}_{i=1}^{\mathcal{N}}$. By inserting (79) in (77), and by denoting

$$\mathcal{Z} = [\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_N] \in \mathbb{R}^{\mathcal{N} \times N} \tag{80}$$

the *transformation matrix* formed by the BEM degrees of freedom related to each RB basis function, we have that

$$\boldsymbol{A}_N(\boldsymbol{\mu}) = \mathcal{Z}^T \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}, \qquad \boldsymbol{b}_N(\boldsymbol{\mu}) = \mathcal{Z}^T \boldsymbol{b}(\boldsymbol{\mu}), \tag{81}$$

where $\boldsymbol{A}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ are the structures of (64). Then, we can recover the full solution simply as $\boldsymbol{u}(\boldsymbol{\mu}) = \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu})$, as well as other outputs of interest, such as the velocity and pressure fields all over the domain, the pressure coefficient and the lift coefficient (see Appendix B). Moreover, we can characterize the error

$$\boldsymbol{e}_N(\boldsymbol{\mu}) = \boldsymbol{u}(\boldsymbol{\mu}) - \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}) \tag{82}$$

between RB and high-fidelity solutions in terms of the high-fidelity residual:

$$\boldsymbol{r}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}). \tag{83}$$

In fact, the following relation holds:

$$\|\mathbf{e}_N(\boldsymbol{\mu})\|_2 \leq \|\boldsymbol{A}^{-1}(\boldsymbol{\mu})\|_2 \, \|\mathbf{r}(\boldsymbol{\mu})\|_2 = \frac{1}{\sigma_{\min}(\boldsymbol{A}(\boldsymbol{\mu}))} \|\mathbf{r}(\boldsymbol{\mu})\|_2,$$

where $\sigma_{\min}(\boldsymbol{A}(\boldsymbol{\mu})) = \sqrt{\lambda_{\min}(\boldsymbol{A}(\boldsymbol{\mu})^T\boldsymbol{A}(\boldsymbol{\mu}))}$ is the smallest singular value of $\boldsymbol{A}(\boldsymbol{\mu})$. The efficient evaluation of error bounds is beyond the scope of this work; the interested reader can refer, e.g., to [10], where he can find similar results related to error bounds for the RB velocity and pressure fields (with respect to a high-fidelity FE approximation). In Section 4.2.2 we will take advantage of a residual-based *a posteriori* error *indicator* to perform the greedy algorithm in a very efficient way.

We point out that although system (78) is nominally of small size N, its assembling would still involve entities associated with our $\mathcal{N}$-dimensional BEM approximation space without the assumption of affine parameter dependence. Yet, a very efficient Offline–Online procedure can be devised by relying on the EIM version of the algebraic structures, that is, by expressing system (78) as

$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu}) \boldsymbol{A}_N^l \right) \boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) \boldsymbol{b}_N^m, \tag{84}$$

where (similarly to (81))

$$\boldsymbol{A}_N^l = \mathcal{Z}^T \boldsymbol{A}^l \mathcal{Z}, \qquad \boldsymbol{b}_N^m = \mathcal{Z}^T \boldsymbol{b}^m. \tag{85}$$

In this way, computation entails an expensive $\boldsymbol{\mu}$-independent Offline stage performed only once, and an Online stage to be run for any chosen parameter value $\boldsymbol{\mu} \in \mathcal{D}$. See e.g. [47] for further details.

4.2. *Reduced basis space construction*

We can rely on either POD or greedy algorithms to construct the RB space – that is, from an algebraic standpoint, to assemble the transformation matrix $\mathcal{Z}$ – in the context of parametrized PDEs.

4.2.1. *Proper orthogonal decomposition*

The Proper Orthogonal Decomposition (POD) technique reduces the dimensionality of a system by transforming the original variables into a new set of orthonormal basis functions (called POD modes), the first few modes ideally retaining most of the *energy* present in all of the original variables, by relying on the use of the singular value decomposition (SVD) algorithm [47]. First introduced in the context of turbulent flows [48], POD has gained a strong importance in time-dependent flow problems and related optimal control application, see e.g. [49,50] just to mention a few, and [51] for applications closer to hydrodynamics.

Let us consider a discrete set of n_{train} snapshot vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_{n_{\text{train}}}\} \in V_h$, and form the snapshot matrix $\mathcal{U} \in \mathbb{R}^{\mathcal{N}\times n_{\text{train}}}$ having them as column vectors:

$$\mathcal{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_{n_{\text{train}}}] \equiv [\boldsymbol{u}(\boldsymbol{\mu}_1), \dots, \boldsymbol{u}(\boldsymbol{\mu}_{n_{\text{train}}})]. \tag{86}$$

Then, let us compute the SVD of $\mathcal{U}$,

$$\mathcal{V}^T \mathcal{U} \mathcal{W} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}, \tag{87}$$

where $\mathcal{W} = [\zeta_1, \zeta_2, \dots, \zeta_{\mathcal{N}}] \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ and $\mathcal{V} = [\Psi_1, \Psi_2, \dots, \Psi_N] \in \mathbb{R}^{n_{\text{train}}\times n_{\text{train}}}$ are orthogonal matrices, whereas $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_r)$ with $\sigma_1 \ge \sigma_2 \ge \dots \ge \sigma_r$; here $r \le n_{\text{train}}$ is the rank of $\mathcal{U}$. For any $N \le n_{\text{train}}$, the POD basis of dimension N is defined as the set of the first N left singular vectors $[\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_N]$ of $\mathcal{U}$, that is,

$$\mathcal{Z} = [\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_N] \in \mathbb{R}^{\mathcal{N}\times N}. \tag{88}$$

By construction, the POD basis is orthonormal. The energy contained in the first N modes is $E(\mathcal{Z}) = \sum_{i=1}^{N} \sigma_i^2$, so that the error in the POD basis is equal to the square of the singular values corresponding to the neglected POD modes. To stop the basis construction, we can select N so that $E(\mathcal{Z}) \le \varepsilon_{POD}^*$, for a prescribed tolerance ε_{POD}^*: it is sufficient to choose N such that

$$\frac{\displaystyle\sum_{i=1}^{N} \sigma_i^2}{\displaystyle\sum_{i=1}^{\mathcal{N}} \sigma_i^2} \ge 1 - \varepsilon_{POD}^*, \tag{89}$$

i.e., we require that the energy retained by the last $\mathcal{N} - N$ modes is equal to ε_{POD}^*.

4.2.2. *Greedy algorithm*

Although being a very general approach, POD entails the evaluation of a huge training set of high-fidelity solutions, which can be, in principle, a very difficult task. The greedy algorithm provides an adaptive technique to overcome this drawback. This technique is a general procedure to approximate each element of a given compact set K (in our case $K = \{u_h(\boldsymbol{\mu}),\ \boldsymbol{\mu} \in \mathcal{D}\}$), by a subspace of properly selected elements of K. A first version of the greedy algorithm to sample the parameter domain based on *a posteriori* error estimation procedures has been presented in [52] for linear affine elliptic and simple nonlinear problems. These features have been extended to a wide class of problems, and nowadays encompass Stokes and Navier–Stokes equations, among others (see e.g. [53,13]).

Let us denote by $\Xi_{\text{train}} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{n_{\text{train}}}\}$ a large train sample of parameter values in $\mathcal{D}$ which will serve to select the RB space, and by ε^*_{RB} a prescribed tolerance. At the nth iteration, the greedy algorithm adds to the retained snapshots that particular candidate snapshot – over all candidate snapshots $u(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \Xi_{\text{train}}$ – which is worst approximated by the already computed basis, that is:

$$\boldsymbol{\mu}^n = \arg \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{n-1}(\boldsymbol{\mu}), \quad n = 2, \dots, N. \tag{90}$$

Here $\Delta_n(\boldsymbol{\mu})$ is a reliable, cheap residual-based error *indicator*, given by

$$\Delta_N(\boldsymbol{\mu}) = \frac{\|\boldsymbol{r}(\boldsymbol{\mu})\|_2^2}{\|\boldsymbol{b}(\boldsymbol{\mu})\|_2^2} = \frac{\|\boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}\boldsymbol{u}_n(\boldsymbol{\mu})\|_2^2}{\|\boldsymbol{b}(\boldsymbol{\mu})\|_2^2}, \tag{91}$$

where $\boldsymbol{u}_n(\boldsymbol{\mu}) \in \mathbb{R}^n$ is the algebraic counterpart of the RB approximation related to the space V_n, $n = 1, \dots, N$. Thanks to the affine expansion

$$\boldsymbol{A}(\boldsymbol{\mu}) = \sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{A}^l, \qquad \boldsymbol{b}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \tag{92}$$

we can express the residual as follows:

$$\begin{aligned}\|\boldsymbol{r}(\boldsymbol{\mu})\|_2^2 = {} & \sum_{m=1}^{M}\sum_{m'=1}^{M} \gamma^m(\boldsymbol{\mu})\gamma^{m'}(\boldsymbol{\mu})\boldsymbol{b}^{m'^T}\boldsymbol{b}^m - 2\sum_{m=1}^{M}\sum_{l=1}^{L}\gamma^m(\boldsymbol{\mu})\gamma^l(\boldsymbol{\mu})\left(\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{b}^m\right)\boldsymbol{u}_N(\boldsymbol{\mu}) \\ & + \sum_{l=1}^{L}\sum_{l'=1}^{L}\gamma^l(\boldsymbol{\mu})\gamma^{l'}(\boldsymbol{\mu})\boldsymbol{u}_N^T(\boldsymbol{\mu})\left(\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{A}^{l'}\mathcal{Z}\right)\boldsymbol{u}_N(\boldsymbol{\mu}).\end{aligned} \tag{93}$$

The evaluation of the $\boldsymbol{\mu}$-independent terms $\boldsymbol{b}^{m'^T}\boldsymbol{b}^m \in \mathbb{R}$, $\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{b}^m \in \mathbb{R}^N$, $\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{A}^{l'}\mathcal{Z} \in \mathbb{R}^{N\times N}$ is independent of $\mathcal{N}$ but depends on the number of affine terms L, M, which for the problem at hand is not small. Such a decomposition is mandatory to efficiently evaluate the error indicator (91) and then to select the snapshots according to (90), although it greatly affects the Offline time required by each step of the greedy algorithm.

