POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione

Dipartimento di Scienze e Tecnologie Aerospaziali

Corso di Laurea Magistrale in
Ingegneria Aeronautica

Reduced-order models for potential flows past parametrized NACA airfoils based on an Isogeometric boundary element method

Advisor:

Prof. Maurizio QUADRIO

Co-Advisors:

Dr. Andrea MANZONI
Dr. Luca HELTAI

Author:

Filippo SALMOIRAGHI
Matr. 783402

Anno Accademico 2012-2013

Contents

List of Figures

List of Tables

Abstract

Several applications of computational fluid dynamics require to simulate many different possible realizations of a system, thus yielding relevant computational challenges and, very often, large demand on computational resources. This is the case, for instance, of optimization, control and design problems in aerodynamics. A possible way to alleviate this computational burden is provided by reduced order models (ROMs), that is, low-dimensional, efficient models which are fast to solve, but also able to approximate well the underlying high-fidelity simulations.

In this work we analyse and implement a Reduced Basis (RB) method 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. This method allows to capture the essential flow features by means of a handful of degrees of freedom, and to keep under control the error with respect to a high-fidelity solution, all over the parameter space.

For the construction of our RB method we rely on a high-fidelity approximation technique given by an Isogeometric Boundary Element Method (IGA-BEM), thus leading to a very efficient Isogeometric Reduced Basis (IGA-RB) Method for the reduction of shape-dependent problems. We have decided to rely on a Galerkin-Boundary Element Method because it enables a preliminary reduction of the problem dimension, through a suitable boundary integral formulation, and the chance to treat external flows in (possibly) infinite domains. On the other hand, Isogeometric Analysis allows a direct interface with CAD tools, in view of possible extensions to complex applications of industrial interest. Moreover, in order to ensure a suitable Offline/Online decomposition between ROM construction and evaluation, a suitable Empirical Interpolation Method has been applied.

We have adopt two different strategies for the construction of the reduced spaces, namely the Proper Orthogonal Decomposition (POD) and a Greedy algorithm, by showing the main analogies and differences for the case at hand, and their computational performances. Finally, we validate the results – obtained both with the high-fidelity IGA-BEM method and the reduced order models – with respect to experimental data and numerical codes (Xfoil), showing in both case a great agreement.

Acknowledgements

In primo luogo, vorrei ringraziare il mio relatore, il Professor Maurizio Quadrio, per la disponibilità mostrata durante lo sviluppo di questa tesi e per l'occasione concessami di crescere dal punto di vista lavorativo e personale. Inoltre, tutto il lavoro fatto non sarebbe stato possibile senza il continuo aiuto, supporto e incitamento da parte dei miei correlatori, Luca Heltai e Andrea Manzoni.

Voglio ringraziare il Professor Antonio De Simone e il resto del gruppo SISSA Mathlab che mi hanno accolto con gentilezza e disponibilità dal primo momento del mio arrivo a Trieste. Grazie anche ai ragazzi dell'ufficio 112 per avermi ospitato ogni volta che l'atmosfera si faceva insostenibile nel mio ufficio; grazie a Marco, per aver affrontato e sostenuto con me le prove e le fatiche di essere tesisti al Mathlab.

Mille ringraziamenti vanno anche alla mia famiglia per avermi sempre supportato e aiutato a trovare la mia strada. Senza di voi non sarei mai arrivato dove sono arrivato.

Dulcis in fundo, un ringraziamento speciale ai miei amici ”buhi di hulo”, Renza e Miguel, per la loro amicizia in questi anni e per aver condiviso i molti momenti belli e brutti della vita.

Sintesi

In questa tesi viene analizzato e implementato un **metodo a basi ridotte** per la soluzione **rapida** e **affidabile** del flusso a potenziale intorno a **profili alari parametrizzati** in funzione dell'angolo d'incidenza del profilo e del numero NACA identificativo della loro forma. Questo metodo consente di catturare le caratteristiche essenziali del comportamento di un sistema descritto da un modello differenziale parametrizzato riducendone il costo computazionale e mantenendo sotto controllo l'errore rispetto alla soluzione ottenuta mediante un metodo ad alta precisione (high-fidelity), come il metodo degli elementi finiti o degli elementi al contorno.

L'idea generale alla base della riduzione di modello consiste nel risolvere il problema combinando un insieme di soluzioni calcolate per particolari valori dei parametri con un metodo di approssimazione *ad alta precisione*. In questo modo, la dimensione del problema ridotto è data dal numero di queste soluzioni (o *funzioni di base*), che in alcuni casi può essere molto piccolo poichè un esiguo numero di modi riesce a descrivere in modo opportuno il comportamento del sistema, al variare del valore dei parametri.

Nei metodi a basi ridotte è dunque possibile suddividere la risoluzione di un problema in due fasi: una fase *offline* (computazionalmente onerosa), in cui si costruisce lo *spazio ridotto*, ovvero si calcola un insieme di funzioni di base risolvendo il problema per valori appositamente selezionati dei parametri, e una fase *online* (poco costosa) in cui è possibile ottenere una soluzione per valori arbitrari dei parametri, in tempo pressoché reale, mediante una proiezione di Galerkin sullo spazio ridotto.

Questa metodologia è particolarmente efficace in tutti quei casi in cui è necessario risolvere il problema un gran numero di volte (ad esempio in problemi di ottimizzazione) oppure nei casi in cui è necessaria una stima certificata della soluzione in tempo reale (ad esempio in problemi di controllo).

Per l'approssimazione numerica del problema e la costruzione delle funzioni di base nel modello ridotto, si utilizza in questo lavoro un **metodo isogeometrico** (isogeometric analysis, IGA) agli **elementi al contorno** (boundary element method, BEM), basato sulla **formulazione integrale**, grazie al quale si può discretizzare e risolvere il problema solo sul bordo del dominio. In questo modo possediamo uno metodo numerico in grado di trattare flussi esterni, oltre che di ridurre di una dimensione il modello completo per la descrizione del flusso attorno al profilo, ottenendo dunque un'ulteriore diminuzione

di complessità. Per quanto riguarda la costruzione di uno spazio ridotto, numerosi approcci sono possibili, a seconda del tipo di problema in esame. Nel caso di problemi parametrizzati gli approcci più utilizzati (e studiati in letteratura) sono un algoritmo di tipo greedy (basato su una stima a posteriori dell'errore) e la decomposizione ortogonale (o proper orthogonal decomposition, POD). Sottolineamo infine che il metodo a basi ridotte non sostituisce il metodo agli elementi al contorno utilizzato per approssimare numericamente il problema in esame, piuttosto è costruito su di esso: in questo modo, la soluzione ridotta non approssima direttamente la soluzione esatta del problema, quanto piuttosto la soluzione approssimata ottenuta con il metodo BEM.

Gli elementi di novità del lavoro sono molteplici:

i. abbiamo sviluppato un metodo isogeometrico agli elementi al contorno (IGA-BEM) **efficiente** per la soluzione del flusso attorno al profilo;

ii. tramite una procedura di interpolazione empirica, abbiamo ridotto il costo computazionale di assemblaggio delle matrici nel metodo IGA-BEM;

iii. abbiamo costruito un metodo a basi ridotte (reduced basis, RBM) sfruttanfo sia un approccio di tipo *POD* che un algoritmo di tipo *Greedy* (IGA-BE-RBM);

iv. infine, abbiamo validato le soluzioni ottenute con il modello ridotto con dati sperimentali provenienti dalla letteratura, mostrando un ottima capacità di previsione dei risultati da parte dei metodi implementati.

Riportiamo in seguito la struttura dell'elaborato.

Nel **capitolo 1** viene introdotto il modello per la descrizione di un flusso intorno a un profilo alare, che sotto opportune condizioni si può ridurre ad un problema di Poisson con condizioni al contorno miste e una condizione di Kutta che assicura la buona posizione del problema in domini bidimensionali. Sfruttandone la sua formulazione integrale, il problema viene riscritto solo sul bordo del dominio (boundary integral equation, BIE). Forniamo inoltre una formulazione variazionale di questo problema, necessaria per la costruzione di un metodo di approssimazione numerica basato su una **proiezione di Galerkin**.

Tale metodo è descritto nel **capitolo 2**. In particolare, consideriamo un metodo isogeometrico agli elementi di contorno (IGA-BEM), che si differenzia dalle tecniche più classiche (come quelle basate sulla distribuzione di vortici sui pannelli) sia per l'approssimazione della geometria che per la procedura di discretizzazione.

L'ingrediente fondamentale di questa tecnica è la possibilità di utilizzare le stesse funzioni per la descrizione della geometria computazionale e della soluzione del problema differenziale. Ciò comporta notevoli vantaggi del punto di vista dell'accuratezza, e la possibilità di trattare geometrie di interesse industriale. I risultati ottenuti con il metodo IGA-BEM sono stati confrontati e validati con dati sperimentali e risultati provenienti da altri metodi numerici, mostrando un'ottima qualità delle soluzioni ottenute con la tecnica implementata.

Essendo interessati alla descrizione del flusso attorno a un profilo alare al variare della sua forma e dell'angolo di incidenza, introduciamo nel **capitolo 3** una formulazione parametrizzata del problema. In questo modo, è possibile descrivere le caratteristiche (o input) fisiche e geometriche del problema mediante parametri (variabili) nel problema differenziale. Grazie alla descrizione isogeometrica del dominio, l'equazione può essere formulata su una configurazione di riferimento (in questo caso il segmento $[0, 1]$), in cui le caratteristiche fisiche e geometriche sono espresse mediante funzioni dei parametri che compaiono nei coefficienti del problema differenziale. Ciò risulta indispensabile, nella costruzione del modello ridotto, per poter combinare soluzioni del problema corrispondenti a differenti configurazioni geometriche. Tuttavia, come spesso accade nel caso di parametrizzazioni geometriche, la dipendenza parametrica nel problema differenziale risulta di tipo non affine, ovvero i coefficienti del problema dipendono, oltre che dai parametri, anche dalle coordinate spaziali. Ciò rende più complicata la separazione delle componenti parametriche dagli operatori differenziali, necessaria per poter disporre di una procedura offline-online efficiente. Per ovviare a questo fatto, e poter dunque costruire un modello ridotto per la soluzione efficiente del problema parametrizzato, consideriamo una tecnica di interpolazione empirica. Ciò permette di approssimare gli operatori parametrizzati mediante un'opportuna combinazione lineare di operatori indipendenti dai parametri, e poter dunque estrarre la dipendenza parametrica dagli integrali che definiscono tali operatori (dipendenza affine). Questo passo è fondamentale per sfruttare a pieno la divisione *offline-online* dei metodi di riduzione del modello e ottenere quindi un algoritmo efficiente dal punto di vista computazionale. Inoltre, questa procedura rende più rapido l'assemblaggio delle matrici del problema IGA-BEM, che risulta di norma particolarmente oneroso nel caso di questi metodi.

Nel **capitolo 4** vengono presentati due metodi a basi ridotte per la soluzione di problemi differenziali parametrizzati, basati su due diversi approcci per la costruzione di uno spazio di basi ridotte. Nel primo caso consideriamo un algoritmo basato sulla **decomposizione ortogonale** (proper orthogonal decomposition, POD), nel secondo invece un algoritmo di tipo **greedy**. In entrambi i casi è possibile costruire una base a partire da un insieme di soluzioni del problema BEM calcolate per opportuni valori dei parametri. Nel primo caso occorre calcolare un vasto numero di tali soluzioni, e operare una decomposizione ai valori singolari della matrice delle soluzioni. Qualora i valori singolari evidenzino un decadimento esponenziale, trattendendo pochi vettori singolari della decomposizione è possibile ottenere in modo immediato una base ridotta per il problema. Nel secondo caso è possibile selezionare, in maniera *adattiva* e *ottimale*, alcuni valori dei parametri, e calcolare solo le soluzioni corrispondenti a questi valori. Per operare tale scelta occorre tuttavia ricorrere a un opportuno stimatore (a posteriori) dell'errore, la cui valutazione potrebbe risultare onerosa rispetto al calcolo delle soluzioni richieste per la POD, in questo caso poco costosa dal momento che il problema IGA-BEM non ha dimensioni elevatissime.

I risultati ottenuti con il metodo BEM, già validati da confronti con dati sperimentali e dati provenienti da altri algoritmi, sono infine confrontati con quelli provenienti dai metodi a basi ridotte nel **capitolo 5**, mostrando un'ottima accuratezza anche in questo caso. Di conseguenza, riusciamo a certificare la bontà di tali metodi non solo nei confronti del metodo IGA-BEM, ma anche con dati sperimentali.

Il lavoro svolto ha dimostrato che l'accoppiamento di un modello isogeometrico agli elementi di contorno con un metodo di riduzione del modello porta a ottimi risultati in termini di accuratezza e velocità di soluzione, nel caso della soluzione di flussi a potenziale intorno a profili alari.

Una naturale evoluzione di quanto presentato in questo elaborato è l'estensione al caso di problemi in tre dimensioni, considerando eventualmente un modello fisico più complesso, che possa tenere conto della presenza dello strato limite e riuscire a prevederne il comportamento. Un'altra possibilità, infine, è rappresentata dallo studio di problemi non stazionari a dall'estensione a questo caso della metodologia IGA-BE-RBM considerata in questo lavoro.

Introduction

In this Master Thesis we analyse and implement a **reduced basis method** 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. This method allow us to capture the essential features of our system, described by a parametrized differential model, by improving computational performances and by keeping the approximation error between the reduced order solution and the full order (or high-fidelity) one under control.

The general idea behind reduced order models (ROMs) is to solve the problem combining the solutions of the full-order (or high-fidelity) problem for some properly selected values of the parameters. In fact, we assume that the behaviour of a system can be well described by a small number of dominant modes. This assumption usually holds in several real world applications. Under this assumption, it is possible to split the numerical approximation in two stages. We first solve the full-order problem only for some instances of parameter values, through a computationally demanding *Offline* stage, in order to construct a *reduced space* of *basis* solutions. In this way, it is possible to perform many low-cost, reduced-order simulations during a very inexpensive *Online* stage for new instances of the parameter values. In fact, we express the reduced solution as a linear combination of the basis solutions and compute it through a Galerkin projection onto this reduced space.

For the numerical approximation of the problem and the construction of the reduced order model basis functions, we use an **isogeometric boundary element method** (IGA-BEM) based on a **boundary integral equation**. This allows us to discretize and solve the problem only on the boundary of the domain at hand, decreasing by one the dimensionality of the problem, and thus leading to a further reduction of the computational complexity.

We highlight that reduced basis methods do not replace the boundary element method. Rather, they build upon, and are measured against (regarding accuracy), a given high-fidelity approximation method: the reduced basis solution does not approximate directly the exact solution, but rather a ‘given’ boundary element solution.

Several novel aspects are proposed in this work. In particular:

i. we have implemented an **efficient** isogeometric boundary element method;
ii. through the empirical interpolation method, we have reduced the computational cost related to the assembling of the discrete model matrices;
iii. we have built a reduced-order model by exploiting a propero orthogonal decomposition (POD) strategy and a greedy algorithm for the basis selection, thus yielding a IGA-BEM method.
iv. finally, we have validated the results obtained through our ROMs with experimental data coming from literature, showing a great agreement and a remarkable computational reduction.

State of the art

This work is based on the coupling of different techniques in an innovative way. Thus, we now provide a brief state of the art of all the main ingredients we deal with, namely panel/boundary element methods, geometry description and parametrization and reduced order models.

Panel/boundary element method

Panel methods have been widely accepted as a useful tool for aerodynamic and hydrodynamic design since pioneering work of Hess and Smith (1962) [25]. A large number of different panel methods have been developed for a variety of applications (Hess 1975 [24]). Until Morino (1974) [46] introduced a panel method based on Green functions in which the primary unknown is the potential, most of the previous works were based on a velocity-based formulation, in which the boundary condition on the body surface is satisfied through the direct computation of the velocity. The Morino potential-based formulation is known to be more stable, and hence more suitable for numerical computation than the velocity method, since the potential is one order less singular than the velocity. A good discussion on the potential-based panel method may be found in Kerwin et al. (1987) [34]. The low-order panel method assumes that the potential is constant over a panel, and hence, to get the velocity distribution on the body surface, this method requires a finite difference scheme, which inevitably introduces numerical differentiation error. This error is most significant near the trailing edge and at the tip of the lift-generating surface, and leads ultimately to the degradation of the accuracy of the low-order method. In [41] (2003) and [35] (2007) Lee et al. developed a higher-order panel method, which allows to improve the prediction of the velocity and pressure in these regions, by employing B-spline basis functions to represent both the geometry and the potential. 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, and so that the inherent limit of the low-order panel method can be resolved.

This is one of the reasons why we have decided to adopt an isogeometric approach to develop our framework. A detailed description of the higher-order panel method based on B-splines was first given by Hsin et al. (1994) [29] and by Maniar (1995) [43] for the two dimensional and three dimensional case respectively.

Geometrical description and parametrization

The isogeometric analysis (IGA) concept for the discretization of partial differential equations (PDEs), introduced by Hughes et al. in [30], was developed for the integration between finite element analysis (FEA) and conventional computer aided design (CAD) tools. The most attractive feature of IGA is its ability to maintain the same exact description of the computational domain geometry throughout the analysis process for the PDE solution space, by using the same class of functions used for geometry parameterization in CAD. IGA is often seen as a generalization of standard FEMs, allowing to employ more regular functions are employed. The additional regularity leads to other advantages with respect to FEA, such as better convergence properties and the ability to treat higher order problems (see, for example, the book by Cottrell et al.. [15] for a comprehensive list of references on the argument).

Nevertheless, while there is a huge amount of literature related to the finite element isogeometric analysis (FE-IGA), only few works deal with boundary element methods (BEM) and boundary integral equations (BIE). Isogeometric boundary element method (IGA-BEM) is very attractive for the solution of homogeneous elliptic PDEs since it requires the solution of integral equations only on the boundary of the domain, which is typically the only information provided by standard CAD tools. In two dimensional problems, there are some works on potential flows, such as [48], [41] and [59]. Three dimensional applications are of great interest in the maritime community, and there are some works based on panel methods to study marine propellers [35] or the wavemaking resistance problem [8].

Furthermore, isogeometric analysis is also attractive for the study of problems dependent on geometric parameters. In fact, we can change the geometry with not much effort by changing the position of some control points. Among the shape parametrization techniques we recall the Free-Form Deformation (FFD), which is based on tensor products of splines and gives a global non affine transformation map [44]. Another well-known technique is related with Radial Basis Functions (RBF), which is a general paradigm for interpolation of scattered data in high dimensions.

Here, we propose a different approach based on B-spline functions and IGA, which allows us to change all the control points position simply by two parameters, thanks to a least square procedure. This feature makes our shape parametrization technique quite different from other geometrical maps, which typically require more parameters to treat a class of shapes/deformations of comparable complexity. In our specific case, IGA allows us to reconstruct profiles within the NACA 4-digits series in an exact fashion, thus enabling to compare our computational results with experimental data, tipically available for several airfoils of the NACA family .

Reduced order models

Numerical methods for Computational Fluid Dynamics are by now essential in engineering applications dealing with flow simulation and control. Despite the constant increase in available computational power, some problems and/or applications can still be very demanding. This effort is even more substantial whenever we are interested in the repeated solution of the fluid equations for different values of model parameters, such as in flow control or optimal design problems (*many-query contexts*), or in *real-time* flow visualization and output evaluation. These problems represent a remarkable challenge to classical numerical approximations techniques. These methods require huge computational efforts, thus making both *many-query* and *real-time* simulations unaffordable. For this reason, we need to rely on suitable *Reduced-Order Models* (ROMs) – that can reduce both the amount of CPU time and storage capacity – in order to enhance the computational efficiency.

During the last three decades, several efforts in theoretical foundation, numerical investigation and methodological improvements of reduced order models have allowed to tackle several problems arising in fluid dynamics. In fact, in the 1980s the reduction strategies were mainly based on ad hoc choices of the basis functions, without the benefit of a formal algorithm. Recent years have seen considerable progress in this field, with several classes of methods emerging. In [9], Benner et al. give a general overview on reduction methods. In this work, we limit ourselves to describe and use two (indeed, very popular) methods for choosing the basis on which to build the reduced order models, namely the *Proper Orthogonal Decomposition* (POD) and the *(greedy) reduced basis* (RB) methods. They have been historically introduced and developed to address different kind of problems: POD has been typically applied to time-dependent problems, whereas greedy RB to parameter dependent problems.

In both cases, the main idea is that the solution of a problem can be obtained by a linear combination of well-chosen solutions for specific choices of the parameters. In particular POD techniques reduce the dimensionality of a system by transforming the original unknowns into new variables (called POD modes or principal components) such that the first few modes retain most of the energy present in the system. POD was introduced in the context of turbulence by Lumley; focusing on fluid dynamics application of POD, we recall the works of Ravindran [52] for optimal control, of Kunish et al. [36] and of Bui-Thanh et al. [13] for blade optimization.

On the other hand, the initial ideas related to parametrized problems grew out of two related research topics dealing with linear/nonlinear structural analysis in the late 70's. In the next decade, different applications, such as incompressible Navier-Stokes equations, have been tackled. As already mentioned, the choice of the solutions was not optimal and adaptive. Finally, in the last decade, much effort has been devoted to the development of *a posteriori* error estimation procedures, in particular rigorous error bounds, and effective sampling strategies ([61]) such as the so-called greedy algorithms. Also an *a priori* theory for RB approximation is available, dealing with a class of single parameter coercive problems [42] and more recently extended also to the multi-parameter case [11]. Thus, if early work on the RB method did not fully exploit the Online-Offline procedure, much work has been devoted to the efficient splitting of this two steps.

Most of the RBM applications deal with physical or engineering parameters, such as viscosity, transport velocity, Peclet number, Biot number, Young modulus or thermal conductivity. There are a few applications dealing also with simple geometrical parameter, such as a length or a thickness that characterize the problem ([50]). The application to complex geometry parametrization is, hence, quite innovative and very recent [**?**]. This is the case of the solution (here through an isogeometric approach) of the problem of potential flows about parametrized airfoils. In [22] Günther has employed a method RBM for the shape optimization of racing car components, where the parameters are only the angle of attack and the thickness of symmetric airfoils. In [53] Rozza treats potential flows but he does not tackle the lifting problem. We highlight that in this work we present for the first time ever the coupling of isogeometric analysis with a reduced basis method, and the possibility to recover solutions of physical meaning.

Furthermore, historically RB methods have been built upon finite element discretization. Only a few applications of RB methods have been developed for boundary element methods. We recall the work of Fares et al. [18], Ganesh et al. [19] and Hesthaven et al. [26] for the electric field integral equation. In these works, the parametric dependence is always related to physical parameters. In our work, for the first time ever, we provide an application of ROM to BEM with (complex) geometric parameter dependence.

Thesis structure

The work has been organised as follows.

In **chapter 1** we introduce the physical model. Under proper assumptions, the flow about an airfoil can be described by a Poisson problem with mixed boundary conditions and a suitable Kutta condition on the trailing edge of the airfoil, in order to ensure the well posedness of the problem in two dimensional domains. Exploiting its integral formulation, we can solve the problem only on the boundary of the domain (boundary integral equation). Moreover, we provide a variational formulation of this problem, which is necessary to build a numerical approximation method based on a **Galerkin projection**.

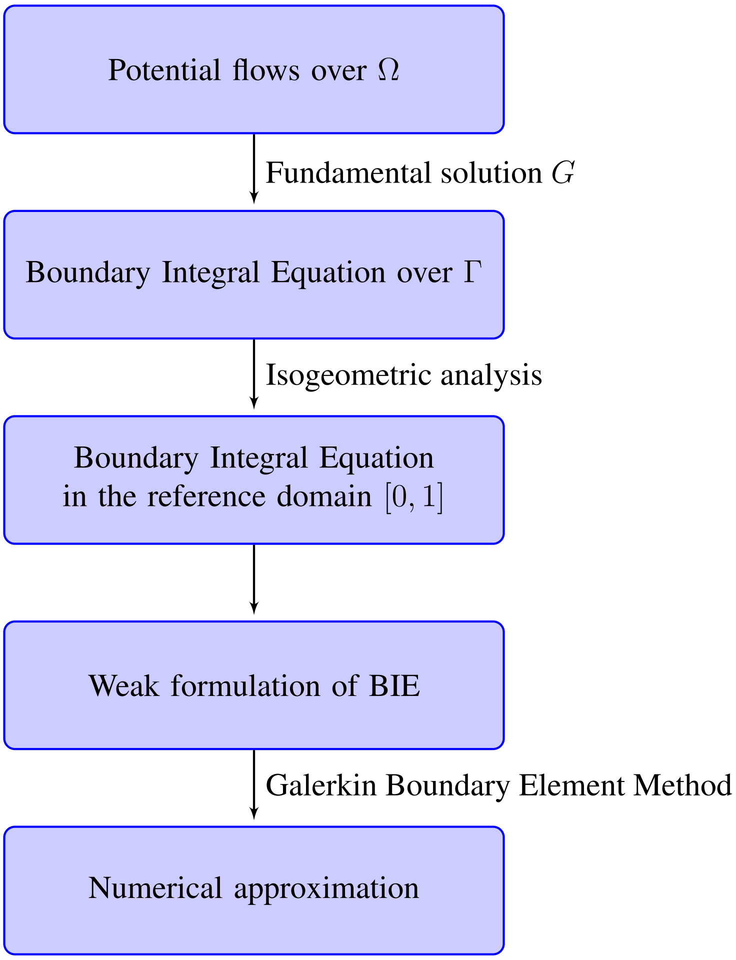

Figure 1: Block diagram of chapter 1 and 2.

This method is described in **chapter 2**. In particular, we consider an isogeometric boundary element method (IGA-BEM). This technique differs from the classical ones (such as those based on the distribution of vortices on the panels) both in terms of geometry approximation and of the discretization procedure. In fact, we use the same basis functions to describe the geometry and the solution; this enhances the accuracy of the numerical method compared to computational cost. In figure 1 we show the block diagram that summarizes the steps carried out in the first two chapters.

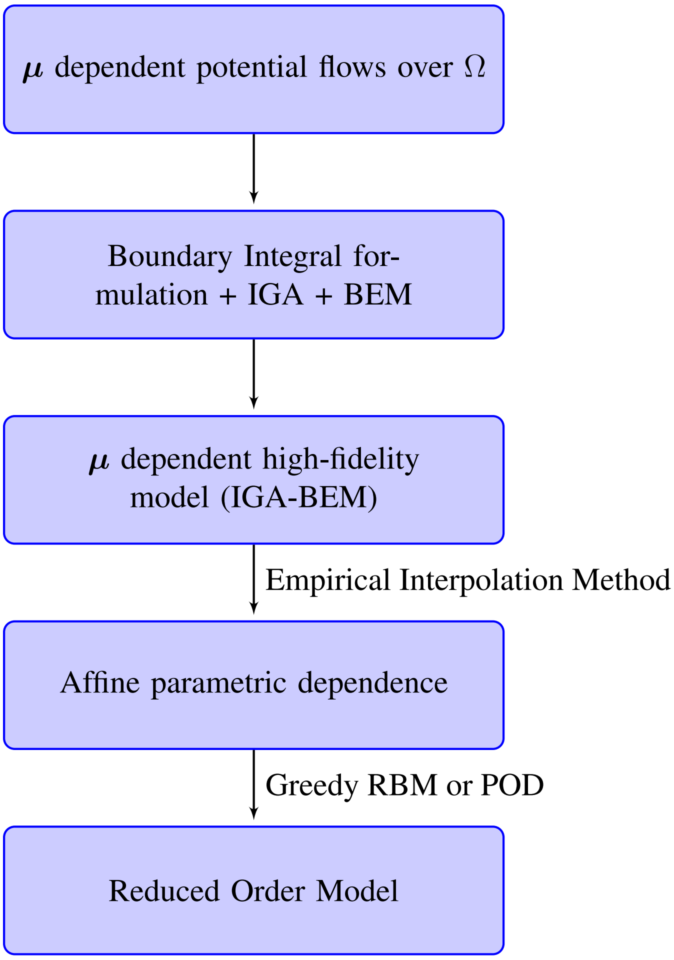

Figure 2: Block diagram of chapter 3 and 4.

In chapters 3 and 4 we derive the parametrized problem and its reduction, following the steps shown in figure 2. Since we are interested in the description of potential flows about an airfoil by considering different shapes and angles of attack, we introduce in **chapter 3** a parametrized formulation of the problem. Thus, we can describe physical and geometrical features of the problem through parameters in the differential problem (figure 3). Thanks to the isogeometric description, we can reformulate the problem on a reference domain $[0, 1]$ where differential operators depend on input parameters through suitable parametrized coefficients. This is necessary for the construction of the reduced-order model, in order to combine solutions of the problem correspondent to different geometry configurations. However, as it often happens for geometrical parametrization, the parametric dependence in the differential problem is non affine, that is, the problem coefficients depend not only on the parameters, but also on the spatial coordinates. Thus, it is not immediate to split parameters from differential operators of the problem. To overcome this problem, and to build an efficient ROM for the solution of the parametrized problem, we consider an empirical interpolation technique, in order to approximate the parametrized operators through a linear combination of parameter independent operators. In this a way, we can extract the parameter dependence from the integrals defining these operators. This operation plays a key role in order to exploit the Offline-Online stratagem and minimize the marginal cost associated with each Online evaluation. Moreover, we highlight that the empirical interpolation method itself reduces the computational cost associated with the assembling of BEM matrices, which is normally very expensive.

In **chapter 4**, we present two different ROMs for the solution of parametrized differential problems, based on two different approaches for the construction of the reduced basis space. First, we consider the **Proper Orthogonal Decomposition** (POD), then a **greedy RB method** (RBM). In both cases, it is possible to build a basis from some *snapshots* of the high-fidelity model, chosen for properly selected parameters values. In the first case it is necessary to compute a wide number of snapshots, and then apply a Singular Value Decomposition of the snapshots matrix. If the singular values show an exponential decay, a few singular vectors immediately provide a reduced basis for the problem at hand. In the second case it is possible to select, in an *adaptive* and *optimal* way, the parameter values for the construction of the basis, and compute only the snapshots correspondent to these values. To carry out this selection in an efficient way, we need to rely on a suitable a posteriori error estimator.

Finally, in **chapter 5**, we compare the results obtained with our ROMs to the ones obtained through the high-fidelity BEM, validated with experimental data and other computational tools. In particular, we show that the reduced order models are reliable not only with respect to high-fidelity model, but also with respect to experimental data.

