Parametric free-form shape design with PDE models and reduced basis method

Toni Lassila[a,b], Gianluigi Rozza[a,*]

[a]*Modelling and Scientific Computing, Institute of Analysis and Scientific Computing, École Fédérale Polytechnique de Lausanne, Station 8, EPFL, CH-1015 Lausanne, Switzerland.*
[b]*Institute of Mathematics, Helsinki University of Technology, P.O. Box 1100, FI-02015 TKK, Finland.*

Abstract

We present a coupling of the reduced basis methods and free-form deformations for shape optimization and design of systems modelled by elliptic PDEs. The free-form deformations give a parameterization of the shape that is independent of the mesh, the initial geometry, and the underlying PDE model. The resulting parametric PDEs are solved by reduced basis methods. An important role in our implementation is played by the recently proposed empirical interpolation method, which allows approximating the non-affinely parameterized deformations with affinely parameterized ones. These ingredients together give rise to an efficient online computational procedure for a repeated evaluation design environment like the one for shape optimization. The proposed approach is demonstrated on an airfoil inverse design problem.

Key words: reduced basis methods, free-form deformations, empirical interpolation, engineering design, shape optimization

1. Introduction and motivation

The solution of partial differential equations in domains of arbitrary shape has become an important part of the computational science and engineering disciplines. A typical application is the simulation and design of structures, such as airfoils, in computational fluid dynamics. The finite element method [7, 54] is ideal for solving elliptic and parabolic PDEs on domains of very general shape. However, when the objective is to solve the same problem repeatedly on different domains, the cost of setting up the problem (meshing, matrix assembly) every time from scratch is too high. This requires more efficient procedures to be developed.

One of our primary interests for solving partial differential equations in domains of varying shapes is to perform shape optimization in the computational fluid dynamics setting. To that end we have to first set up the framework of parametric shapes. Assume that we have a bounded reference domain, which is given to us in the form of a triangular mesh and its nodal points. We consider small perturbations of this reference

---

*Corresponding author
Email addresses:* `toni.lassila@epfl.ch` (Toni Lassila), `gianluigi.rozza@epfl.ch` (Gianluigi Rozza)

domain. The description of the perturbations of a domain can fall roughly in two categories: variational [15, 17, 48] or parametric [2, 23, 34]. In this work we consider only parametric deformations, i.e. the perturbed domain is a function of a finite number of real parameters. The role of the parameters is to allow the efficient study of several and different configurations and settings for the problem considered (different shapes for many different target functions and state solutions). Once the shape has been parameterized with a small number of parameters, our aim is to perform shape optimization w.r.t the parameters by solving a partial differential state equation on the parametric domain that models some relevant physical phenomena. One of the goals of this paper is to propose a combined methodology to reduce the geometrical complexity (i.e. to reduce the number of geometrical parameters needed for shape optimization) as well as to reduce the dimension of the resulting discretized problem in terms of linear system dimension. The requirements set for the domain parameterization are that it should be expressive and versatile enough to describe a wide variety of shapes of interest while at the same time being computationally effective and efficient enough for real-time solution of the parametric PDEs.

For a review of shape parameterization techniques in engineering design we refer the reader to [23, 43]. Two different but typical approaches for parametric domains are:

- **Basis shapes** are a well-chosen set of shapes that have relevance to the underlying model – a parameterization can be obtained by blending smoothly between the basis shapes. For problems, such as aerodynamic design, where the qualities of the desired target shapes are limited to a small subset of shapes, this has been a common choice. These parameterizations are usually *low-dimensional*, and consequently the methods will only explore a limited variety of shapes [43, 49].

- **Splines** can be used to describe smooth boundaries of shapes with arbitrary accuracy [8, 30]. The approximation properties of splines are well understood and there are several families of splines to choose from. To obtain a parametric shape we can take the location of the control points and their weights as the parameters. The drawback to this approach is that the number of parameters used to describe the shapes is a priori *high-dimensional* [4, 43] and often outside the practical realm of nonlinear programming techniques such as Sequential Quadratic Programming (SQP) and various quasi-Newton methods [6]. For this reason many people have turned to pseudo-optimization methods such as evolutionary algorithms to solve problems of optimal engineering design [4, 12, 32].

Neither the shape bases nor the boundary splines are absolutely the best choice to fulfill our requirements set for shape parameterization. The shape bases require expert knowledge of the problem in constructing the basis shapes that might not be available a priori. On the other hand, the high number of degrees of freedom in a boundary spline based parameterization makes model reduction schemes for efficient solution of the PDEs not viable. Earlier works on reduced models in shape optimization have been presented in [50]

```
┌─────────────────────────────────────────────────────────────────┐
│  ┌─────┐                                                         │
│  │ FFD │   versatile parameterization of geometry                │
│  └─────┘                                                         │
│     ↓                                                            │
│  ┌─────┐                                                         │
│  │ EIM │   offline-online computational decomposition            │
│  └─────┘                                                         │
│     ↓                                                            │
│  ┌─────┐                                                         │
│  │ RB  │   efficient reduced order solution for the PDE          │
│  └─────┘                                                         │
│     ↓                                                            │
│  ┌─────┐                                                         │
│  │ OPT │   shape optimization algorithm                          │
│  └─────┘                                                         │
└─────────────────────────────────────────────────────────────────┘
```
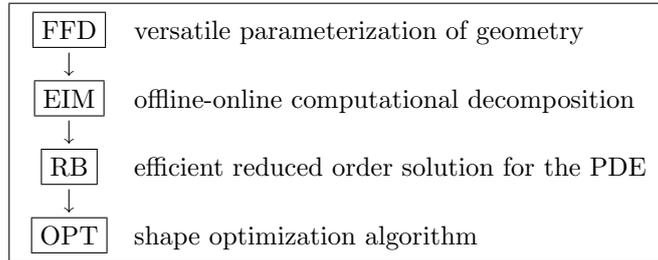
Figure 1: The building blocks of our shape design approach

for the structural shape optimization and in [21] for shape optimization of airfoils by Proper Orthogonal Decomposition (POD).