4.3. Computational strategy

To conclude this section, we provide some details about the computational stages required to set up a reduced-order model taking advantage of the techniques we have described so far. The cornerstone of the methodology is the chance to separate the computational effort between a first expensive Offline phase, to be run once for all, and a very inexpensive Online phase, whose evaluation provides the solution of the problem for any new ($\boldsymbol{\mu}$-dependent configuration). In particular:

i. during the Offline phase, the IGA-BEM high-fidelity approximation has to be set, including the parametrization of the problem. Once all the structures corresponding to the IGA-BEM problem (64) have been assembled, the EIM algorithm has to be run in order to recover affinely parametrized structures appearing in (66). Then we can construct the reduced basis space – that is, the transformation matrix $\mathcal{Z}$ – by relying on either the POD approach, or the greedy algorithm. In this latter case we take advantage of the decomposition (93) for the evaluation of the dual norms of the residuals. At the same stage, the reduced-order structures (85) are assembled and stored;

ii. during the Online phase, for any query corresponding to a new $\boldsymbol{\mu}$ value, the $N \times N$ RB problem (84) has to be assembled starting from the reduced-order structures already stored. The weights $\{\gamma^l(\boldsymbol{\mu})\}_{l=1}^L$ (resp. $\{\gamma^m(\boldsymbol{\mu})\}_{m=1}^M$) appearing in (84) have to be evaluated by solving a $L \times L$ (resp. $M \times M$) linear system, corresponding to the Online interpolation problem related to the EIM procedure. Once the RB problem (84) has been solved, we can recover also velocity and pressure fields, and other outputs of interest.

The interplay among the different components of the algorithm presented in the paper is sketched in the diagram reported in Fig. 6.

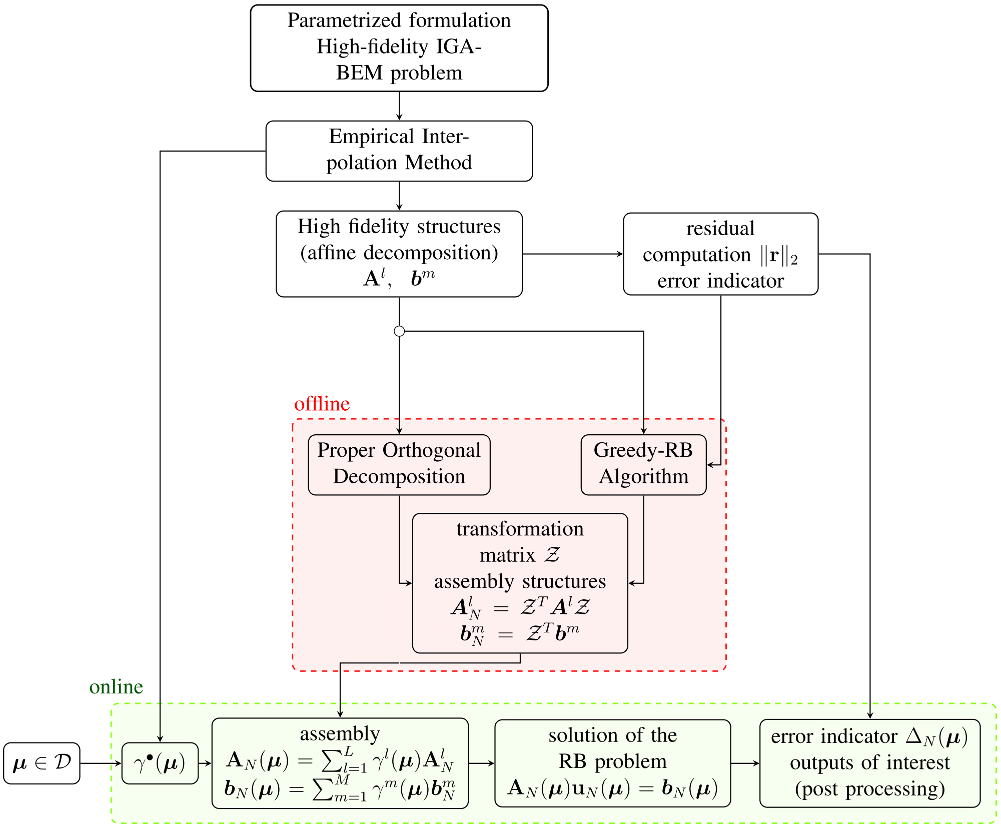
Parametrized formulation High-fidelity IGA-BEM problem
Empirical Interpolation Method
High fidelity structures (affine decomposition) $\mathbf{A}^l, \quad \boldsymbol{b}^m$
residual computation $\|\mathbf{r}\|_2$ error indicator
offline
Proper Orthogonal Decomposition
Greedy-RB Algorithm
transformation matrix $\mathcal{Z}$ assembly structures
$\boldsymbol{A}_N^l = \mathcal{Z}^T \boldsymbol{A}^l \mathcal{Z}$
$\boldsymbol{b}_N^m = \mathcal{Z}^T \boldsymbol{b}^m$
online
$\boldsymbol{\mu} \in \mathcal{D}$
$\gamma^\bullet(\boldsymbol{\mu})$
assembly
$\mathbf{A}_N(\boldsymbol{\mu}) = \sum_{l=1}^L \gamma^l(\boldsymbol{\mu})\mathbf{A}_N^l$
$\boldsymbol{b}_N(\boldsymbol{\mu}) = \sum_{m=1}^M \gamma^m(\boldsymbol{\mu})\boldsymbol{b}_N^m$
solution of the RB problem
$\mathbf{A}_N(\boldsymbol{\mu})\mathbf{u}_N(\boldsymbol{\mu}) = \boldsymbol{b}_N(\boldsymbol{\mu})$
error indicator $\Delta_N(\boldsymbol{\mu})$ outputs of interest (post processing)

Fig. 6. Scheme of the RB-IGA coupling algorithm. In the outlined box we collected all operations that can be performed *offline*, while the lower outlined box contains the *online* operations.

5. Numerical results and validation with experimental data

In order to validate the reliability of our framework in the characterization of potential flows about airfoils, we test our algorithms for two different NACA 4-digits profiles (NACA 0012 and NACA 4412) for which several experimental results are available in the literature [18,54,55]. We validate our solver by comparing with the low order solver Xfoil [24], which is a simple linear-vorticity stream function panel method, and with a B-spline based collocation solver [24]. We first provide results obtained through the IGA-BEM method (Section 5.1). In Section 5.2, we show the results related to the EIM approximation of non affine structures. Finally, in Section 5.3, we show the results obtained with the ROMs we have implemented, by comparing their computational performances.

5.1. IGA-BEM validation

In the following we present a validation of the IGA-BEM high-fidelity method, exploiting the results shown in [24]. The method presented in that work uses the same basis functions we use here, with two notable differences: they adopt a collocation method and they do not introduce the wake. By contrast, in our work we use a Galerkin approach and we introduce the wake, in view of the extension of this work to three dimensional problems. The convergence analysis presented in [24] for third order B-splines, shows that using a number of degrees of freedom ≥ 70 to describe the high-fidelity approximation is enough to reach convergence of the results. In what follows, we used $\mathcal{N} = 77$ basis functions. For an in-depth comparison between our framework and [24], see [17]. To make a fare comparison with the low order panel method, we have set Xfoil (in its inviscid version) to use $\mathcal{N} = 200$ degrees of freedom, which was shown in [24] to be necessary in order to obtain the same level of accuracy as a high order IGA-BEM method with about sixty basis functions. In Table 1 we summarize some numerical details related to our IGA-BEM implementation.

Table 1
Choice of some relevant numerical parameters for IGA-BEM algorithm.

B-spline order	3
Degrees of freedom	76 + 1 (Kutta condition)
Parametrization of the geometry	arc length
External integral quadrature formula	Standard Gauss
Number of points (external integral)	4
Internal integral quadrature formula	Gauss Telles
Number of points (internal integral)	10

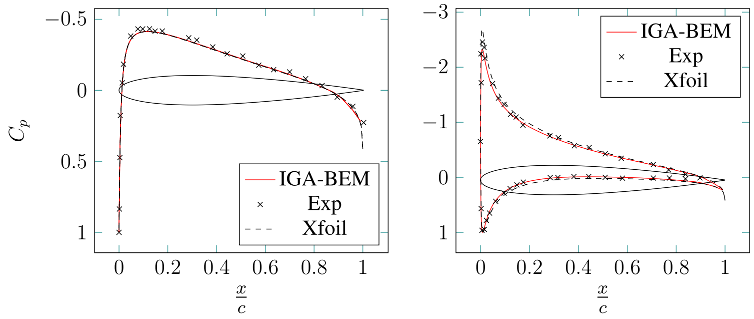

Fig. 7. Pressure coefficient for NACA 0012: IGA-BEM, experimental data [54] and Xfoil for different angles of attack: $\alpha = 0°$ (left), $6°$ (right).

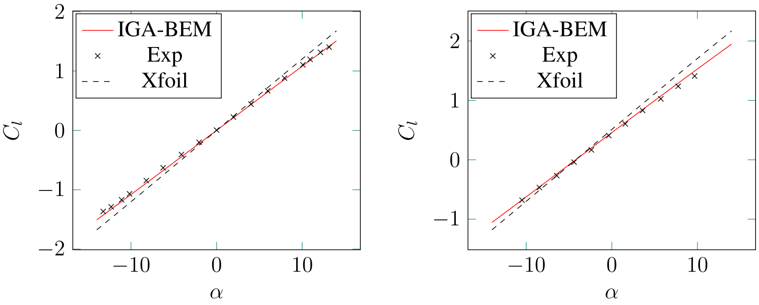

Fig. 8. $C_L(\alpha)$ curves (IGA-BEM, experimental data [18] and Xfoil) for two different NACA airfoils: NACA 0012 (left), NACA 4412 (right).