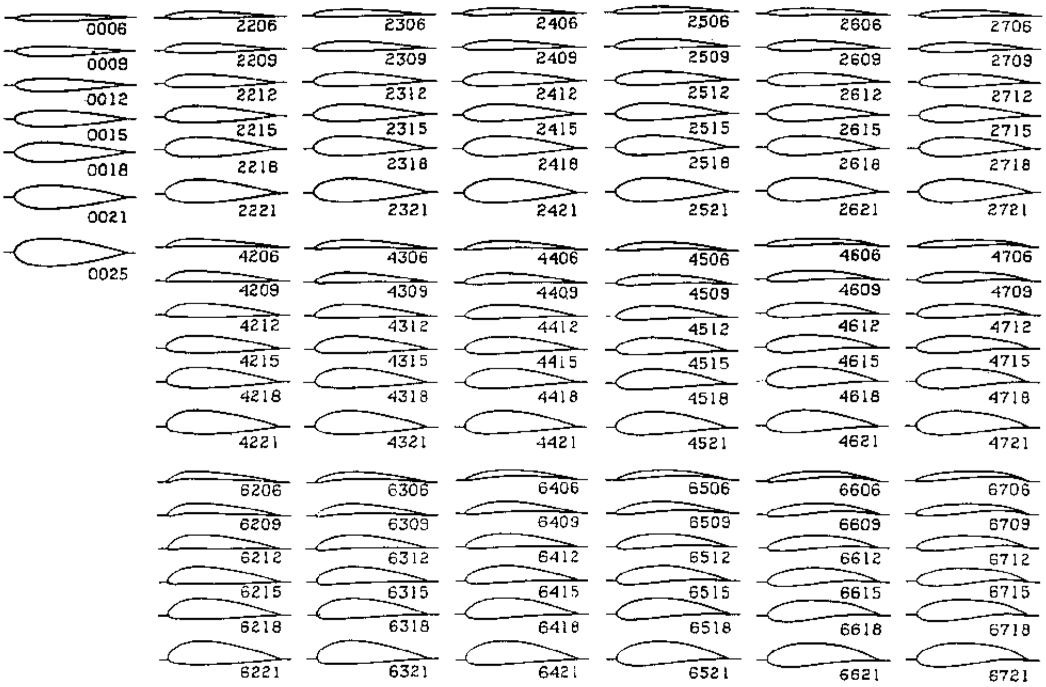

Figure 3: Examples of parametrized airfoil of the NACA 4-digits family.

The work carried out has shown that coupling an isogeometric boundary element method with a reduced order method for potential flows about an airfoil problem gives great results in term of accuracy and solution velocity.

A natural evolution of this work is the extension of the proposed framework to three dimensional problems, possibly with a more complex physical model, that should handle the presence of boundary layers. Another possibility is given by the study of unsteady problems, which would entail the combination of POD and RBM for the sake of time-parameter sampling in the reduced space construction [50].

Chapter 1

Boundary integral formulation of potential flows

In this chapter we derive a model for the description of potential flows about an airfoil and a formulation of this problem through a suitable boundary integral equation (BIE). First, we briefly show how under certain flow conditions, Navier-Stokes equations simplify to Laplace equation for a potential field. We introduce suitable boundary conditions and a Kutta condition on the trailing edge of the airfoil, in order to ensure the well posedness of the problem in two dimensional domains. Then, we derive a boundary integral equation to express the perturbation potential. Finally, we show how to rewrite the boundary integral equation in weak form, in view of using Galerkin method for the numerical approximation. The aim of this procedure is to simplify the description of potential flows: on the one hand, the reduction to a boundary integral formulation has the advantage of diminishing the dimensionality of the problem by one; on the other hand, it yields a correct treatment of problems in infinite domains. Finally, we write the BIE in a reference domain and we introduce its variational formulation, in view of the use, for the numerical solution, of an isogeometric approach based on a Galerkin projection.

1.1 Basic notation and governing equations

We now provide a brief derivation of a potential model for the description of a flow about airfoils. In order to describe the motion of a incompressible viscous Newtonian fluid in a spatial domain $\Omega \subseteq \mathbb{R}^2$, we should solve the *incompressible Navier-Stokes equations*

$$\begin{aligned} &\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V} \cdot \nabla)\boldsymbol{V} + \frac{\nabla p}{\rho} - \nu \nabla^2 \boldsymbol{V} = \boldsymbol{g} \\ &\nabla \cdot \boldsymbol{V} = 0 \end{aligned} \tag{1.1}$$

where $\boldsymbol{V}$ and p denote the fluid velocity and pressure respectively, ρ the fluid density, $\boldsymbol{g}$ the external forces (for unit of mass) and ν the kinematic viscosity. The first equation of (1.1) is the *momentum balance equation* whereas the second one expresses the as *mass conservation*, which, for incompressible flows, translates into the well-known incompressibility condition. Usually, it is very expensive to solve Navier-Stokes equations. Thus, we aim at simplifying the problem, by using a less complex model.

By neglecting the viscous term $-\nu\nabla^2\boldsymbol{V}$ in (1.1), we obtain the so-called *Euler equations*

$$\begin{aligned}&\frac{\partial \boldsymbol{V}}{\partial t}+(\boldsymbol{V}\cdot\nabla)\boldsymbol{V}+\frac{\nabla p}{\rho}=\boldsymbol{g}\\&\nabla\cdot\boldsymbol{V}=0.\end{aligned}\tag{1.2}$$

We remark that, by neglecting viscosity, the model can capture inviscid features, as lift, but not viscous effects, like turbulence and boundary layers. Moreover, we further assume that the fluid motion is irrotational, so that $\psi\equiv\nabla\times\boldsymbol{V}=0$. This ensures the existence of a scalar function Φ such that $\boldsymbol{V}=\nabla\Phi$. Φ is known as *(kinetic) potential* or simply *potential*. From (1.2) we can write the following system

$$\begin{aligned}&\frac{\partial \Phi}{\partial t}+\frac{1}{2}|\nabla\Phi|^2+\frac{p}{\rho}+\chi=C(t)\\&\nabla^2\Phi=0,\end{aligned}\tag{1.3}$$

where $\boldsymbol{g}\equiv-\nabla\chi$, χ is the external (conservative) force potential and $C(t)$ an arbitrary function not depending on space. The first equation of (1.3) is the so-called *Bernoulli equation*, whereas the second one is the *Laplace equation*. Moreover, we want to focus on steady flows, and to neglect body forces. In fact, in several aerodynamics problems $\boldsymbol{g}$ is the gravity force, and it does not affect the solution of the problem. System (1.3) thus becomes

$$\begin{aligned}&\frac{1}{2}|\nabla\Phi|^2+\frac{p}{\rho}=C\\&\nabla^2\Phi=0.\end{aligned}\tag{1.4}$$

Under the assumption of irrotational motion, we find the potential Φ from the incompressibility condition, that is now represented by Laplace equation. Once the potential has been determined, we can find the pressure from the first equation of (1.4). Since (1.4) is a linear problem, we can express the potential as

$$\Phi=\phi_\infty+\phi=\boldsymbol{V}_\infty\cdot\boldsymbol{x}+\phi,\tag{1.5}$$

where $\boldsymbol{V}_\infty$ is the inflow velocity of the fluid, ϕ is denoted as *perturbation potential* and $\boldsymbol{x}$ is the vector of the spatial coordinates. In the following we will solve the perturbation potential problem; in fact, exploiting the linearity of Laplace operator ∇^2, we have that

$$\nabla^2\Phi=0\leftrightarrow\nabla^2\phi=0.\tag{1.6}$$

Once we have obtained the perturbation potential ϕ, we can easily recover the full potential Φ. For the complete treatment of this derivation, we refer to [7].

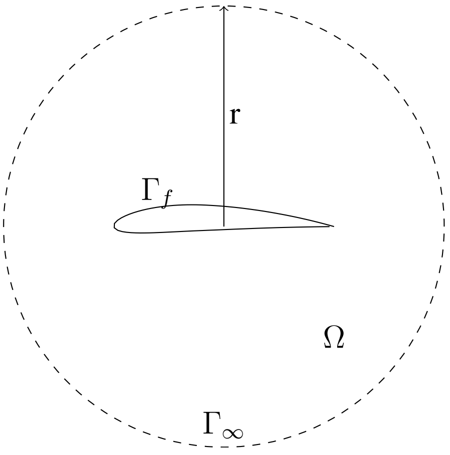

Figure 1.1: Original domain Ω.

1.1.1 Boundary, wake and Kutta conditions

Our aim is to describe potential flows about an airfoil, exploiting Laplace equation (1.6), in the domain Ω. We denote $\Omega \subset \mathbb{R}^2$ as a planar region surrounding the airfoil delimited by the outer boundary Γ_∞ and the airfoil boundary Γ_f (figure 1.1), and by $\partial\Omega$ the boundary of Ω.

In order to solve the problem, we need to introduce a set of suitable boundary conditions. Far from the airfoil, we want the perturbation potential to be zero. Therefore, we impose a homogeneous Dirichlet boundary condition on Γ_∞, that is,

$$\lim_{|r|\to\infty} u(r) = 0 \quad \text{on } \Gamma_\infty. \tag{1.7}$$

On the other hand, on the airfoil Γ_f, we impose a non penetration condition

$$\nabla\Phi \cdot \boldsymbol{n} = 0 \quad \text{that is} \quad \boldsymbol{V}_\infty \cdot \boldsymbol{n} + \nabla\phi \cdot \boldsymbol{n} = 0; \tag{1.8}$$

hence, we have

$$\frac{\partial\phi}{\partial n} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n} \quad \text{on } \Gamma_f. \tag{1.9}$$

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

$$\begin{array}{lll} -\nabla^2\phi = 0 & \text{in} & \Omega \\ \dfrac{\partial\phi}{\partial n} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n} & \text{on} & \Gamma_f \\ \lim_{|r|\to\infty}\phi(r) = 0 & \text{on} & \Gamma_\infty. \end{array} \tag{1.10}$$

The solution of (1.10) has always null circulation, which is not physical. From the mathematical standpoint, in order to overcome this problem, we need to introduce a cut

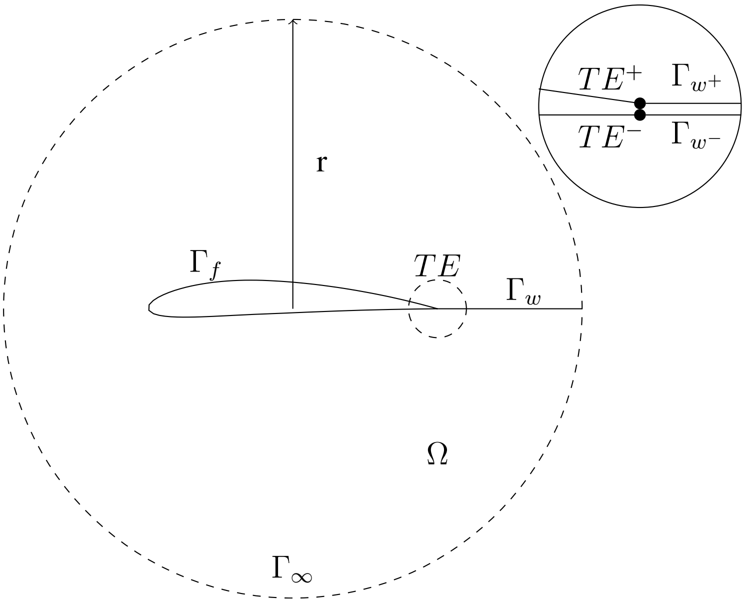

Figure 1.2: Domain with the wake.

Γ_w in the domain (figure 1.2). This operation makes the domain Ω simply connected and allows the potential ϕ to be discontinuous when crossing the cut. This cut is nothing but the well-known wake of the airfoil. 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 (that is, a vortex sheet) detaching from the trailing edge. Here the vorticity is released into the fluid as a jump in the potential ϕ. Thus, the flow is almost everywhere irrotational, except on the wake.

Hence, let us define:

- Γ_{w^+} and Γ_{w^-} as

$$\Gamma_{w^+} := \Gamma_w + \frac{\epsilon}{2}\boldsymbol{n} \quad \text{and} \quad \Gamma_{w^-} := \Gamma_w - \frac{\epsilon}{2}\boldsymbol{n}, \tag{1.11}$$

  such that

$$\lim_{\epsilon\to 0}\Gamma_{w^+} = \lim_{\epsilon\to 0}\Gamma_{w^-} = \Gamma_w, \tag{1.12}$$

  where $\boldsymbol{n}$ is the versor normal to Γ_w;

- TE^+ and TE^- as

$$TE^+ := TE + \frac{\epsilon}{2}\boldsymbol{n} \quad \text{and} \quad TE^- := TE - \frac{\epsilon}{2}\boldsymbol{n}, \tag{1.13}$$

  such that

$$\lim_{\epsilon\to 0}TE^+ = \lim_{\epsilon\to 0}TE^- = TE. \tag{1.14}$$

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

$$\begin{gathered}\phi(TE^+) \neq \phi(TE^-) \\ \phi(\Gamma_{w^+}) \neq \phi(\Gamma_{w^-}).\end{gathered} \tag{1.15}$$

In particular, since we deal with a steady flow, on the wake we simply impose the potential jump to be equal to the one at the trailing edge, that is,

$$[\phi] = \phi(TE^+) - \phi(TE^-) \quad \text{on } \Gamma_w, \tag{1.16}$$

where, form now on, we will use $[\cdot]$ to express the jump operator, defined as the difference between the quantity $\cdot$ on Γ_{w^+} and Γ_{w^-}. We denote equation (1.16) as the *wake condition*, which represents the equation to be solved on the wake. We highlight that on the wake we consider only the potential jump and not the potential itself. In section 1.2 we will show how we end up with a problem where the unknowns are the potential ϕ and the potential jump $[\phi]$ on Γ_f and Γ_w, respectively.

On the wake, we impose a Neumann boundary condition for the potential jump. In fact, we impose that there is no mass accumulation in the wake, that is,

$$[\nabla\phi \cdot \boldsymbol{n}] = \left[\frac{\partial\phi}{\partial n}\right] = 0 \quad \text{on } \Gamma_w. \tag{1.17}$$

The problem we have derived so far thus reads: given $\boldsymbol{V}_\infty$, find ϕ such that

$$\begin{array}{lll} -\nabla^2\phi = 0 & \text{in} & \Omega \\ [\phi] - \phi(TE^+) + \phi(TE^-) = 0 & \text{on} & \Gamma_w \\ \dfrac{\partial\phi}{\partial n} = -\boldsymbol{V}_\infty \cdot \boldsymbol{n} & \text{on} & \Gamma_f \\ \left[\dfrac{\partial\phi}{\partial n}\right] = 0 & \text{on} & \Gamma_w \\ \lim_{|r|\to\infty}\phi(r) = 0 & \text{on} & \Gamma_\infty. \end{array} \tag{1.18}$$

Problem (1.18) leads to in an infinite number of solutions, differing by a solution ϕ_0, satisfying the same Neumann boundary conditions, but showing a non null circulation around the airfoil (see [33] for further details). Only one of them is physically plausible. A possible way to overcome this inconvenient, and to select the physical solution of the potential ϕ solving (1.18), is to introduce the well-known *Kutta condition* (see e.g. [6] for a detailed review about this condition).

The Kutta condition can be of kinematic or dynamic type. The former enforces the velocity at the trailing edge by requiring that it is zero in the steady flow. Since this requirement is too strong in the numerical sense (and too far from the real flow observations), we shall consider a weaker condition by requiring that the velocity amplitude coincides on the upper and lower sides of the trailing edge, that is,

$$V(TE^+) = V(TE^-). \tag{1.19}$$

Equivalently, we may ask that

$$(\boldsymbol{\tau}\cdot(\boldsymbol{V}_\infty+\nabla_s\phi))^+=-(\boldsymbol{\tau}\cdot(\boldsymbol{V}_\infty+\nabla_s\phi))^-,\tag{1.20}$$

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

$$\nabla_s\phi:=\nabla\phi-(\boldsymbol{n}\cdot\nabla\phi)\boldsymbol{n},\tag{1.21}$$

where $\boldsymbol{n}$ is the versor normal to the curve, $I-(\boldsymbol{n}\otimes\boldsymbol{n})$ is the projection operator and I is the identity operator. We can see condition (1.19) as a constraint ensuring that the pressure jump on trailing edge is zero. This is nothing but the so-called dynamic Kutta condition.

Finally, we can summarize the problem we have derived so far as follow: given $\boldsymbol{V}_\infty$, find ϕ such that

$$\begin{array}{lll}
-\nabla^2\phi=0 & \text{in} & \Omega\\
[\phi]-\phi(TE^+)+\phi(TE^-)=0 & \text{on} & \Gamma_w\\
\frac{\partial\phi}{\partial n}=-\boldsymbol{V}_\infty\cdot\boldsymbol{n} & \text{on} & \Gamma_f\\
\left[\frac{\partial\phi}{\partial n}\right]=0 & \text{on} & \Gamma_w\\
\lim_{|r|\to\infty}\phi(r)=0 & \text{on} & \Gamma_\infty,
\end{array}\tag{1.22}$$

subject to the constraint

$$V_{TE^+}-V_{TE^-}=0\quad \text{on}\,TE.\tag{1.23}$$

Problem (1.22) is well posed because of the presence of the Kutta condition, which ensures the uniqueness of the solution. Further details on the formulation and the well posedness of problem (1.22) can be found e.g. in [28].

1.2 Boundary integral formulation of Laplace equation

In order to use a boundary element method for the numerical solution of (1.22), in this section we introduce a boundary integral formulation of the Laplace equation. We remark that this strategy allows to decrease the dimensionality of the problem by one, as well as to treat problems set in infinite domains like in our case.

This procedure will lead to an integral problem, whose solution gives the potential ϕ on the boundary Γ_f. On Γ_w we compute the solution by (1.16); once we have computed the distribution of the potential on Γ, defined as $\Gamma=\Gamma_f\cup\Gamma_w$, we are able to recover the velocity all over the domain Ω and to compute some outputs of interest. To this end, we operate a sequence of formal transformations of Laplace equation, by assuming to deal with regular functions.

Let us multiply Laplace equation (that is, the first equation of (1.22)) by a function (so far arbitrary) G, and integrate over the domain:

$$\int_\Omega -\nabla^2\phi G = 0 \quad \forall G. \tag{1.24}$$

By integrating by parts twice in (1.24) and using the divergence theorem, we obtain

$$\int_\Omega \nabla\phi\nabla G - \int_{\partial\Omega} G\frac{\partial\phi}{\partial n_y} = 0 \quad \forall G \tag{1.25}$$

that is,

$$\int_{\partial\Omega} \phi\frac{\partial G}{\partial n_y} - \int_\Omega \phi\nabla^2 G - \int_{\partial\Omega} G\frac{\partial\phi}{\partial n_y} = 0 \quad \forall G, \tag{1.26}$$

where n_y is the normal to $\partial\Omega$ pointing outwards the domain. Since we have assumed that on the boundary Γ_∞ the potential $\phi \to 0$ as well as the flux $\frac{\partial\phi}{\partial n_y} \to 0$ as $r \to \infty$, we can simplify (1.26) and obtain

$$\int_\Omega \phi\nabla^2 G = \int_\Gamma \phi\frac{\partial G}{\partial n_y} - \int_\Gamma G\frac{\partial\phi}{\partial n_y}. \tag{1.27}$$

Let us now choose a particular function $G : \mathbb{R}^2 \to \mathbb{R}$ such that

$$-\nabla^2 G(\boldsymbol{x}-\boldsymbol{y}) = \delta(\boldsymbol{x}-\boldsymbol{y}); \tag{1.28}$$

G is called *Green's function* and represents the *fundamental solution* of the Laplace equation, that is, the potential $\Phi(\boldsymbol{x})$ given a pointwise source $\delta(\boldsymbol{x}-\boldsymbol{y})$ concentrated at $\boldsymbol{y}$. Here $\delta(\boldsymbol{x}-\boldsymbol{y})$ denotes the Dirac delta distribution. In the case of a kinetic potential G is nothing else but the Rankine source [7]. In the two dimensional case, G takes the following form:

$$G(\boldsymbol{x}-\boldsymbol{y}) = -\frac{1}{2\pi} ln|\boldsymbol{x}-\boldsymbol{y}|, \tag{1.29}$$

so that

$$\frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} = \frac{1}{2\pi}\frac{\boldsymbol{x}-\boldsymbol{y}}{|\,\boldsymbol{x}-\boldsymbol{y}\,|^2}\boldsymbol{n}_y. \tag{1.30}$$

Hence, we can rewrite (1.27) as

$$\phi(\boldsymbol{x}) = \int_\Gamma G\frac{\partial\phi}{\partial n_y} - \int_\Gamma \phi\frac{\partial G}{\partial n_y}; \tag{1.31}$$

note that this equation is valid for any $\boldsymbol{x} \in \Omega$. We call the two integrals

$$\int_\Gamma G\frac{\partial\phi}{\partial n_y} \quad \text{and} \quad \int_\Gamma \phi\frac{\partial G}{\partial n_y} \tag{1.32}$$

single layer potential (SLP) and *double layer potential* (DLP), respectively.

In particular, it can be shown that the SLP is continuous as the point $\boldsymbol{x}$ approaches and then crosses Γ. By contrast, as $\boldsymbol{x}$ crosses Γ, the DLP undergoes a jump discontinuity (see e.g. [58]). If we assume that Γ is smooth, when $\boldsymbol{x}$ approaches Γ from outside Ω, we find

$$\lim_{\boldsymbol{x}\to\Gamma^+} \int_\Gamma \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy = \int_\Gamma^{PV} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy - \frac{1}{2}\phi(\boldsymbol{x}), \tag{1.33}$$

whereas when $\boldsymbol{x}$ approaches Γ from inside Ω

$$\lim_{\boldsymbol{x}\to\Gamma^-} \int_\Gamma \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy = \int_\Gamma^{PV} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy + \frac{1}{2}\phi(\boldsymbol{x}), \tag{1.34}$$

where the superscript PV means that we are taking the principal value of the integral [1]. We can now rearrange the terms of (1.31) in the following form

$$\frac{1}{2}\phi(\boldsymbol{x}) = \int_\Gamma G \frac{\partial \phi}{\partial n_y} - \int_\Gamma^{PV} \phi \frac{\partial G}{\partial n_y}. \tag{1.36}$$

Instead, when Γ is not smooth (this is e.g. the case of a corner point like the trailing edge), we must modify (1.36) as follows:

$$\alpha\phi = \int_\Gamma G \frac{\partial \phi}{\partial n_y} - \int_\Gamma^{PV} \phi \frac{\partial G}{\partial n_y}, \tag{1.37}$$

where the quantity α is the fraction of the angle (respectively solid angle), in the two dimensional (respectively three dimensional) case, by which the point $\boldsymbol{x}$ sees the fluid domain Ω. If the boundary is smooth, the angle, in the two dimensional case, is equal to π, and thus we find back $\alpha = \frac{1}{2}$.

For the problem at hand, we can rewrite the SLP and DLP separately. Let us define

$$u := \begin{cases} \phi & \text{on} \quad \Gamma_f \\ [\phi] & \text{on} \quad \Gamma_w \end{cases}, \tag{1.38}$$

and

$$h := \begin{cases} \frac{\partial \phi}{\partial n_y} & \text{on} \quad \Gamma_f \\ \left[\frac{\partial \phi}{\partial n_y}\right] & \text{on} \quad \Gamma_w \end{cases}, \tag{1.39}$$

---

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

$$\int_\Gamma^{PV} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy := \lim_{\epsilon\to 0} \int_{\Gamma\setminus B_\epsilon} \phi(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy, \tag{1.35}$$

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

where $[\phi]$ and $\left[\frac{\partial\phi}{\partial n_y}\right]$ are constant real functions on Γ_w. Exploiting the Neumann boundary conditions expressed by (1.9) and (1.17), we can write

$$h = \begin{cases} -\boldsymbol{V}_\infty \cdot \boldsymbol{n} & \text{on} \quad \Gamma_f \\ 0 & \text{on} \quad \Gamma_w. \end{cases} \tag{1.40}$$

Thus, the two integrals appearing in (1.37) become

$$\begin{aligned} \int_{\partial\Omega} G\frac{\partial\phi}{\partial n_y} &= \int_{\Gamma_f} G\frac{\partial\phi}{\partial n_y} + \int_{\Gamma_{w+}} G\frac{\partial\phi}{\partial n_y} + \int_{\Gamma_{w-}} G\frac{\partial\phi}{\partial n_y} \\ &= \int_{\Gamma_f} G\frac{\partial\phi}{\partial n_y} + \int_{\Gamma_w} G\left[\frac{\partial\phi}{\partial n_y}\right] = \int_\Gamma Gh \end{aligned} \tag{1.41}$$

and

$$\begin{aligned} \int_{\partial\Omega}^{PV} \phi\frac{\partial G}{\partial n_y} &= \int_{\Gamma_f}^{PV} \phi\frac{\partial G}{\partial n_y} + \int_{\Gamma_{w+}}^{PV} \phi\frac{\partial G}{\partial n_y} + \int_{\Gamma_{w-}}^{PV} \phi\frac{\partial G}{\partial n_y} \\ &= \int_{\Gamma_f}^{PV} \phi\frac{\partial G}{\partial n_y} + \int_{\Gamma_w}^{PV} [\phi]\frac{\partial G}{\partial n_y} = \int_\Gamma^{PV} u\frac{\partial G}{\partial n_y}, \end{aligned} \tag{1.42}$$

respectively. If we exploit the boundary condition over Γ_f and Γ_w in (1.37), we can reformulate problem (1.22) as follows: given h, find u such that

$$\begin{aligned} &\alpha(\boldsymbol{x})u(\boldsymbol{x}) + \int_\Gamma^{PV} u(\boldsymbol{y})\frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y}dy = \int_\Gamma G(\boldsymbol{x}-\boldsymbol{y})h(\boldsymbol{y})dy && \text{on}\,\Gamma_f \\ &u(\boldsymbol{x}) - u(TE^+) + u(TE^-) = 0 && \text{on}\,\Gamma_w \end{aligned} \tag{1.43}$$

subject to the following constraint, representing the Kutta condition:

$$\boldsymbol{\tau}(TE^+)\cdot\nabla_s u|_{TE^+} + \boldsymbol{\tau}(TE^-)\cdot\nabla_s u|_{TE^-} = \boldsymbol{V}_\infty\cdot(\boldsymbol{\tau}(TE^+)+\boldsymbol{\tau}(TE^-)) \quad \text{on}\,TE. \tag{1.44}$$

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

Thus, we end up with an integral equation, which we can solve on the boundary Γ_f, an algebraic equation, which can be solved on Γ_w, and a constraint, which has to be satisfied on the trailing edge. For the sake of simplicity, from now on, we will omit the superscript ^{PV} representing the principal value integral in (1.43), so that

$$\int_\Gamma u(\boldsymbol{y})\frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y}dy := \int_\Gamma^{PV} u(\boldsymbol{y})\frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y}dy. \tag{1.45}$$

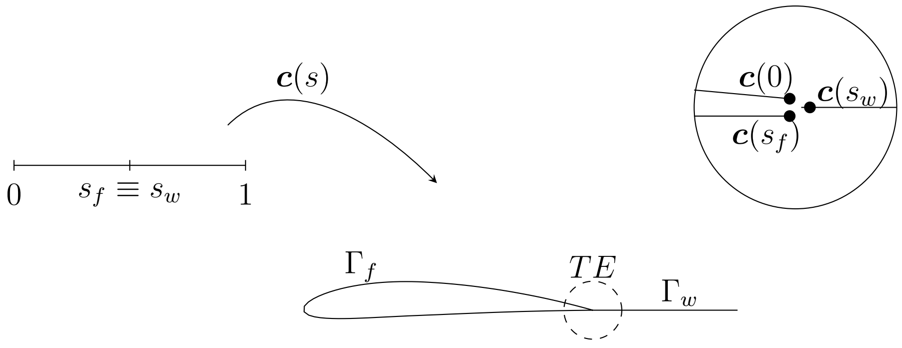

Figure 1.3: Mapping of $[0, 1]$ in Γ.

1.3 BIE in a reference domain

In view of setting an isogeometric boundary element method, it is more convenient to formulate the BIE (1.43) 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. Further details can be found e.g. in [15].

Let us denote by $\boldsymbol{c} : (0, 1) \setminus \{s_f\} \rightarrow \Gamma \subset \mathbb{R}^2$ the map giving the transformation of coordinates (see figure 1.3), with $\boldsymbol{c}' > 0$ (hence invertible), such that

$$\Gamma = \boldsymbol{c}(s), \quad s \in (0, 1) \setminus \{s_f\}. \tag{1.46}$$

In particular we want to map the interval $(0, s_f)$ and $(s_w, 1)$ with $s_f \equiv s_w$ in Γ_f and Γ_w, respectively, that is,

$$\begin{aligned} \Gamma_f &= \{\boldsymbol{c}(s) | s \in (0, s_f)\} \\ \Gamma_w &= \{\boldsymbol{c}(s) | s \in (s_w, 1)\}\,. \end{aligned} \tag{1.47}$$

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

$$\begin{aligned} TE^+ &= \lim_{s \to 0^+} \boldsymbol{c}(s) \\ TE^- &= \lim_{s \to s_f^-} \boldsymbol{c}(s). \end{aligned} \tag{1.48}$$

respectively. From now on, for the sake of readability, we will omit the limit and we will simply write

$$\begin{aligned} \boldsymbol{c}(0) &= \lim_{s \to 0^+} \boldsymbol{c}(s) \\ \boldsymbol{c}(s_f) &= \lim_{s \to s_f^-} \boldsymbol{c}(s) \\ \boldsymbol{c}(s_w) &= \lim_{s \to s_w^+} \boldsymbol{c}(s). \end{aligned} \tag{1.49}$$

In this way, by walking through the airfoil in counter clockwise direction from TE^+ to TE^-, $(0, s_f)$ is mapped onto the airfoil. As already shown in section 1.1, the airfoil and the wake are geometrically coincident at the trailing edge, that is,

$$\boldsymbol{c}(0) \equiv \boldsymbol{c}(s_f) \equiv \boldsymbol{c}(s_w), \tag{1.50}$$

but we allow the solution u at the trailing edge to be discontinuous, that is,

$$u(0) \neq u(s_f) \neq u(s_w). \tag{1.51}$$

Therefore, we can write $\tilde{f} = f \circ \boldsymbol{c}$ such that

$$\tilde{f}(s) = f(\boldsymbol{c}(s)) = f(\boldsymbol{x}) \mid_{\Gamma}; \tag{1.52}$$

thus we have

$$\boldsymbol{x} = \boldsymbol{c}(s); \quad \boldsymbol{y} = \boldsymbol{c}(q), \tag{1.53}$$

where s and q are reference variables. Let us apply now the change of coordinates to the integrals appearing in (1.43): we obtain

$$\begin{aligned} \int_{\Gamma} u(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy &= \int_0^1 u(q) \frac{\partial G}{\partial n_q}(\boldsymbol{c}(s) - \boldsymbol{c}(q)) J(q) dq \\ \int_{\Gamma} G(\boldsymbol{x}-\boldsymbol{y}) h(\boldsymbol{y}) dy &= \int_0^1 G(\boldsymbol{c}(s) - \boldsymbol{c}(q)) h(q) J(q) dq, \end{aligned} \tag{1.54}$$

where J is the determinant of the Jacobian of the transformation.

We can now rewrite problem (1.43) in the following way: given h, find u such that

$$\alpha u(s) + (Nu)(s) = (Dh)(s) \qquad \text{in}(0, s_f) \tag{1.55}$$

$$u(s) - u(0) + u(s_f) = 0 \qquad \text{in}(s_w, 1), \tag{1.56}$$

subject to the constraint

$$\boldsymbol{\tau}(0) \cdot \nabla_s u|_0 + \boldsymbol{\tau}(s_f) \cdot \nabla_s u|_{s_f} = \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0) + \boldsymbol{\tau}(s_f)) \quad \text{on } \{0, s_f\}, \tag{1.57}$$

where the operators N and D are defined as follows:

$$(Nu)(s) \; := \; \int_0^1 u(q) \frac{\partial G}{\partial n_q}(\boldsymbol{c}(s) - \boldsymbol{c}(q)) J(q) dq \tag{1.58}$$

$$(Dh)(s) \; := \; \int_0^1 G(\boldsymbol{c}(s) - \boldsymbol{c}(q)) h(q) J(q) dq, \tag{1.59}$$

In the next section we will use these operators to define the bilinear form of the variational formulation of problem (1.55). Moreover, in the next chapter we will exploit the isogeometric framework to characterize the map $\boldsymbol{c}$, and then to define a parametrized description of the computational domain.