Our proposed approach to parametric shape design is as follows. First we define the mathematical setting of the parametric PDE problem. A suitably low-dimensional parameterization of the reference domain is obtained by considering free-form deformations (FFD) [20, 47] that are independent from the description of the reference domain and meshes. FFDs are a powerful tool for representing smooth global deformations of the reference domain and allowing the reduction of a great number of shape parameters. They allow the construction of a mesh-independent parameterization, since for a given geometry and computational mesh we can construct a parameterization of arbitary flexibility without the need for remeshing. These deformations lead to non-affinely parameterized transformations of the computational domain. The non-affinely parameterized transformations are then approximated by affinely parameterized transformations obtained with the recently developed empirical interpolation method (EIM) [5, 13, 22, 25, 26, 40]. This permits an offline-online computational procedure for the parametric PDE using reduced basis approximations (RB), which were originally considered in [3, 11, 27, 31], and which have since been treated in more analytical detail for example in [29, 33, 41]. As the shape optimization problem can be recast into an optimal control framework, we recall previous works where these kind of problems have been solved by reduced basis methods, see e.g. [14, 36, 37]. A diagram of our computational "tools" properly linked is displayed in Fig. 1. The result is an efficient method for shape design, where the parametric deformation of shapes is independent from the reference geometry, the PDE model, or the chosen mesh. We will demonstrate the efficiency of the chosen approach with an inverse design problem in the NACA family of airfoil shapes [1]. Recent works suggest that free-form deformations are a promising approach to airfoil design [32, 44, 45]. In [4] a thorough investigation into boundary B-spline vs. FFD based parameterizations in airfoil design was made, and it was observed that with a comparative number of optimization variables the FFD parameterizations obtained better results. The reduced basis method could however be coupled with other parameterization methods such as boundary splines [12], radial basis functions [24] etc.

Our approach to shape optimization is related to the surrogate model optimization [38] approach. In

surrogate model optimization an expensive model is replaced with a cheaper reduced model (in this case the reduced basis approximation), which is then used for evaluations of cost function in the optimization iteration. The surrogate models proposed in literature are often based on approximation techniques such as local polynomial fitting, radial basis functions [12, 32], or neural network techniques. The common factor with all of these is that the state equations still need to be solved using the expensive finite element method for many different parameter values, typically to the order of 10^3 times, during the course of the optimization iteration. In contrast, in the reduced basis technique the finite element "truth approximations" are not used once during the optimization process, owing to the all the "preparation" that has already been done offline in a precomputation stage. The flexibility of changing the cost functional and the constraints to better suit the design criteria is therefore greater.

We argue that free-form deformations and the reduced basis framework together can be considered a combination of basis shapes and spline representations. Free-form deformations have the good expressiveness of splines, while the reduced basis methodology permits efficient computations by reducing the solution of the PDE to small number of snapshot configurations, which can be thought of as our set of basis shapes. By allowing the user to choose the density of the control point grid and the degrees of freedom to be used we keep the dimension of the parameterization low for practical optimization methods. The reduced basis framework has to be applicable to the flow model equation in the sense that reliable and sharp a posteriori error estimates are available for the reduced basis solutions. A posteriori estimates for potential flows are fairly straightforward [39] using the existing theory for coercive problems, and work on reduced basis methods for the Stokes [40, 42] and Navier-Stokes [9, 35, 52] equations has been carried out and their use with a coupled FFD technique is possible and an object of future study.

An alternative approach to parameterizing the domain perturbations is to build piecewise affine geometry transformations based on domain subdivisions. A detailed description of this technique was given in [41]. The piecewise affine geometry parameterization however suffers from two drawbacks:

- The smoothness of the shape is not necessarily preserved. This is sometimes important, especially in applications such as aerodynamic design.

- Discontinuities in the transformation tensor mean that the mesh has to conform to the domain subdivision, otherwise we need to perform partial remeshing or at the very least do local refinements near the boundaries of the subdivisions.

For problems with simple domains and pure sizing optimization the affine parameterizations are workable. For more complex problems, including shape optimization and inverse design, the geometry and the mesh take a lot of expert time to construct. If possible we want to avoid remeshing or readjusting the mesh. In these cases it is beneficial to have a procedure which allows us to parameterize the geometry independently

4

from the PDE model or its discrete formulation. In this framework we propose an approach based on three coupled tools: free-form deformations, the empirical interpolation method, and reduced basis methods.