5.1.1. NACA 0012 profile

As a first test, we consider the symmetric NACA 0012 airfoil. In Fig. 7, we show the comparison of the pressure coefficient among our IGA-BEM, experimental data and Xfoil calculations for different angles of attack. We can observe that the obtained results are very close both to experimental data and Xfoil calculations. In Fig. 8, we show the lift coefficient curve and we compare it with experimental data and Xfoil calculations.

5.1.2. NACA 4412 profile

As a second test, we consider the NACA 4412 profile. This choice is driven by the fact that a numerical comparison with a similar B-spline method for this profile is available in [24]. First, in Fig. 9, we show the comparison of the pressure coefficient among IGA-BEM, experimental data and Xfoil calculations for different angles of attack. Also for the NACA 4412 profile, the results obtained are very close to experimental data and Xfoil calculations. In Fig. 8, we show the lift coefficient curve and we compare it with experimental data and Xfoil calculations.

We have also tested our IGA-BEM implementation against the B-splines based method presented in [24]. In Fig. 9, we provide this test in terms of pressure coefficient. We note that there is a discrepancy in the results: this difference can be remarked also in Fig. 8, in terms of gap between Xfoil and IGA-BEM lift coefficients. Thus, if we compare the

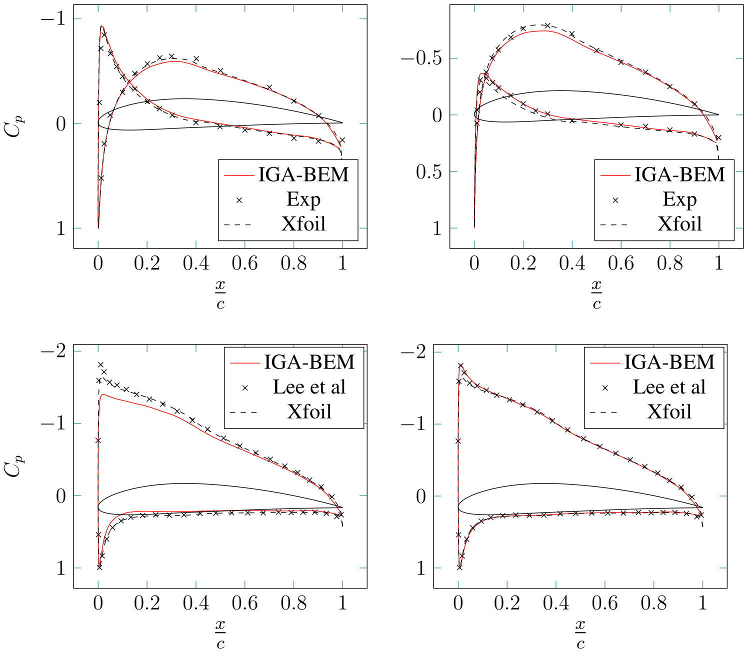

Fig. 9. Pressure coefficient for NACA 4412: IGA-BEM, experimental data from [55] and Xfoil (top); $\alpha = -2°$ (left), 0° (right); IGA-BEM, B-splines based method from [24] and Xfoil (bottom); $\alpha = 5°$ (left), $C_l = 1.11$ (right).

pressure coefficient not for a given angle of attack α but for a given lift coefficient C_l, we recover a great accordance for the pressure coefficient curves, as depicted in Fig. 9. The results provided so far show the reliability of our IGA-BEM framework.

5.2. Empirical interpolation method

In this section we show how the parameter dependent terms appearing in (73) can be approximated by a series of affine terms, through the EIM procedure of Section 3.3 and we analyze the impact of the EIM matrix/vector approximation on the IGA-BEM high-fidelity solution, by a stability estimation.

First of all, we show how the greedy algorithm of EIM selects the values of the parameter $\boldsymbol{\mu}$ and how the error decreases according to the number of affine terms taken, both for the matrix $\boldsymbol{N}(\boldsymbol{\mu})$ and the vector $\boldsymbol{b}(\boldsymbol{\mu})$. To initialize the EIM procedure, we prescribe a tolerance $\varepsilon^*_{EIM} = 10^{-8}$ and a random train of 600 parameter values. We chose this tolerance so that the error contribution Δ^{EIM} due to EIM approximation can be neglected when accounting for the error between RB model and the high-fidelity approximation. In other words, the EIM error Δ^{EIM} is orders of magnitude smaller than the predicted error by the indicator $\Delta_N(\boldsymbol{\mu})$. By recalling the stability estimate for linear systems in the form (64), we can consider the linear system corresponding to the EIM assembling of the operators as the perturbed system

$$(\boldsymbol{A} + \delta\boldsymbol{A})\,(\boldsymbol{u} + \delta\boldsymbol{u}) = \boldsymbol{b} + \delta\boldsymbol{b}, \tag{94}$$

where the errors due to EIM approximation are given by $\delta\boldsymbol{A}$ and $\delta\boldsymbol{b}$ for the matrix and the right hand side, respectively. Then, the following relation holds (see e.g. [56]):

$$\frac{\|\delta\boldsymbol{u}\|}{\|\boldsymbol{u}\|} = \frac{K(\boldsymbol{A})}{1 - K(\boldsymbol{A})\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\|}\left(\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} + \frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}\right), \tag{95}$$

where $K(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$ is the condition number of $\boldsymbol{A}$ and provided that $K(\boldsymbol{A})\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\| < 1$. For the case at hand, we have $K(\boldsymbol{A}) = O(10^3)$, $\|\delta\boldsymbol{b}\|/\|\boldsymbol{b}\| = O(10^{-6})$ and $\|\delta\boldsymbol{A}\|/\|\boldsymbol{A}\| = O(10^{-6})$, so that

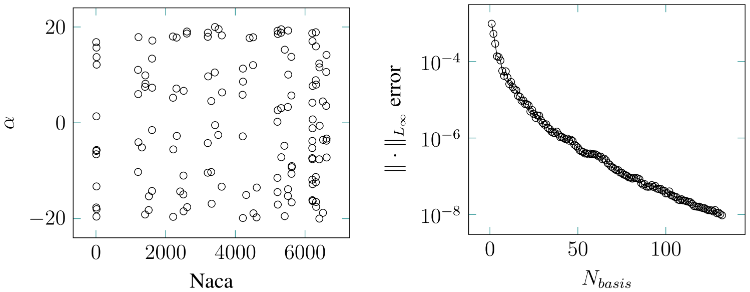

Fig. 10. EIM approximation of $\boldsymbol{N}(\boldsymbol{\mu})$: sample values in the parameter space (left) and convergence of the greedy-EIM algorithm (right); $\varepsilon^*_{EIM} = 10^{-8}$.

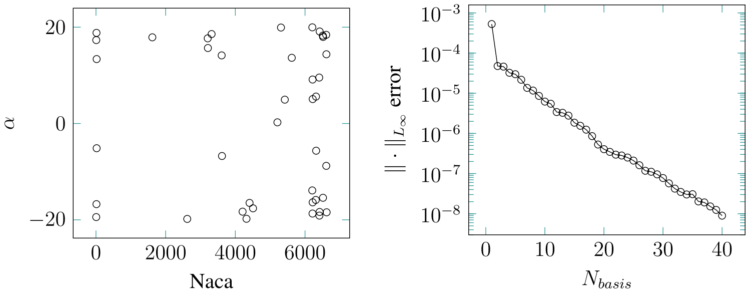

Fig. 11. EIM approximation of $\boldsymbol{b}(\boldsymbol{\mu})$: sample values in the parameter space (left) and convergence of the greedy-EIM algorithm (right); $\varepsilon^*_{EIM} = 10^{-8}$.

Table 2
Some features of the empirical interpolation method.

EIM samples number	600
EIM tolerance ε^*_{EIM}	10^{-8}
Number of elements of $\boldsymbol{N}$	5776
Affine matrix components L	132
Number of elements of $\boldsymbol{b}$	76
Affine r.h.s. components M	40
Time for structures assembling (IGA-BEM)	0.9 s
Time for structures assembling (EI-IGA-BEM)	0.6 s $\begin{cases} 0.1 \text{ s for } \boldsymbol{N} \\ 0.5 \text{ s for } \boldsymbol{b} \end{cases}$

$$\frac{\|\boldsymbol{u}(\boldsymbol{\mu}) - \boldsymbol{u}^{EIM}(\boldsymbol{\mu})\|}{\|\boldsymbol{u}(\boldsymbol{\mu})\|} \leq \Delta^{EIM} = O(10^{-3}) \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{96}$$

where $\boldsymbol{u}^{EIM}(\boldsymbol{\mu})$ is the solution to problem (64) obtained by considering the EIM approximation of the operators.

In Fig. 10 we show the parameter values selected during the EIM procedure and the error (in L_∞-norm) related to the approximation of matrix $\boldsymbol{N}(\boldsymbol{\mu})$, respectively; in Fig. 11 we report the same quantities regarding the construction of right hand side $\boldsymbol{b}(\boldsymbol{\mu})$. We note that the EIM algorithm selects only 132 points over $76 \times 76 = 5776$ elements of matrix $\boldsymbol{N}(\boldsymbol{\mu})$ for its approximation, to reach a precision of 10^{-8}. This leads to a great computational saving concerning the matrix assembling. On the contrary, we note that the greedy algorithm of EIM selects 40 points over 76 elements of $\boldsymbol{b}(\boldsymbol{\mu})$ for its approximation to reach the same precision, entailing a much less tangible saving in the case of the right hand side.