1.4 Weak formulation and well posedness

The reduced order models developed in this work are built over a Galerkin boundary element method (Galerkin-BEM) for the numerical approximation of the boundary integral equation. The Galerkin boundary element method is based on the variational (or weak) formulation of the integral equation. Thus, in view of the numerical approximation, we introduce in this section the weak formulation of (1.55).

Let us introduce the Sobolev space $V := H^{\frac{1}{2}}([0,1]\backslash\{s_f\})$ and its dual V'. Functions in V constant on the wake look like the one depicted in figure 1.4.

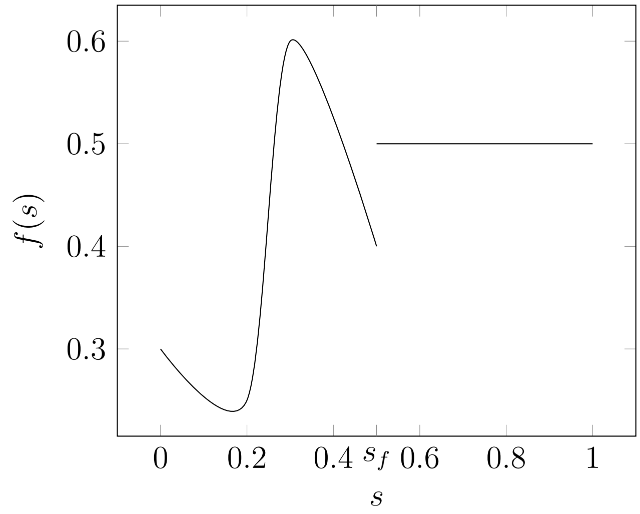

Figure 1.4: General function $f(s) \in V$.

In the variational formulation we condense (1.55) in a single bilinear form. If we introduce the indicator function χ_A defined as

$$\chi_A(s) = \begin{cases} 1 & \text{for } s \in A \\ 0 & \text{otherwise} \end{cases}, \tag{1.60}$$

we can write the variational formulation of (1.55) as follows: given $h \in V'$, find $u \in V$ such that

$$a(u,v) = d(h,v) \quad \forall v \in V \tag{1.61}$$

with the constraint that

$$V_{TE^+} - V_{TE^-} = 0 \quad \text{on } \{0, s_f\}, \tag{1.62}$$

where

$$\begin{aligned} a(u,v) &:= \alpha m(u,v) + n(u,v) \\ &+ w(u,v). \end{aligned} \tag{1.63}$$

The bilinear forms

$$\begin{aligned} m(\cdot,\cdot) &: V \times V \to \mathbb{R} \\ n(\cdot,\cdot) &: V \times V \to \mathbb{R} \\ w(\cdot,\cdot) &: V \times V \to \mathbb{R} \\ d(\cdot,\cdot) &: V' \times V \to \mathbb{R} \end{aligned} \tag{1.64}$$

appearing in (1.63) and (1.61) are given by

$$\begin{aligned} m(u,v) &:= \int_0^1 v(s)u(s)\chi_{[0,s_f]}(s)ds \quad \forall u,v \in V \\ n(u,v) &:= \int_0^1 v(s)(Nu)(s)\chi_{[0,s_f]}(s)ds \quad \forall u,v \in V \\ w(u,v) &:= \int_0^1 v(s)\left[u(s) - u(0) + u(s_f)\right]\chi_{[s_f,1]}(s)ds \quad \forall u,v \in V \\ d(h,v) &:= \int_0^1 v(s)(Dh)(s)\chi_{[0,s_f]}(s)ds \quad \forall v \in V, h \in V'. \end{aligned} \tag{1.65}$$

We can provide an alternative formulation, which includes the Kutta condition as a constraint enforced through a Lagrange multiplier $\pi \in \mathbb{R}$:

$$\begin{cases} a(u,v) \;+\; b(v,\lambda) \;=\; d(h,v) & \forall v \in V \\ b(u,\pi) \;=\; \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0) + \boldsymbol{\tau}(s_f)) & \forall \pi \in \mathbb{R}, \end{cases}' \tag{1.66}$$

where

$$b(u,\pi) = \left[\boldsymbol{\tau}(0) \cdot \nabla_s u|_0 + \boldsymbol{\tau}(s_f) \cdot \nabla_s u|_{s_f}\right]\pi. \tag{1.67}$$

Problem (1.66) is typically referred to as a *saddle point* problem.

In the next chapter, we will exploit (1.65) and (1.61) to derive a Galerkin method for the numerical solution of the problem at hand.

Chapter 2

Numerical approximation: isogeometric boundary element method

In this chapter we present a boundary element method (BEM) for the discretization of potential flows based on the Boundary Integral Equation (1.55) obtained in the previous chapter. In section 2.1 we give a general introduction to isogeometric analysis. In view of setting a parametrized version of this problem, by taking into account shape variations of the airfoil within the 4-digits NACA family, we introduce an isogeometric description of the geometry in section 2.2. We introduce an isogeometric boundary element method (IGA-BEM) based on a Galerkin projection for the numerical approximation of potential flows in section 2.3 by deriving the corresponding discretized formulation of the problem and showing its peculiarities. Finally, starting from the solution on the boundary, we show how to recover the solution (in terms of velocity and pressure) in the whole domain around the airfoil, as well as some physical outputs of interest in section 2.5.

2.1 Isogeometric description with B-splines

In view of taking into account shape variations of the airfoil, we adopt an isogeometric description of the geometry. In fact, isogoemetric analysis allows to easily change the shape of a profile and to deal always with smooth geometries.

In traditional Finite Element Analysis (FEA) we use low-order Lagrange polynomials defined on a suitable triangulation of the domain (or mesh) as basis functions, whereas computer aided geometry modeling is based on techniques like spline functions. As a matter of fact, a model conversion is necessary if we have to analyse a geometry by FEA that has been designed in a computer aided design (CAD) program. The main property of spline functions is their capability to represent the exact geometry; this is one of the main reasons why splines are nowadays widely used in the CAD

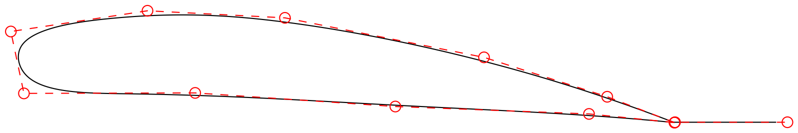

Figure 2.1: Control polygon and B-spline curve for NACA 4412 airfoil geometry; coarse knot vector.

context. For the same reason, IGA provides advantages concerning accuracy compared to computational cost, since it does not introduce mesh errors.

We now characterize the geometrical map introduced in section 1.3 to rewrite the problem over the reference configuration $[0, 1]$. We recall that our goal is to express the boundary Γ as the image of the interval $[0, 1]$ through a map $\boldsymbol{c} : [0, 1] \rightarrow \Gamma \subset \mathbb{R}^2$ (as shown in section 1.3).

We 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* $\{\xi_i(s)\}_{i=1}^{\mathcal{N}}$ defined recursively as follows:

$$\xi_{i,0}(s) = \begin{cases} 1 & \theta_i \leq s \leq \theta_{i+1} \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

where

$$\xi_{i,k}(s) = \frac{s - \theta_i}{\theta_{i+k} - \theta i}\xi_{i,k-1}(s) + \frac{\theta_{i+k+1} - s}{\theta_{i+k+1} - \theta_{i+1}}\xi_{i+1,k-1}(s) \quad k = 1 : p; \tag{2.2}$$

here

- p is the polynomial order of the chosen basis functions;
- $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_{n+p+1}\}$, $\theta_i \in \mathbb{R}$, is called *knot vector*, and it is a non-decreasing set of coordinates in the *s* parameter space.

Then, we can define the map $\boldsymbol{c}$ as follows:

$$\boldsymbol{c}(s) = \sum_{i=1}^{\mathcal{N}} \xi_i(s)\boldsymbol{P}_i, \quad \boldsymbol{P}_i \in \mathbb{R}^2. \tag{2.3}$$

The choice of $\boldsymbol{\theta}$ affects the continuity of the basis function and, as a consequence, the continuity of the B-spline curve. A knot vector is said to be *open* if its first and last knot values appear $p + 1$ times. Basis functions formed from open knot vectors are interpolatory at the ends of the parameter space interval, $[\theta_1, \theta_{n+p+1}]$ but they are not, in general, interpolatory at interior knots.

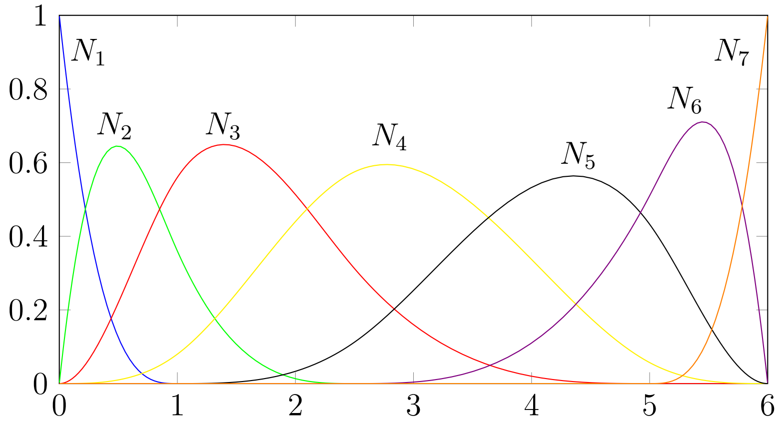

Figure 2.2: The seven cubic B-spline basis functions for the open knot vector $\boldsymbol{\theta} = \{0, 0, 0, 0, 1, 2.5, 5, 6, 6, 6, 6\}$.

Let us recall some peculiarities of the basis functions:

- the basis $\{\xi_i\}_{i=1}^{\mathcal{N}}$ constitutes a partition of unity, that is, $\sum_{i=1}^{\mathcal{N}} \xi_i(s) = 1 \quad \forall s \in [0, 1]$. Each basis function is pointwise nonnegative over the entire domain, that is, $\xi_i(s) \geq 0 \quad \forall s \in [0, 1]$ (see figure 2.2);
- the support of the B-spline functions of order p is always given by $p + 1$ knot spans, that is, B-spline functions have a compact support;
- basis functions of order p have $p - m_i$ continuous derivatives across the knot θ_i, where m_i is the multiplicity of the value of θ_i in the knot vector. The use of a non-uniform knot vector, thus, allows to obtain a richer behaviour of the B-spline than the one obtained from a uniform knot vector;
- the derivatives of B-spline basis functions can be efficiently evaluated in terms of B-spline lower order bases. In this way, we can write:

$$\frac{d^k \xi_{i,p}(s)}{d^k s} = \frac{p!}{(p-k)!} \sum_{j=0}^{k} \alpha_{k,j} \xi_{i+j,p-k}(s), \tag{2.4}$$

with

$$\alpha_{0,0} = 1 \qquad\qquad \alpha_{k,0} = \frac{\alpha_{k-1,0}}{s_{i+p-k+1} - s_i}$$

$$\alpha_{k,j} = \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{s_{i+p+j-k+1} - s_{i+j}} \quad j = 1 : k-1 \qquad \alpha_{k,k} = \frac{-\alpha_{k-1,k-1}}{s_{i+p+1} - s_{i+k}}.$$

In the following we show some peculiarities of the B-spline functions which are relevant to the setting of a geometrical parametrization to deal with a class of shapes, like the NACA family:

- an affine transformation of a B-spline curve is obtained by applying the transformation directly to the control points;
- a B-spline curve will have at least as many continuous derivatives across an element boundary as its basis functions have across the corresponding knot value;
- due to the compact support of the B-spline basis functions, moving a single control point can affect the geometry of no more than $p+1$ elements of the curve.

Moreover, let us now provide the expression of the superficial gradient (1.21) in the isogeometric framework, which will be deeply exploited in the following. Exploiting the properties of B-splines, we can also write

$$\nabla_s u := J^{-1}(s) u'(s) \boldsymbol{\tau}(s) \tag{2.5}$$

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 versor tangent to Γ. We will exploit this formula to express Kutta condition in section 2.3 and the pressure coefficient in section 2.5. A detailed description of B-spline properties can be found e.g. in [15].

2.2 B-splines description of NACA 4-digits profiles

Let us now focus on the description of the airfoil geometry, given by the NACA 4-digits family and its angle of attack α, by taking advantage of the isogeometric framework addressed in the previous section. For the detailed description of NACA airfoils, see e.g. [2].

For this kind of airfoils, the shape can be expressed analytically as a function of the maximum camber, the maximum camber location and the maximum thickness of the airfoil. In fact, the first digit indicates the maximum value of the mean-line ordinate in percentage of the chord, the second integer indicates the distance from the leading edge to the location of the maximum camber in tens of the chord, whereas the last two integers indicate the section thickness in percent of the chord. Thus, for example, 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 figure 2.2.

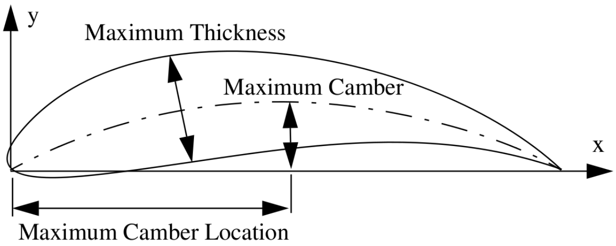

Figure 2.3: 4-digits NACA profiles are expressed as function of their maximum camber, maximum camber location and maximum thickness.

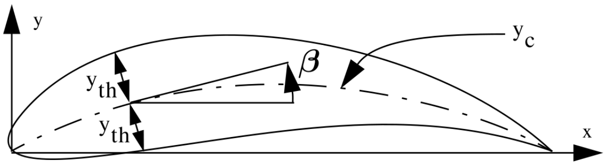

Figure 2.4: Schematic representation of the construction of the airfoil given the analytic expression of its thickness y_{th}, mean line y_c and camber θ.

Hence, by denoting (x_U, y_U) and (x_L, y_L) the points on the upper and lower surface of the airfoil respectively, we can write (see figure 2.2)

$$\begin{cases} x_U = x - y_{th}\text{cos}\beta \\ y_U = y_c + y_{th}\text{sin}\beta \end{cases} \tag{2.6}$$

and

$$\begin{cases} x_L = x + y_{th}\text{cos}\beta \\ y_L = y_c - y_{th}\text{sin}\beta. \end{cases} \tag{2.7}$$

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

$$\begin{aligned} y_{th} &= 5\tau c(0.2969\sqrt{\frac{x}{c}} - 0.126\frac{x}{c} - 0.3537\left(\frac{x}{c}\right)^2 \\ &+ 0.2843\left(\frac{x}{c}\right)^3 - 0.1015\left(\frac{x}{c}\right)^4) \end{aligned} \tag{2.8}$$

$$y_c = \begin{cases} \frac{m}{p^2}\left(2p\frac{x}{c} - \left(\frac{x}{c}\right)^2\right) & \text{for} \quad \frac{x}{c} \leq p \\ \frac{m}{(1-p)^2}\left(1 - 2p + 2p\frac{x}{c} - \left(\frac{x}{c}\right)^2\right) & \text{for} \quad \frac{x}{c} \geq p \end{cases} \tag{2.9}$$

$$\beta = \begin{cases} \text{atan}\left(\frac{2m}{cp^2}\left(p - \frac{x}{c}\right)\right) & \text{for} \quad \frac{x}{c} \leq p \\ \text{atan}\left(\frac{2m}{c(1-p)^2}\left(p - \frac{x^2}{c}\right)\right) & \text{for} \quad \frac{x}{c} \geq p, \end{cases} \tag{2.10}$$

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 a suitable least-square procedure.

Let us denote by $\boldsymbol{\gamma}$ a set of points on the airfoil distributed according to their arc length

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_M \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_M & y_M \end{bmatrix} \quad \text{with } M \geq \mathcal{N} \tag{2.11}$$

and denote by $\boldsymbol{c}(s), s \in [0,1]$ the B-spline description of the airfoil Γ_f. In order to find the control points position by a least-square procedure, we want to find

$$\text{arg min}_{\{\boldsymbol{P}_i\}_{i=1}^{\mathcal{N}}} \sum_{i=1}^{M} \frac{1}{2} \left| \boldsymbol{\gamma}_i - \sum_{j=1}^{\mathcal{N}} \boldsymbol{P}_j \xi_j(s_i) \right|^2 \tag{2.12}$$

where $s_i := \frac{i}{M+1}$ for $i = 1, \cdots, M$. The minimum of the quadratic functional in (2.12) yields to the following first-order optimality condition:

$$\frac{\partial}{\partial \boldsymbol{P}_j} \left(\sum_{i=1}^{M} \frac{1}{2} \left(\boldsymbol{\gamma}_i - \sum_{k=1}^{\mathcal{N}} \boldsymbol{P}_k \xi_k(s_i) \right) \cdot \left(\boldsymbol{\gamma}_i - \sum_{l=1}^{\mathcal{N}} \boldsymbol{P}_l \xi_l(s_i) \right) \right) = 0 \quad \text{for } j = 1, \cdots, \mathcal{N} \tag{2.13}$$

that is

$$\begin{aligned} &\frac{\partial}{\partial \boldsymbol{P}_j} \sum_{i=1}^{M} \left(\left(\sum_{l=1}^{\mathcal{N}} \boldsymbol{P}_l \xi_l(s_i) \right) \cdot \left(\sum_{k=1}^{\mathcal{N}} \boldsymbol{P}_k \xi_k(s_i) \right) - \boldsymbol{\gamma}_i \cdot \left(\sum_{k=1}^{\mathcal{N}} \boldsymbol{P}_k \xi_k(s_i) \right) \right) \\ &\quad = \sum_{i=1}^{M} \sum_{k=1}^{\mathcal{N}} \boldsymbol{P}_k \xi_k(s_i) \xi_j(s_i) - \sum_{i=1}^{M} \boldsymbol{\gamma}_i \xi_j(s_i) = 0 \quad \text{for } j = 1, \cdots, \mathcal{N}, \end{aligned} \tag{2.14}$$

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

$$B_{ij} := \xi_j(s_i), \tag{2.15}$$

equation (2.13) translates to

$$\boldsymbol{B}^T \boldsymbol{B} \boldsymbol{P} = \boldsymbol{B}^T \boldsymbol{\gamma}, \tag{2.16}$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_1 \\ \vdots \\ \boldsymbol{P}_{\mathcal{N}} \end{bmatrix}. \tag{2.17}$$

System (2.16) gives nothing but the so-called normal equations related to the least square minimization (2.12).

Hence, given $\boldsymbol{\gamma}$, we obtain automatically the control points position $\boldsymbol{P}$ (for null angle of attack) by solving (2.16). Exploiting the properties of the B-splines (provided in the previous section), we can easily apply a rotation to the control polygon in order to rotate the airfoil according to its angle of attack and to obtain the airfoil geometry for the problem at hand. The possibility to modify the airfoil control points position simply by changing its NACA 4-digits number and the angle of attack will be broadly exploited in chapter 3 for the construction of a parametrized formulation, in order to describe potential flows in varying domains.

2.3 Isogeometric Galerkin Boundary Element Method

In this section we derive a numerical approximation of the variational formulation (1.61) based on a Galerkin boundary element method. We apply a Galerkin method for its superior properties of stability, consistency and convergence, compared to the alternatives, such as collocation [56]. Basically, we restrict the solution of (1.61) to live in the finite dimensional space $V^h \subset V$, $V^{h'} \subset V'$, so that the problem becomes: given $h_h \in V^{h'}$, find $u_h \in V^h$ such that

$$a(u_h, v_h) = d(h_h, v_h) \quad \forall v_h \in V^h. \tag{2.18}$$

Here we can express

$$u_h(x) = \sum_{j=1}^{\mathcal{N}} \varphi_j(x) u_j, \tag{2.19}$$

where $\varphi_j(x)$ and u_j are called *basis functions* and *degrees of freedom* (DOF), respectively; $\mathcal{N}$ denotes the dimension of V^h. By choosing $v_h(x) = \varphi_i(x)$ for $i = 1 : \mathcal{N}$, (2.18) rewrites as the following linear system:

$$\sum_{j=1}^{\mathcal{N}} a(\varphi_j, \varphi_i) u_j = d(h_h, \varphi_i), \quad i = 1, \cdots, \mathcal{N}. \tag{2.20}$$

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 (this is nothing but the so called *isogeometric concept*). By following this approach, we can express the solution of (2.18) as

$$u_h(s) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) u_i, \tag{2.21}$$

that is, we define the finite dimensional space $V^h \subset V$ as follows:

$$V^h = \text{span}\{\xi_i, i = 1 : \mathcal{N}\}. \tag{2.22}$$

Thanks to the properties of B-splines, if we take $p+1$ times the knot s_f in the knot vector θ, we allow the solution to be discontinuous in s_f. In the same way, the approximate version of the Neumann boundary condition is given by

$$h_h(s) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) h_i. \tag{2.23}$$

Here u_i and h_i are respectively the control values of the potential u and of the Neumann boundary condition h. Moreover, let us denote by N_f and N_w the number of basis functions for the airfoil and the wake, respectively.

2.3.1 Assembling of the linear system

In order to show how to assemble the linear system (2.20), let us recall that (2.18) can be expressed as

$$\alpha m(u_h, v_h) + n(u_h, v_h) + w(u_h, v_h) = d(h_h, v_h) \quad \forall v_h \in V^h. \tag{2.24}$$

where $m(u,v)$, $n(u,v)$, $w(u,v)$ and $d(h,v)$ are defined in (1.65).

Thus, by exploiting (2.21) and (2.23), and the definition of indicator function (1.60), the linear system (2.20) can be written as follows:

$$\widehat{\boldsymbol{A}}\boldsymbol{u} = \widehat{\boldsymbol{f}}, \tag{2.25}$$

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

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{\alpha M} + \boldsymbol{N} \\ \boldsymbol{W} \end{bmatrix}, \quad \widehat{\boldsymbol{f}} = \begin{Bmatrix} \boldsymbol{d} \\ 0 \end{Bmatrix} \tag{2.26}$$

and

$$\begin{aligned}
M_{ij} &= m(\xi_j, \xi_i) = \int_0^1 \xi_i(s)\xi_j(s)ds, \quad \text{for } i = 1 : N_f, j = 1 : \mathcal{N} \\
N_{ij} &= n(\xi_j, \xi_i) = \int_0^1 \xi_i(s)(N\xi_j)(s)ds, \quad \text{for } i = 1 : N_f, j = 1 : \mathcal{N} \\
W_{ij} &= w(\xi_j, \xi_i) = \int_0^1 \xi_i(s)[\xi_j(s) - \xi_j(0) + \xi_j(s_f)]ds, \quad \text{for } i = N_f + 1 : \mathcal{N}, j = 1 : \mathcal{N}
\end{aligned} \tag{2.27}$$

are the elements of the (rectangular) matrices $\boldsymbol{M}^f$, $\boldsymbol{N}$ and $\boldsymbol{W}$, respectively, whereas

$$d_i = d(h_h, \xi_i) = \int_0^1 \xi_i(s) \left(D \sum_{j=1}^{\mathcal{N}} \xi_j(q) h_j \right)(s)ds, \quad \text{for } i = 1 : N_f \tag{2.28}$$

are the components of the vector $\boldsymbol{d}$. Here $\boldsymbol{M}^f$, $\boldsymbol{N}$ and $\boldsymbol{W}$ represent the airfoil mass matrix, the Neumann matrix, and the wake condition matrix, respectively. Note that the indicator function χ_A plays a fundamental role in the splitting of matrix $\widehat{\boldsymbol{A}}$ in two (rectangular) submatrices.

We can proceed to the imposition of the Kutta condition (1.20) at the trailing edge. We recall that Kutta condition can be written as

$$(\boldsymbol{\tau} \cdot (\boldsymbol{V}_\infty + \nabla_s \phi))^+ = -(\boldsymbol{\tau} \cdot (\boldsymbol{V}_\infty + \nabla_s \phi))^-, \tag{2.29}$$

where $\boldsymbol{\tau}$ denotes the unit tangential vector along the surface, pointing in the counterclockwise direction, ∇_s is the superficial gradient operator, $V = |\boldsymbol{V}|$ and the superscript $^+$ and $^-$ refer to TE^+ and TE^-, respectively. As shown in section 2.1, the superficial gradient in the isogeometric framework can be written as (see equation (2.5)),

$$\nabla_s u(s) \cdot \boldsymbol{\tau} = J^{-1} u'(s). \tag{2.30}$$

where $J(s)$ is the determinant of the Jacobian of the transformation and $u'(s)$ is the first derivative of the potential. Hence, we can express the (dynamic) Kutta condition as

$$J^{-1}(0)u'(0) + J^{-1}(s_f)u'(s_f) = -(\boldsymbol{\tau}(0) + \boldsymbol{\tau}(s_f)) \cdot \boldsymbol{V}_\infty. \tag{2.31}$$

We can enforce this condition as a constraint for linear system (2.25) by using the Lagrange multiplier method. Exploiting the property of the B-splines derivatives shown in section 2.1, we can set

$$\boldsymbol{k} = \sum_{i=1}^{\mathcal{N}} J^{-1}(0)\xi_i'(0) + J^{-1}(s_f)\xi_i'(s_f) \tag{2.32}$$

Hence, by introducing a Lagrange multiplier λ, system (2.25) becomes

$$\begin{bmatrix} \widehat{\boldsymbol{A}} & \boldsymbol{k}^T \\ \boldsymbol{k} & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \widehat{\boldsymbol{f}} \\ -(\boldsymbol{\tau}(0) + \boldsymbol{\tau}(s_f)) \cdot \boldsymbol{V}_\infty \end{Bmatrix}. \tag{2.33}$$

This is nothing but the algebraic formulation related to the saddle point problem (1.66). System (2.33) can be represented in a more compact form as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}. \tag{2.34}$$

From now on, we will focus on (2.34).

For the solution of linear system (2.34) we use the Matlab \ solver since its dimension is, for the problem at hand, quite small. Once we have solved (2.34) we can recover the approximate solution $u_h(s)$ by equation (2.21).

Some properties of BEM

We now provide some general considerations and remarks related to the Boundary Element Method. BEM can be used to find solutions of boundary value problems in exterior unbounded domains. Since we want to solve the potential u only on the boundary of the domain, BEM requires the discretization of the geometry only into boundary elements. This decreases dramatically the number of degrees of freedom of the system to be solved.

On the other hand, dealing with the boundary element method entails some computational bottlenecks, since the matrix $\boldsymbol{A}$ is full, and assembling its terms can be rather expensive because the integrals are very complicated and can contains singularities (integrands of $O(ln(r))$, $O(1/r)$, $O(1/r^2)$ inside $(Nu)(s)$ and $(Dh)(s)$) that are numerically expensive to be evaluated. These singularities can greatly affects the accuracy of the numerical solution. Moreover, since the matrix $\boldsymbol{A}$ is full, also the solution of the linear system can be very expensive, when dealing with a large dimension approximation space. The fact that the system matrix is dense is usually considered a drawback of BEM, however, for isogeometric methods, it has its positive side: high order and low order B-splines produce the same matrix bandwidth, making high order B-splines the ideal candidates for isogeometric BEM approximations. On the contrary, in finite element methods, for example, this is not the case: high order B-splines effect the sparsity of the matrix more than the low order ones.

Concerning integration techniques, we adopt the so-called Telles algorithm [60], briefly explained in next section. This algorithm is self-adaptive, improves the accuracy of Gaussian quadrature schemes within the near-singularity range and which is by far more efficient (in terms of computational cost) than classical techniques used for the evaluation of singular integrals (see e.g. [37]).

2.4 Numerical evaluation of boundary element integrals

For the sake of numerical evaluation of integrals over the interval $[0, 1]$, we usually adopt quadrature formulas, such that

$$\int_0^1 f(t)dt \approx \sum_{q=1}^{N_q} f(t_q)w_q, \tag{2.35}$$

where t_q and w_q are properly chosen points, called *quadrature points*, and coefficients, called *quadrature weights*, respectively. Usual quadrature formulas, such as Gauss ones (see e.g. [51]) are based on polynomial integration.

When $f(t)$ is singular in s (as the example depicted in figure 2.4) with well defined improper integral, that is,

$$\lim_{t \to s} f(t) = \infty \tag{2.36}$$

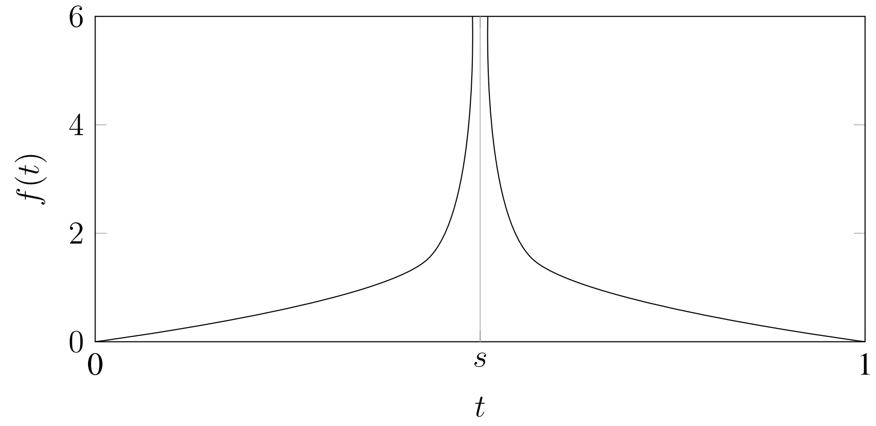

Figure 2.5: Function $f(t)$ singular in $t = s$.

but

$$\lim_{t \to s} \int_0^1 f(t) < \infty, \tag{2.37}$$

we need to exploit special techniques for the evaluation of integrals.

The so-called Telles algorithm [60] is one of these techniques, which allows to remove the singularity by a local change of coordinates. Let us consider a third-degree polynomial regular transformation $\xi(\eta) : [0,1] \to [0,1]; t = \xi(\eta)$

$$\xi(\gamma) = a\eta^3 + b\eta^2 + c\eta + d, \tag{2.38}$$

such that the following requirements are met:

$$\xi''(\overline{\eta}) = 0 \; \xi'(\overline{\eta}) = 0 \; \xi(0) = 0 \; \xi(1) = 1, \tag{2.39}$$

where $\overline{\eta}$ is simply the value of η which satisfies $s = \xi(\overline{\eta})$. Thus, we can write

$$\int_0^1 f(t)dt = \int_0^1 f(\xi(\eta))\xi'(\eta)d\eta \tag{2.40}$$

where $\xi'(\eta)$ is nothing but the determinant of the Jacobian of the transformation $\xi(\eta)$. Thanks to requirements (2.39), $\xi'(\eta)$ is null in the singularity position $\overline{\eta}$. Such a property cancels the singularity and then standard Gaussian integration can be employed, so that

$$\int_0^1 f(\xi(\eta))\xi'(\eta)d\eta \approx \sum_{q=1}^{N_q} f(\xi(\eta_q))\xi'(\eta_q)w_q. \tag{2.41}$$

The coefficients of (2.38) are given by:

$$a = \frac{1}{Q}, \quad b = -\frac{3\overline{\eta}}{Q}, \quad c = \frac{3\overline{\eta}^2}{Q}, \quad d = -b, \tag{2.42}$$

where $Q = 1 + 3\overline{\eta}^2$ and

$$\overline{\eta} = \sqrt[3]{st^* + |t^*|} + \sqrt[3]{st^* - |t^*|} + s, \tag{2.43}$$

where $t^* = s^2 - 1$.

Hence, for the numerical evaluation of variational forms $m(u, v)$, $n(u, v)$, $w(u, v)$ and $d(h, v)$ in (1.65), we use the standard Gauss quadrature formula, whereas for $(Nu)(s)$ and $(Dh)(s)$ in (1.65) we use the Telles quadrature formula.