2. Elliptic PDE problem with parametric domain

Let $\Omega_0 \subset \mathbb{R}^d$ be a fixed reference domain, and $\boldsymbol{T} : \Omega_0 \times \mathcal{D} \to \mathbb{R}^d$ an invertible and continuously differentiable parametric map, where $\mathcal{D} \subset \mathbb{R}^P$ is the admissible parameter range. For any given $\boldsymbol{\mu} \in \mathcal{D}$ we want to solve the elliptic partial differential equation

$$\widetilde{a}(\widetilde{u}^e, v) = \widetilde{f}(v) \quad \text{for all } v \in X^e(\Omega) \tag{1}$$

in some Hilbert space $X^e(\Omega)$ defined on the domain $\Omega(\boldsymbol{\mu})$ with prescribed boundary conditions. Here $\widetilde{a}(\cdot, \cdot)$ is a coercive continuous symmetric bilinear form, and $\widetilde{f}$ a bounded linear form. By $\widetilde{u}^e(\boldsymbol{\mu}) \in X^e$ we denote the exact solution. Without loss of generality we can always transform equation (1) back onto the reference domain and consider instead the *parametric partial differential equation*

$$a(u^e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \text{for all } v \in X^e(\Omega_0), \tag{2}$$

where $a(\cdot, \cdot; \boldsymbol{\mu})$ is also a coercive continuous symmetric parametric bilinear form and $f(\cdot; \boldsymbol{\mu})$ a bounded linear functional for all $\boldsymbol{\mu} \in \mathcal{D}$.

By using finite element discretization with Galerkin projection we arrive at a corresponding finite-dimensional equation

$$a(u^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \text{ for all } v \in X^{\mathcal{N}} \subset X^e(\Omega_0), \tag{3}$$

where the dimension $\mathcal{N}$ of the finite element space $X^{\mathcal{N}}$ can be chosen large enough so that the difference between the exact solution $u^e(\boldsymbol{\mu})$ and the finite element solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ is negligible. We call $u^{\mathcal{N}}(\boldsymbol{\mu})$ a "truth approximation" to be obtained, and forget about the exact solution as it is typically unobtainable. We next discuss reduced basis methods for obtaining reliable approximate solutions to (3).

3. Reduced basis methods for parametric elliptic PDEs

The reduced basis method is an efficient way to compute approximations to the finite element "truth approximation" $u^{\mathcal{N}}(\boldsymbol{\mu})$ by considering only a small subspace of $X^{\mathcal{N}}$ spanned by well-chosen solutions of equation (3). If the dependence of the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$ on the parameter $\boldsymbol{\mu}$ is smooth the parametric manifold of solutions in X is smooth. Let us take a relatively small collection of parameter values $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N$ and the corresponding FE solutions $u^{\mathcal{N}}(\boldsymbol{\mu}^1), \ldots, u^{\mathcal{N}}(\boldsymbol{\mu}^N)$ (called snapshot solutions). From these snapshot solutions we can construct with Gram-Schmidt orthonormalization (using a suitable inner product, see [29] Ch. 3.2.1) a basis $\{\zeta_n\}_{n=1}^N$ spanning an N-dimensional subspace $X_N^{\mathcal{N}} \subset X^{\mathcal{N}}$, where typically $N \ll \mathcal{N}$. Using Galerkin projection onto this subspace we obtain the reduced basis formulation to find $u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}}$ s.t.

$$a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \text{ for all } v \in X_N^{\mathcal{N}} \tag{4}$$

and this reduced basis solution can be written in the basis spanned by the reduced basis functions as

$$u_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{n=1}^{N} u_n \zeta_n$$

with coefficients u_n, $n = 1, \ldots, N$. The choice of the snapshot solutions plays a vital role in the accuracy of the reduced basis approximation. A parameter-independent inner product and norm can be obtained by choosing a fixed $\bar{\boldsymbol{\mu}} \in \mathcal{D}$ and defining as

$$(u, v)_X := a(u, v; \bar{\boldsymbol{\mu}}) \quad \text{for all } u, v \in X^e,$$

$$||u||_X := \sqrt{(u, u)_X} \quad \text{for all } u \in X^e.$$

The following greedy algorithm for choosing the basis functions ζ_n has been proposed [53]. Let $\Xi_{train} \subset \mathcal{D}$ be a finite training sample of parameter points chosen according to uniform or log-uniform distribution containing a sufficiently varied set of training points. Given the first snapshot parameter value $\boldsymbol{\mu}^1$ and a sharp, rigorous, and inexpensive a posteriori error bound $\Delta_n(\boldsymbol{\mu})$ for the norm $|| \cdot ||_X$ such that

$$||u^{\mathcal{N}}(\boldsymbol{\mu}) - u_n^{\mathcal{N}}(\boldsymbol{\mu})||_X \leq \Delta_n(\boldsymbol{\mu}) \quad \text{for all } \boldsymbol{\mu} \in \mathcal{D},$$

we choose the remaining snapshot parameter values as the solutions

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

The quality of the reduced basis approximation depends crucially on the quality of the a posteriori error estimator. We introduce the residual as

$$r_n(v; \boldsymbol{\mu}) := f(v; \boldsymbol{\mu}) - a(u_n^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) \in X'$$

and use the residual-based estimator

$$\Delta_n(\boldsymbol{\mu}) := \frac{||r_n(\cdot; \boldsymbol{\mu})||_{X'}}{\alpha_{LB}(\boldsymbol{\mu})},$$

where $|| \cdot ||_{X'}$ is the dual norm of the residual and is defined as [41]

$$||r_n(\cdot; \boldsymbol{\mu})||_{X'} := \sup_{v \in X^{\mathcal{N}}} \frac{r_n(v; \boldsymbol{\mu})}{||v||_X},$$

and $\alpha_{LB}(\boldsymbol{\mu})$ is a computable lower bound for the coercivity constant $\alpha_0(\boldsymbol{\mu})$ [19], i.e.