In Table 2 we summarize some numerical details related to the EIM procedure. As already mentioned, EIM reduces the computational cost associated with the assembling of BEM matrices and vectors, which is normally very expensive. By imposing a strict tolerance, EIM selects many affine terms, which require more time to be evaluated.

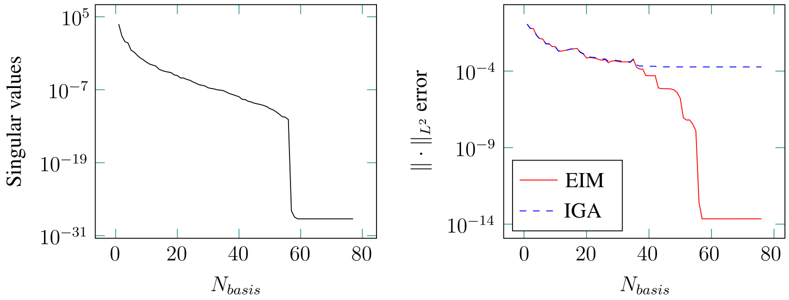

Fig. 12. Eigenvalues of the Singular Value Decomposition (left) and error convergence between POD and EIM-IGA-BEM and POD and IGA-BEM solutions for a random train of 50 parameter values (right).

However, we highlight that the first aim of EIM is not to reduce the computational cost, but rather to approximate the non affine terms through an affine expansion, in view of developing efficient ROMs.

5.3. Reduced order models

We now present the numerical results obtained through the ROMs implemented in this work (POD and greedy-RB algorithm). We show some details about the offline construction of the reduced basis spaces, a convergence test for the error, and some online evaluations, compared with the results of IGA-BEM and experimental data. In the end, we provide a direct comparison between POD and greedy-RB algorithm in terms of computational performances.

5.3.1. Proper orthogonal decomposition

For the problem at hand, we start from a set of 600 snapshots obtained by solving the IGA-BEM problem on a random sample Ξ_{train}. We apply the SVD algorithm obtaining the singular values depicted (in decreasing order) in Fig. 12. Thanks to (89), the energy information contained in the returned POD modes is proportional to these singular values.

As a first test, we assess the quality of the POD solution, by evaluating, on a set of 50 random values of the parameters, the error between the POD and the high-fidelity solution. Concerning this latter, we consider both the original IGA-BEM formulation and the one obtained by considering the EIM approximation of the operators (EIM-IGA-BEM solutions). We recall that the POD solution exploits the EIM version of the operators because of the Offline/Online decomposition. The test sample is the same also for the RB convergence tests, shown later on. For all possible choices of the reduced space dimension N, we compute the error (in L^2-norm) for all IGA-BEM and EIM-IGA-BEM solutions in the sample. In figure 12 we show the mean errors between POD and EIM-IGA-BEM solutions, and between POD and IGA-BEM solutions. As expected, the approximation error has the same behavior of the singular values of the snapshot matrix.

From this comparison, we can also see where the error between IGA-BEM and EIM-IGA-BEM solutions, namely Δ^{EIM}, affects the reduction error, that is, the error between POD and IGA-BEM solutions. POD modes are computed by considering the SVD of a snapshot matrix, each snapshot being computed by considering the EIM approximation of the operators. If we take less than 30 basis functions, the errors have the same trend. If we take more than 30 basis functions, the error between POD and IGA-BEM remain constant equal to Δ^{EIM}, that is, the error due to the approximation of the operators is the dominant contribution. By considering a tolerance of $O(10^{-4})$ in the selection of the POD modes (see (89)), we retain 9 basis functions in our reduced space so that we are able to neglect the EIM contribution to the approximation error Δ^{EIM}. We also provide a comparison between the pressure coefficient computed through POD algorithm, IGA-BEM and experimental data, first for the NACA 0012 airfoil (Fig. 13), then for the NACA 4412 airfoil (Fig. 14). In Table 3 we summarize some computational details related to POD reduced order model, as well its performance in terms of CPU time and dimension of the reduced system.

5.3.2. Greedy-RB algorithm

We now consider the numerical results obtained through the greedy-RB algorithm. We take a train of 600 (random) parameter values and we apply the greedy algorithm by prescribing a tolerance $\varepsilon_{RB}^{*} = 3 \cdot 10^{-4}$ on the stopping criterion. For the case at hand, greedy-RB algorithm selects 18 basis functions. In Fig. 15 we show the convergence

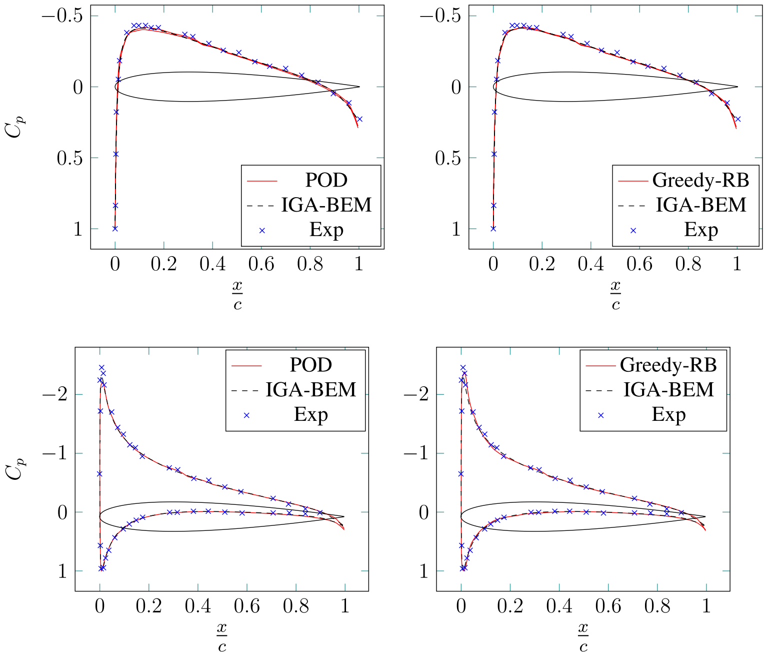

Fig. 13. Pressure coefficient for NACA 0012: IGA-BEM, POD and experimental data (from [18]) (left column), IGA-BEM, Greedy-RB and experimental data (from [18]) (right column); $\alpha = 0°, 6°$.

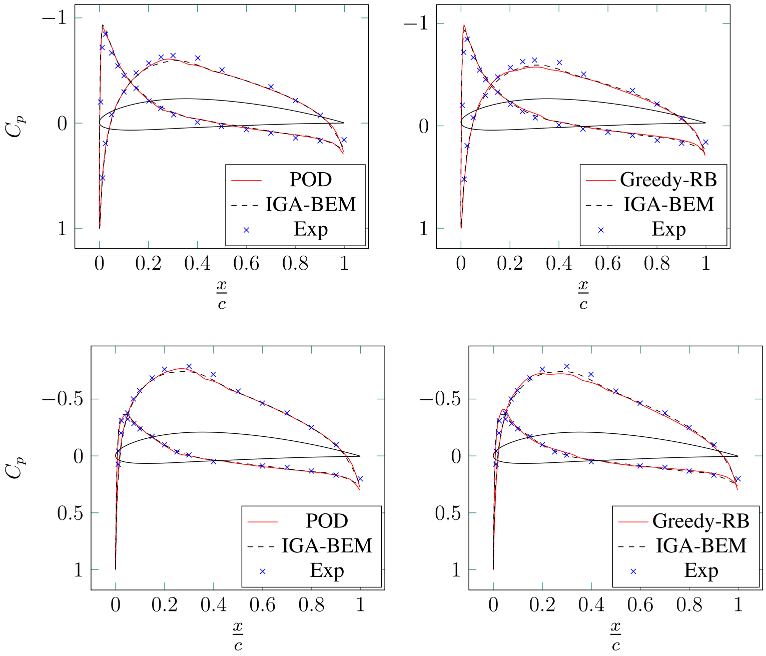

Fig. 14. Pressure coefficient for NACA 4412: IGA-BEM, POD and experimental data (from [18]) (left column), IGA-BEM, Greedy-RB and experimental data (from [18]) (right column); $\alpha = -2°, 0°$.

of the greedy-RB algorithm in terms of the error indicator $\max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_N(\boldsymbol{\mu})$, $N = 1, 2 \ldots$ until convergence and the retained snapshots. We can also observe that the greedy algorithm selects a few snapshots in any subregion of the parameter space, thus yielding the possibility to represent all configurations as combinations of the basis functions. Only two snapshots are selected for ‘large’ angles of attack whereas several snapshots correspond to the

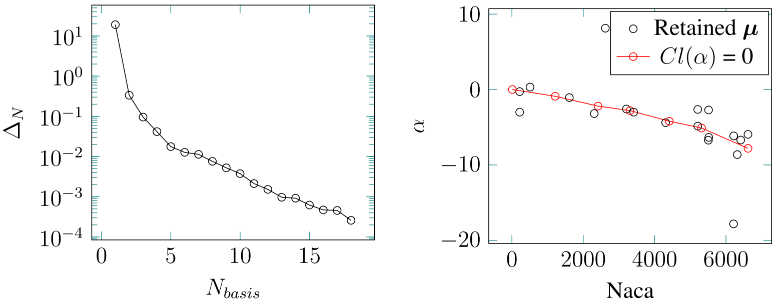

Fig. 15. Greedy algorithm: convergence of the error estimator Δ_N (left), retained snapshots in the parameter space (right); $\varepsilon^*_{RB} = 3 \times 10^{-4}$.