2.5 Post processing

Once we have computed the solution u on the boundary Γ by solving system (2.34), 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:

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2}, \tag{2.44}$$

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

$$p + \frac{1}{2}\rho V^2 = p_\infty + \frac{1}{2}\rho V_\infty^2 \quad \text{so} \quad p = p_\infty + \frac{1}{2}\rho V_\infty^2 - \frac{1}{2}\rho V^2 \tag{2.45}$$

so that, exploiting (2.45), we can write the pressure coefficient as follows:

$$C_p = 1 - \frac{V^2}{V_\infty^2} = 1 - \frac{|\nabla\Phi|^2}{V_\infty^2}. \tag{2.46}$$

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

$$\begin{aligned} \nabla\Phi &= \nabla_s\Phi = \nabla_s u + \nabla_s(\boldsymbol{V}_\infty \cdot \boldsymbol{x}) \\ &= J^{-2}u'(s)\frac{\partial \boldsymbol{c}(s)}{\partial s} + J^{-2}V_\infty\frac{\partial c_x(s)}{\partial s}\frac{\partial \boldsymbol{c}(s)}{\partial s} \end{aligned} \tag{2.47}$$

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

$$\begin{aligned} |\nabla \Phi|^2 &= J^{-4}\left(u'(s) + V_\infty \frac{\partial c_x(s)}{\partial s}\right)^2 \frac{\partial \boldsymbol{c}(s)}{\partial s}^2 \\ &= J^{-2}\left(u'(s) + V_\infty \frac{\partial c_x(s)}{\partial s}\right)^2, \end{aligned} \tag{2.48}$$

so that the pressure coefficient can be evaluated by introducing (2.48) in (2.46), and by exploiting the representation of the derivatives of u in the isogeometric framework (see section 2.1). The pressure coefficient is one of the most important outputs related to airfoil characterization, since it gives local informations about the behaviour 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

$$C_l = \int_{\Gamma_f} C_p \boldsymbol{n} ds \cdot \boldsymbol{n}_\perp, \tag{2.49}$$

where $\boldsymbol{n}$ is the versor normal to the airfoil surface and $\boldsymbol{n}_\perp$ is the versor 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

$$L = -\rho_\infty V_\infty K, \tag{2.50}$$

exploiting Kutta-Joukowski theorem, or as

$$L = \frac{1}{2}\rho_\infty V_\infty^2 C_l. \tag{2.51}$$

Hence, we obtain

$$C_l = -2\frac{K}{V_\infty}. \tag{2.52}$$

In chapter 5 we fully exploit these quantities to compute the C_p and C_l coming from the high-fidelity model (IGA-BEM) and the reduced order model (IGA-BE-RBM), and to compare them with experimental and Xfoil data.

Finally, we want also to recovery the velocity field in the whole external domain. Recalling the boundary integral equation (1.31), we have the following representation formula:

$$u(\boldsymbol{x}) = -\int_\Gamma u(\boldsymbol{y}) \frac{\partial G(\boldsymbol{x} - \boldsymbol{y})}{\partial n_y} dy + \int_\Gamma G(\boldsymbol{x} - \boldsymbol{y}) h(\boldsymbol{y}) dy, \tag{2.53}$$

valid for any $\boldsymbol{x} \in \Omega$. By definition of potential u, the velocity is

$$\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{V}_\infty + \nabla_x u(\boldsymbol{x}), \tag{2.54}$$

so that

$$\boldsymbol{V}(\boldsymbol{x}) = \boldsymbol{V}_\infty - \int_\Gamma u(\boldsymbol{y}) \nabla_x \frac{\partial G(\boldsymbol{x}-\boldsymbol{y})}{\partial n_y} dy - \int_\Gamma \nabla_x G(\boldsymbol{x}-\boldsymbol{y}) h(\boldsymbol{y}) dy. \tag{2.55}$$

In the same way, we can obtain the pressure field from (2.45).

We recall that in chapter 4 we will compute an error bound for the potential u. Similar results for the estimation of error bounds for the velocity $\boldsymbol{V}$ and pressure p (with respect to a high-fidelity FE approximation) can be found in e.g. [53].

Chapter 3

Parametrized formulation of potential flows

Our ultimate goal is the efficient description of potential flows about airfoils in several different scenarios, from both a physical and geometrical standpoint. For this reason, we introduce in this chapter a parametrized version of (1.43), depending on two input parameters which represent the shape of the airfoil and its angle of attack. We derive the algebraic formulation of the parametrized problem, by highlighting the role of the parameters. Then, we show how to express in a suitable way the parametric dependence in the structures (matrices and vectors) of the derived problem, through the so-called empirical interpolation method (EIM). This is mandatory in order to build a ROM able to solve the problem for any new instance of the input parameters, in a very efficient way. We show how to implement the EIM procedure in a BEM context in two alternative ways. This is a further, relevant achievement of this thesis, since EIM itself manages to reduce the computational cost related to the assembling stage of the BEM structures, typically very expensive, in the case of multiple queries.

3.1 Parametric dependence

Let us now assume that the geometry of the problem described so far is parameter dependent. This entails a parametric dependence on the whole potential flow problem, and ultimately, on the potential ϕ and our output of interest. Let us define the parameter vector

$$\boldsymbol{\mu} = \begin{Bmatrix} \text{Naca} \\ \alpha \end{Bmatrix} \in \mathcal{D} \subset \mathbb{R}^2 \tag{3.1}$$

where α is the angle of attack and Naca is the four digit number used for the description of the shape of an airfoil developed by National Advisory Committee for Aeronautics (NACA), as explained in section 2.2.

For each value of NACA number (for null angle of attack), we can find the position of the control points from (2.16) by solving the normal equations

$$\boldsymbol{B}^T \boldsymbol{B} \boldsymbol{P}(\text{Naca}) = \boldsymbol{B}^T \boldsymbol{\gamma}(\text{Naca}), \tag{3.2}$$

where

$$\boldsymbol{P}(\text{Naca}) = \begin{bmatrix} \boldsymbol{P}_1(\text{Naca}) \\ \vdots \\ \boldsymbol{P}_{\mathcal{N}}(\text{Naca}) \end{bmatrix} \tag{3.3}$$

and

$$\boldsymbol{\gamma}(\text{Naca}) = \begin{bmatrix} \boldsymbol{\gamma}_1(\text{Naca}) \\ \vdots \\ \boldsymbol{\gamma}_M(\text{Naca}) \end{bmatrix} = \begin{bmatrix} x_1 & y_1(\text{Naca}) \\ \vdots & \vdots \\ x_M & y_M(\text{Naca}) \end{bmatrix}. \tag{3.4}$$

Then, we can easily apply a rotation to the control polygon in order to rotate the airfoil according to its angle of attack α, so that, in the end, $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{\mu})$. Thus, we end up with a parametrized geometry described by

$$\boldsymbol{c}(s; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) \boldsymbol{P}_i(\boldsymbol{\mu}). \tag{3.5}$$

In particular, we highlight that we can change all the control points position simply by two parameters thanks to the least square procedure (2.16) and the subsequent rotation. This feature makes this shape parametrization technique quite different from other strategies recently considered, such as Free Form Deformation (FFD) or Radial Basis Function (RBF) [44], which would typically require more parameters to represent shapes/deformations of comparable complexity. Moreover, the IGA parametrization we have considered allows to recover exactly the whole family of NACA airfoils, by prescribing directly its NACA number. This makes it possible to compare the results with experimental data, tipically available for airfoils inside the NACA family, and to easily obtain a description of the flow in a very wide family of configurations simply by prescribing one parameter. This is a very remarkable achievement of the present work.

Since we want to neglect some 4-digits NACA profile which are not interesting from an engineering standpoint (figure 3.1), we restrict the Naca domain, namely $[0000, 9999]$, to 7 not connected subregions within the ranges $[i208, i620]$, $i = 0, \cdots 6$. We remark that, for symmetric profile, that is, when $i = 0$, the second digit is meaningless. Thus, for example, NACA 0012 profile has the very same shape of NACA 0212 one. For the same reason, we neglect angles of attack too high and we restrict the domain of α to $[-20\,\text{deg}, 20\,\text{deg}]$.

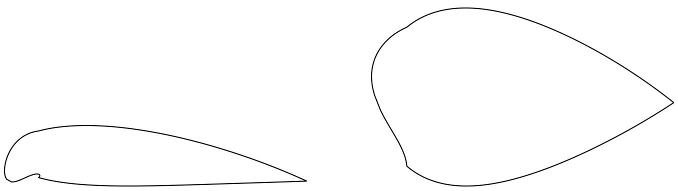

Figure 3.1: NACA 9120 and NACA 2160 profiles are not interesting from an engineering standpoint.

We now introduce the parametrized version of (1.22) and, after all the steps carried out in chapter 1, we introduce the weak formulation of the parametrized problem. In the following, for the sake of simplicity, we will only show where the parametric geometry affects the problem at hand and, then, we just provide the final weak form.

3.1.1 Weak formulation for parametrized problem

Starting from (1.55) and (1.57), let us analyse which terms are affected by the parameter $\boldsymbol{\mu}$:

- the mapping $\boldsymbol{c}$ is now $\boldsymbol{\mu}$-dependent, so that the Green function G, its derivative $\frac{\partial G}{\partial n_q}$ and the determinant of the Jacobian of the transformation J are now $\boldsymbol{\mu}$-dependent:

$$\begin{aligned} G &= G(\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu})) \\ \frac{\partial G}{\partial n_q} &= \frac{\partial G}{\partial n_q}(\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu})) \\ J &= J(q;\boldsymbol{\mu}); \end{aligned} \tag{3.6}$$

- the normal $\boldsymbol{n}$ and tangent $\boldsymbol{\tau}$ versors to Γ are now $\boldsymbol{\mu}$-dependent, so that Neumann boundary condition (1.40) and Kutta condition (1.19) are now $\boldsymbol{\mu}$-dependent:

$$\begin{gathered} h = h(q;\boldsymbol{\mu}) \\ V(TE^+) = V(TE^+;\boldsymbol{\mu}) \quad \text{and} \quad V(TE^-) = V(TE^-;\boldsymbol{\mu}). \end{gathered} \tag{3.7}$$

Moreover, let us introduce the parametrized operators N and D (1.58) as follows:

$$\begin{aligned} (Nu)(s;\boldsymbol{\mu}) &= \int_0^1 u(q)\frac{\partial G}{\partial n_q}(\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu}))J(q;\boldsymbol{\mu})dq \\ (Dh)(s;\boldsymbol{\mu}) &= \int_0^1 G(\boldsymbol{c}(s;\boldsymbol{\mu}) - \boldsymbol{c}(q;\boldsymbol{\mu}))h(q;\boldsymbol{\mu})J(q;\boldsymbol{\mu})dq. \end{aligned} \tag{3.8}$$

We can now write the parametric version of the weak form (1.66) as follows: given $\boldsymbol{\mu} \in \mathcal{D}$, $h \in V'$, find $u \in V$, $\lambda \in \mathbb{R}$ such that

$$\begin{cases} a(u, v; \boldsymbol{\mu}) \; + \; b(v, \lambda; \boldsymbol{\mu}) \; = \; d(h(\boldsymbol{\mu}), v; \boldsymbol{\mu}) & \forall v \in V \\ b(u, \pi; \boldsymbol{\mu}) \; = \; \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) & \forall \pi \in \mathbb{R}, \end{cases} \tag{3.9}$$

where

$$\begin{aligned} b(u, \pi; \boldsymbol{\mu}) &:= \left[\boldsymbol{\tau}(0, \boldsymbol{\mu}) \cdot \nabla_s u|_0 + \boldsymbol{\tau}(s_f, \boldsymbol{\mu}) \cdot \nabla_s u|_{s_f}\right] \pi \\ d(h(\boldsymbol{\mu}), v; \boldsymbol{\mu}) &:= \int_0^1 v(s)(Dh)(s; \boldsymbol{\mu}) \chi_{[0, s_f]}(s) ds \quad \forall v \in V, h \in V', \end{aligned} \tag{3.10}$$

and, recalling (1.63), $a(u, v; \boldsymbol{\mu}) = \alpha m(u, v) + n(u, v; \boldsymbol{\mu}) + w(u, v)$ where

$$\begin{aligned} m(u, v) &:= \int_0^1 v(s) u(s) \chi_{[0, s_f]}(s) ds \quad \forall u, v \in V \\ n(u, v; \boldsymbol{\mu}) &:= \int_0^1 v(s)(Nu)(s; \boldsymbol{\mu}) \chi_{[0, s_f]}(s) ds \quad \forall u, v \in V, \boldsymbol{\mu} \in \mathcal{D} \\ w(u, v) &:= \int_0^1 v(s) \left[u(s) - u(0) + u(s_f)\right] \chi_{[s_f, 1]}(s) ds \quad \forall u, v \in V. \end{aligned} \tag{3.11}$$

We can now derive a Galerkin method for the numerical approximation of the problem at hand.

3.1.2 Numerical approximation for parametrized problem

In this section we derive a numerical approximation of (3.9) based on a Galerkin boundary element method. Following the same procedure used in section 2.3, we assume that the solution is expressed as

$$u_h(s; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) u_i(\boldsymbol{\mu}) \tag{3.12}$$

and that the Neumann boundary condition is given by

$$h_h(s; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} \xi_i(s) h_i(\boldsymbol{\mu}), \tag{3.13}$$

so that the problem reads as follows: given $\boldsymbol{\mu} \in \mathcal{D}$, $h_h \in V^{h'}$, find $u_h \in V^h$, $\lambda \in \mathbb{R}$ such that

$$\begin{cases} a(u_h, v_h; \boldsymbol{\mu}) \; + \; b(v_h, \lambda; \boldsymbol{\mu}) \; = \; d(h_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) & \forall v_h \in V^h \\ b(u_h, \pi; \boldsymbol{\mu}) \; = \; \boldsymbol{V}_\infty \cdot (\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f; \boldsymbol{\mu})) & \forall \pi \in \mathbb{R}. \end{cases} \tag{3.14}$$

The corresponding algebraic formulation of this problem is the following linear system:

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}). \tag{3.15}$$

the matrix $\boldsymbol{A}(\boldsymbol{\mu})$, and indeed the right hand side $\boldsymbol{b}(\boldsymbol{\mu})$, show a block structure, as already mentioned in section 2.3. Let us now focus on the matrix structure, in order to better understand which submatrices depend on the parameter $\boldsymbol{\mu}$. We can write $\boldsymbol{A}(\boldsymbol{\mu})$ as follows:

$$\boldsymbol{A}(\boldsymbol{\mu}) = \begin{bmatrix} \begin{array}{c} [\boldsymbol{\alpha M} + \boldsymbol{N}(\boldsymbol{\mu})] \\ \boldsymbol{W} \end{array} & \boldsymbol{k}^T(\boldsymbol{\mu}) \\ \boldsymbol{k}(\boldsymbol{\mu}) & 0 \end{bmatrix}, \tag{3.16}$$

where

$$\begin{aligned}
M_{ij} &= m(\xi_j, \xi_i) = \int_0^1 \xi_i(s)\xi_j(s)ds, \quad i = 1 : N_f, j = 1 : \mathcal{N} \\
N_{ij} &= n(\xi_j, \xi_i; \boldsymbol{\mu}) = \int_0^1 \xi_i(s)(N\xi_j)(s; \boldsymbol{\mu})ds, \quad i = 1 : N_f, j = 1 : \mathcal{N} \\
W_{ij} &= w(\xi_j, \xi_i) = \int_0^1 \xi_i(s)[\xi_j(s) - \xi_j(0) + \xi_j(s_f)]ds, \quad i = N_f + 1 : \mathcal{N}, j = 1 : \mathcal{N}
\end{aligned} \tag{3.17}$$

are the elements of the (rectangular) matrices $\boldsymbol{M}$, $\boldsymbol{N}(\boldsymbol{\mu})$ and $\boldsymbol{W}$, respectively, and

$$\boldsymbol{k}(\boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} J^{-1}(0; \boldsymbol{\mu})\xi_i'(0) + J^{-1}(s_f; \boldsymbol{\mu})\xi_i'(s_f) \tag{3.18}$$

is the vector corresponding to the (Kutta) constraint.

Concerning the right hand side $\boldsymbol{b}(\boldsymbol{\mu})$, we can write

$$\boldsymbol{b} = \begin{Bmatrix} \boldsymbol{d}(\boldsymbol{\mu}) \\ 0 \\ -(\boldsymbol{\tau}(0; \boldsymbol{\mu}) + \boldsymbol{\tau}(s_f); \boldsymbol{\mu}) \cdot \boldsymbol{V}_\infty \end{Bmatrix}, \tag{3.19}$$

where

$$d_i(\boldsymbol{\mu}) = d(h_h(\boldsymbol{\mu}), \xi_i; \boldsymbol{\mu}) = \int_0^1 \xi_i(s) \left(D \sum_{j=1}^{\mathcal{N}} \xi_j(q) h_j(\boldsymbol{\mu}) \right)(s; \boldsymbol{\mu})ds, \quad i = 1 : N_f. \tag{3.20}$$

Since some of the terms are parameter dependent and some are not we will introduce a splitting in order to highlight this difference. We can rewrite the matrix $\boldsymbol{A}(\boldsymbol{\mu})$ as

$$\boldsymbol{A}(\boldsymbol{\mu}) = \boldsymbol{A}^M + \boldsymbol{A}^w + \boldsymbol{A}^k(\boldsymbol{\mu}) + \tilde{\boldsymbol{N}}(\boldsymbol{\mu}) \tag{3.21}$$

where:

- $\boldsymbol{A}^M$ is the mass matrix, corrected with $\boldsymbol{\alpha}$ on the diagonal, with null elements in the rows and columns corresponding to the ones of the wake and Kutta condition
$$\begin{bmatrix} \boldsymbol{\alpha M} & 0 \\ 0 & 0 \end{bmatrix}; \tag{3.22}$$
- $\boldsymbol{A}^w$ is the wake matrix with null elements everywhere, except from the rows and columns corresponding to the wake
$$\begin{bmatrix} 0 & 0 \\ \boldsymbol{W} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.23}$$
- $\boldsymbol{A}^k(\boldsymbol{\mu})$ is the matrix with null elements everywhere except from the rows and columns corresponding to Kutta condition
$$\begin{bmatrix} 0 & \boldsymbol{k}^T(\boldsymbol{\mu}) \\ \boldsymbol{k}(\boldsymbol{\mu}) & 0 \end{bmatrix}; \tag{3.24}$$
- $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ is Neumann matrix with null elements in the rows and columns correspondent to the ones of the wake and Kutta condition
$$\begin{bmatrix} \boldsymbol{N}(\boldsymbol{\mu}) & 0 \\ 0 & 0 \end{bmatrix}. \tag{3.25}$$

This representation will be very useful in the following. In fact, when the parameter value changes, only $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$, $\boldsymbol{A}^k(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ are affected by parameter variations.

In order to fully exploit the Offline-Online splitting in the ROMs presented in chapter 4, we need to write system (3.15) in the form
$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l + \boldsymbol{A}^M + \boldsymbol{A}^k(\boldsymbol{\mu}) + \boldsymbol{A}^w\right)\boldsymbol{u}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \tag{3.26}$$
thus expressing the matrix $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and the vector $\boldsymbol{b}(\boldsymbol{\mu})$ as a linear combination of $\boldsymbol{\mu}$-dependent functions and $\boldsymbol{\mu}$-independent matrices (and vectors).

This form is obtained naturally if the problem has an affine parametric dependence. Unfortunately, this is not our case: the isogeometric parametrization entails a non affine parametric dependence. Thus, we need to approximate the nonaffinely parameter dependent terms. This procedure is carried out by the Empirical Interpolation Method (EIM), shown in the following section. Since we want to impose Kutta condition exactly, we will not apply EIM to $\boldsymbol{A}^k(\boldsymbol{\mu})$ but we will compute it directly during the online stage. By doing this, we do not increase the computational cost of the online stage because the elements related to Kutta condition are very few, and very fast to be evaluated. In our specific case, we need to compute only four elements of $\boldsymbol{A}^k(\boldsymbol{\mu})$ and one element of the right hand side.

3.2 Empirical Interpolation Method

The assumption of affine parametric dependence of the problem is of fundamental importance in order to exploit the Offline-Online stratagem, and then minimize the marginal cost associated with each Online evaluation. This is straightforward if the problem depends affinely on the parameters. However, also nonaffine problems can be efficiently treated in the RB framework, by relying on the so-called Empirical Interpolation Method. This is an interpolation method for parametric functions based on adaptively chosen interpolation points and global shape functions, allowing to recover the affine structure, thanks to a suitable approximation of those parameter-dependent functions appearing in the problem operators. We remark that EIM itself enables to reduce the computational cost required to solve the BEM problem: in fact, it speeds up the assembling of BEM matrices, which is an expensive part of every BEM algorithm. In the following subsections we provide a short presentation of the Empirical Interpolation Method based on [5].

3.2.1 Idea and formulation

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 $\boldsymbol{\mu}$. Under the assumption of affine parameter dependence $g(\boldsymbol{x},\boldsymbol{\mu})$ can be expressed as

$$g(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{j=1}^{M} \gamma^j(\boldsymbol{\mu}) g_j(\boldsymbol{x}). \tag{3.27}$$

If the parametric dependence is nonaffine, we want to find an approximate expansion of the form

$$g_M(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{j=1}^{M} \gamma^j(\boldsymbol{\mu}) g_j(\boldsymbol{x}) + e_{EIM}(\boldsymbol{\mu}), \tag{3.28}$$

where $\gamma^j(\boldsymbol{\mu}), j = 1 : M$ are $\boldsymbol{\mu}$-dependent functions, $g_j(\boldsymbol{x}), j = 1 : M$ are $\boldsymbol{\mu}$-independent functions (denoted also as *shape functions*) and $e_{EIM}(\boldsymbol{\mu})$ is an error term, which we want to keep as small as possible. Being an interpolation procedure, EIM seeks a sequence of (nested) sets of interpolation points $T_M = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M\}$ also called *magic points*, with $\boldsymbol{p}_j \in \Omega$ for each $j = 1 : M$, and a set of shape functions $g_j(\boldsymbol{x})$, in order to compute (3.28) by solving the following linear system:

$$\sum_{j=1}^{M} B_{i,j}^{M} \gamma^j(\boldsymbol{\mu}) = g(\boldsymbol{p}_i, \boldsymbol{\mu}), \quad \forall i = 1 : M, \tag{3.29}$$

where $B^M \in R^{M\times M}$ is defined as $B^M_{ij} := g_j(\boldsymbol{p}_i), \forall i, j = 1 : M$. Linear system (3.29), also denoted as Lagrange interpolation problem, stems by the imposition of interpolation constraints:

$$g_M(\boldsymbol{p}_i, \boldsymbol{\mu}) = g(\boldsymbol{p}_i, \boldsymbol{\mu}) \quad \forall i = 1 : M. \tag{3.30}$$

EIM usually selects both the interpolation points and basis functions by means of a greedy algorithm. Let us denote by $\Xi^{EIM}_{train} \subset \mathcal{D}$ 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. To begin the EIM procedure we have to chose an 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 (3.29) to approximate each snapshot of the generating function, and choosing the parameter of the snapshot that is the worst approximated to be

$$\boldsymbol{\mu}^M := \arg \max_{\boldsymbol{\mu}\in\Xi^{EIM}_{train}} \inf_{v\in G_{M-1}} ||g(\cdot, \boldsymbol{\mu}) - v||_{L^\infty(\Omega)}. \tag{3.31}$$

The snapshot $g(\boldsymbol{x}, \boldsymbol{\mu}^M)$ is then normalized to form the M-th basis function, that is,

$$\begin{array}{rcl} S_M & = & S_{M-1} \cup \boldsymbol{\mu}^M \\ \zeta_M(\mathbf{x}) & = & g(\mathbf{x}; \boldsymbol{\mu}^M) \\ G_{M-1} \cup \{\zeta_1\} = G_M & := & \text{span}(\zeta_1, \ldots, \zeta_M). \end{array} \tag{3.32}$$

The greedy procedure stops when the EIM error is below the tolerance ϵ^*_{EIM}, for any $\boldsymbol{\mu} \in \Xi^{EIM}_{train}$. The whole procedure is shown in the following Algorithm 1

$Algorithm 1$

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

Then, we construct sets of nested interpolation points $T_M = \{\mathbf{p}_1, \cdots, \mathbf{p}_1\}$, with $M = 1 : M_{max}$. 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 $\boldsymbol{p}_1$, that is:

$$\mathbf{p}_1 := \arg \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} |\zeta_1(\mathbf{x})|, \quad q_1 = \zeta_1(\mathbf{x})/\zeta_1(\mathbf{p}_1), \quad B^1_{11} = 1. \tag{3.33}$$

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

$$\sum_{j=1}^{M-1} \sigma_j^{M-1} q_j(\mathbf{p}_i) = \zeta_M(\mathbf{p}_i), \qquad i = 1, \ldots, M-1 \tag{3.34}$$

and compute the residual

$$r_M(\mathbf{x}) := \zeta_M(\mathbf{x}) - \sum_{j=1}^{M-1} \sigma_j^{M-1} g_j(\mathbf{x}); \tag{3.35}$$

the m-th magic point is given by

$$\mathbf{p}_M := \arg \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} |r_M(\mathbf{x})|, \tag{3.36}$$

and then we set:

$$\begin{aligned} q_M(\mathbf{x}) &= r_M(\mathbf{x})/r_M(\mathbf{p}_M) \\ B_{ij}^M &= q_j(\mathbf{p}_i), \quad i, j = 1, \ldots, M. \end{aligned} \tag{3.37}$$