$$\alpha_{LB}(\boldsymbol{\mu})||u||_X^2 \leq \alpha_0(\boldsymbol{\mu})||u||_X^2 \leq a(u, u; \boldsymbol{\mu}) \quad \text{for all } u \in X^{\mathcal{N}}.$$

For efficient and reliable methods of computing both the dual norm of the residual and a lower bound for the coercivity we refer the reader to [29, 41]. In the greedy basis construction algorithm we usually fix a priori an error tolerance ε_{tol}^{RB} and then continue the process until the stopping tolerance is achieved

$$\Delta_N(\boldsymbol{\mu}) \leq \varepsilon_{tol}^{RB} \quad \text{for all } \boldsymbol{\mu} \in \Xi_{train}.$$

Once the system (4) has been assembled and solved we want to estimate the error of the solution for a chosen $\boldsymbol{\mu} \in \mathcal{D}$. Because of the difficulty of giving a priori error estimates for reduced basis methods for many problems, we decide to provide a reliable way to compute a posteriori error estimates instead. For this reason we use the same residual-based estimators. See [41] for details.

We define another concept that is important for the computational efficiency of the reduced basis approximation; we say that the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$ and the form $f(\cdot; \boldsymbol{\mu})$ are *affinely parameterized* if they are of the form

$$
a(w, v; \boldsymbol{\mu}) = \sum_{m=1}^{M_a} \Theta_a^m(\boldsymbol{\mu}) a^m(w, v)
$$
$$
f(v; \boldsymbol{\mu}) = \sum_{m=1}^{M_f} \Theta_f^m(\boldsymbol{\mu}) f^m(v)
\tag{5}
$$

for suitable integers M_a and M_f, and functions $\Theta_a^m, \Theta_f^m : \mathcal{D} \to \mathbb{R}$. This assumption allows us to separate the computation of solutions for (4) into online and offline stages. By substituting the discrete form of (5) into (4) we obtain

$$
\sum_{m=1}^{M_a} \Theta_a^m(\boldsymbol{\mu}) a^m(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v) = \sum_{m=1}^{M_f} \Theta_f^m(\boldsymbol{\mu}) f^m(v)
\tag{6}
$$

for all $v \in X_N^{\mathcal{N}}$, and as $u_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{n=1}^{N} u_n(\boldsymbol{\mu}) \zeta_n$ for a unique set of coefficients $u_n(\boldsymbol{\mu})$, this equation becomes

$$
\sum_{n=1}^{N} u_n \sum_{m=1}^{M_a} \Theta_a^m(\boldsymbol{\mu}) a^m(\zeta_n, \zeta_{n'}) = \sum_{m=1}^{M_f} \Theta_f^m(\boldsymbol{\mu}) f^m(\zeta_{n'})
\tag{7}
$$

for all $n' = 1, \ldots, N$. We see now that the computational effort is split into two parts: a $\boldsymbol{\mu}$-independent offline stage and an online stage for any chosen $\boldsymbol{\mu} \in \mathcal{D}$. The basis functions ζ_n, the matrix forms of $a^m(\zeta_n, \zeta_{n'})$ (which we denote from now on as $\boldsymbol{A}^m$), and the vectors $f^m(\zeta_n)$ are computed in the offline stage and stored. The small $N \times N$ linear system (7) can then be assembled and solved efficiently during the online stage.

Until now we have not discussed the parameterization of the domains, but we have assumed it to be given. Next we detail one possible way of obtaining a shape parameterization by using free-form deformations that turns out to have some good properties in shape optimization and engineering design.

4. Free-form deformations for automatic parametric geometries

We consider a free-form deformation (FFD) method [47] based on tensor products of splines. A simple version of the FFD is defined as follows. For simplicity we cover only the two-dimensional case, extensions to three dimensions are straightforward. Let $D_0 \supset \Omega_0$ be a rectangle that contains the reference domain, and $\Psi : (x_1, x_2) \mapsto (x_1', x_2')$ an affine map from D_0 to the unit square, $\Psi(D_0) = (0, 1) \times (0, 1)$. The FFD shall be defined in the reference coordinates (x_1', x_2') of the unit square. Select a regular grid of control points

$\boldsymbol{P}^0_{\ell,k} \in (0,1) \times (0,1)$ (the unperturbed configuration) where $\ell = 0, \ldots, L$ and $k = 0, \ldots, K$ so that

$$\boldsymbol{P}^0_{\ell,k} = \begin{bmatrix} \ell/L \\ k/K \end{bmatrix}.$$

A perturbation of the control points is given by a set of $(L+1)(K+1)$ parameter vectors $\boldsymbol{\mu}_{\ell,k}$ where the perturbed positions of the control points are

$$\boldsymbol{P}_{\ell,k}(\boldsymbol{\mu}_{\ell,k}) = \boldsymbol{P}^0_{\ell,k} + \boldsymbol{\mu}_{\ell,k}.$$

It is possible that not all the control points are allowed to move freely - typically we fix several rows/columns of control points to obtain desired levels of continuity and to fix certain parts of the domain. It is also possible to allow some control points to move in the x'_1- or x'_2-direction, but not both. We refer to the effectively free scalar-valued parameters using the same variables but different indexing

$$\mu_1, \ldots, \mu_p, \quad p = 1, \ldots, P$$

and call P the number of *degrees of freedom*. From now on we assume that the parameterization involves only those μ corresponding to actual degrees of freedom omitting the parameters that are fixed.