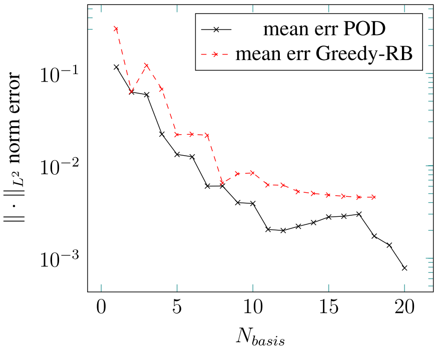

Fig. 16. Error convergence comparison between POD and greedy-RB for a random train of parameters.

zero lift coefficient angle of attack (red line in Fig. 15) and different NACA profiles. This means that, between the two parameters of interest, the reduction is harder to perform for the shape than for the angle of attack. This is not surprising, since the IGA parametrization entails much more involved operations to describe the shape of a wide family of airfoils, than their rotation.

As a first test, we take the same set of 50 IGA-BEM solutions already exploited for the POD case and we perform a convergence test in the same way as in the POD case. In Fig. 16 we show the mean error between Greedy-RB and IGA-BEM solutions. We provide a comparison between the pressure coefficient computed through Greedy-RB algorithm, IGA-BEM and experimental data, first for NACA 0012 airfoil (Fig. 13), then for NACA 4412 airfoil (Fig. 14). In Table 3 we summarize some details related to the greedy-RB algorithm, and its computational performance.

5.3.3. Comparison of POD and greedy-RB computational performances

We now compare the performance of the two ROMs we have considered (POD and greedy-RB) and the high-fidelity IGA-BEM approximation. Since the dimension of the high-fidelity problem at hand is quite small, for the sake of comparison we provide also some details related to a second high-fidelity solution, obtained by considering a larger amount of degrees of freedom.

In Fig. 16, we compare the online errors (evaluated on a test sample set Ξ_{test} of 50 parameters point, not included in the training sample Ξ_{train}) convergence for POD and the greedy-RB methods, whereas in Table 3 we summarize some numerical details of our ROMs. Convergence rate is similar in both cases, showing an exponential decay with respect to the dimension N. Both the methods give great results in terms of accuracy with respect to the high-fidelity IGA-BEM method. Not only, both ROMs yield a computational saving of about 4 times for each input/output evaluation. We only point out a strong difference between the CPU times required for the Offline stage by the two ROMs. This aspect is related mainly to the following reasons:

i. on the one hand, IGA-BEM for two dimensional problems is a very efficient and rapid tool, i.e., it is not too expensive to evaluate 600 snapshots to initialize the POD algorithm. For more complex problems, this evaluation could be very expensive, or even impossible;
ii. on the other hand, the number of affine terms increases dramatically the computational cost in the Offline stage of the greedy-RB method. In fact, an efficient evaluation of the norm of the residuals according to (93) is very

Table 3
Performance comparison among IGA-BEM, POD and RB.

Approximation data	IGA-BEM 1	IGA-BEM 2
Number of parameters	2	2
Affine matrix components L	132	120
Affine r.h.s. components M	40	41
Sample train (both POD and greedy-RB)	600	600
IGA-BEM space dimension $\mathcal{N}$	77	147
Greedy-RB space dimension N	9	9
POD space dimension N	9	9
Greedy-RB tolerance	3×10^{-4}	3×10^{-4}
POD tolerance	2×10^{-4}	2×10^{-4}
IGA-BEM evaluation time	3.6 s	8 s
Greedy-RB construction time (offline)	2 h 50 min	5 h
Greedy-RB evaluation time (online)	0.95 s	1.9 s
POD construction time (offline)	8 min	16 min
POD evaluation time (online)	1.1 s	1.9 s
Computational speedup greedy-RB	3.8	4.2
Computational speedup POD	3.3	4.2

expensive when dealing with several (say 50 or more) affine terms, as in our case. Since the error indicators have to be computed for any $\mu \in \Xi_{\text{train}}$, this affects the speed of the greedy algorithm.

As a final, general remark, we compare our results to a previous work [11] on RB approximation of potential flows. In this work, the problem of potential flows past airfoils, parametrized with respect to the angle of attack and thickness, does not consider any lift; moreover, a finite element method is used as the high-fidelity approximation. Furthermore, only affine geometrical maps are considered, yielding simple deformations such as scaling/sizing based on airfoil thickness and rotation. Even if the ROM performances during the online stage are comparable, involving in each case $O(10)$ basis functions, the offline stage is in our case 10 times less expensive. In addition, our framework allows to:

i. evaluate physically meaningful output thanks to the boundary integral formulation;
ii. characterize a wide NACA family in exact terms from a geometrical standpoint, thanks to our IGA parametrization;
iii. perform the offline stage in a much more efficient way.

Both our ROMs allow to deal with systems of $O(10)$ starting from a high-fidelity method with $O(10^2)$ degrees of freedom, which is already a rather small problem. We expect to obtain a much significant model reduction if we consider a larger dimension of the high-fidelity BEM problem (see Table 3 for a first example), or three dimensional problems.

6. Conclusions and future works

The main goal of the present work has been the analysis and implementation of reduced order models for the rapid and reliable solution of potential flows past airfoils parametrized with respect to the angle of attack and the NACA number identifying their shape. The high-fidelity model we have employed is an Isogeometric boundary element method (IGA-BEM) for the solution of external flows around arbitrary geometries. The isogeometric paradigm is very attractive when coupled with boundary element methods, because it allows the analysis of industrial CAD designs (typically described by NURBS surfaces), by direct coupling with CAD data structures without the generation of volumetric NURBS meshes (an outstanding open problem of isogeometric finite element analysis). We have also implemented a technique that allows the automatic generation of B-spline patches describing the NACA 4-digits family of airfoils at arbitrary angle of attack. Such a parametrization leads to a non affine parametric dependence in the boundary integral equation that describes the flow, and it requires an Empirical Interpolation Method to deal with parametrized operators efficiently. We have applied this technique directly to the algebraic structures of the IGA-BEM approximation, yielding a speed up of the assembly of parameter dependent IGA-BEM matrices by one order of magnitude.

We have constructed two different reduced spaces, by using a Proper Orthogonal Decomposition and a greedy-RB method. We have successfully tested the results coming from both these techniques for two different airfoils of the NACA 4-digits family (NACA 0012 and NACA 4412), comparing them with experimental results and other numerical tools (Xfoil and a B-splines based method presented in [24]). Numerical tests have shown a great agreement with both experimental and numerical data, as well as a good computational saving with respect to the high-fidelity model.

Although the construction of reduced spaces using POD and greedy-RB algorithm is very different, we have not noticed relevant differences in their accuracy and rapidity in the prediction of the results during the Online stage, for the problem at hand. On the contrary, their behavior during the Offline stage is quite different: in the problems we analyzed, POD is much faster than greedy-RB algorithm. This difference is mostly related to the number of affine terms and to the dimension of the high-fidelity model. A huge number of affine terms increases dramatically the cost of the greedy-RB algorithm (due to the evaluation of the error indicator over the whole training sample, at each iteration), whereas the fast speed of the high fidelity model (IGA-BEM) makes the computation of a great number of snapshots (required by POD) a reasonable strategy. This ultimately makes the use of greedy-RB algorithm unfavorable for the case we have considered. Nevertheless, solving a great number of full order problems could be very expensive, or even impossible, for more complex problems—this could make the greedy-RB strategy preferable, depending on the case at hand.

In conclusion, the main achievement of this work is the coupling of reduced order models with isogeometric boundary element methods. The work carried out shows that this coupling can be very effective to characterize flow problems defined on complex parametrized geometries. Although we have focused on fluid flows past airfoils, the technology outlined in the paper is well suited for several other applications related to aero and hydro dynamics problems that are interesting from both academic and industrial standpoints. Not only, two relevant contexts where the proposed method would have a significant impact in terms of computational speedup are optimal flow control and optimal design problems—in this respect, maximizing aero/hydro dynamics performances is crucial to several applications. Other possible extensions include the treatment of three-dimensional problems [57], free surface boundary conditions and the wave resistance problem [58,28] and other flow regimes, such as viscous flows [40]. Of particular note is the case of swimming [59–61] and motility problems [62], where real time simulations are necessary whenever aiming at driving artificial microswimmers and crawlers using optimal control tools.

Acknowledgments

This work has been supported by the SHARM 2012-2014 SISSA post-doctoral research grant on the project "Reduced Basis Methods for shape optimization in computational fluid dynamics", by the project OpenViewSHIP, "Sviluppo di un ecosistema computazionale per la progettazione idrodinamica del sistema elica-carena", supported by Regione FVG - PAR FSC 2007-2013, Fondo per lo Sviluppo e la Coesione and by the SISSA "Research training fellowship for undergraduate students", 2013.

Appendix A. An isogeometric description of NACA 4-digits airfoils

The NACA 4-digits airfoils [18] are a family of airfoils developed by the National Advisory Committee for Aeronautics (NACA) whose shape is uniquely and analytically defined by four digit integers, indicating respectively the maximum value of the mean-line ordinate in percentage of the chord, the distance from the leading edge to the location of the maximum camber in tens of the chord, and the section thickness in percentage of the chord. For instance, the NACA 4412 has 4% camber located at 40% of the chord from the leading edge, and is 12% thick. The wing section is obtained by combining the camber line and the thickness distribution as shown in Fig. A.1.

If we denote (x_U, y_U) and (x_L, y_L) the points on the upper and the lower surface of the airfoil, respectively, we can write (see Fig. A.2)

$$\begin{cases} x_U = x - y_{th}\cos\beta \\ y_U = y_c + y_{th}\sin\beta \end{cases} \tag{A.1}$$

and

$$\begin{cases} x_L = x + y_{th}\cos\beta \\ y_L = y_c - y_{th}\sin\beta. \end{cases} \tag{A.2}$$

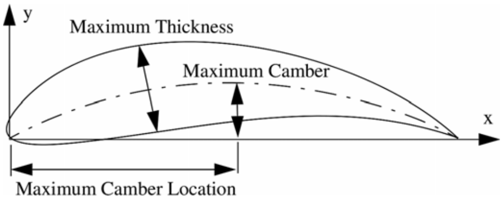

Fig. A.1. 4-digits NACA profiles are expressed as function of their maximum camber, maximum camber location and maximum thickness.