All these operations are summarized in the following 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 $B_{11}^1 = 1$;
for $M = 2 : M_{\text{max}}$
    solve $\sum_{j=1}^{M-1} \sigma_j^{M-1} q_j(\mathbf{p}_i) = \zeta_M(\mathbf{p}_i), \qquad i = 1, \ldots, M-1$;
    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 $q_M(\mathbf{x}) = r_M(\mathbf{x})/r_M(\mathbf{p}_M), \quad B_{ij}^M = q_j(\mathbf{p}_i), \quad i, j = 1, \ldots, M$;
end.
```

In the following we present two different ways to apply EIM to the problem at hand. The first one is more classical, and it deals with the application of previous algorithms to those parameter dependent functions appearing in the integrals (3.8); the second one deals instead with the algebraic structures related to the BEM parametrized problem, directly. A brief comparison between the two approaches is also provided.

3.2.2 Application to potential flows about an airfoil

Let us now analyze how the EIM can be applied to the problem at hand. In particular, we focus on the matrix $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and the right hand side $\boldsymbol{b}(\boldsymbol{\mu})$, since $\boldsymbol{A}^M$ and $\boldsymbol{A}^w$ are parameter-independent and we compute $\boldsymbol{A}^k(\boldsymbol{\mu})$ during the online stage.

We can express $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ as follows

$$\begin{aligned}\tilde{N}_{ij}(\boldsymbol{\mu}) &= \begin{cases}\int_0^1 \xi_i(s)\Psi_j(s,\boldsymbol{\mu})ds, & i = 1:N_f, j = 1:\mathcal{N}\\ 0, & i = N_f+1:\mathcal{N}, j = 1:\mathcal{N}\end{cases}\\ b_i(\boldsymbol{\mu}) &= \begin{cases}\int_0^1 \xi_i(s)\Theta(s,\boldsymbol{\mu})ds, & i = 1:N_f\\ 0, & i = N_f+1:\mathcal{N},\end{cases}\end{aligned} \tag{3.38}$$

where

$$\begin{aligned}\Psi_j(s;\boldsymbol{\mu}) &= (N\xi_j)(s;\boldsymbol{\mu}),\\ \Theta(s;\boldsymbol{\mu}) &= \left(D\sum_{j=1}^{\mathcal{N}}\xi_j(q)h_j(\boldsymbol{\mu})\right)(s;\boldsymbol{\mu}).\end{aligned} \tag{3.39}$$

For the sake of simplicity, in this section, we neglect the last column and row of $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$, as well as the last term of $\boldsymbol{b}(\boldsymbol{\mu})$, which are related to the Kutta condition; as already mentioned, we do not apply EIM to this terms.

In order to obtain an affine expansion of $\Psi_j(s,\boldsymbol{\mu}), j = 1:\mathcal{N}$ and $\Theta(s,\boldsymbol{\mu})$, we apply the Empirical Interpolation Method to these functions. The two terms can be approximated by

$$\Psi_j(s;\boldsymbol{\mu}) = \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})\beta_j^l(s) \quad \text{and} \quad \Theta(s;\boldsymbol{\mu}) = \sum_{m=1}^{M}\gamma^m(\boldsymbol{\mu})\beta^m(s), \tag{3.40}$$

respectively. Let us analyse all the functions of (3.38) in order to better understand how the EIM changes the building of the structures $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$. Let us focus only on the non null elements. We can write

$$\begin{aligned}\tilde{N}_{ij}(\boldsymbol{\mu}) &= \int_0^1 \xi_i(s)\Psi_j(s;\boldsymbol{\mu})ds \approx \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})\int_0^1 \xi_i(s)\beta_j^l(s)ds\\ &= \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})h_{ij}^l, \quad i = 1:N_f, j = 1:\mathcal{N},\end{aligned} \tag{3.41}$$

where

$$h_{ij}^l = \int_0^1 \xi_i(s)\beta_j^l(s)ds, \quad i = 1:N_f, j = 1:\mathcal{N}. \tag{3.42}$$

With a more compact notation, we can write the matrix $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ as follows

$$\tilde{\boldsymbol{N}}(\boldsymbol{\mu}) = \sum_{l=1}^{L}\gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l, \tag{3.43}$$

where

$$\boldsymbol{N}^l = \begin{bmatrix} h^l_{11} & \cdots & h^l_{1\mathcal{N}} \\ \vdots & \ddots & \vdots \\ h^l_{N_f 1} & \cdots & h^l_{N_f \mathcal{N}} \\ 0 & \cdots & 0 \end{bmatrix}, \quad l = 1 : L, \tag{3.44}$$

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. In the same way, we have

$$\begin{aligned} b_i(\boldsymbol{\mu}) &= \int_0^1 \xi_i(s)\Theta(s;\boldsymbol{\mu})ds \approx \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) \int_0^1 \xi_i(s)\beta^m(s)ds \\ &= \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) h_i^m, \quad i = 1 : N_f, \end{aligned} \tag{3.45}$$

where

$$h_i^m = \int_0^1 \xi_i(s)\beta^m(s)ds, \quad i = 1 : N_f. \tag{3.46}$$

More compactly:

$$\boldsymbol{b}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \tag{3.47}$$

where

$$\boldsymbol{b}^m = [h_1^m, \cdots, h_{N_f}^m, 0]^T \quad m = 1 : M, \tag{3.48}$$

and M is the number of EIM terms.

During the Offline stage, we compute the terms β^l and β^m, so that we need to assemble the structures $\boldsymbol{N}^l$ and $\boldsymbol{b}^m$ only once. During the Online stage, for any new value of the parameter $\boldsymbol{\mu}$, we solve two lower (and small) triangular system (3.29) to obtain $\gamma^l(\boldsymbol{\mu})$ and $\gamma^m(\boldsymbol{\mu})$, and then assemble the $\boldsymbol{\mu}$-dependent structures $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$.

3.2.3 An alternative EIM version for matrices and vectors

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

$$\begin{aligned} \tilde{N}_{ij}(\boldsymbol{\mu}) &= \begin{cases} \sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu}) h_{ij}^l, & i = 1 : N_f, j = 1 : \mathcal{N} \\ 0, & i = N_f + 1 : \mathcal{N}, j = 1 : \mathcal{N} \end{cases} \\ b_i(\boldsymbol{\mu}) &= \begin{cases} \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) h_i^m, & i = 1 : N_f \\ 0, & i = N_f + 1 : \mathcal{N}, \end{cases} \end{aligned} \tag{3.49}$$

that is,

$$\tilde{\boldsymbol{N}}(\boldsymbol{\mu}) = \sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu}) \boldsymbol{N}^l, \tag{3.50}$$

where

$$\boldsymbol{N}^l = \begin{bmatrix} h^l_{11} & \cdots & h^l_{1\mathcal{N}} \\ \vdots & \ddots & \vdots \\ h^l_{N_f 1} & \cdots & h^l_{N_f \mathcal{N}} \\ 0 & \cdots & 0 \end{bmatrix}, \quad l = 1 : L, \tag{3.51}$$

and L is the number of EIM basis functions, $\mathcal{N}$ is the dimension of the problem at hand and N_f is the number of control values for the airfoil. In the same way, we have

$$\boldsymbol{f}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) \boldsymbol{b}^m, \tag{3.52}$$

where

$$\boldsymbol{b}^m = [h^m_1, \cdots, h^m_{N_f}, 0]^T \quad m = 1 : M, \tag{3.53}$$

and M is the number of EIM basis functions.

Basically, the two methods differ in two aspects. On the one hand, the second EIM strategy returns directly the structures $\boldsymbol{N}^l$ and $\boldsymbol{b}^m$. On the other hand, instead of choosing some *magic points* for the evaluation of linear system (3.29), in the second case EIM selects at each step a couple of *magic indices* (i, j). These features are neither pros or cons of the two techniques, but from now on we will exploit the second approach to deal with nonaffinely parametrized operators, since we noted that it selects less affine terms, for a prescribed tolerance, by making the second strategy more efficient and it reduces the computational cost associated with the assembling of BEM matrices and vectors assembling, which is normally very expensive.

Once EIM procedure has been performed, we can rewrite problem (3.15) under the following affine form:

$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu}) \boldsymbol{N}^l + \boldsymbol{A}^M + \boldsymbol{A}^k(\boldsymbol{\mu}) + \boldsymbol{A}^w \right) \boldsymbol{u}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu}) \boldsymbol{b}^m. \tag{3.54}$$

We have obtained so far all the necessary ingredients for the construction of efficient reduced order models, which is the topic of the following chapter.

Chapter 4

Reduced order models for parametrized potential flows

In this chapter we discuss all the details related to the construction of Reduced Order Models (ROMs) for parametrized PDEs. Following [50], [53] and [54], in section 4.1 we introduce the basic components of a ROM. Then, in section 4.2, we show how to write the algebraic (discrete) RB problem, given the RB basis functions. Moreover, we show how to take advantage of an efficient Offline-Online splitting in section 4.2.1, thanks to the Empirical Interpolation procedure introduced in chapter 3. In section 4.3 we present two different strategies for the construction of the reduced space, namely the Proper Orthogonal Decomposition (POD) and the (greedy) reduced basis method (RBM), this latter procedure relying on an (a posteriori) error bound, whose main features are briefly sketched in section 4.4. Finally, in section 4.5, we show how the problem of potential flows about an airfoil, obtained in chapter 3, can be solved by taking advantage of these two ROMs.

4.1 Main components of Reduced Order Models

The goal of a ROM is to compute a low-dimensional approximation of the high-fidelity (parameter dependent) solution in an inexpensive way. The reduced solution is obtained by a projection onto a small subspace, made by *global* (properly chosen) basis functions, instead of a large space of *local* (but generic) basis functions. In fact, we assume that the family of the full-order solutions, obtained for different values of the parameter $\boldsymbol{\mu}$ through a suitable high-fidelity approximation technique, is smooth. Therefore, we expect any high-fidelity solution (that is, for any value of $\boldsymbol{\mu} \in \mathcal{D}$) to be well approximated in terms of these (properly selected) basis functions. We can summarize the main ingredient of a ROM as follows:

i. *High-fidelity method*: in our case, reduced order models do not replace our high-fidelity IGA-BEM approximation technique. Rather, they build upon this high-fidelity (or full-order) approximation method. In other words, the reduced basis solution does not approximate directly the exact solution, but rather a 'given' IGA-BEM solution. As seen in chapter 3, the full-order problem reads: $\mathcal{N}$), find $u_h(\mu) \in V^h$ such that

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \quad \forall v_h \in V^h, \tag{4.1}$$

where V^h is a finite dimensional space of (possibly large) dimension $\mathcal{N}$. The solution of (4.1) can be kept as close as desired to the physical solution u by choosing a suitable discretization space.

ii. *Galerkin projection*: a ROM consists in selecting a (reduced) basis $\{\zeta_i\}_{i=1}^N$ and finding the reduced order solution $u_N(\boldsymbol{\mu})$ as a linear combination of these basis functions. The choice of the basis functions differs for the different methods. The space where we seek the reduced solution is

$$V^N = \text{span}\left\{\zeta_i, i = 1, \cdots N\right\}, \tag{4.2}$$

with $N \ll \mathcal{N}$ [1]. Therefore, the reduced order problem reads: find $u_N(\boldsymbol{\mu}) \in V^N$ such that

$$a(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}) \quad \forall v_N \in V^N \tag{4.3}$$

iii. *Offline-Online splitting*: the goal is to perform the extensive snapshots generation only once (Offline stage), and then a rapid evaluation of the solution for any new $\boldsymbol{\mu}$ instance (Online stage). Unfortunately, although problem (4.3) is nominally of small size N, assembling its structures (matrix and right hand side) involves entities associated with our $\mathcal{N}$-dimensional BE approximation space. To overcome this problem, and then to construct a $\mathcal{N}$-independent Online stage, we can rely in our case on the EIM to recover an affine structure of our operators.

iv. *Error estimation procedure*: rigorous, sharp and inexpensive error bounds such that

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu}) \tag{4.4}$$

might be easy to characterize, depending on the problem at hand. This error bound might be also employed to generate a clever parameter sampling for the construction of the reduced space. If we have applied an Empirical Interpolation Method in the algorithm (which is our case) the error bound becomes more involved, in

---

[1] Each V^N is a (hierarchy assumption) N-dimensional subspace of V^h, and we further suppose that $V^1 \subset V^2 \subset \cdots \subset V^N \subset V^h$.

order to take into account also the contribution of the EIM error. In this case, we can split (4.4) in two terms, and write it as

$$\begin{aligned} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V &\leq \|u_h(\boldsymbol{\mu}) - u_h^{EIM}(\boldsymbol{\mu})\|_V + \|u_h^{EIM}(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \\ &\leq \Delta^{EIM}(\boldsymbol{\mu}) + \Delta_N(\boldsymbol{\mu}), \end{aligned} \tag{4.5}$$

where $u_h^{EIM}(\boldsymbol{\mu})$ is the solution of (3.54), that is, the solution of the IGA-BEM problem once the Empirical Interpolation Method has been applied (EI-IGA-BEM).

In chapter 3 we have already provided all the details related to the high-fidelity model. In the following, we provide more details about the other ingredients of the construction of the ROMs.

4.2 Algebraic Reduced Basis problem

In this section, we first introduce the algebraic version of the RB method, no matters the method we use to generate the reduced space. Then, we discuss the algebraic connection between the reduced order problem and the high-fidelity one.

As shown in the previous section, we assume that the reduced order solution is given by

$$u_N(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_N^m(\boldsymbol{\mu})\zeta^m; \tag{4.6}$$

by inserting (4.6) into (4.3) and taking $v_N = \zeta^n, n = 1 : N$, we obtain the RB algebraic system

$$\sum_{m=1}^{N} a(\zeta^m, \zeta^n; \boldsymbol{\mu}) u_N^m(\boldsymbol{\mu}) = f(\zeta^n; \boldsymbol{\mu}), \quad n = 1 : N. \tag{4.7}$$

Hence, we can write the algebraic reduced order problem as

$$\boldsymbol{A}_N(\boldsymbol{\mu})\boldsymbol{u}_N(\boldsymbol{\mu}) = \boldsymbol{b}_N(\boldsymbol{\mu}), \tag{4.8}$$

where

$$\begin{aligned} (\boldsymbol{A}_N(\boldsymbol{\mu}))_{nm} &= a(\zeta^m, \zeta^n; \boldsymbol{\mu}), \\ (\boldsymbol{b}_N(\boldsymbol{\mu}))_n &= f(\zeta^n; \boldsymbol{\mu}) \end{aligned} \tag{4.9}$$

and $u_N^m(\boldsymbol{\mu}) = (u_N^1(\boldsymbol{\mu}), \cdots, u_N^N(\boldsymbol{\mu}))^T$ is the vector of the degrees of freedom related to the RB solution. Since each basis function ζ^n belongs to the BE space V^h, we can express each of them as

$$\zeta^n = \sum_{i=1}^{\mathcal{N}} \zeta_i^n \xi_i, \quad n = 1 : N, \tag{4.10}$$

that is, as a linear combination of the BE basis functions $\{\xi_i\}_{i=1}^{\mathcal{N}}$. Then, by inserting (4.10) in (4.7), and by denoting

$$\mathcal{Z} = [\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_N] \in \mathbb{R}^{\mathcal{N}\times N} \tag{4.11}$$

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

$$\begin{array}{rcl} \boldsymbol{A}_N(\boldsymbol{\mu}) & = & \mathcal{Z}^T \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z} \\ \boldsymbol{b}_N(\boldsymbol{\mu}) & = & \mathcal{Z}^T \boldsymbol{b}(\boldsymbol{\mu}), \end{array} \tag{4.12}$$

where $\boldsymbol{A}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ are the structures given by the BE discretization of the problem:

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{u}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}). \tag{4.13}$$

A graphical sketch related to the assembling of RB structures is provided in figure 4.1.

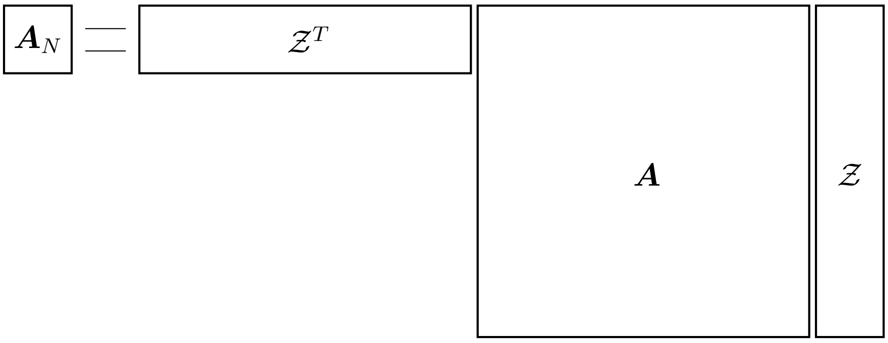

Figure 4.1: Schematic representation of the RB matrix assembling.

We can now characterize the error between the RB solution and the high-fidelity one

$$\boldsymbol{e}_N(\boldsymbol{\mu}) = \boldsymbol{u}(\boldsymbol{\mu}) - \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}) \tag{4.14}$$

in terms of the high-fidelity residual of the RB solution. This latter is given by

$$\boldsymbol{r}(\boldsymbol{\mu}) = \boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}). \tag{4.15}$$

In fact, the following relation (straightforward to show) holds:

$$\boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{e}_N(\boldsymbol{\mu}) = \boldsymbol{r}(\boldsymbol{\mu}) \tag{4.16}$$

In section 4.4 we will use these quantities in order to obtain a suitable error bound, required by the greedy algorithm for the construction of the reduced space. Before doing that, we make some comment about the efficient Offline-Online procedure.

4.2.1 Offline-Online procedure

System (4.8) is nominally of small size: a set of N linear algebraic equations in N unknowns. However, the formation of the matrix and the right hand side, involves entities associated with our $\mathcal{N}$-dimensional BE approximation space. Fortunately, we can rely on affine parameter dependence (or onto the Empirical Interpolation already performed) to construct very efficient Offline-Online procedure. In particular, system (4.8) can be expressed in matrix form as

$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{A}_N^l\right) \boldsymbol{u}_N(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}_N^m. \tag{4.17}$$

Thus, we can write

$$\boldsymbol{A}_N^l = \mathcal{Z}^T \boldsymbol{A}^l \mathcal{Z}, \quad \boldsymbol{b}_N^m = \mathcal{Z}^T \boldsymbol{b}^m. \tag{4.18}$$

In this way, computation entails an expensive $\boldsymbol{\mu}$-independent Offline stage performed only once, and an Online stage for any chosen parameter value $\boldsymbol{\mu} \in \mathcal{D}$. During the former the BE structures $\{\boldsymbol{H}^l\}_{l=1}^L$ and $\{\boldsymbol{h}^m\}_{m=1}^M$, as well as the basis matrix $\mathcal{Z}$, are computed and stored. In the latter, for any given $\boldsymbol{\mu}$, all the $\gamma_l(\boldsymbol{\mu})$ and $\gamma_m(\boldsymbol{\mu})$ coefficients are evaluated, and the $N \times N$ linear system (4.17) is assembled and solved, in order to get the RB approximation $\boldsymbol{u}_N(\boldsymbol{\mu})$. Then, the full solution can be recovered simply as

$$\boldsymbol{u}(\boldsymbol{\mu}) = \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}) \tag{4.19}$$

The Online operation count is $O(LN^2)$ to assemble the matrix, $O(N^3)$ to invert it and $O(MN)$ to get the right hand side in (4.17). The Online solution requires - thanks to the hierarchy assumption - only $O(LN_{max}^2) + O(MN_{max})$ operations: for any given N, we may extract the necessary RB $N \times N$ matrices as principal submatrices of the corresponding $N_{max} \times N_{max}$ quantities. The Online cost to evaluate $\boldsymbol{\mu} \rightarrow \boldsymbol{u}^N(\boldsymbol{\mu})$ is thus independent of $\mathcal{N}$.

4.3 Strategies for reduced order space construction

In the last decades a considerable progress has been made in strategies for reduced order space construction, with several classes of methods emerging. In [9] Benner et al. give a general overview about these methods. We limit ourselves to describe (and use) two main methods for choosing the basis (and then build our reduced order models), namely the *Proper Orthogonal Decomposition* (POD) and the *(greedy) reduced basis* (RB) methods, which are the most suitable for parametrized problems.

4.3.1 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 uncorrelated variables (called POD modes, or principal components), the first few modes ideally retaining most of the 'energy' present in all of the original variables. The POD method relies on the use of the singular value decomposition (SVD) algorithm (see e.g. [62]-[49]. Consider a discrete set of n_{train} snapshot vectors $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{n_{\text{train}}}\} \in V^h$, and form the snapshot matrix $\mathcal{U} \in \mathbb{R}^{\mathcal{N} \times n_{\text{train}}}$ having them as column vectors:

$$\mathcal{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{n_{\text{train}}}] \equiv [\boldsymbol{u}(\boldsymbol{\mu}_1), \cdots, \boldsymbol{u}(\boldsymbol{\mu}_{n_{\text{train}}})]. \tag{4.20}$$

The SVD of $\mathcal{U}$ reads

$$\mathcal{V}^T \mathcal{U} \mathcal{W} = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \tag{4.21}$$

where

$$\mathcal{W} = [\zeta_1, \zeta_2, \cdots, \zeta_{\mathcal{N}}] \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}} \tag{4.22}$$

and

$$\mathcal{V} = [\Psi_1, \Psi_2, \cdots, \Psi_N] \in \mathbb{R}^{n_{\text{train}} \times n_{\text{train}}} \tag{4.23}$$

are orthogonal matrices, whereas

$$\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_r) \tag{4.24}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$; here $r \leq n_{\text{train}}$ is the rank of $\mathcal{U}$.

For any $N \leq n_{\text{train}}$, the POD basis of dimension N is defined as the set of the first N left singular vectors $[\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_N]$ of $\mathcal{U}$, that is, the basis matrix is given by

$$\mathcal{Z} = [\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_N] \in \mathbb{R}^{\mathcal{N} \times N}. \tag{4.25}$$

By construction, the POD basis is orthonormal. Furthermore, it can be shown that the energy contained in the first N modes is

$$E(\mathcal{Z}) = \sum_{i=N+1}^{\mathcal{N}} \sigma_i^2, \tag{4.26}$$

so that the error in the POD basis is equal to the squares of the singular values corresponding to the neglected POD modes. In this way, we can select N so that $E(\mathcal{Z}) \leq \epsilon_{tol}^*$, for a prescribed tolerance ϵ_{tol}^*. To do this, it is sufficient to choose N such that

$$I(N) = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{\mathcal{N}} \sigma_i^2} \geq 1 - \delta, \tag{4.27}$$

that is, by requiring that the energy retained by the last $\mathcal{N} - N$ modes is equal to δ, being δ as small as desired; $I(N)$ is referred to as the relative information content of the POD basis.

Typically a POD approach to built an RB space is more expensive than the greedy approach. As we will see, in this latter, we only need to compute the N - typically very few – high-fidelity retained snapshots, whereas in the POD approach we must compute n_{train} – typically very many – high-fidelity candidate snapshots, as well as the solution of a SVD problem. Besides, (4.26) provides information about the amount of energy neglected by the selected POD modes, that is an indication in the L^2-norm. On the other hand, in the case of the greedy algorithm, we rely on an a posteriori error estimator.

4.3.2 Greedy algorithm

A greedy algorithm is a general procedure to approximate each element of a compact set $K \in V$ by a subspace of properly selected elements of K. We refer to [49] for a general overview about the algorithm.

Let us denote by $\Xi_{train} = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{n_{\text{train}}}\}$ a large train sample of parameter values in $\mathcal{D}$ which will serve to select the RB space. Moreover, let us denote by ϵ^*_{tol} a prescribed tolerance for the stopping criterion of the greedy algorithm. The idea behind this sampling strategy is that, at the n-th iteration, the greedy algorithm adds to the retained snapshots that particular candidate snapshot – over all candidate snapshots $u(\boldsymbol{\mu}), \boldsymbol{\mu} \in \Xi_{train}$ – which is worst approximated by the already computed basis V_{n-1}.

The greedy sampling strategy can be implemented as follows (see e.g. [50]):