From now on we denote by $\boldsymbol{x}$ the coordinates on the reference domain Ω_0 and by $\widetilde{\boldsymbol{x}}$ the coordinates on the parametric domain $\Omega(\boldsymbol{\mu})$. We construct a parametric domain map $\widetilde{\boldsymbol{T}} : D_0 \to \mathbb{R}^2$ as

$$\widetilde{\boldsymbol{T}}(\boldsymbol{x}; \boldsymbol{\mu}) = \Psi^{-1} \left(\sum_{\ell=0}^{L} \sum_{k=0}^{K} b^{L,K}_{\ell,k}(\Psi(\boldsymbol{x})) \boldsymbol{P}_{\ell,k}(\boldsymbol{\mu}_{\ell,k}) \right) = \widetilde{\boldsymbol{x}}(\boldsymbol{\mu}), \tag{8}$$

where

$$b^{L,K}_{\ell,k}(x'_1, x'_2) = \binom{L}{\ell}\binom{K}{k}(1 - x'_1)^{(L-\ell)}(x'_1)^{\ell}(1 - x'_2)^{(K-k)}(x'_2)^{k}$$

are tensor products of the 1-d Bernstein basis polynomials

$$b^{L}_{\ell}(x'_1) = \binom{L}{\ell}(1 - x'_1)^{(L-\ell)}(x'_1)^{\ell},$$

$$b^{K}_{k}(x'_2) = \binom{K}{k}(1 - x'_2)^{(K-k)}(x'_2)^{k}$$

defined on the unit square with local variables $(x'_1, x'_2) \in (0,1) \times (0,1)$. A diagram of of the process is shown in Fig. 2.

Bernstein basis polynomials are used because they have the properties of *partition of unity*

$$\sum_{\ell=1}^{L} b^{L}_{\ell}(x'_1) \equiv \sum_{k=1}^{K} b^{K}_{k}(x'_2) \equiv 1 \tag{9}$$

and *positivity*

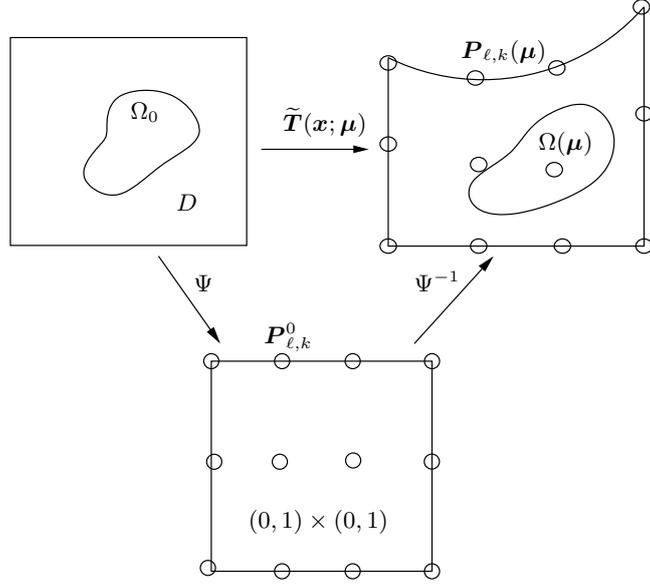$$b^{L,K}_{\ell,k}(x'_1, x'_2) \geq 0 \quad \text{for all } x'_1, x'_2$$

Figure 2: Schematic diagram of the control points $\boldsymbol{P}_{\ell,k}$, the free-form deformation map $\widetilde{\boldsymbol{T}}(\boldsymbol{x}; \boldsymbol{\mu})$, and the resulting deformation when applied to the reference domain Ω_0.

in $(0,1) \times (0,1)$. Also, they can be evaluated in a numerically stable fashion by the de Casteljau algorithm [10]. Extensions of free-form deformations to Non-Uniform and Rational B-Spline (NURBS) basis functions are available [4, 20, 44, 45, 46]. These permit the exact representation of shapes that are conic sections, for example. The simple polynomial basis functions have derivatives that are expressible in terms of tensor products of lower order Bernstein polynomials using the well known derivative formula [10]:

$$\nabla b_{\ell,k}^{L,K}(x_1', x_2') = \begin{bmatrix} L \left[b_{\ell-1}^{L-1}(x_1') - b_{\ell}^{L-1}(x_1') \right] b_k^K(x_2') \\[2mm] K \left[b_{k-1}^{K-1}(x_2') - b_k^{K-1}(x_2') \right] b_{\ell}^L(x_1') \end{bmatrix}^T. \tag{10}$$

Denote by $\boldsymbol{J_F}$ the Jacobian matrix of any function $\boldsymbol{F} : \mathbb{R}^2 \to \mathbb{R}^2$. We compute the Jacobian of the map $\widetilde{\boldsymbol{T}}$ by using the gradient formula (10) and the partition of unity property (9) and obtain

$$\boldsymbol{J}_{\widetilde{\boldsymbol{T}}}(\boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{J}_{\Psi}^{-1} \left[\boldsymbol{I} + \sum_{\ell=0}^{L} \sum_{k=0}^{K} \nabla b_{\ell,k}^{L,K}(\Psi(\boldsymbol{x})) \boldsymbol{\mu}_{\ell,k} \right] \boldsymbol{J}_{\Psi}. \tag{11}$$

Thus not only the map $\widetilde{\boldsymbol{T}}$ but also its Jacobian $\boldsymbol{J}_{\widetilde{\boldsymbol{T}}}$ can be stably evaluated. Finally, the parameterized domain $\Omega(\boldsymbol{\mu})$ is obtained using the restriction $\boldsymbol{T} = \widetilde{\boldsymbol{T}}|_{\Omega_0}$ as $\Omega(\boldsymbol{\mu}) = \boldsymbol{T}(\Omega_0)$.