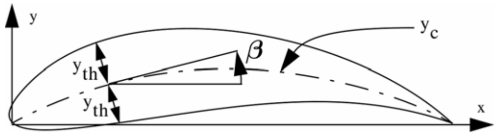

Fig. A.2. Schematic representation of the construction of the airfoil given the analytic expression of its thickness y_{th}, mean line y_c and camber θ.

The thickness distribution y_{th}, the camber line y_c and the angle β are given by:

$$y_{th} = 5\tau c\left(0.2969\sqrt{\frac{x}{c}} - 0.126\frac{x}{c} - 0.3537\left(\frac{x}{c}\right)^2 + 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4\right) \tag{A.3}$$

$$y_c = \begin{cases} \dfrac{m}{p^2}\left(2p\dfrac{x}{c} - \left(\dfrac{x}{c}\right)^2\right) & \text{for } \dfrac{x}{c} \le p \\ \dfrac{m}{(1-p)^2}\left(1 - 2p + 2p\dfrac{x}{c} - \left(\dfrac{x}{c}\right)^2\right) & \text{for } \dfrac{x}{c} \ge p \end{cases} \tag{A.4}$$

$$\beta = \begin{cases} \operatorname{atan}\left(\dfrac{2m}{cp^2}\left(p - \dfrac{x}{c}\right)\right) & \text{for } \dfrac{x}{c} \le p \\ \operatorname{atan}\left(\dfrac{2m}{c(1-p)^2}\left(p - \dfrac{x^2}{c}\right)\right) & \text{for } \dfrac{x}{c} \ge p, \end{cases} \tag{A.5}$$

where c is the airfoil chord length, m is the maximum camber, p is the maximum camber location and τ is the maximum thickness.

In order to provide a B-spline description of the airfoil shape, we start from a knot vector $\boldsymbol{\theta}$ and a set of basis functions $\{\xi_i(s)\}_{i=1}^{\mathcal{N}}$, by choosing the control points $\{\boldsymbol{P}_i\}_{i=1}^{\mathcal{N}}$ through the following least-square procedure. Let us denote by $\boldsymbol{\gamma}(\text{Naca}) \in \mathbb{R}^{M\times 2}$ a matrix whose rows are a set of points on the given NACA airfoil, distributed according to their arc length:

$$\boldsymbol{\gamma}(\text{Naca}) = \begin{bmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_M \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_M & y_M \end{bmatrix} \quad \text{with } M \ge \mathcal{N}. \tag{A.6}$$

Let us denote by $\boldsymbol{c}(s; \text{Naca})$, $s \in [0, 1]$ the B-spline description of the given airfoil Γ_f. We obtain its control points by the following least-square procedure:

$$\boldsymbol{P}(\text{Naca}) := \arg\min_{\boldsymbol{P}} \sum_{i=1}^{M} \frac{1}{2}\left|\boldsymbol{\gamma}_i(\text{Naca}) - \sum_{j=1}^{\mathcal{N}} \boldsymbol{P}_j \xi_j(s_i),\right|^2 \tag{A.7}$$

where $s_i := \frac{i}{M+1}$ for $i = 1, \ldots, M$. This is equivalent to solving the following problem:

$$\frac{\partial}{\partial \boldsymbol{P}_j}\left(\sum_{i=1}^{M}\frac{1}{2}\left(\boldsymbol{\gamma}_i(\text{Naca}) - \sum_{k=1}^{\mathcal{N}}\boldsymbol{P}_k\xi_k(s_i)\right)\cdot\left(\boldsymbol{\gamma}_i(\text{Naca}) - \sum_{l=1}^{\mathcal{N}}\boldsymbol{P}_l\xi_l(s_i)\right)\right) = 0 \quad \text{for } j = 1, \ldots, \mathcal{N}, \tag{A.8}$$

that is

$$\frac{\partial}{\partial \boldsymbol{P}_j}\sum_{i=1}^{M}\left(\left(\sum_{l=1}^{\mathcal{N}}\boldsymbol{P}_l\xi_l(s_i)\right)\cdot\left(\sum_{k=1}^{\mathcal{N}}\boldsymbol{P}_k\xi_k(s_i)\right) - \boldsymbol{\gamma}_i\cdot\left(\sum_{k=1}^{\mathcal{N}}\boldsymbol{P}_k\xi_k(s_i)\right)\right)$$
$$= \sum_{i=1}^{M}\sum_{k=1}^{\mathcal{N}}\boldsymbol{P}_k\xi_k(s_i)\xi_j(s_i) - \sum_{i=1}^{M}\boldsymbol{\gamma}_i\xi_j(s_i) = 0 \quad \text{for } j = 1, \ldots, \mathcal{N}, \tag{A.9}$$

where $(\cdot,\cdot)$ denotes the (Euclidean) scalar product in $\mathbb{R}^2$. If we define $\boldsymbol{B}$ such that

$$B_{ij} := \xi_j(s_i), \tag{A.10}$$

Eq. (A.8) translates to

$$\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{P}(\text{Naca}) = \boldsymbol{B}^T\boldsymbol{\gamma}(\text{Naca}). \tag{A.11}$$

Given $\boldsymbol{\gamma}(\text{Naca})$, we obtain automatically the control points position $\boldsymbol{P}(\text{Naca})$ (for null angle of attack) by solving (A.11). Afterwards, we can easily apply a rotation to the control polygon in order to rotate the airfoil according to its angle of attack, obtaining the control points $\boldsymbol{P}(\boldsymbol{\mu})$ of the current airfoil geometry. In this framework, based on isogeometric analysis and B-spline functions, the position of the entire set of control points can be changed by simply varying a very few (compared to the number of control points) parameters thanks to a suitable least square procedure. This feature makes the proposed shape parametrization different from other geometrical maps, such as FFD or Radial Basis Functions (RBF) techniques [6,5,63], which typically require many more parameters to treat a class of shapes/deformations of comparable complexity, and very often do not provide any clear geometrical meaning to these parameters.

Appendix B. Post processing

Once we have computed the solution u on the boundary Γ by solving system (64), we have to post process it in order to evaluate both the velocity field all over the domain Ω, and some outputs of interest, such as pressure and lift coefficients. Let us recall the definition of the pressure coefficient C_p:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2}, \tag{B.1}$$

where p is the pressure, p_∞, ρ and V_∞ are the upstream infinity pressure, density and velocity magnitude. By applying Bernoulli equation over a streamline between the upstream infinity and a point on the airfoil surface, we obtain

$$p + \frac{1}{2}\rho V^2 = p_\infty + \frac{1}{2}\rho V_\infty^2, \quad \text{that is, } p = p_\infty + \frac{1}{2}\rho V_\infty^2 - \frac{1}{2}\rho V^2 \tag{B.2}$$

so that, exploiting (B.2), we can write the pressure coefficient as follows:

$$C_p = 1 - \frac{V^2}{V_\infty^2} = 1 - \frac{|\nabla\Phi|^2}{V_\infty^2}. \tag{B.3}$$

Let us focus on $\nabla\Phi$, and let us analyze how it can be written, exploiting some properties ensured by the isogeometric framework. In particular, we can express

$$\nabla\Phi = \nabla_s\Phi = \nabla_s u + \nabla_s(\boldsymbol{V}_\infty\cdot\boldsymbol{x}) = J^{-2}u'(s)\frac{\partial\boldsymbol{c}(s)}{\partial s} + J^{-2}V_\infty\frac{\partial c_x(s)}{\partial s}\frac{\partial\boldsymbol{c}(s)}{\partial s} \tag{B.4}$$

where $u'(s)$ is the first derivative of the potential, J is the Jacobian of the transformation, $\frac{\partial \boldsymbol{c}(s)}{\partial s}$ and $\frac{\partial c_x(s)}{\partial s}$ are the first derivatives of the mapping and of the first component of the mapping $\boldsymbol{c}(s)$. Hence we obtain

$$|\nabla \Phi|^2 = J^{-4}\left(u'(s) + V_\infty \frac{\partial c_x(s)}{\partial s}\right)^2 \frac{\partial \boldsymbol{c}(s)}{\partial s}^2 = J^{-2}\left(u'(s) + V_\infty \frac{\partial c_x(s)}{\partial s}\right)^2, \tag{B.5}$$

so that the pressure coefficient can be evaluated by inserting (B.5) in (B.3), and by exploiting the representation of the derivatives of u in the isogeometric framework (see Section 3). The pressure coefficient is one of the most important outputs related to airfoil characterization, since it gives local informations about the behavior of the pressure and (through Bernoulli equation) about the velocity field close to the airfoil. Once we have the pressure coefficient, we can obtain the lift coefficient by integrating the expression of C_p on the airfoil surface Γ_f. Assuming that the profile's chord c is equal to 1, we can write

$$C_l = \int_{\Gamma_f} C_p \boldsymbol{n} \mathrm{d}s \cdot \boldsymbol{n}_\perp, \tag{B.6}$$

where $\boldsymbol{n}$ is the unit vector normal to the airfoil surface and $\boldsymbol{n}_\perp$ is the unit vector normal to $\boldsymbol{V}_\infty$. Alternatively, we can compute the lift coefficient through the circulation value. We recall that the circulation on the airfoil is given by $K = u_{N_f} - u_1$. Thus, we can write the lift force both as

$$L = -\rho_\infty V_\infty K, \tag{B.7}$$

exploiting Kutta–Joukowski theorem, or as

$$L = \frac{1}{2}\rho_\infty V_\infty^2 C_l. \tag{B.8}$$

Hence, we obtain

$$C_l = -2\frac{K}{V_\infty}. \tag{B.9}$$

Finally, we can also recover the velocity field in the whole external domain Ω. Recalling the boundary integral equation (20), we have the following representation formula:

$$u(\boldsymbol{x}) = -\int_\Gamma u(\boldsymbol{y})\frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy + \int_\Gamma G(\boldsymbol{x}-\boldsymbol{y})h(\boldsymbol{y})dy, \tag{B.10}$$

valid for any $\boldsymbol{x} \in \Omega$. By definition of potential u, the velocity is $\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{V}_\infty + \nabla_x u(\boldsymbol{x})$, so that

$$\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{V}_\infty - \int_\Gamma u(\boldsymbol{y})\nabla_x \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy - \int_\Gamma \nabla_x G(\boldsymbol{x}-\boldsymbol{y})h(\boldsymbol{y})dy. \tag{B.11}$$

In the same way, we can obtain the pressure field from (B.2).