```
$S_1 = \{\boldsymbol{\mu}^1\}$;
compute $u_h(\boldsymbol{\mu}^1)$;
$V_1 = \text{span}\{u_h(\boldsymbol{\mu}^1)\}$;
for $N = 2, \dots$
    $\boldsymbol{\mu}^N = \arg\max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \Delta_{N-1}(\boldsymbol{\mu})$;
    $\varepsilon_{N-1} = \Delta_{N-1}(\boldsymbol{\mu}^N)$;
    if $\varepsilon_{N-1} \leq \varepsilon^*_{\text{tol}}$                    (4.28)
        $N_{\text{max}} = N - 1$;
    end;
    compute $u_h(\boldsymbol{\mu}^N)$;
    $S_N = S_{N-1} \cup \{\boldsymbol{\mu}^N\}$;
    $V_N = V_{N-1} \cup \text{span}\{u_h(\boldsymbol{\mu}^N)\}$;
end.
```

Here $\Delta_N(\boldsymbol{\mu})$ is a reliable, accurate and cheap error estimator, whose construction is the object of the following section.

4.4 A posteriori error estimation

Effective a posteriori error bounds for the solution are crucial both for the efficiency of the basis construction and the reliability of the RB approximation. This error bound must show some relevant features:

i. it must be rigorous, that is, it must be valid for all N and for all parameter values in the parameter domain $\mathcal{D}$;
ii. it must be reasonably sharp. In fact, a too much conservative error bound can yield inefficient approximations, or bring to unnecessary safety margins;
iii. it must be very efficient, that is, the online cost for its evaluation must be independent of $\mathcal{N}$.

We now provide a basic derivation of the a posteriori error bound in the case of a linear elliptic problem from an algebraic standpoint. If we start from (4.13), we can write the following relation:

$$\boldsymbol{A}(\boldsymbol{\mu})\left(\boldsymbol{u}(\boldsymbol{\mu}) - \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu})\right) = \boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}). \tag{4.29}$$

Exploiting (4.14) and (4.15) in (4.29) we can obtain the error as

$$\begin{aligned} \|\boldsymbol{e}(\boldsymbol{\mu})\| &= \|\boldsymbol{A}^{-1}(\boldsymbol{\mu})\boldsymbol{r}(\boldsymbol{\mu})\| \\ &\leq \|\boldsymbol{A}^{-1}(\boldsymbol{\mu})\|\|\boldsymbol{r}(\boldsymbol{\mu})\|, \end{aligned} \tag{4.30}$$

where $\|\boldsymbol{A}^{-1}(\boldsymbol{\mu})\|$ plays the role of the stability factor. When dealing with the V-norm (or the energy norm), the stability factor is nothing but the $\boldsymbol{\mu}$-dependent coercivity constant.

In the following, we adopt a ‘surrogate’ error estimator based only on the residual in L^2 norm:

$$\Delta_N(\boldsymbol{\mu}) = \frac{\|\boldsymbol{r}(\boldsymbol{\mu})\|_2^2}{\|\boldsymbol{b}(\boldsymbol{\mu})\|_2^2} = \frac{\|\boldsymbol{b}(\boldsymbol{\mu}) - \boldsymbol{A}(\boldsymbol{\mu})\mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu})\|_2^2}{\|\boldsymbol{b}(\boldsymbol{\mu})\|_2^2}. \tag{4.31}$$

Since we can express $\boldsymbol{A}(\boldsymbol{\mu})$ and $\boldsymbol{b}(\boldsymbol{\mu})$ through the following affine expansions:

$$\boldsymbol{A}(\boldsymbol{\mu}) = \sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{A}^l, \quad \boldsymbol{b}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m, \tag{4.32}$$

we can compute the norm of the residual as follows:

$$\begin{aligned} \|\boldsymbol{r}\|_2^2 &= (\boldsymbol{r}, \boldsymbol{r}) = (\boldsymbol{b} - \boldsymbol{A}\mathcal{Z}\boldsymbol{u}_N, \boldsymbol{b} - \boldsymbol{A}\mathcal{Z}\boldsymbol{u}_N) \\ &= (\boldsymbol{b}, \boldsymbol{b}) - 2\,(\boldsymbol{b}, \boldsymbol{A}\mathcal{Z}\boldsymbol{u}_N) + (\boldsymbol{A}\mathcal{Z}\boldsymbol{u}_N, \boldsymbol{A}\mathcal{Z}\boldsymbol{u}_N) \\ &= \mathcal{C}_b + \mathcal{C}_m + \mathcal{C}_a, \end{aligned} \tag{4.33}$$

where, for the sake of readability, we have omitted the dependence on $\boldsymbol{\mu}$ in these terms. Exploiting the Offline-Online splitting, we can compute $\mathcal{C}_b$, $\mathcal{C}_m$, $\mathcal{C}_a$ as follows:

$$\begin{aligned}
\mathcal{C}_b &= \sum_{m=1}^{M}\sum_{m'=1}^{M}\gamma^m(\boldsymbol{\mu})\gamma^{m'}(\boldsymbol{\mu})\left(\boldsymbol{b}^m,\boldsymbol{b}^{m'}\right)\\
\mathcal{C}_m &= -2\sum_{m=1}^{M}\sum_{l=1}^{L}\gamma^m(\boldsymbol{\mu})\gamma^{l}(\boldsymbol{\mu})\left(\boldsymbol{b}^m,\boldsymbol{A}^l\mathcal{Z}\right)\boldsymbol{u}_N \qquad (4.34)\\
\mathcal{C}_a &= \sum_{l=1}^{L}\sum_{l'=1}^{L}\gamma^l(\boldsymbol{\mu})\gamma^{l'}(\boldsymbol{\mu})\boldsymbol{u}_N^T\left(\boldsymbol{A}^l\mathcal{Z},\boldsymbol{A}^{l'}\mathcal{Z}\right)\boldsymbol{u}_N,
\end{aligned}$$

thus by evaluating (and then storing) all the $\boldsymbol{\mu}$-dependent quantities appearing in (4.34). We can now summarize equations (4.33) and (4.34) in the residual final expression:

$$\begin{aligned}
\|\boldsymbol{r}\|_2^2 &= \sum_{m=1}^{M}\sum_{m'=1}^{M}\gamma^m(\boldsymbol{\mu})\gamma^{m'}(\boldsymbol{\mu})\boldsymbol{b}^{m'^T},\boldsymbol{b}^{m'}\\
&- 2\sum_{m=1}^{M}\sum_{l=1}^{L}\gamma^m(\boldsymbol{\mu})\gamma^{l}(\boldsymbol{\mu})\left(\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{b}^m\right)\boldsymbol{u}_N\\
&+ \sum_{l=1}^{L}\sum_{l'=1}^{L}\gamma^l(\boldsymbol{\mu})\gamma^{l'}(\boldsymbol{\mu})\boldsymbol{u}_N^T\left(\mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{A}^{l'}\mathcal{Z}\right)\boldsymbol{u}_N. \qquad (4.35)
\end{aligned}$$

Thus, during the Offline stage, we end up to evaluate

$$\begin{aligned}
&M^2 \text{ terms} \quad \boldsymbol{b}^{m'^T}\boldsymbol{b}^m \in \mathbb{R}\\
&M\times L \text{ terms} \quad \mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{b}^m \in \mathbb{R}^N \qquad (4.36)\\
&L^2 \text{ terms} \quad \mathcal{Z}^T\boldsymbol{A}^{l^T}\boldsymbol{A}^{l'}\mathcal{Z} \in \mathbb{R}^{N\times N}. \qquad (4.37)
\end{aligned}$$

Although the mere residual is not a rigorous error bound, it gives good results in the selection of the reduced basis. Terms (4.36) are of dimension N but, for the particular problem at hand, we have many affine terms. This increases the time needed for the Offline stage of the greedy RB method; in chapter 5 we will provide some results and comparisons with respect to the POD Offline stage.

4.5 The case of potential flows about NACA profiles

In this section we apply the ideas discussed in the previous sections to our problem of interest, given by the parametrized system (3.54). We recall that system (3.54) has been obtained after the empirical interpolation method, and reads as:

$$\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l + \boldsymbol{A}^M + \boldsymbol{A}^k(\boldsymbol{\mu}) + \boldsymbol{A}^w\right)\boldsymbol{u}(\boldsymbol{\mu}) = \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{b}^m; \tag{4.38}$$

note that this system is of dimension $\mathcal{N} \times \mathcal{N}$.

Once we have computed the basis functions either by POD or (greedy) RB, we can exploit (4.18) to assemble RB matrices and right hand side in order to obtain the reduced order system

$$\begin{aligned}\left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l_N + \boldsymbol{A}^M_N + \boldsymbol{A}^k_N(\boldsymbol{\mu}) + \boldsymbol{A}^w_N\right)\boldsymbol{u}_N(\boldsymbol{\mu}) &= \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{h}^m_N \\ \boldsymbol{A}_N(\boldsymbol{\mu})\boldsymbol{u}_N(\boldsymbol{\mu}) &= \boldsymbol{b}_N(\boldsymbol{\mu}),\end{aligned} \tag{4.39}$$

where

$$\begin{aligned}\boldsymbol{A}_N(\boldsymbol{\mu}) &= \left(\sum_{l=1}^{L} \gamma^l(\boldsymbol{\mu})\boldsymbol{N}^l_N + \boldsymbol{A}^M_N + \boldsymbol{A}^k_N(\boldsymbol{\mu}) + \boldsymbol{A}^w_N\right) \\ \boldsymbol{b}_N(\boldsymbol{\mu}) &= \sum_{m=1}^{M} \gamma^m(\boldsymbol{\mu})\boldsymbol{h}^m_N.\end{aligned} \tag{4.40}$$

Now the system is of dimension $N \times N$, with $N \ll \mathcal{N}$. In figure 4.2 we sketch the global construction/evaluation of the proposed ROM for the parametrized problem at hand; we recall that we decided not to apply EIM to the elements related to Kutta condition, in order to impose this latter exactly. Moreover, the elements related to Kutta condition are very few and very fast to be evaluated. Thus, we can compute them during the Online stage without any sensible efficiency loss.

Once we have computed the RB solution, we can recover the full-order solution as

$$\boldsymbol{u}(\boldsymbol{\mu}) = \mathcal{Z}\boldsymbol{u}_N(\boldsymbol{\mu}), \tag{4.41}$$

and through the post processing procedures shown in section 2.5 we can compute all the outputs of interest, such as the pressure coefficient, the velocity and pressure fields all over the domain.

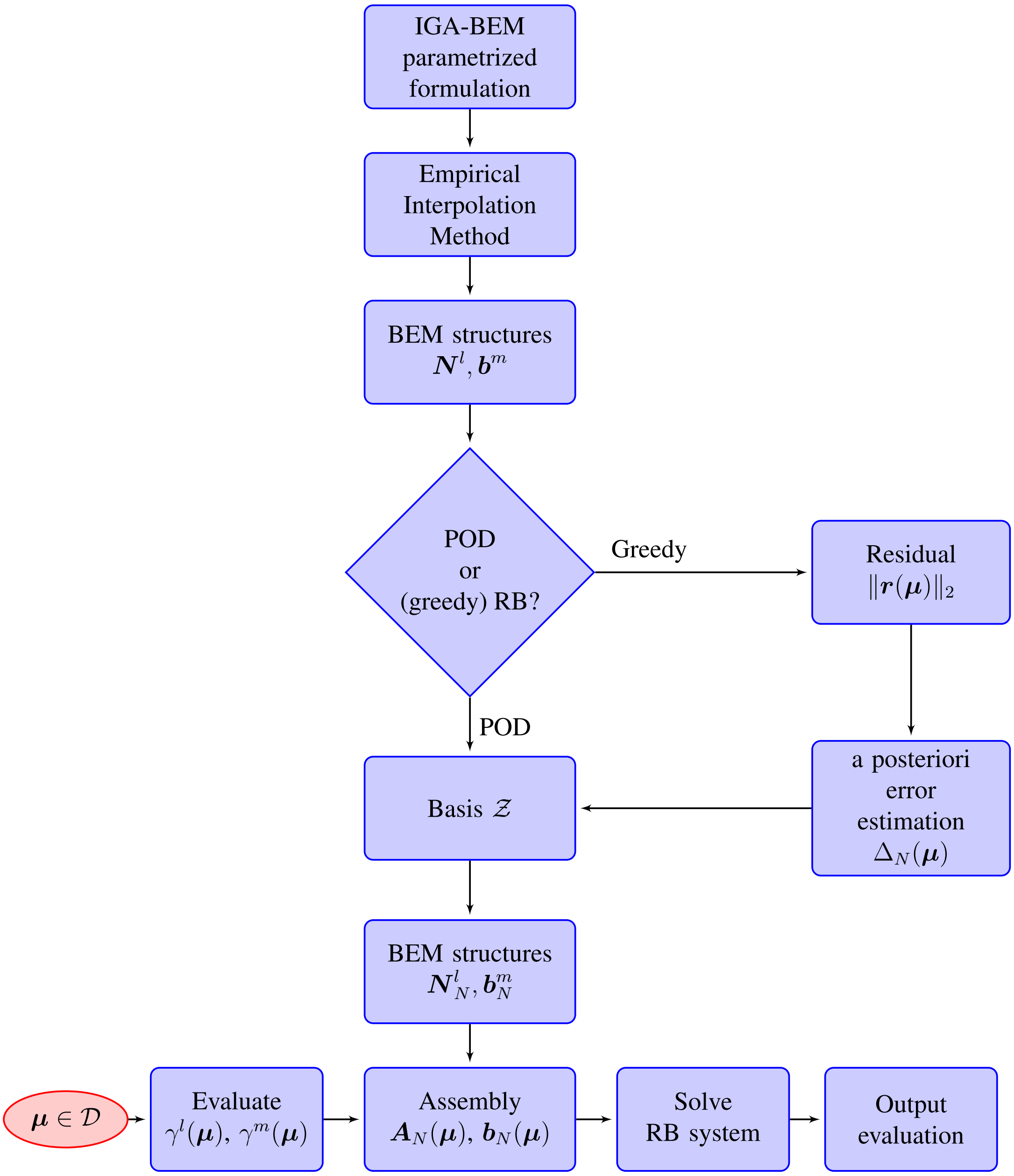

Figure 4.2: Numerical procedure for the solution of nonaffinely parametrized problems through reduced order models.

Chapter 5

Numerical results

In this chapter we show the results obtained through the numerical framework presented so far, as well as some details related to its computational performance.

In particular, 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 (NACA 0012 and NACA 4412) profiles. These airfoils are well-known benchmarks in the aeronautical literature, thus we can rely on several references in order to evaluate the methods we have developed in this work. In particular we refer to [1], [2], [21] and [47] for the experimental results; we refer to [17] and [41] for the numerical results. Among numerical results considered for comparison and validation, the former (Xfoil) deals with a classical panel method, the latter deals with a B-splines method, more similar to ours.

First, we provide results obtained through the IGA-BEM method (section 5.1), then, in section 5.2, we show the results of the empirical interpolation method for the affine approximation of the parameter dependent terms of the problem, as well as the results of EI-IGA-BEM (Empirical Interpolation, Isogeometric Analysis, Boundary Element Method). Finally, in section 5.3, we show the results obtained with the ROMs we have considered in this work, namely POD and RBM methods, and we compare their computational performances.

Throughout the work, we have used the software *Matlab* [45]. We have taken this decision mainly for two reason:

i. we could use some Empirical Interpolation Method and (greedy) Reduced Basis method libraries already implemented in Matlab [31], as starting blocks to develop our framework;
ii. since this work deals with the coupling of different techniques for the first time ever (such as in the case of IGA-RBM coupling and BEM-RBM coupling in presence of complex geometrical parametrization), the choice of a software that allows easy debugging and data handling in an intuitive way is of crucial importance.

5.1 IGA-BEM validation

In the following we present the results and comparisons for the high-fidelity method, namely the Isogeometric Boundary Element Method (IGA-BEM).

First, after some preliminary tests, we have tuned the values of some numerical parameters that allow to obtain good results. On the one hand, we noted that, in order to obtain a good description of the geometry, it is sufficient to take third degree B-spline basis functions, with 76 control points. Moreover, we noticed that the choice of the parametrization is of fundamental importance for the accuracy of IGA-BEM, as well depicted in figure 5.1. For this reason, we adopted an arc length parametrization.

On the other hand, for the integrals in (2.27) and (2.28), we need to take 4 Gauss points for the external integrals and 10 Telles points for the internal integrals. We recall that we exploit Telles algorithm in order to overcome the problem of integration of singular functions. As explained in [60], this algorithm is self-adaptive, improves the accuracy of Gaussian quadrature schemes within the near-singularity range and it is more efficient (in terms of computational cost) than other techniques used for the evaluation of singular integrals [37]. Since the problem at hand is subject to a constraint (Kutta condition), the degrees of freedom are not 76, but 77. In table 5.1 we summarize some numerical details of IGA-BEM. In particular, by taking only 77 degrees of freedom, we already perform a first numerical reduction, compared to the 200 degrees of freedom necessary to Xfoil to compute the results. In fact, as already mentioned, isogeometric analysis yields a remarkable improvement regarding computational efficiency and accuracy.

Table 5.1: 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

In the following we present the results in terms of pressure coefficient, lift coefficient and streamline visualization, as explained in section 2.5.

5.1.1 Reparametrization of the geometry

In the isogeometric framework, there are several ways to parametrize the airfoil geometry. This choice effects the results, especially in terms of pressure coefficient. As can be seen in figure 5.1, the parametrization may give an oscillation (related to the determinant of the Jacobian of the mapping) in the results. In order to overcome this problem, we have decided to adopt an *arc length* parametrization. Note that the lift coefficient, which is an integral output, is less affected by these oscillations.

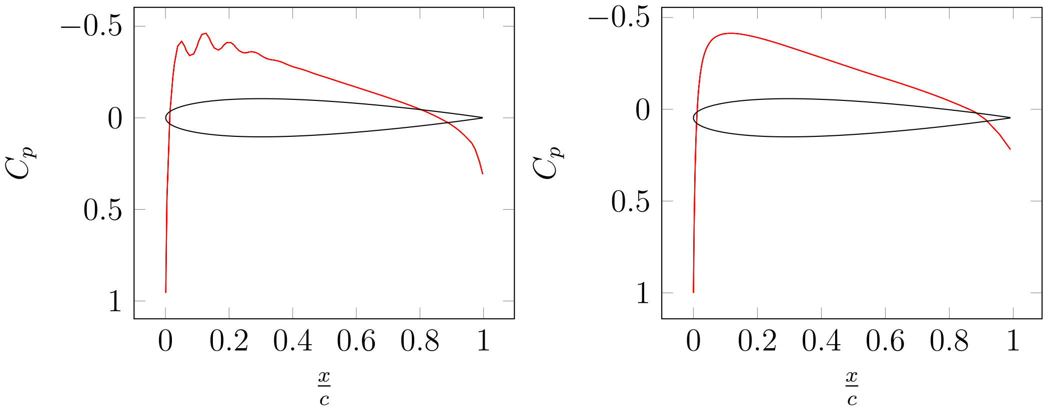

Figure 5.1: Pressure coefficient for NACA 0012: uniform parametrization (left), arc length parametrization (right); $\alpha = 0\,\mathrm{deg}$.

5.1.2 NACA 0012 profile

As a first test, we consider the NACA 0012 airfoil. We remark that the most relevant peculiarity of NACA 0012 is that it is a symmetric profile. Hence, we have to check that at null angle of attack both the lift is null and the pressure coefficient is equal on the lower and upper surfaces. This is the first numerical test for the validation of IGA-BEM (figure 5.1 and 5.3). In figure 5.2, we show the comparison of the pressure coefficient among IGA-BEM, experimental data and Xfoil calculations for some angles of attack. We remark that we use Xfoil in its inviscid version. This is coherent with the fact that also IGA-BEM is an inviscid model, based on the solution of the Laplace equation.

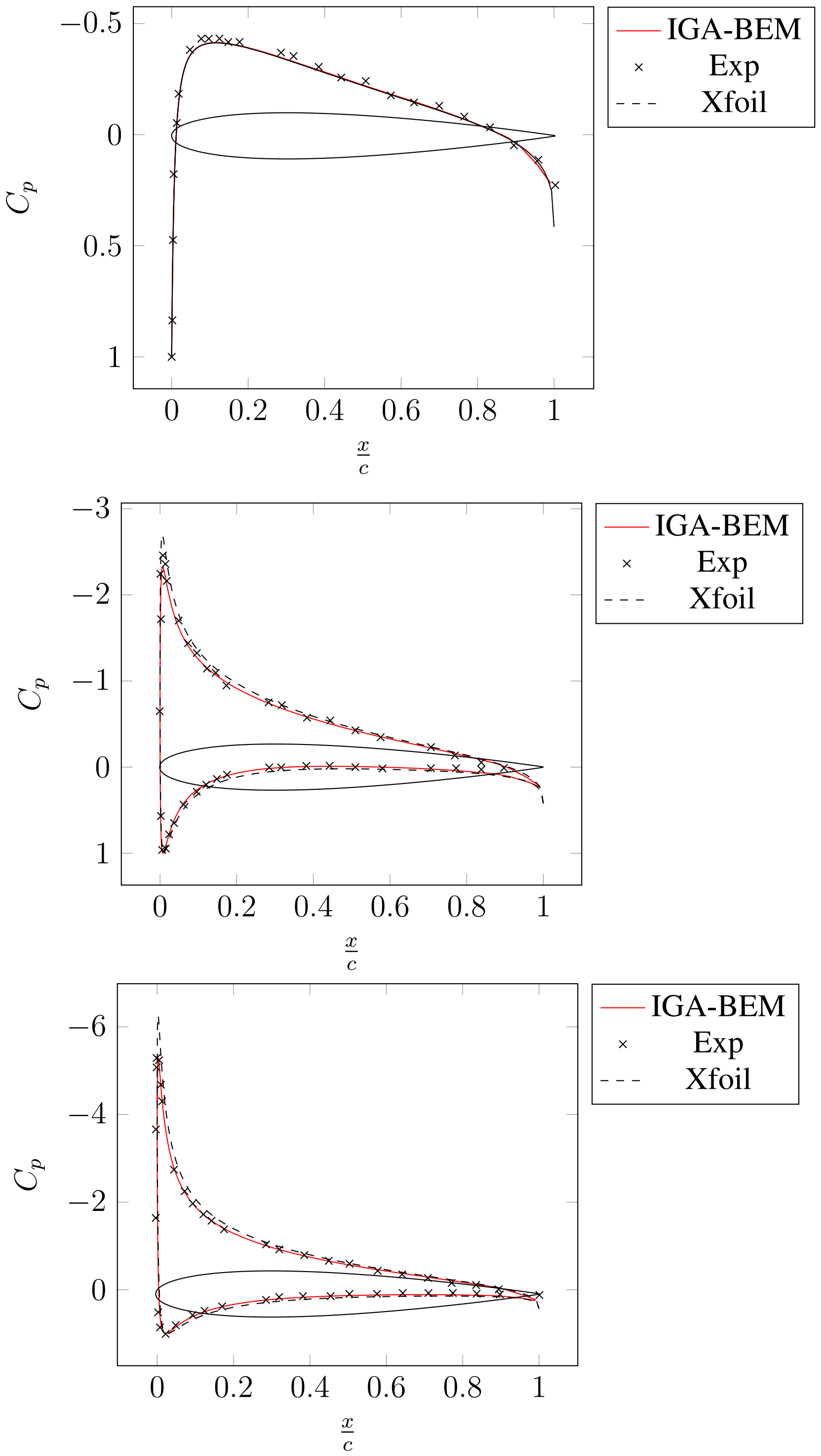

Figure 5.2: Comparison of the pressure coefficient for NACA 0012: IGA-BEM, experimental data [21] and Xfoil; $\alpha = 0\,\text{deg}, 6\,\text{deg}, 10\,\text{deg}$.

The results obtained are very close both to experimental data and Xfoil calculations.

In figure 5.3, we show the lift coefficient curve and we compare it with experimental data and Xfoil calculations. We note that the Xfoil curve prediction is steeper than our prediction and experimental data.

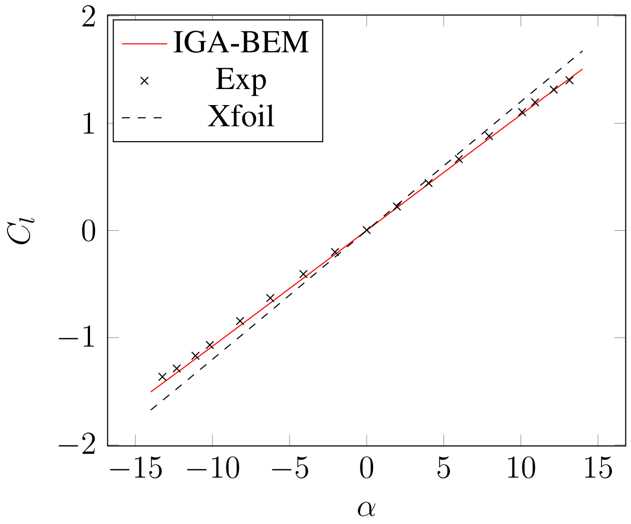

Figure 5.3: $C_L(\alpha)$ curve comparison for NACA 0012: IGA-BEM, experimental data [2] and Xfoil.

Finally, in figure 5.4, we provide the visualization of the streamlines for four different angles of attack. This is a useful output in order to understand the general behaviour of the flow around the airfoil.

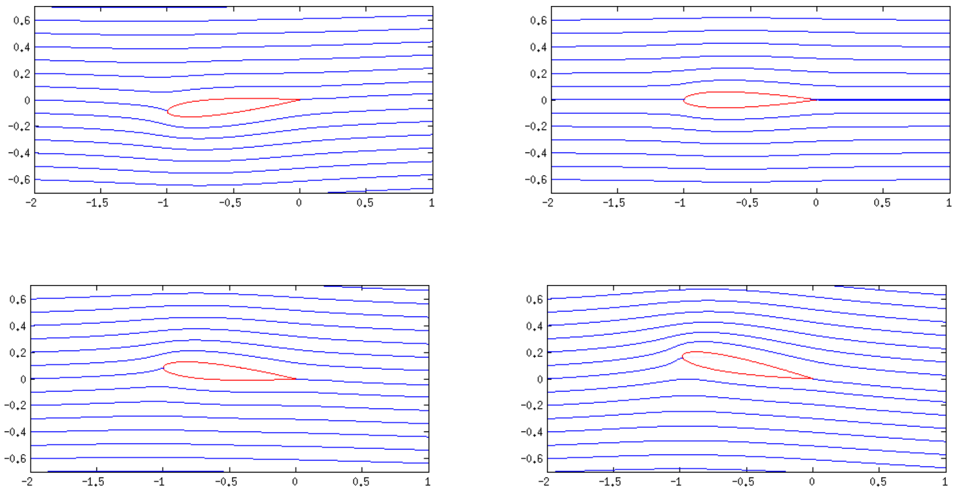

Figure 5.4: Streamlines visualization for Naca 0012; $\alpha = -5\,\mathrm{deg}, 0\,\mathrm{deg}, 5\,\mathrm{deg}, 10\,\mathrm{deg}$.

5.1.3 NACA 4412 profile