Once the parametric map $\boldsymbol{T}$ has been obtained we can use its inverse to transform the problem back to the reference domain. For example, when equation (3) is the Poisson equation with homogeneous Dirichlet boundary conditions on a parametric domain we use the transformation map $\boldsymbol{T}$ and its Jacobian $\boldsymbol{J_T}$ to define the *transformation tensor* of the problem

$$\boldsymbol{\nu_T}(\boldsymbol{x}; \boldsymbol{\mu}) = \boldsymbol{J_T}^{-T} \boldsymbol{J_T}^{-1} |\det(\boldsymbol{J_T})| \tag{12}$$

9

and the problem is transformed to the reference domain as

$$\int_{\Omega_0} \boldsymbol{\nu_T}(\boldsymbol{x}; \boldsymbol{\mu}) \nabla u^{\mathcal{N}} \cdot \nabla v \, d\Omega_0 = \int_{\Omega_0} fv |\det(\boldsymbol{J_T})| \, d\Omega_0, \tag{13}$$

for all $v \in X^{\mathcal{N}}$.

We have now a flexible parameterization method for small deformations of Ω_0 that can be adjusted by the user by choice of the control points and degrees of freedom. Compared to boundary splines, the parameterization can be chosen to be relatively low-dimensional without sacrificing any of the accuracy in describing the geometry because the map $\boldsymbol{T}$ is independent of the reference domain.

The tensor $\boldsymbol{\nu_T}$ is not affinely parameterized in the sense of (5) (nor is $|\det(\boldsymbol{J_T})|$ appearing in the RHS in general), so further work is needed to obtain an efficient computational procedure. Assume that we have for each component $i, j = 1, 2$ of the transformation tensor an expansion given by

$$[\boldsymbol{\nu_T}]_{i,j} = \sum_{m=1}^{\widetilde{M}_{i,j}} \Theta_{i,j}^m(\boldsymbol{\mu}) \xi_{i,j}^m(\boldsymbol{x}) + \varepsilon_{i,j}(\boldsymbol{x}; \boldsymbol{\mu}) \tag{14}$$

and for the right-hand side an expansion given by

$$f(\boldsymbol{x}) |\det(\boldsymbol{J_T}(\boldsymbol{x}; \boldsymbol{\mu}))| = \sum_{m=1}^{\widetilde{M}_f} \Theta_f^m(\boldsymbol{\mu}) \xi_f^m(\boldsymbol{x}) + \varepsilon_f(\boldsymbol{x}; \boldsymbol{\mu}) \tag{15}$$

where the error terms are guaranteed to be under some tolerance,

$$||\varepsilon_{i,j}(\cdot; \boldsymbol{\mu})||_\infty \leq \varepsilon_{tol}^{EIM} \text{ for all } \boldsymbol{\mu} \in \mathcal{D}, \tag{16}$$

$$||\varepsilon_f(\cdot; \boldsymbol{\mu})||_\infty \leq \varepsilon_{tol}^{EIM} \text{ for all } \boldsymbol{\mu} \in \mathcal{D}. \tag{17}$$

We require all the $\Theta_{i,j}^m$'s, $\xi_{i,j}^m$'s, Θ_f^m's, and ξ_f^m's to be efficiently computable scalar functions. We shall discuss in Sect. 5 a way of approximating the non-affine transformations with affine ones using the empirical interpolation procedure as proposed in [5, 22]. Substituting into (13) and dropping the error terms we obtain for the Laplacian

$$\int_{\Omega_0} \Big[\frac{\partial u^{\mathcal{N}}}{\partial x_1} ([\boldsymbol{\nu_T}]_{1,1} \frac{\partial v}{\partial x_1} + [\boldsymbol{\nu_T}]_{1,2} \frac{\partial v}{\partial x_2})$$

$$+ \frac{\partial u^{\mathcal{N}}}{\partial x_2} ([\boldsymbol{\nu_T}]_{2,1} \frac{\partial v}{\partial x_1} + [\boldsymbol{\nu_T}]_{2,2} \frac{\partial v}{\partial x_2}) \Big] d\Omega_0$$

$$= \sum_m \int_{\Omega_0} \{ \frac{\partial u^{\mathcal{N}}}{\partial x_1} \Big[\Theta_{1,1}^m \xi_{1,1}^m \frac{\partial v}{\partial x_1} + \Theta_{1,2}^m \xi_{1,2}^m \frac{\partial v}{\partial x_2} \Big]$$

$$+ \frac{\partial u^{\mathcal{N}}}{\partial x_2} \Big[\Theta_{2,1}^m \xi_{2,1}^m \frac{\partial v}{\partial x_2} + \Theta_{2,2}^m \xi_{2,2}^m \frac{\partial v}{\partial x_2} \Big] \} d\Omega_0$$

$$= \sum_{i=1}^2 \sum_{j=1}^2 \sum_{m=1}^{\widetilde{M}_{i,j}} \Theta_{i,j}^m(\boldsymbol{\mu}) a_{i,j}^m(u^{\mathcal{N}}, v) = \sum_{m=1}^{\widetilde{M}_f} \Theta_f^m(\boldsymbol{\mu}) f^m(v),$$