Appendix C. Empirical interpolation method

Let us denote by $g(\boldsymbol{x}; \boldsymbol{\mu})$ a general scalar function depending on both the spatial coordinates $\boldsymbol{x}$ and the parameters vector $\boldsymbol{\mu}$. The empirical interpolation method enables to express $g(\boldsymbol{x}, \boldsymbol{\mu})$ through an approximate expansion $g(\boldsymbol{x}, \boldsymbol{\mu}) = g_M(\boldsymbol{x}, \boldsymbol{\mu}) + e_{EIM}(\boldsymbol{\mu})$, where

$$g_M(\boldsymbol{x}, \boldsymbol{\mu}) = \sum_{j=1}^{M} \gamma^j(\boldsymbol{\mu})\zeta_j(\boldsymbol{x}), \tag{C.1}$$

$\gamma^j(\boldsymbol{\mu}), j = 1, \dots, M$ are $\boldsymbol{\mu}$-dependent functions, $g_j(\boldsymbol{x}), j = 1, \dots, M$ are $\boldsymbol{\mu}$-independent functions (or *shape functions*) and $e_{EIM}(\boldsymbol{\mu})$ is an error term, which we want to keep as small as possible over the parameter space. To do this, EIM seeks a sequence of (nested) sets of interpolation points $T_M = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_M\}$ also called *magic points*, with $\boldsymbol{p}_j \in \Omega$ for each $j = 1, \dots, M$, and a set of shape functions $g_j(\boldsymbol{x})$, such that $g_M(\boldsymbol{p}_j, \boldsymbol{\mu}) = g(\boldsymbol{p}_j, \boldsymbol{\mu})$ for each $j = 1, \dots, M$, that is,

$$\sum_{j=1}^{M} B^M_{i,j} \gamma^j(\boldsymbol{\mu}) = g(\boldsymbol{p}_i, \boldsymbol{\mu}), \qquad \forall i = 1, \dots, M, \tag{C.2}$$

where $B^M_{ij} := \zeta_j(\boldsymbol{p}_i)$, $\forall i, j = 1, \dots, M$. During the online stage, for any prescribed parameter value, the solution of the linear system (C.2) – also denoted as Lagrange interpolation problem – gives the expression of the expansion $g_M(\boldsymbol{x}, \boldsymbol{\mu})$, provided the shape functions and the magic points selected. These two operations are run in the Offline stage, according to a greedy algorithm, described below.

Let us denote by $\Xi^{EIM}_{\text{train}} \subset \mathcal{D}$ a large training set, representing a discrete version of the parameter space, $M_{\max}$ the maximum number of terms in the expansion, ε^*_{EIM} a prescribed tolerance. First, we seek a set of shape functions. We start from a (random) initial parameter value $\boldsymbol{\mu}^1$, define $S_1 = \{\boldsymbol{\mu}^1\}$, compute the initial basis function $\zeta_1(\mathbf{x}) = g(\mathbf{x}; \boldsymbol{\mu}^1)$ and store $G_1 = \text{span}\{\zeta_1\}$. Then, for $M \geq 2$, we use (C.2) to approximate each snapshot of the generating function, and choosing the parameter of the snapshot that is the worst approximated to be

$$\boldsymbol{\mu}^M := \arg \max_{\boldsymbol{\mu} \in \Xi^{EIM}_{\text{train}}} \inf_{v \in G_{M-1}} \|g(\cdot, \boldsymbol{\mu}) - v\|_{L^\infty(\Omega)}. \tag{C.3}$$

This problem can be rewritten as a least-square problem, and has to be solved for any $\boldsymbol{\mu} \in \mathcal{D}$. The snapshot $g(\boldsymbol{x}, \boldsymbol{\mu}^M)$ is then normalized to form the Mth basis function, so that $\zeta_M(\mathbf{x}) = g(\mathbf{x}; \boldsymbol{\mu}^M)$, $G_{M-1} \cup \{\zeta_1\} = G_M := \text{span}(\zeta_1, \dots, \zeta_M)$. The greedy procedure stops when the EIM error is below the tolerance ε^*_{EIM}, for any $\boldsymbol{\mu} \in \Xi^{EIM}_{\text{train}}$. The whole procedure is shown in Algorithm 1:

Algorithm 1

```
$\zeta_1(\mathbf{x}) := g(\mathbf{x}, \boldsymbol{\mu}^1)$;   $G_1 := \text{span}(\zeta_1)$;
for  $M = 2 : M_{\max}$
    solve $\boldsymbol{\mu}^M := \arg\max_{\boldsymbol{\mu} \in \Xi^{EIM}_{\text{train}}} \inf_{v \in G_{M-1}} \|g(\cdot, \boldsymbol{\mu}) - v\|_{L^\infty(\Omega)}$
    set $\zeta_M(\mathbf{x}) := g(\mathbf{x}, \boldsymbol{\mu}^M)$,    $G_M := \text{span}(\zeta_1, \dots, \zeta_M)$;
    if  $\max_{\boldsymbol{\mu} \in \Xi^{EIM}_{\text{train}}} \inf_{v \in G_M} \|g(\cdot, \boldsymbol{\mu}) - v\|_{L^\infty(\Omega)} < \varepsilon^*_{EIM}$
        $M_{\max} = M - 1$;
    end;
end.
```

Then, we construct a set of nested interpolation points $T_M = \{\mathbf{p}_1, \dots, \mathbf{p}_1\}$, with $M = 1, \dots, M_{\max}$, according to a greedy algorithm. We select the first magic point $\boldsymbol{p}_1$ and the first basis function $q_1(\mathbf{x})$ normalized to 1 for the magic point $\boldsymbol{p}_1$:

$$\mathbf{p}_1 := \arg \operatorname*{ess\,sup}_{\mathbf{x} \in \Omega} |\zeta_1(\mathbf{x})|, \qquad g_1 = \zeta_1(\mathbf{x})/\zeta_1(\mathbf{p}_1), \qquad B^1_{11} = 1. \tag{C.4}$$

Then, for $M = 2 : M_{\max}$, we solve the linear system

$$\sum_{j=1}^{M-1} \sigma^{M-1}_j g_j(\mathbf{p}_i) = \zeta_M(\mathbf{p}_i), \quad i = 1, \dots, M-1 \tag{C.5}$$

and compute the residual $r_M(\mathbf{x}) := \zeta_M(\mathbf{x}) - \sum_{j=1}^{M-1} \sigma^{M-1}_j g_j(\mathbf{x})$; the mth magic point is given by

$$\mathbf{p}_M := \arg \operatorname*{ess\,sup}_{\mathbf{x} \in \Omega} |r_M(\mathbf{x})|, \tag{C.6}$$

and then we set $g_M(\mathbf{x}) = r_M(\mathbf{x})/r_M(\mathbf{p}_M)$, $B^M_{ij} = g_j(\mathbf{p}_i)$ for any $i, j = 1, \dots, M$. All these operations are summarized in Algorithm 2:

Algorithm 2

```
compute $\mathbf{p}_1 := \arg\operatorname{ess\,sup}_{\mathbf{x}\in\Omega} |\zeta_1(\mathbf{x})|$;
$q_1 = \zeta_1(\mathbf{x})/\zeta_1(\mathbf{p}_1)$, set $B^1_{11} = 1$;
for $M = 2 : M_{\max}$
    solve $\sum_{j=1}^{M-1} \sigma_j^{M-1} q_j(\mathbf{p}_i) = \zeta_M(\mathbf{p}_i), \qquad i = 1, \dots, M-1$;
    compute (*residual*) $r_M(\mathbf{x}) := \zeta_M(\mathbf{x}) - \sum_{j=1}^{M-1} \sigma_j^{M-1} g_j(\mathbf{x})$;
    compute $\mathbf{p}_M := \arg\operatorname{ess\,sup}_{\mathbf{x}\in\Omega} |r_M(\mathbf{x})|$;
    set $q_M(\mathbf{x}) = r_M(\mathbf{x})/r_M(\mathbf{p}_M), \quad B^M_{ij} = q_j(\mathbf{p}_i), \quad i, j = 1, \dots, M$;
end.
```

As a matter of fact, the interpolation points $\{\boldsymbol{p}_j\}_{j=1}^M$ and the basis functions $\{g_j(\boldsymbol{x})\}_{j=1}^M$ are obtained, at each step M, as the maximum point and the normalization of the residual function $r_M(\boldsymbol{x})$ resulting from the interpolation of $\zeta_M(\boldsymbol{x})$ by (C.5). Further details about the EIM algorithm, and its application in the context of RB methods, can be found e.g. in [41,63].

References

[1] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives, in: A. Quarteroni, G. Rozza (Eds.), Reduced Order Methods for Modeling and Computational Reduction, vol. 9, Springer, 2013, pp. 235–274. MS&A Series.

[2] G. Rozza, D. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (3) (2008) 229–275.

[3] A. Quarteroni, G. Rozza, A. Manzoni, Certified reduced basis approximation for parametrized partial differential equations and applications, J. Math. Ind. 1 (3) (2011).