As a second test, among several very popular NACA 4-digits non symmetric airfoils, we have chosen 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 [41]. First, in figure 5.6, we show the comparison of the pressure coefficient among IGA-BEM, experimental data and Xfoil calculations for some angles of attack; as before we use Xfoil in its inviscid version. Also for the NACA 4412 profile, the results obtained are very close to experimental data and Xfoil calculations. In figure 5.5, we show the lift coefficient curve and we compare it with experimental data and Xfoil calculations. We highlight that the Xfoil curve prediction is steepest than our prediction and experimental data also for the NACA 4412 profile. Moreover, we want to test IGA-BEM with respect to the B-splines based method presented in [41]. Thus, in figure 5.7, we provide the pressure coefficient comparison with Xfoil and the already mentioned method [41]. We note that there is a discrepancy in the results: this difference can be remarked also in figure 5.5, in terms of gap between Xfoil and IGA-BEM lift coefficients. Thus, if we compare the 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 figure 5.7. Finally, we provide the flow visualization through streamlines in figure 5.8. We remark that this is a qualitative result, which is useful in order to understand the behaviour of the flow. The results provided so far show the reliability of our IGA-BEM framework. This means that the implemented IGA-BEM performs in a very good way in predicting the results in terms of pressure and lift coefficients. Moreover, it uses less than a half of the Xfoil degrees of freedom. After the preliminary validation of the high-fidelity method, we can proceed now with all the steps that leads to the construction and validation of the reduced order models.

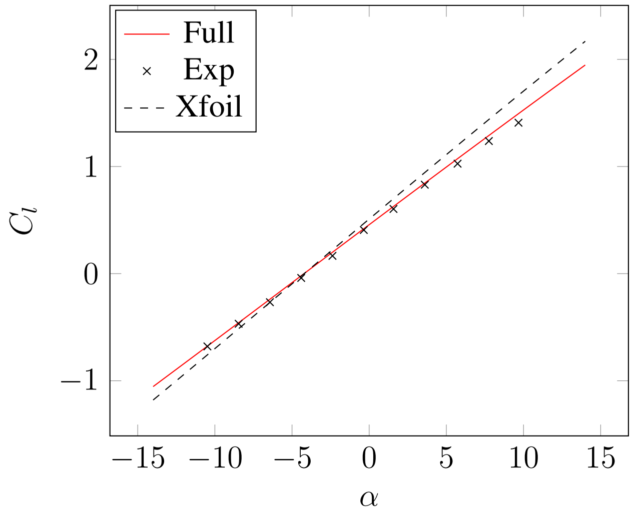

Figure 5.5: $C_L(\alpha)$ curve comparison for NACA 4412: IGA-BEM, experimental data [2] and Xfoil.

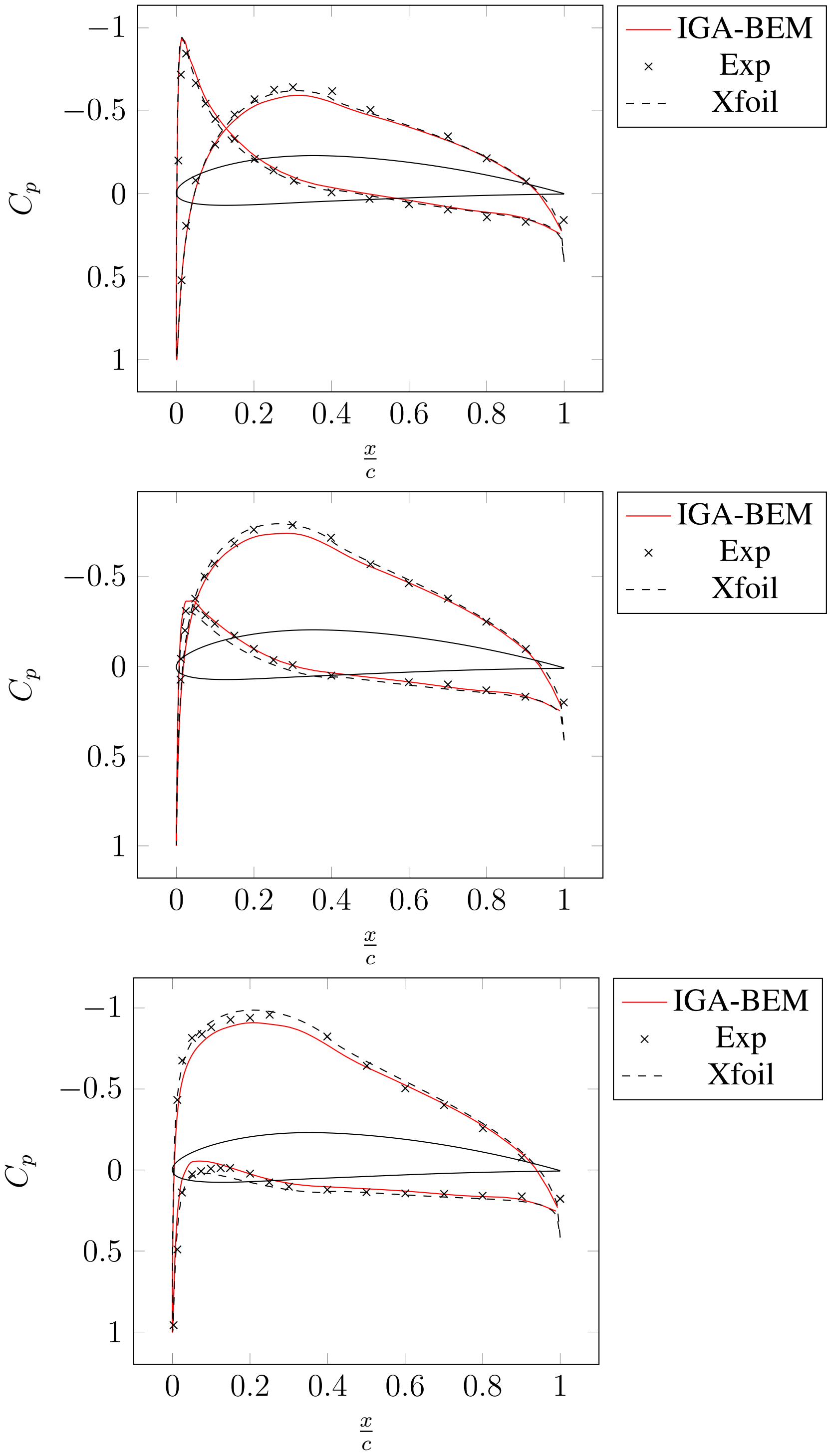

Figure 5.6: Comparison of the pressure coefficient for NACA 4412: IGA-BEM, experimental data [47] and Xfoil; $\alpha = -2\,\mathrm{deg}, 0\,\mathrm{deg}, 1.87\,\mathrm{deg}$.

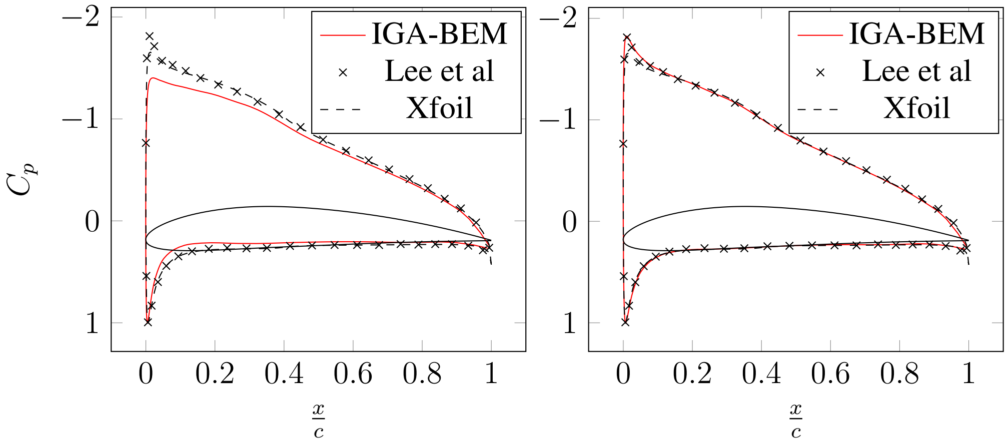

Figure 5.7: Comparison of the pressure coefficient for NACA 4412: IGA-BEM, B-splines based method [41] and Xfoil; $\alpha = 5\,\mathrm{deg}$ (left), $C_l = 1.11$ (right).

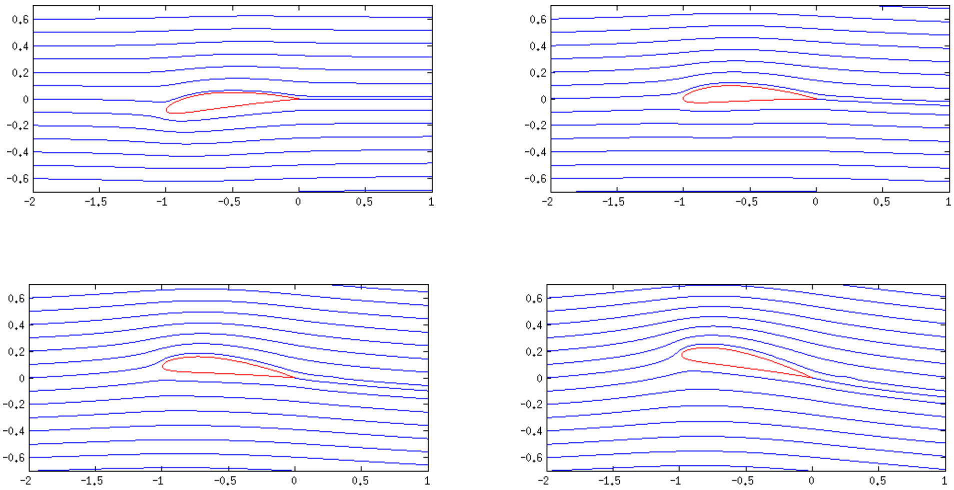

Figure 5.8: Streamlines visualization for Naca 4412; $\alpha = -5\,\mathrm{deg}, 0\,\mathrm{deg}, 5\,\mathrm{deg}, 10\,\mathrm{deg}$.

5.2 Empirical interpolation method

In this section we present the results for the empirical interpolation method. First we show how the parameter dependent terms appearing in (3.17)-(3.19) are approximated by a series of affine terms, through the EIM procedure of section 3.2. Then, we analyse the impact of the EIM matrix/vector approximation on the IGA-BEM high-fidelity solution, by recovering some of the results already presented for IGA-BEM. Moreover, we highlight how the empirical interpolation method reduces the computational cost of the problem. This is related to the efficiency of the online EIM step in building the system structures, in particular the system matrix, in the following reduced-order models.

5.2.1 Approximation of parameter dependent terms

First of all, we show how the greedy algorithm of EIM selects the values of the parameter $\boldsymbol{\mu}$ and how the error decrease according to the number of affine terms taken, both for the matrix $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ and the vector $\boldsymbol{b}(\boldsymbol{\mu})$. To initialize the greedy algorithm, we have prescribed a tolerance $\epsilon^*_{EIM} = 10^{-8}$ and a random train of 600 parameter values. We have decided to take a tolerance of 10^{-8} in order to neglect the error estimator Δ^{EIM} in the estimation of the ROM error in (4.5). In this way, Δ^{EIM} is orders of magnitude smaller than the estimator Δ_N. In the following, we will provide a deeper explanation of this.

In figure 5.9 we show the choice of the parameter values and the error (in L_∞-norm) related to $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$, respectively. In the same way, in figure 5.10 we show the choice of the parameter values and the error (in L_∞-norm) related to $\boldsymbol{b}(\boldsymbol{\mu})$, respectively. We recall that our parameter domain is made by 7 not connected subregions within the ranges $[i208, i620] \times [-20\,\text{deg}, 20\,\text{deg}]$, $i = 0, \cdots, 6$.

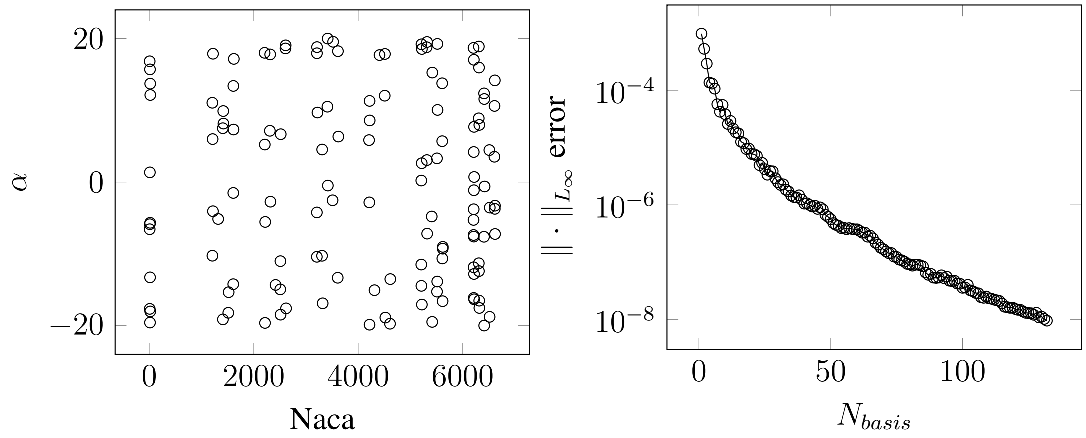

Figure 5.9: EIM approximation of $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$: sample values in the parameter space (left) and convergence of the greedy-EIM algorithm (right); $\epsilon^*_{EIM} = 10^{-8}$.

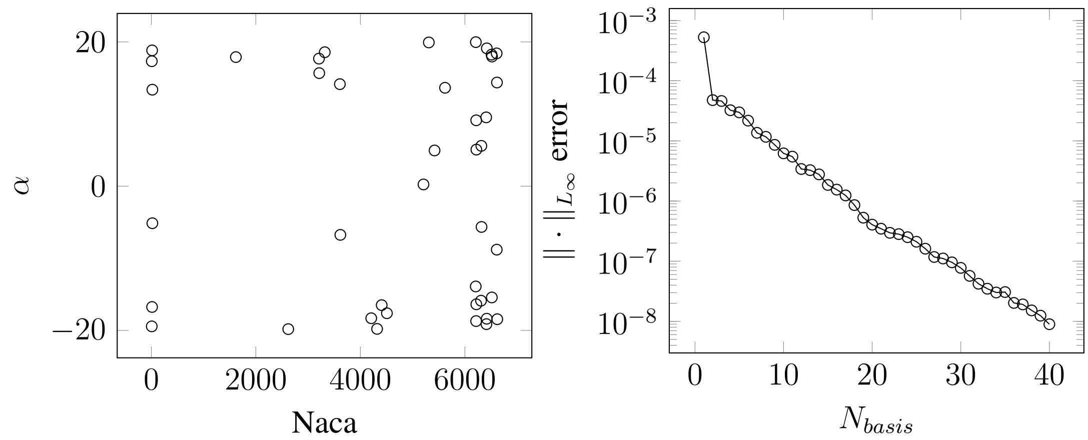

Figure 5.10: EIM approximation of $\boldsymbol{b}(\boldsymbol{\mu})$: sample values in the parameter space (left) and convergence of the greedy-EIM algorithm (right); $\epsilon^*_{EIM} = 10^{-8}$.

We recall that for each retained parameter value, EIM selects a couple of magic index i, j for the matrix (and a singe magic index i for the vector), corresponding to the ij-th (i-th) element of the matrix (vector) to be evaluated online. We note that the greedy algorithm of EIM selects only 132 point over $76 \times 76 = 5776$ elements of matrix $\tilde{\boldsymbol{N}}(\boldsymbol{\mu})$ for its approximation, to reach a precision of 10^{-8}. Concerning the matrix assembling, this leads to a great computational saving.

On the contrary, we note that the greedy algorithm of EIM selects 40 point over 76 elements of $\boldsymbol{b}(\boldsymbol{\mu})$ for its approximation, to read the same precision. Thus, concerning the right hand side assembling, this leads to a computational saving much less tangible.

From figure 5.10, we can see that EIM tend to select the parameters close to the boundaries of the parameter domain $\mathcal{D}$. This is a typical behaviour of every greedy algorithm.

5.2.2 Numerical results of EI-IGA-BEM

We now present the comparison between the results obtained through the EI-IGA-BEM and the IGA-BEM, that is, between the full-order problems where the algebraic structures are assembled relying on EIM or not. First, in order to check the reliability of the EIM, we have computed the error between the results of the two methods for a random set Ξ_{sample} of n_{sample} parameter, in order to estimate Δ^{EIM}.

We recall briefly the stability estimation for linear systems in the form (2.34). For a deeper knowledge on the matter, we refer to [51]. We can write the perturbed system, i.e. the linear system corresponding to the EIM assembling of the operators, as

$$(\boldsymbol{A} + \delta\boldsymbol{A})\,(\boldsymbol{u} + \delta\boldsymbol{u}) = \boldsymbol{b} + \delta\boldsymbol{b}, \tag{5.1}$$

and we want an estimate of $\delta\boldsymbol{u}$ in terms of $\delta\boldsymbol{A}$ and $\delta\boldsymbol{b}$. Here $\delta\boldsymbol{A}$ and $\delta\boldsymbol{b}$ play the role of EIM errors made on the approximation of the matrix $\boldsymbol{A}$ and the right hand side $\boldsymbol{b}$, respectively. We can write the following stability estimate

$$\frac{\|\delta \boldsymbol{u}\|}{\|\boldsymbol{u}\|} = \frac{K(\boldsymbol{A})}{1 - K(\boldsymbol{A})\|\delta \boldsymbol{A}\|/\|\boldsymbol{A}\|} \left(\frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|} + \frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|} \right), \tag{5.2}$$

where $K(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$ is the conditioning number of $\boldsymbol{A}$ and, for the problem at hand,

$$K(\boldsymbol{A}) = O(10^3). \tag{5.3}$$

Let us assume that the train sample Ξ_{sample} properly spans $\mathcal{D}$. Thus, we can obtain the error estimations $\|\delta \boldsymbol{A}\|$ and $\|\delta \boldsymbol{b}\|$ as follows

$$\begin{array}{lll} \|\delta \boldsymbol{A}\| & = \max_i \left\{ \left\| \boldsymbol{A}_E^i - \boldsymbol{A}_F^i \right\|_{L^\infty} \right\}_{i=1}^{n_{\text{sample}}} & \approx O(\epsilon_{EIM}^*) \\ \|\delta \boldsymbol{b}\| & = \max_i \left\{ \left\| \boldsymbol{b}_E^i - \boldsymbol{b}_F^i \right\|_{L^\infty} \right\}_{i=1}^{n_{\text{sample}}} & \approx O(\epsilon_{EIM}^*), \end{array} \tag{5.4}$$

respectively. Here, with the subscripts $_E$ and $_F$ we refer to the matrix and the right hand side of EI-IGA-BEM and IGA-BEM, respectively. For the problem at hand, (5.4) leads to

$$\begin{aligned} \frac{\|\delta \boldsymbol{A}\|}{\|\boldsymbol{A}\|} &\approx O(10^{-6}) \\ \frac{\|\delta \boldsymbol{b}\|}{\|\boldsymbol{b}\|} &\approx O(10^{-6}). \end{aligned} \tag{5.5}$$

We thus can provide an estimation Δ^{EIM} for the error $\frac{\|\delta \boldsymbol{u}\|}{\|\boldsymbol{u}\|}$ related to the EIM procedure. In fact, exploiting (5.2), we can write:

$$\frac{\|\delta \boldsymbol{u}\|}{\|\boldsymbol{u}\|} \leq \Delta^{EIM} \approx O(10^{-3}). \tag{5.6}$$

We obtain the same result if we directly compute for the sample previously chosen:

$$\frac{\|\delta \boldsymbol{u}\|}{\|\boldsymbol{u}\|} = \frac{\max_i \left\{ \|\boldsymbol{u}_E^i - \boldsymbol{u}_F^i\|_{L^\infty} \right\}_{i=1}^{n_{\text{sample}}}}{\max_i \left\{ \|\boldsymbol{u}^i\|_{L^\infty} \right\}_{i=1}^{n_{\text{sample}}}} \approx O(10^{-3}). \tag{5.7}$$

The choice of the EIM tolerance $\epsilon_{EIM}^* = 10^{-8}$ is thus motivated by this analysis. In table 5.2 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. We can see that, in this case, the computational reduction is moderate. In fact, the aim to impose a strict tolerance has leaded to the selection of many affine terms, which require more time to be evaluated. However, we highlight that the first aim of EIM is not to reduce the computational cost, but rather to express the non affine terms through an affine expansion, in view of developing efficient ROMs. For the sake of completeness, in the following we show the results obtained by means of the EI-IGA-BEM procedure for some configurations already analysed in the IGA-BEM framework in section 5.1. We can remark that the reconstruction of the pressure coefficient is absolutely equivalent.

Table 5.2: Some features of the empirical interpolation method.

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

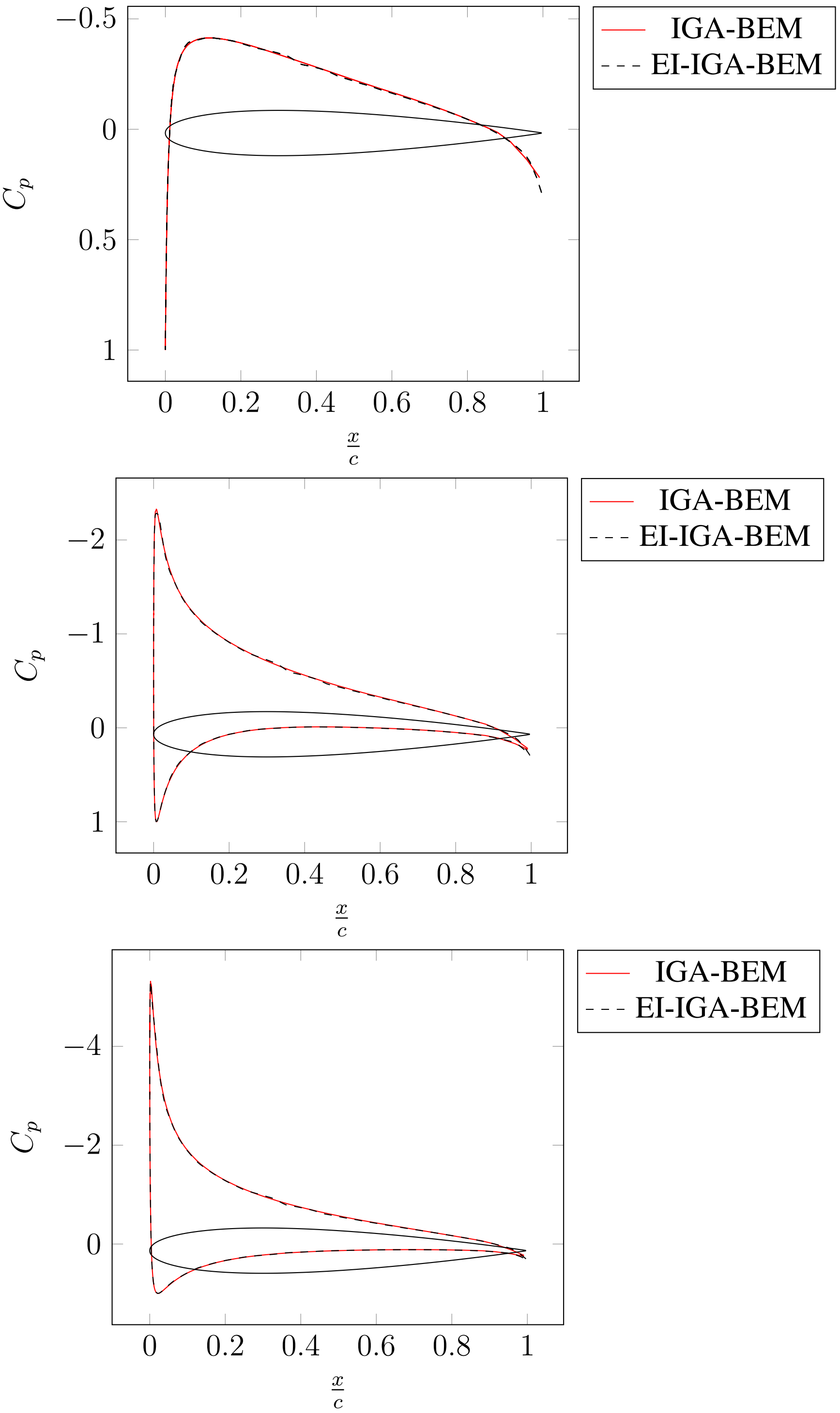

Figure 5.11: Comparison of the pressure coefficient for NACA 0012: IGA-BEM and EI-IGA-BEM; $\alpha = 0\,\mathrm{deg}, 6\,\mathrm{deg}, 10\,\mathrm{deg}$.

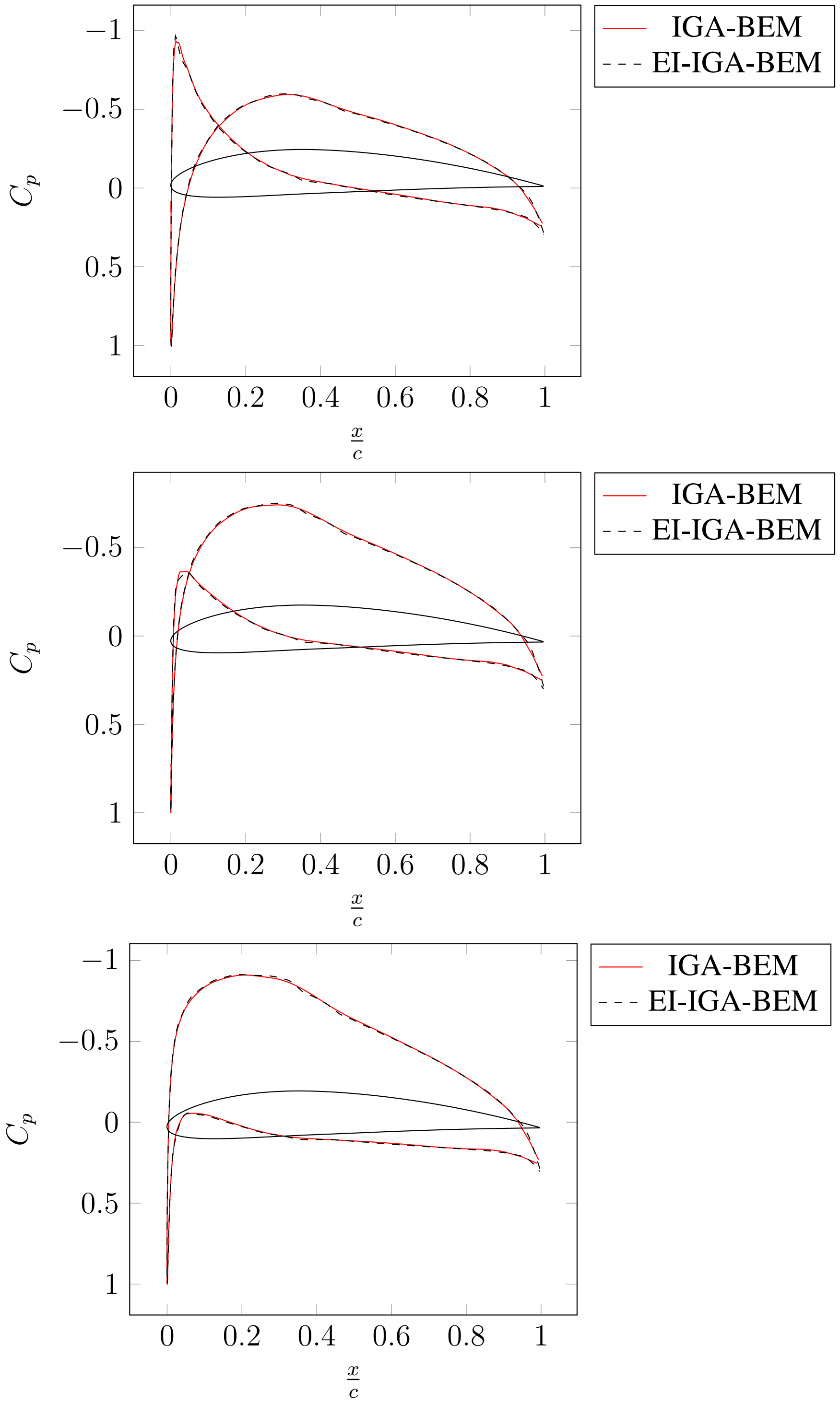

Figure 5.12: Comparison of the pressure coefficient for NACA 4412: IGA-BEM and EI-IGA-BEM; $\alpha = -2\,\mathrm{deg}, 0\,\mathrm{deg}, 1.87\,\mathrm{deg}$.

5.3 Reduced Order Models

We now present the numerical results obtained through the reduced order models implemented in this work, by considering first the proper orthogonal decomposition, then what concerns the greedy reduced basis method.

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, EI-IGA-BEM and experimental data. In the end, we provide a direct comparison between POD and RBM in terms of computational performances.

5.3.1 Proper Orthogonal Decomposition

We briefly recall the idea behind POD. Basically, during the offline stage, we first compute a (large) set of snapshots and then, through a Singular Value Decomposition, we transform the original variable into the so-called POD modes, the first few modes retaining most of the 'energy' present in the original variables. During the online stage, through (4.27), we choose the dimensionality of the reduced basis and we solve the new (reduced) system, thanks to a Galerkin projection.

For the problem at hand, we have taken a set of 600 snapshots and we have applied the SVD algorithm. The resulting singular values are depicted in figure 5.13. We remark that, through (4.26), the energy information contained in the chosen POD modes is related to these singular values. In particular, we point out that $O(10)$ POD modes are sufficient in order to explain almost all the energy of the system, thanks to (4.27).

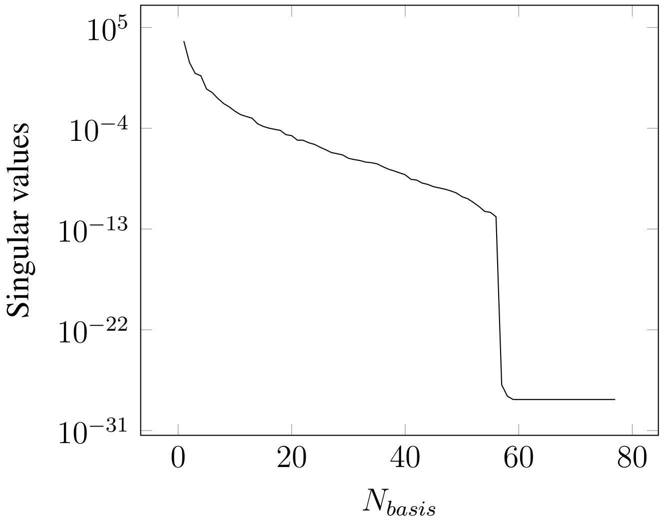

Figure 5.13: Eigenvalues of the Singular Value Decomposition.

As a first test, we have taken a set of 50 IGA-BEM and EI-IGA-BEM solutions for 50 random values of the parameters in order to asses the quality of the POD basis. Then, for these parameter values, we compute the POD solution. Moreover, this sample is the same also for the RBM convergence tests, shown later on. For all possible choices of the reduced space dimension, we compute the error (in L^2-norm) for all IGA-BEM and EI-IGA-BEM solutions in the sample. In figure 5.14 and 5.15 we show the maximum, minimum and mean errors between POD and EI-IGA-BEM solutions, and between POD and IGA-BEM solutions, respectively. From figure 5.14 we can see that the error has the same shape of the SVD eigenvalues in figure 5.13, which is what we expect.

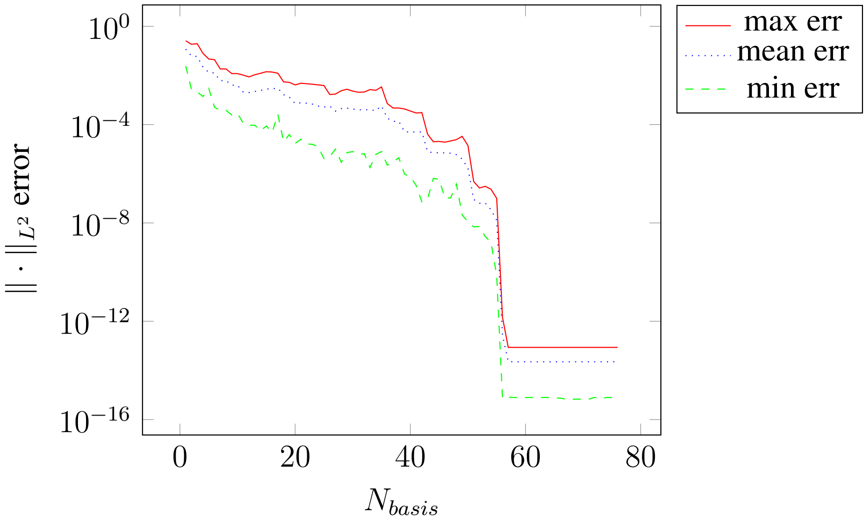

Figure 5.14: Error convergence between POD and EI-IGA-BEM solutions for a random train of 50 parameter values.

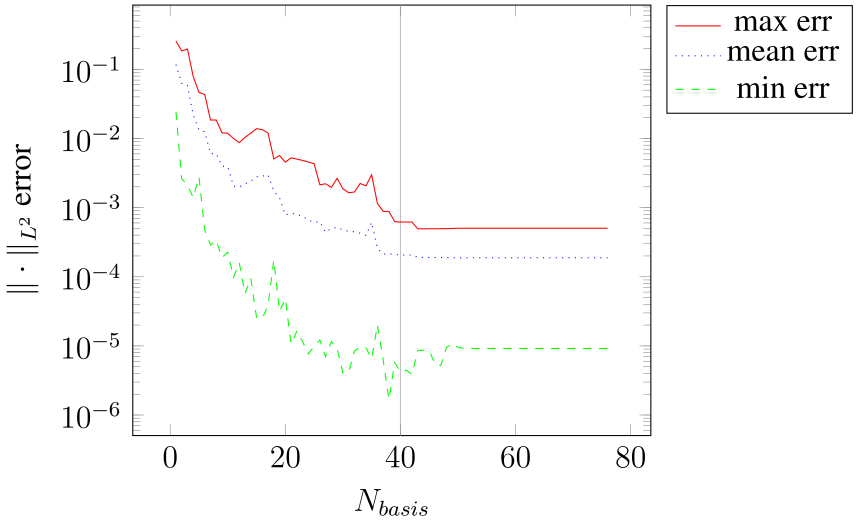

Figure 5.15: Error convergence between IGA-BEM and POD solutions for a random train of 50 parameter values. Δ^{EIM} does not permit to reach null error.

From the comparison between figure 5.14 and figure 5.15 we can see where the error between IGA-BEM and EI-IGA-BEM solutions, namely Δ^{EIM}, affects the error between POD and IGA-BEM solutions. We recall that POD modes are computed by considering the SVD of a snapshot matrix, each snapshot being computed by considering the EIM approximation of the operators. In fact, if we take less than 40 basis functions, the errors have the same trend. On the contrary, if we take more than 40 basis functions, the error between POD and IGA-BEM is not null, as it is in the case of POD and EI-IGA-BEM. Therefore, we can say that the POD error is governed by the reduced order technique for $N < 40$ and by EIM for $N > 40$. However, during the online evaluations, it is sufficient to retain only 9 basis functions and this aspect will be negligible from now on. From (4.27) we know that, if we consider 9 basis functions, we are keeping 99.9998% of the system energy. Thus, by considering a POD tolerance of $O(10^{-4})$, we retain 9 basis solutions (or modes) in our reduced space. We provide a comparison between the pressure coefficient computed through POD algorithm and IGA-BEM, EI-IGA-BEM and experimental data, first for NACA 0012 airfoil (figures 5.16 and 5.17), then for NACA 4412 airfoil (figures 5.18 and 5.19). In table 5.26 we summarize some computational details of POD, as well its efficiency parameters, in term of time and dimension of the reduced system.

We remark that, in terms of pressure coefficient, the results are almost equal, although we use about $\frac{1}{10}$ of the degrees of freedom of IGA-BEM.

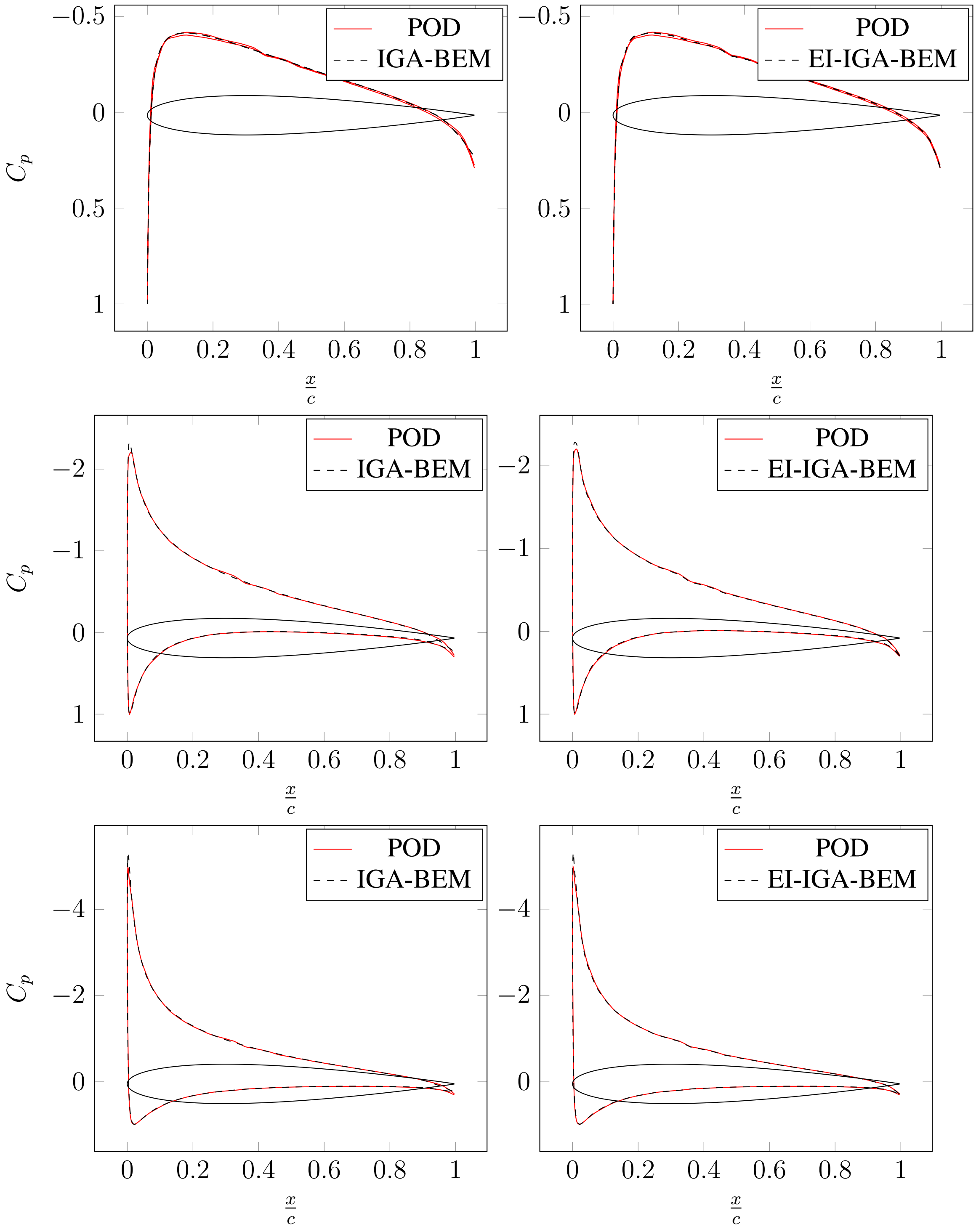

Figure 5.16: Comparison of the pressure coefficient for NACA 0012: IGA-BEM and POD (left column), EI-IGA-BEM and POD (right column); $\alpha = 0\,\mathrm{deg}, 6\,\mathrm{deg}, 10\,\mathrm{deg}$.

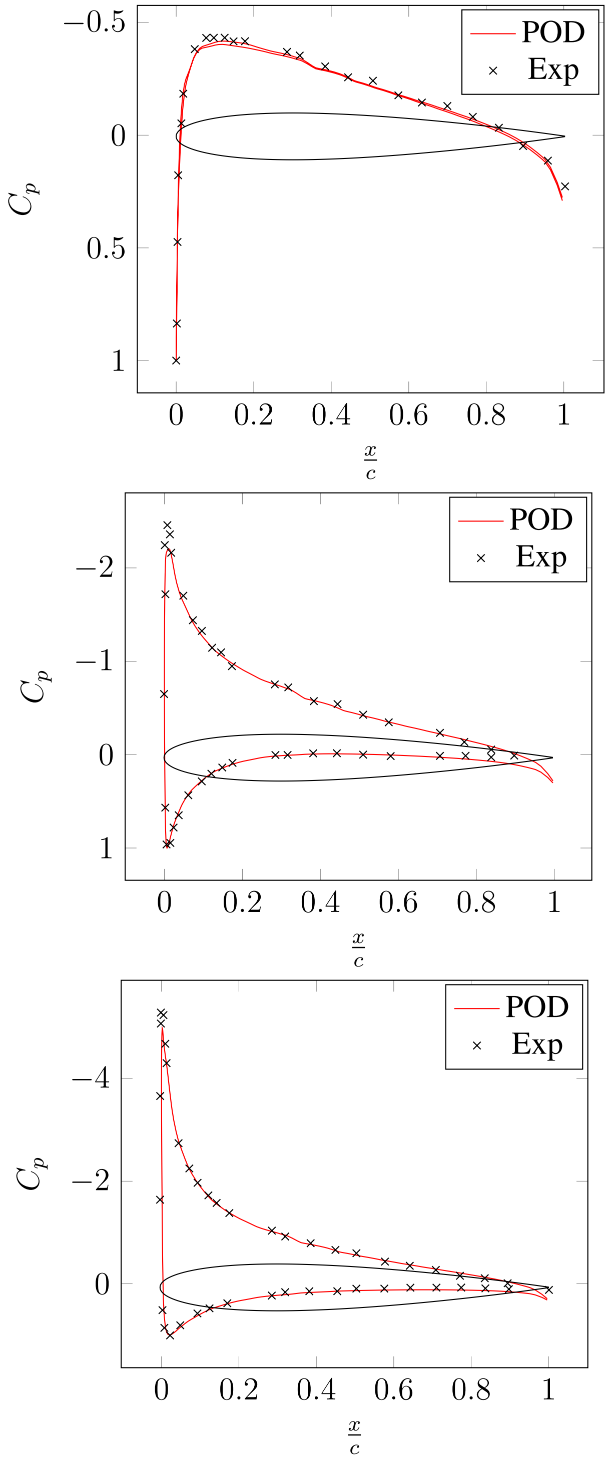

Figure 5.17: Comparison of pressure coefficient for NACA 0012: experimental data [21] and POD; $\alpha = 0\,\mathrm{deg}, 6\,\mathrm{deg}, 10\,\mathrm{deg}$.

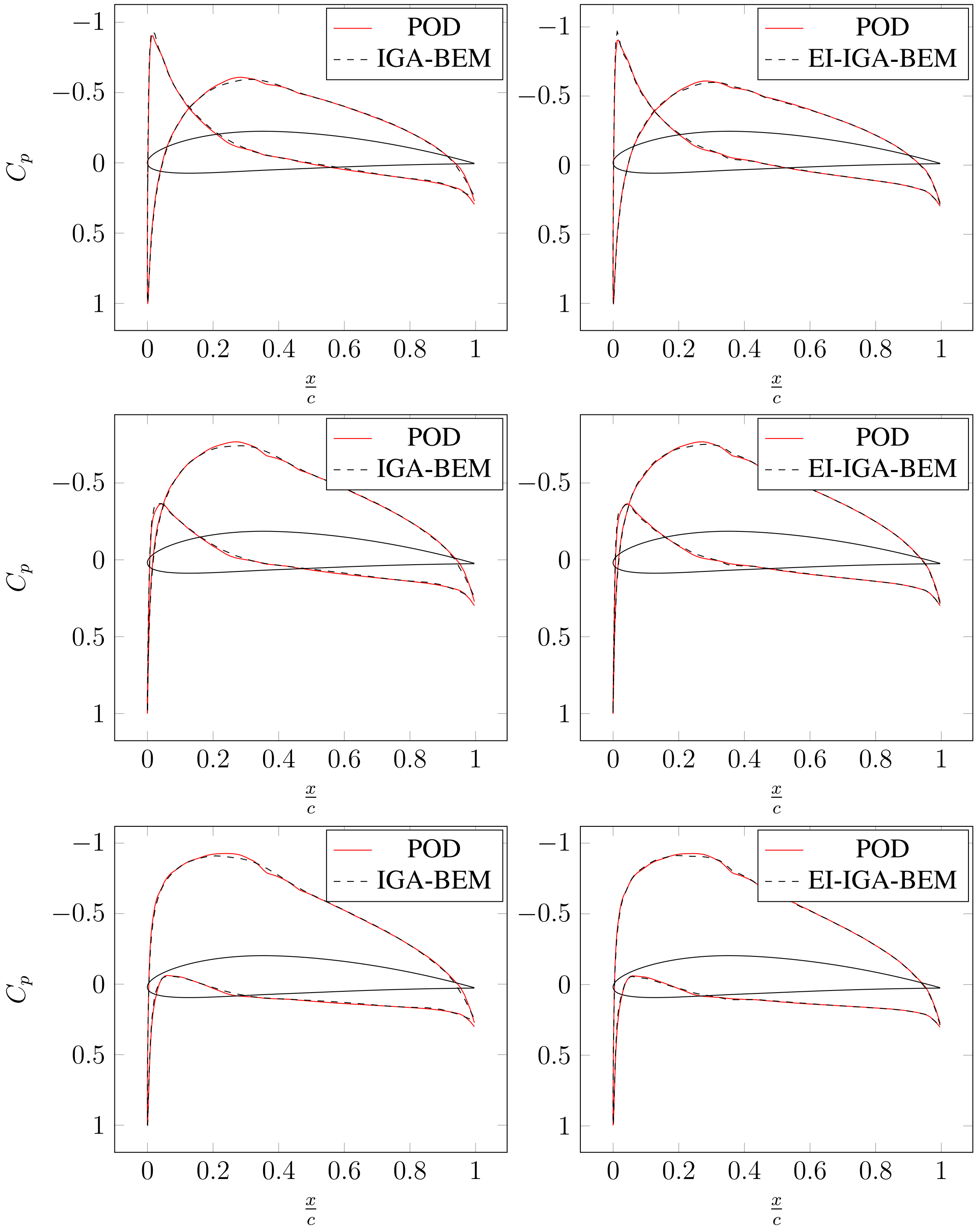

Figure 5.18: Comparison of the pressure coefficient for NACA 4412: full IGA-BEM and POD (left column), EI-IGA-BEM and POD (right column); $\alpha = -2\,\mathrm{deg}, 0\,\mathrm{deg}, 1.87\,\mathrm{deg}$.

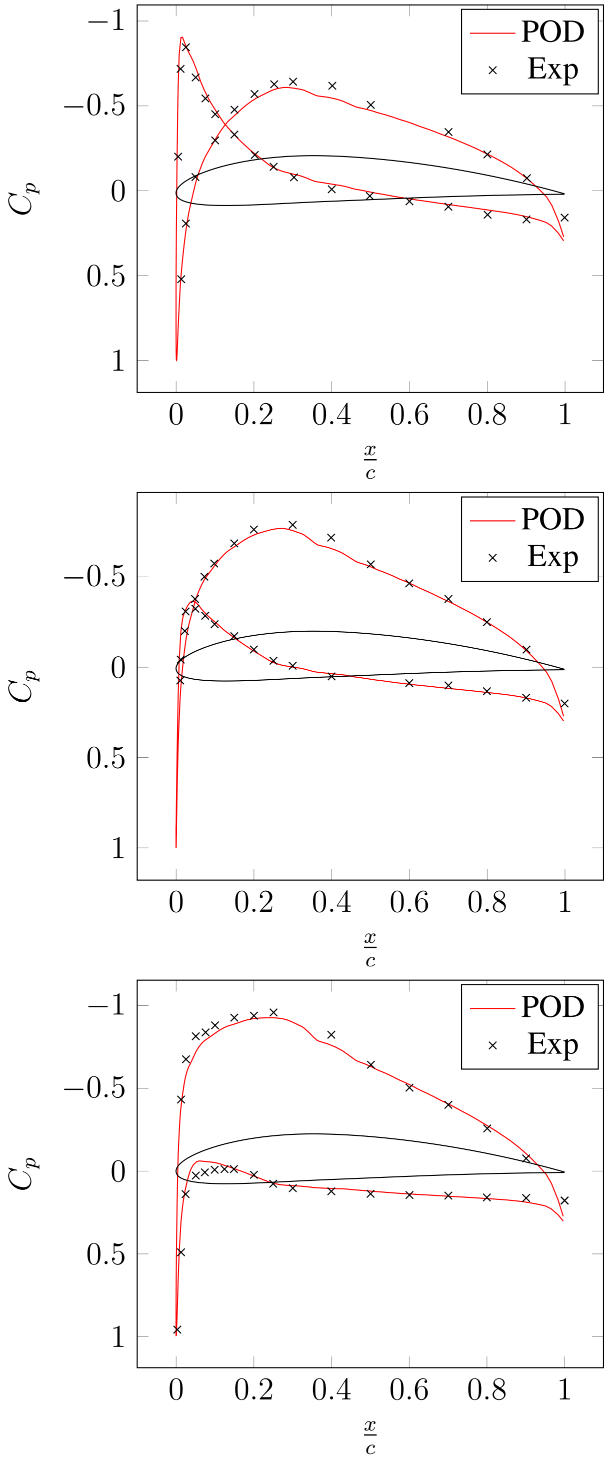

Figure 5.19: Comparison of the pressure coefficient for NACA 4412: experimental data [47] and POD; $\alpha = -2\,\mathrm{deg}, 0\,\mathrm{deg}, 1.87\,\mathrm{deg}$.

5.3.2 Greedy algorithm

We now consider the numerical results obtained through the greedy algorithm. Basically, during the offline stage, at every iteration n, the greedy algorithm expands the already computed basis V_{n-1} with that solution – over all candidate $u(\boldsymbol{\mu}), \boldsymbol{\mu} \in \Xi_{train}$ – which is least well approximated by V_{n-1}, where 'least well approximated' has to be intended in the sense of the error estimator Δ_N. The algorithm continues to iterate until the error estimator is under the required tolerance ϵ^*_{tol}, all over the parameter space. Then, during the Online stage, we solve the new (reduced) system thanks to a Galerkin projection.