10

for all $v \in X^{\mathcal{N}}$, where the bilinear forms

$$a_{i,j}^m(w,v) = \int_{\Omega_0} \xi_{i,j}^m(\boldsymbol{x}) \frac{\partial w}{\partial x_i} \frac{\partial v}{\partial x_j} \, d\Omega_0$$

are continuous, as are the linear forms

$$f^m(v) = \int_{\Omega_0} \xi_f^m(\boldsymbol{x}) v \, d\Omega_0.$$

Now we have an efficient online-offline computational procedure for the matrix assembly. In the offline stage the matrices $\boldsymbol{\mathcal{A}}_{i,j}^m$ corresponding to the bilinear forms $a_{i,j}^m$ are computed and stored. The complete stiffness matrix can then be cheaply assembled in the online stage. For any given $\boldsymbol{\mu}$ we evaluate the $\Theta_{i,j}^m$ and Θ_f^m, and sum together the contributions from each $\boldsymbol{\mathcal{A}}_{i,j}^m$ and respectively $\boldsymbol{\mathcal{F}}^m$. The same thing is done for the vectors $\boldsymbol{\mathcal{F}}^m$ representing the right-hand side. Note, however, that the dimension of the system is still $\mathcal{N} \times \mathcal{N}$, and even with optimal multigrid solvers costs $O(\mathcal{N})$ to solve online. This is too expensive for a repeated evaluation design environment. The solution is therefore approximated with the reduced basis method and equation (7), so that the final form of the reduced basis problem with affine parameterization can be written as

$$\sum_{n=1}^{N} u_n \sum_{m=1}^{\widetilde{M}_{i,j}} \sum_{i=1}^{2} \sum_{j=1}^{2} \Theta_{i,j}^m(\boldsymbol{\mu}) a_{i,j}^m(\zeta_n, \zeta_{n'}) = \sum_{m=1}^{\widetilde{M}_f} \Theta_f^m(\boldsymbol{\mu}) f^m(\zeta_{n'}), \tag{18}$$

for all $n' = 1, \dots, N$. In practice N and the $\widetilde{M}_{i,j}, \widetilde{M}_f$ can be chosen to be quite modest and the solution of this system will still be close to the finite element solution. This is especially true if the number of parameters P is small (say 10-20), as is the case when we use free-form deformations.

5. Empirical interpolation method for nonpolynomial affine approximations

For many linear PDEs modelling physical phenomena the parameters $\boldsymbol{\mu}$ enter affinely into the bilinear form. However, with shape design this is not usually the case unless we are dealing with a pure optimal sizing problem. The possibility of finding good affine approximations of the form (14) for the coefficients of generic parametric PDEs has been considered in [5, 13, 22, 25, 26]. Applications of the proposed *empirical interpolation method* for reduced basis methods can be found in [16, 40, 51].

The empirical interpolation method has two stages. In the first stage a set of hierarchical interpolation spaces is constructed. For each term $i, j = 1, 2$ of the transformation tensor $[\boldsymbol{\nu_T}]_{i,j} \in C^\infty(\Omega_0 \times \mathcal{D}, \mathbb{R})$ we choose an initial parameter value $\boldsymbol{\mu}_{i,j}^1$ and define the first shape function as $\xi_{i,j}^1(\boldsymbol{x}) = [\boldsymbol{\nu_T}(\boldsymbol{x}, \boldsymbol{\mu}_{i,j}^1)]_{i,j}$, and the interpolation space $W_{i,j}^1 = \mathrm{span}\{\xi_{i,j}^1\}$. The hierarchical interpolation spaces $W_{i,j}^{\widetilde{M}}, \widetilde{M} = 2, \dots, \widetilde{M}_{i,j}$ are then built iteratively by solving the problem

$$\boldsymbol{\mu}_{i,j}^m = \underset{\boldsymbol{\mu} \in \Xi^{train}}{\operatorname{argmax}} \inf_{w \in W_{i,j}^{m-1}} ||[\boldsymbol{\nu_T}(\cdot, \boldsymbol{\mu})]_{i,j} - w||_{L^\infty(\Omega_0)} \tag{19}$$

to obtain the parameter values $\boldsymbol{\mu}_{i,j}^m$. Here $\Xi^{train} \subset \mathcal{D}$ is a finite training set of parameter values and thus the problem (19) is a linear programming problem. The $\widetilde{M}_{i,j}$'s are chosen so that the tolerance criteria (16) is achieved for each component of the tensor $\boldsymbol{\nu_T}$. We then define the mth shape function $\xi_{i,j}^m(\boldsymbol{x}) = [\boldsymbol{\nu_T}]_{i,j}(\boldsymbol{x}, \boldsymbol{\mu}_{i,j}^m)$ and the $\widetilde{M}_{i,j}$th interpolation space $W_{i,j}^{\widetilde{M}} = \mathrm{span}\{\xi_{i,j}^m, m = 1, \ldots, \widetilde{M}_{i,j}\}$. By $\widehat{M} := \sum_{i,j=1}^{2} \widetilde{M}_{i,j} + \widetilde{M}_f$ we denote the total number of terms in the expansion of the transformation tensor and right-hand side.