[4] F. Negri, G. Rozza, A. Manzoni, A. Quarteroni, Reduced basis method for parametrized elliptic optimal control problems, SIAM J. Sci. Comput. 35 (5) (2013) A2316–A2340.

[5] A. Manzoni, A. Quarteroni, G. Rozza, Shape optimization of cardiovascular geometries by reduced basis methods and free-form deformation techniques, Internat. J. Numer. Methods Fluids 70 (5) (2012) 646–670.

[6] T. Lassila, G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, Comput. Methods Appl. Mech. Engrg. 199 (23–24) (2010) 1583–1592.

[7] T. Lassila, A. Manzoni, A. Quarteroni, G. Rozza, A reduced computational and geometrical framework for inverse problems in haemodynamics, Int. J. Numer. Methods Biomed. Eng. 29 (7) (2013) 741–776.

[8] T.J. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (39) (2005) 4135–4195.

[9] M. Barrault, Y. Maday, N.C. Nguyen, 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) (2004) 667–672.

[10] G. Rozza, Reduced basis approximation and error bounds for potential flows in parametrized geometries, Commun. Comput. Phys. 9 (2011) 1–48.

[11] C. Günther, Reduced Basis Method for the Shape Optimization of Racing Car Components (Master Thesis), Ecole Polytechnique Fédérale de Lausanne, 2008.

[12] S. Pagani, A. Manzoni, Model reduction of PDE-constrained parametric optimization problems by an adjoint-based approach, Technical report, 2014, submitted for publication.

[13] A. Manzoni, An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier–Stokes flows, ESAIM Math. Modelling Numer. Anal. 48 (2014) 1199–1226.

[14] M. Fares, J.S. Hesthaven, Y. Maday, B. Stamm, The reduced basis method for the electric field integral equation, J. Comput. Phys. 230 (14) (2011) 5532–5555.

[15] M. Ganesh, J. Hesthaven, B. Stamm, A reduced basis method for electromagnetic scattering by multiple particles in three dimensions, J. Comput. Phys. 231 (23) (2012) 7756–7779.

[16] J.S. Hesthaven, B. Stamm, S. Zhang, Certified reduced basis method for the electric field integral equation, SIAM J. Sci. Comput. 34 (3) (2012) A1777–A1799.

[17] F. Salmoiraghi, Reduced order models for potential flows past parametrized NACA airfoils based on an isogeometric boundary element method (Master thesis), Politecnico di Milano, 2014.

[18] I.H. Abbott, A. Von Doenhoff, Theory of Wing Sections: Including A Summary of Airfoil Data, Dover Publications, 1959.

[19] K. Karamcheti, Principles of Ideal-Fluid Aerodynamics, Wiley, New York, 1966.

[20] P. Bassanini, C. Casciola, M. Lancia, R. Piva, Edge singularities and Kutta condition in 3D aerodynamics, Meccanica 34 (3) (1999) 199–229.

[21] G.C. Hsiao, W.L. Wendland, Boundary Integral Equations, Springer, 2008.

[22] O. Steinbach, Numerical Approximation Methods For Elliptic Boundary Value Problems: Finite and Boundary Elements, Springer, 2008.

[23] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Comput. Methods Appl. Mech. Engrg. 254 (2013) 197–221.

[24] C.-S. Lee, J.E. Kerwin, A b-spline higher-order panel method applied to two-dimensional lifting problem, J. Ship Res. 47 (4) (2003) 290–298.

[25] R.R. Simpson, S.S. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis, Comput. Methods Appl. Mech. Engrg. 209–212 (2012) 87–100.

[26] T. Takahashi, T. Matsumoto, An application of fast multipole method to isogeometric boundary element method for Laplace equation in two dimensions, Eng. Anal. Bound. Elem. 36 (12) (2012) 1766–1775.

[27] G.-D. Kim, C.-S. Lee, J. Kerwin, A B-spline based higher order panel method for analysis of steady flow around marine propellers, Ocean Eng. 34 (14) (2007) 2045–2060.

[28] A. Ginnis, K. Kostas, C. Politis, P. Kaklis, K. Belibassakis, T. Gerostathis, M. Scott, T. Hughes, Isogeometric boundary-element analysis for the wave-resistance problem using T-splines, Comput. Methods Appl. Mech. Engrg. 279 (2014) 425–439.

[29] R. Simpson, M. Scott, M. Taus, D. Thomas, H. Lian, Acoustic isogeometric boundary element analysis, Comput. Methods Appl. Mech. Engrg. 269 (2014) 265–290.

[30] K. Li, X. Qian, Isogeometric analysis and shape optimization via boundary integral, Comput.-Aided Design 43 (11) (2011) 1427–1437.

[31] J.A. Cottrell, T.J. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD And FEA, John Wiley & Sons, 2009.

[32] J.L. Hess, A. Smith, Calculation of Non-lifting Potential Flow About Arbitrary Three-dimensional Bodies, Technical Report, DTIC Document, 1962.

[33] J.L. Hess, Review of integral-equation techniques for solving potential-flow problems with emphasis on the surface-source method, Comput. Methods Appl. Mech. Engrg. 5 (2) (1975) 145–196.

[34] L. Morino, C.-C. Kuot, Subsonic potential aerodynamics for complex configurations: a general theory, AIAA J. 12 (2) (1974) 191–197.

[35] J.E. Kerwin, S.A. Kinnas, J.-T. Lee, W.-Z. Shih, A surface panel method for the hydrodynamic analysis of ducted propellers, SNAME Trans. 95 (1987) 93–122.

[36] C. Hsin, J. Kerwin, J. Newman, A Higher-Order Panel Method Based on B-splines, 1994.

[37] H.D. Maniar, A three dimensional higher order panel method based on B-splines (PhD thesis), Massachusetts Institute of Technology, 1995.

[38] J. Telles, A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals, Internat. J. Numer. Methods Engng. 24 (5) (1987) 959–973.

[39] H.R. Kutt, Quadrature Formulae For Finite-Part Integrals, Technical Report, WISK, 178, National Research Institute for Mathematical Sciences, Pretoria, South Africa, 1975.

[40] L. Heltai, M. Arroyo, A. DeSimone, Nonsingular isogeometric boundary element method for Stokes flows in 3D, Comput. Methods Appl. Mech. Engrg. 268 (2014) 514–539.

[41] M. Grepl, Y. Maday, N. Nguyen, A. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM Math. Modelling Numer. Anal. 41 (3) (2007) 575–605.

[42] P. Benner, S. Gugercin, K. Willcox, A Survey of Model Reduction Methods for Parametric Systems, Technical Report 13–14, Max Planck Institute Magdeburg, 2013.

[43] K. Carlberg, R. Tuminaro, P. Boggs, Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics, 2014, preprint arXiv:1401.8044.

[44] D. Wirtz, D. Sorensen, B. Haasdonk, A posteriori error estimation for deim reduced nonlinear dynamical systems, SIAM J. Sci. Comput. 36 (2) (2014) A311–A338.

[45] D. Amsallem, A. Manzoni, F. Negri, Parametrized Matrices Interpolation Based on Discrete Empirical Interpolation for Efficient Model Reduction, Technical Report, 2014, submitted for publication.

[46] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[47] A. Quarteroni, Numerical Models for Differential Problems, in: Modeling Simulation and Applications (MS&A), vol. 9, Springer, 2014.

[48] G. Berkooz, P. Holmes, J. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1) (1993) 539–575.

[49] S.S. Ravindran, Proper orthogonal decomposition in optimal control of fluids, Int. J. Numer. Methods Fluids 34 (1999) 425–448.

[50] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2002) 492–515.

[51] T. Bui-Thanh, K. Willcox, O. Ghattas, B. van BloemenWaanders, Goal-oriented, model-constrained optimization for reduction of large-scale systems, J. Comput. Phys. 224 (2) (2007) 880–896.

[52] K. Veroy, C. Prud'homme, D. Rovas, A. Patera, A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations, in: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, vol. 3847, 2003.

[53] G. Rozza, D. Huynh, A. Manzoni, Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf–sup stability constants, Numer. Math. 125 (1) (2013) 115–152.

[54] N. Gregory, C. O'reilly, Low-speed aerodynamic characteristics of NACA 0012 aerofoil section, including the effects of upper-surface roughness simulating hoar frost, National Physical Laboratory Teddington, England, 1970.

[55] F. Moyers, A comparison of theoretical and experimental pressure distributions at high speed about the NACA 4412 airfoil (Master thesis), California Institute of Technology, 1940.

[56] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer-Verlag, New York, 2000.

[57] J. Gu, J. Zhang, G. Li, Isogeometric analysis in BIE for 3-D potential problem, Eng. Anal. Bound. Elem. 36 (5) (2012) 858–865.

[58] A. Mola, L. Heltai, A. DeSimone, A stable and adaptive semi-Lagrangian potential model for unsteady and nonlinear ship-wave interactions, Eng. Anal. Bound. Elem. 37 (1) (2013) 128–143.

[59] F. Alouges, A. DeSimone, L. Heltai, Numerical strategies for stroke optimization of axisymmetric microswimmers, Math. Models Methods Appl. Sci. 21 (02) (2011) 361–387.

[60] F. Alouges, A. DeSimone, L. Heltai, Optimally swimming stokesian robots, Discrete Contin. Dyn. Syst. Ser. B (DCDS-B) 18 (5) (2013) 1189–1215.

[61] M. Arroyo, L. Heltai, D. Millán, A. DeSimone, Reverse engineering the euglenoid movement, Proc. Natl. Acad. Sci. USA 109 (44) (2012) 17874–17879.

[62] P. Gidoni, G. Noselli, A. DeSimone, Crawling on directional surfaces, Int. J. Non-Linear Mech. 61 (2014) 65–73.

[63] A. Manzoni, Reduced models for optimal control, shape optimization and inverse problems in haemodynamics (PhD thesis), Ecole Polyechnique Fédérale de Lausanne, 2012.