Thus, we have taken a train of 600 (random) parameter values and we have applied the greedy algorithm with tolerance $\epsilon^*_{tol} = 3 * 10^{-4}$. Under these requirements, greedy algorithm selects 18 basis functions. In figure 5.20 we show the resulting error estimator Δ_N, and how the greedy algorithm selects the parameter values for the retained snapshots.

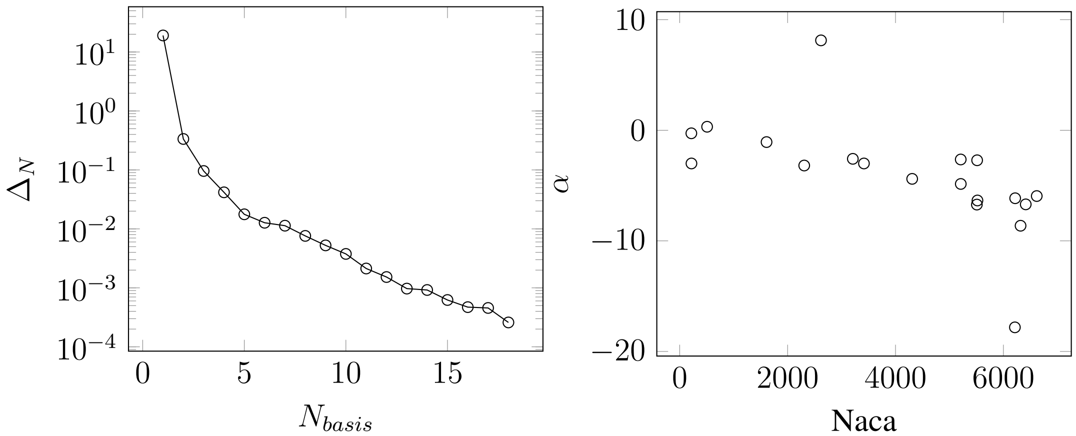

Figure 5.20: Greedy algorithm: convergence of the error estimator Δ_N (left), retained snapshots in the parameter space; $\epsilon^*_{tol} = 3 * 10^{-4}$.

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. Not only, just two snapshots are selected for 'large' angles of attack whereas several snapshots correspond to $\alpha \approx 0$ and different NACA profiles. This means that, between the two parameters, the reduction is harder for the shape than for the angle of attack. This is not surprising, since the IGA parametrization entails much more involved operations in order to describe the shape of a wide family of airfoils, than their rotation.

As a first test, we have taken the same set of 50 IGA-BEM solutions already exploited in the POD framework and we have performed a convergence test in the same way as in the POD case. For all possible choices of the RB space dimension, we compute the error (in L^2-norm) for all IGA-BEM solutions in the sample. In figure 5.21 we show the maximum, minimum and mean errors between RBM and IGA-BEM solutions.

We provide a comparison between the pressure coefficient computed through RBM algorithm and IGA-BEM, EI-IGA-BEM and experimental data, first for NACA 0012 airfoil (figures 5.22 and 5.23), then for NACA 4412 airfoil (figures 5.24 and 5.25). Also in the case of RBM, in terms of pressure coefficient the results are almost equal, although we use about $\frac{1}{10}$ of the degrees of freedom of IGA-BEM. In table 5.26 we summarize some computational details of RBM, as well its efficiency parameters, in term of time and dimension of the reduced system.

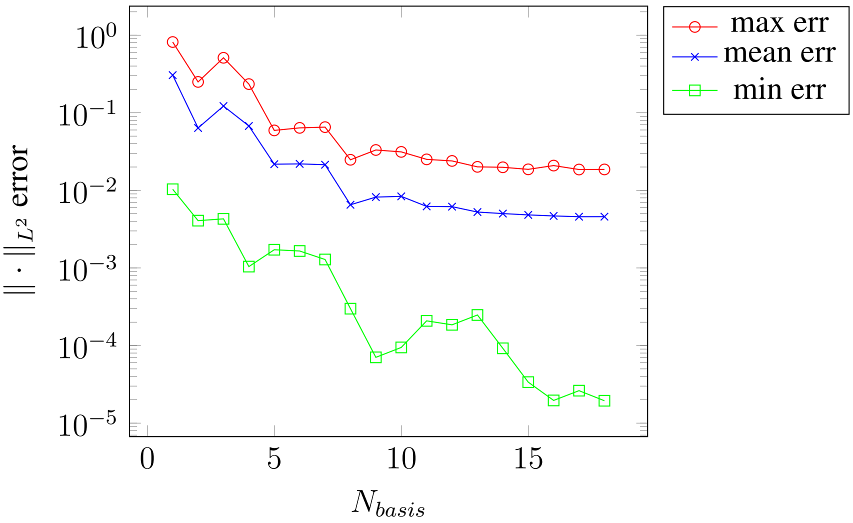

Figure 5.21: Error convergence between IGA-BEM and RBM solutions for a random train of 50 parameter values.

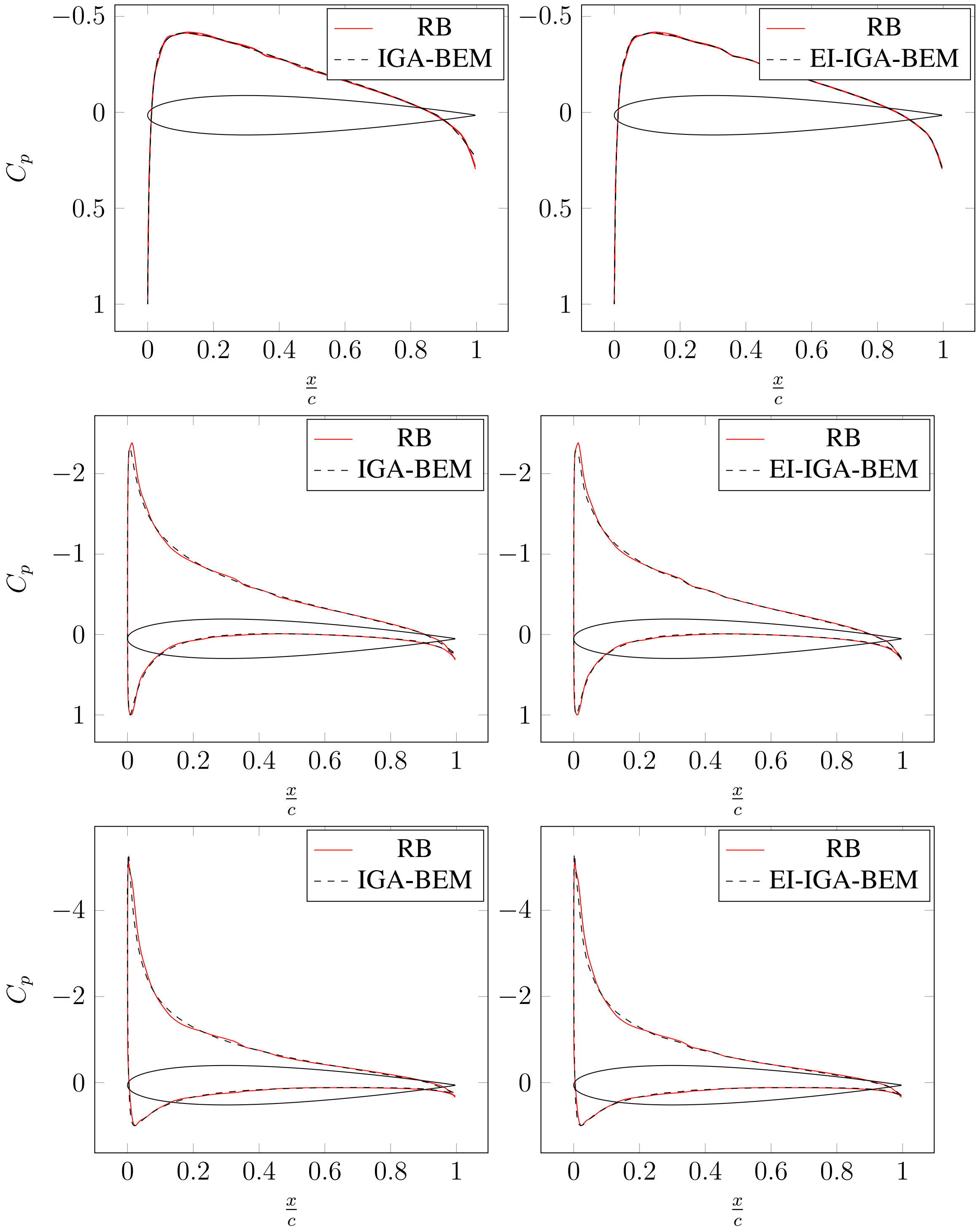

Figure 5.22: Comparison of the pressure coefficient for NACA 0012: IGA-BEM and (greedy) RB (left column), EI-IGA-BEM and (greedy) RB(right column); $\alpha = 0\,\mathrm{deg}, 6\,\mathrm{deg}, 10\,\mathrm{deg}$.

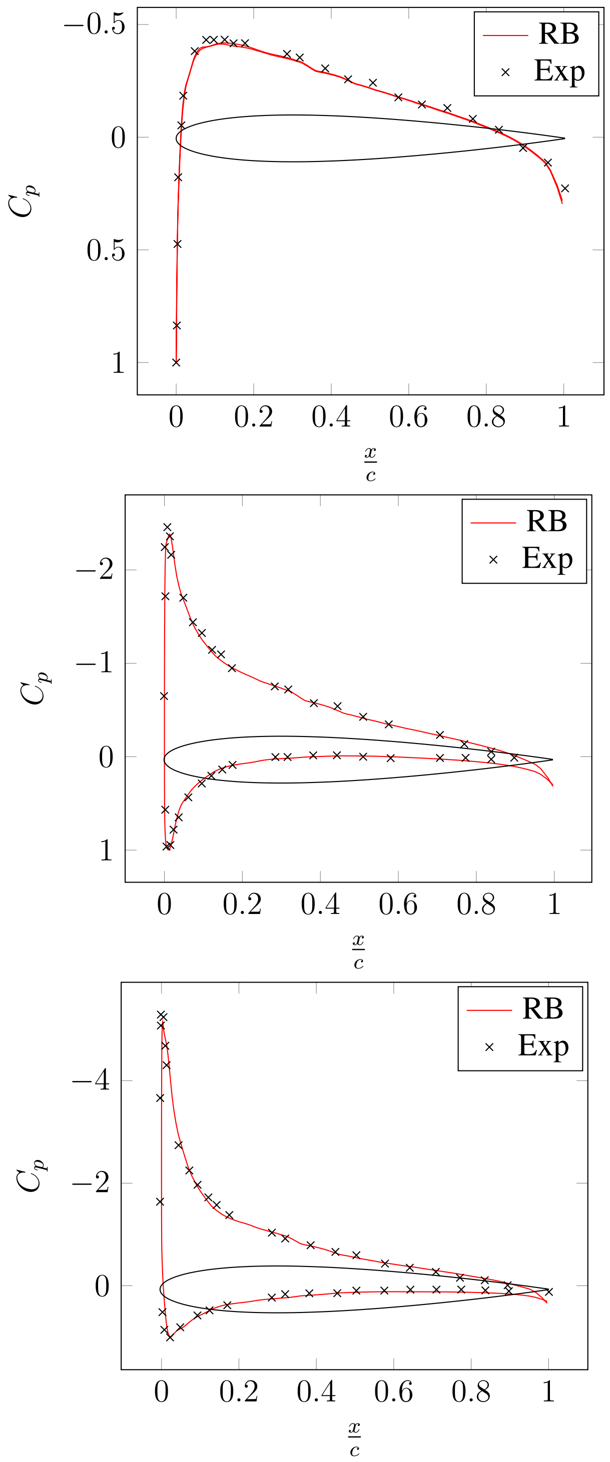

Figure 5.23: Comparison of the pressure coefficient for NACA 0012: experimental data and (greedy) RB; $\alpha = 0\,\mathrm{deg}, 6\,\mathrm{deg}, 10\,\mathrm{deg}$.

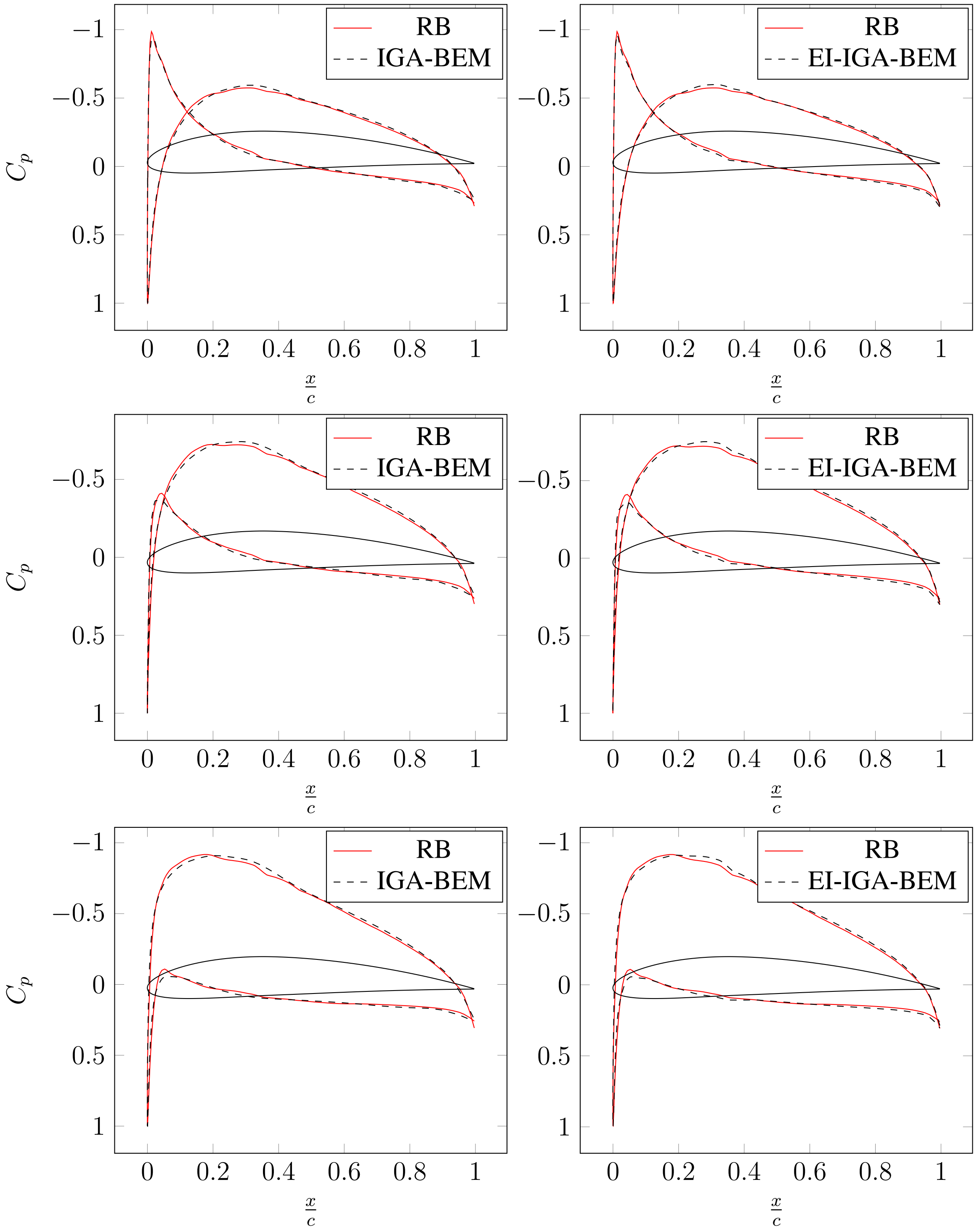

Figure 5.24: Comparison of the pressure coefficient for NACA 4412: IGA-BEM and (greedy) RB (left column), EI-IGA-BEM and (greedy) RB (right column); $\alpha = -2\,\mathrm{deg}, 0\,\mathrm{deg}, 1.87\,\mathrm{deg}$.

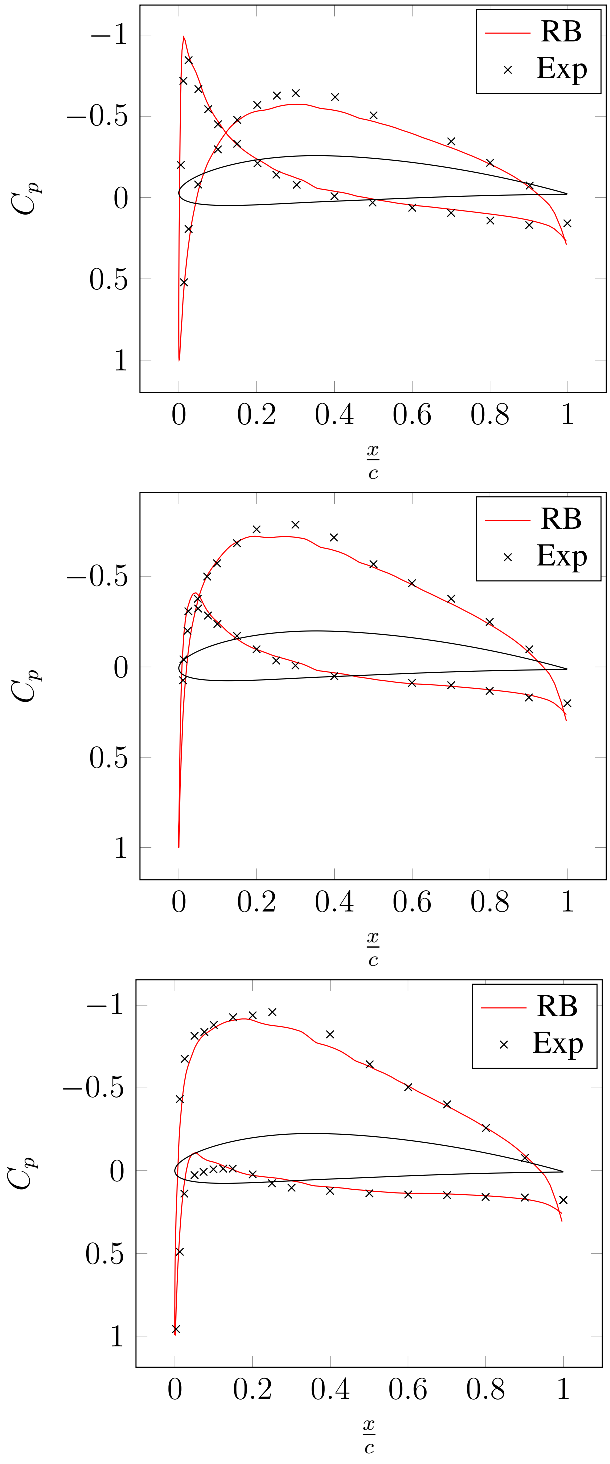

Figure 5.25: Comparison of the pressure coefficient for NACA 4412: experimental data and (greedy) RB; $\alpha = -2\,\text{deg}, 0\,\text{deg}, 1.87\,\text{deg}$.

5.3.3 Comparison of POD and RBM computational performances

We now compare the results obtained through POD, RBM and IGA-BEM. Since the dimension of the high-fidelity problem at hand is quite small, we provide also some details related to a second high-fidelity solution, obtained by considering a larger amount of degrees of freedom, for the sake of the comparison of numerical performances. In fact, we remember that the choice of a boundary element method, is already a first strategy to reduce the dimension of the full-order problem.

In figure 5.26, we compare the convergence test for POD and RBM, whereas in table 5.3 we summarize some numerical details of our ROMs. The curves in 5.26 are similar and, as it can be also seen in figures 5.22, 5.23, 5.24, 5.25, 5.16, 5.17, 5.18 and 5.19, both the methods give great results in terms of accuracy with respect to the high-fidelity IGA-BEM method. Moreover, the two methods lead to a remarkable computational saving for each input/output evaluation, which is very similar. 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 problem is a very efficient and rapid tool, i.e., it is not too expensive to evaluate 600 solution for the POD technique. For more complex problems, performing 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 RBM. In fact, the evaluation of the norm of the residual (4.35) is very expensive if we have several affine terms, as in our case, and this reflects on the greedy algorithm, requiring at each step the identification of the worst-case solution in terms of error indicator.

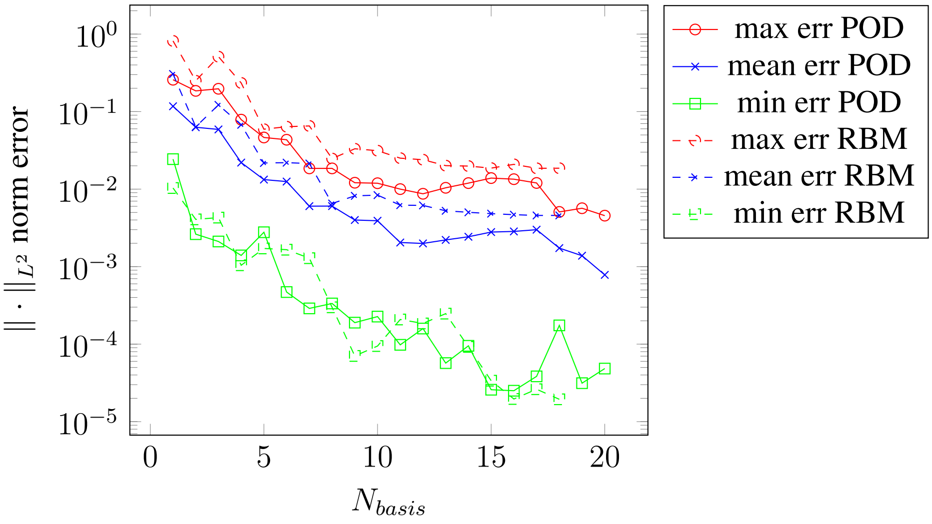

Figure 5.26: Error convergence comparison between POD and RBM for a random train of parameters.

Table 5.3: 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 rhs components M	40	41
Sample train (both POD and RBM)	600	600
IGA-BEM space dimension $\mathcal{N}$	77	147
RBM space dimension N	9	9
POD space dimension N	9	9
RBM tolerance	$3 * 10^{-4}$	$3 * 10^{-4}$
POD tolerance	$2 * 10^{-4}$	$2 * 10^{-4}$
IGA-BEM evaluation time	3.6 s	8 s
RBM construction time	2h 50min	5h
RBM evaluation time	0.95s	1.9s
POD construction time	8min	16min
RBM evaluation time	1.1s	1.9s
Computational speedup RBM	3.8	4.2
Computational speedup POD	3.3	4.2

As a final, general comparison, let us consider the work by Günther [22]. 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 to compute a high-fidelity approximation. Although the problem analysed in this work is much more involved, we can derive some general considerations. First of all, FEM is about 10 times more expensive than BEM in this case. On the contrary, the performance of the ROM during the online stage are comparable involving in each case $O(10)$ basis functions to represent parametrized solutions during the online stage. Moreover, our framework allows to:

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

All the ingredients considered in our framework play a key role in order to obtain an efficient characterization of potential flows past parametrized NACA airfoils, and related outputs of interest.

Both the ROMs implemented 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. However, we expect to obtain a much significant model reduction if we consider a larger dimension of the high-fidelity BEM problem (see table 5.3 for a first example), or three dimensional problems.

Conclusions

The main goal of the present work has been the analysis and implementation of a reduced basis method 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 used 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 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 differential problem, and it requires special techniques in order to apply a Reduced Order Model (ROM). We used a variation of the Empirical Interpolation Method (EIM) applied directly to the algebraic structures of IGA-BEM, which approximates the non affine parametric dependence with an affine one and speeds up the assembly of parameter dependent IGA-BEM matrices by one order of magnitude.

We exploited the EIM technique applied to IGA-BEM to construct two different reduce order models, namely the Proper Orthogonal Decomposition and the (greedy) Reduced Basis 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 both with experimental results and with other numerical tools (Xfoil and a B-splines based method presented in [41]). These airfoils were selected because they are standard benchmark problems in the aeronautical literature, for which several experimental data are available. 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. Moreover, we noted that the empirical interpolation method itself reduces the computational cost associated with the assembling of BEM structures. Although the construction of reduced spaces using POD and RBM is very different, we have not noticed relevant differences in their accuracy and rapidity in the prediction of the results of the problem at hand during the Online stage.

On the contrary, their behaviour during the Offline stage is quite different: in the problems we analysed, POD is much faster than RBM. This difference is mostly related to the number of affine terms and to the dimension of the full-order model. A great number of affine terms increases dramatically the cost of RBM, and the speed at which it is possible to solve the full order model (IGA-BEM) makes the computation of a great number of snapshots (required by POD) a reasonable strategy, ultimately making the use of RBM unfavorable in the case at hand.

The main achievement of this work is the coupling of reduced order models with isogeometric boundary elements, through a variation of the empirical interpolation method. The work carried out shows that this coupling can be very effective, and it is of great interest also in view of future developments.

Bibliography

[1] I. H. Abbott and A. Von Doenhoff. *Theory of wing sections: including a summary of airfoil data*. Dover Publications, 1959.

[2] I. H. Abbott, A. E. Von Doenhoff, and L. Stivers Jr. Summary of airfoil data. Technical report, 1945.

[3] N. Aubry. On the hidden beauty of the proper orthogonal decomposition. *Theor. Comp. Fluid. Dyn.*, 2:339–352, 1991.

[4] W. Bangerth, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and T. D. Young. The `deal.ii` library, version 8.1. *arXiv preprint* *`http://arxiv.org/abs/1312.2266v4`*, 2013.

[5] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

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

[7] G. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library, 1973.

[8] K. Belibassakis, T. P. Gerostathis, K. Kostas, C. Politis, P. Kaklis, A. Ginnis, and C. Feurer. A bem-isogeometric method with application to the wavemaking resistance problem of ships at constant speed. In *ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering*, pages 95–102. American Society of Mechanical Engineers, 2011.

[9] P. Benner, S. Gugercin, and K. Willcox. A survey of model reduction methods for parametric systems. Technical Report 13–14, Max Planck Institute Magdeburg, 2013.

[10] G. Berkooz, P. Holmes, and J. Lumley. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annu. Rev. Fluid Mech.*, 25(1):539–575, 1993.

[11] A. Buffa, Y. Maday, A. T. Patera, C. Prudhomme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Math. Model. Numer. Anal.*, 46(03):595–603, 2012.

[12] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.

[13] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *J. Comp. Phys.*, 224(2):880–896, 2007.

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

[15] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[16] L. B. da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h-p-k-refinement in isogeometric analysis. *Num. Math.*, 118(2):271–305, 2011.

[17] M. Drela and H. Youngren. Xfoil 6.94 user guide. *Massachusetts Institute of Technology, Cambridge, Massachusetts*, 2001.

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

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

[20] N. Giuliani. An hybrid boundary element method for free surface flows. Master Thesis, Politecnico di Milano (Italy), 2013.

[21] N. Gregory and 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.

[22] C. Günther. Reduced Basis Method for the Shape Optimization of Racing Car Components. Master Thesis, Ecole Polyechnique Féd'erale de Lausanne (Switzerland), 2008.

[23] L. Heltai, M. Arroyo, and A. DeSimone. Nonsingular Isogeometric Boundary Element Method for Stokes Flows in 3d. Technical Report 35/2012/M, SISSA, 2012.

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

[25] J. L. Hess and A. Smith. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. Technical report, DTIC Document, 1962.

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

[27] G. C. Hsiao. Boundary element methods–an overview. *Appl. Numer. Math.*, 56(10):1356–1369, 2006.

[28] G. C. Hsiao and W. L. Wendland. *Boundary integral equations*. Springer, 2008.

[29] C. Hsin, J. Kerwin, and J. Newman. A higher-order panel method based on b-splines, 1994.

[30] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39):4135–4195, 2005.

[31] D. Huynh, N. Nguyen, A. Patera, and G. Rozza. Rapid reliable solution of the parametrized partial differential equations of continuum mechanics and transport. Available at `http://augustine.mit.edu`, ©MIT 2008-2014.

[32] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf–sup stability constants. *Comptes Rendus Mathematique*, 345(8):473–478, 2007.

[33] K. Karamcheti. *Principles of ideal-fluid aerodynamics*. Wiley New York, 1966.

[34] J. E. Kerwin, S. A. Kinnas, J.-T. Lee, and W.-Z. Shih. A surface panel method for the hydrodynamic analysis of ducted propellers. *SNAME Transaction*, 95:93–122, 1987.

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

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

[37] H. R. Kutt. Quadrature formulae for finite-part integrals. Technical report, WISK, 178, National Research Institute for Mathematical Sciences, Pretoria, South Africa, 1975.

[38] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9, pages 235–274. Springer, MS&A Series, 2013.

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

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

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

[42] Y. Maday, A. T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17(1-4):437–446, 2002.

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

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

[45] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

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

[47] 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.

[48] C. Politis, A. I. Ginnis, P. D. Kaklis, K. Belibassakis, and C. Feurer. An isogeometric BEM for exterior potential-flow problems in the plane. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 349–354. ACM, 2009.

[49] A. Quarteroni. *Numerical Models for Differential Problems*, volume 2 of *Modeling, Simulation and Applications (MS&A)*. Springer, 2009.

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

[51] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*. Springer-Verlag, New York, 2000.

[52] S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International journal for numerical methods in fluids*, 34(5):425–448, 2000.

[53] G. Rozza. Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Comm. Comput. Phys.*, 9:1–48, 2011.

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

[55] S. Salsa. *Partial differential equations in action: from modelling to theory*. Springer, 2008.

[56] S. A. Sauter and C. Schwab. *Boundary element methods*. Springer, 2011.

[57] L. Sirovich. Turbulence and the dynamics of coherent structures. I - Coherent structures. II - Symmetries and transformations. III - Dynamics and scaling. *Quart. Appl. Math.*, 45:561–571, 1987.

[58] O. Steinbach. *Numerical approximation methods for elliptic boundary value problems: Finite and Boundary Elements*. Springer, 2008.

[59] T. Takahashi and T. Matsumoto. An application of fast multipole method to isogeometric boundary element method for Laplace equation in two dimensions. *Engineering Analysis with Boundary Elements*, 36(12):1766–1775, 2012.

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

[61] K. Veroy, C. Prudhomme, D. Rovas, and 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*, volume 3847, 2003.

[62] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modeling. *Lecture Notes, University of Konstanz*, 2012.

[63] C. Wauquiez. *Shape optimization of low speed airfoils using Matlab and Automatic Differentiation*. VDM Publishing, 2009.