The interpolation spaces $W_{i,j}^{\widetilde{M}}$ having been constructed, we define a set of Lagrange interpolation points also known as "magic points". The first interpolation point is chosen as

$$\boldsymbol{z}_{i,j}^1 = \operatorname*{argsup}_{\boldsymbol{x} \in \Omega_0} |\xi_{i,j}^1(\boldsymbol{x})|$$

and the corresponding normed basis function as

$$\widehat{\xi}_{i,j}^1(\boldsymbol{x}) = \xi_{i,j}^1(\boldsymbol{x})/\xi_{i,j}^1(\boldsymbol{z}_{i,j}^1).$$

For $2 \leq \widetilde{M} \leq \widetilde{M}_{i,j}$ we obtain the interpolation points iteratively by solving the interpolation problem at the already obtained magic points

$$\sum_{m=1}^{\widetilde{M}-1} \sigma_{\widetilde{M}-1,i,j}^m \widehat{\xi}_{i,j}^m(\boldsymbol{z}_{i,j}^{m'}) = [\boldsymbol{\nu_T}]_{i,j}(\boldsymbol{z}_{i,j}^{m'}, \boldsymbol{\mu}_{i,j}^{\widetilde{M}}) \quad \text{for } 1 \leq m' \leq \widetilde{M}-1$$

for the coefficients $\sigma_{\widetilde{M}-1,i,j}^m$, $m = 1, \ldots, \widetilde{M}-1$, and then looking at the residual function

$$r_{i,j}^{\widetilde{M}}(\boldsymbol{x}) = \xi_{i,j}^{\widetilde{M}}(\boldsymbol{x}) - \sum_{m=1}^{\widetilde{M}-1} \sigma_{\widetilde{M}-1,i,j}^m \widehat{\xi}_{i,j}^m(\boldsymbol{x}).$$

The other interpolation points are obtained from

$$\boldsymbol{z}_{i,j}^{\widetilde{M}} = \operatorname*{argsup}_{\boldsymbol{x} \in \Omega_0} |r_{i,j}^{\widetilde{M}}(\boldsymbol{x})|$$

and the corresponding normed basis functions

$$\widehat{\xi}_{i,j}^{\widetilde{M}}(\boldsymbol{x}) = r_{i,j}^{\widetilde{M}}(\boldsymbol{x})/r_{i,j}^{\widetilde{M}}(\boldsymbol{z}_{i,j}^{\widetilde{M}}).$$

Denote by $\boldsymbol{B}_{i,j}$ the interpolation matrices

$$[\boldsymbol{B}_{i,j}]_{m',m} = \widehat{\xi}_{i,j}^m(\boldsymbol{z}_{i,j}^{m'}), \quad 1 \leq m, m' \leq \widetilde{M}_{i,j}$$

and by $\boldsymbol{z}_{i,j}$ the vector of all the interpolation points. Finally, the non-affinely parameterized bilinear form is replaced with an affinely parameterized approximation

$$a(u, v; \boldsymbol{\mu}) \approx \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{m=1}^{\widetilde{M}_{i,j}} \Theta_{i,j}^m(\boldsymbol{\mu}) a_{i,j}^m(u, v), \tag{20}$$

where the matrix versions of

$$a_{i,j}^m(w,v) = \int_{\Omega_0} \widehat{\xi}_{i,j}^m(\boldsymbol{x}) \frac{\partial w}{\partial x_i} \frac{\partial v}{\partial x_j} \, d\Omega_0 \tag{21}$$

are computed once in the offline stage, and the coefficient functions $\Theta_{i,j}^m(\boldsymbol{\mu})$ can be evaluated by solving $\widetilde{M}_{i,j} \times \widetilde{M}_{i,j}$, $1 \leq i, j \leq 2$, linear systems of the form

$$\boldsymbol{B}_{i,j}\boldsymbol{\Theta}_{i,j}(\boldsymbol{\mu}) = [\boldsymbol{\nu_T}(\boldsymbol{z}_{i,j};\boldsymbol{\mu})]_{i,j}$$

for each desired value of $\boldsymbol{\mu}$, where $\boldsymbol{\Theta}_{i,j}(\boldsymbol{\mu})$ is a vector of the values $\Theta_{i,j}^m(\boldsymbol{\mu})$, $m = 1, \ldots, \widetilde{M}_{i,j}$. The same treatment can be performed to the right-hand side to obtain expansion (15), which we omit for brevity. The efficient online-offline decomposition leading to equation (7) is then recovered.

We make some remarks about the empirical interpolation method:

- The interpolation points $\boldsymbol{z}_{i,j}^m$ are chosen by a greedy procedure and are suboptimal, but in effect are not far away from the optimal interpolations points (in cases where these can be computed); for a demonstration see [22].

- The basis functions $\widehat{\xi}_{i,j}^m$ only need to be evaluated offline at the mesh points. Compared to the analytical expressions for the $[\boldsymbol{\nu_T}]_{i,j}$'s, which are complicated rational functions[1] and expensive to evaluate even with the recursive algorithm, they are also computationally more efficient.

- In our case the transformation tensor $\boldsymbol{\nu_T}$ is constructed with Bernstein polynomials and the interpolation basis functions $\widehat{\xi}_{i,j}^m$ are rational functions. If the greedy procedure for constructing the interpolation spaces uses a large enough training set Ξ^{train}, we can expect that approximation (20) is not worse than those given by standard rational interpolation techniques.

6. Inverse airfoil design in the NACA family

We provide a possible application of the computational methods presented in the previous sections by considering an inverse airfoil design problem in 2-d potential flow. Fig. 3 shows one reference domain Ω_0 that surrounds a NACA0012 airfoil [1] with 5° angle of attack.

The potential flow equation around the airfoil follows from the (unphysical) assumption that the flow field $\boldsymbol{u}$ is steady and irrotational, i.e. $\partial_t \boldsymbol{u} = 0$ and $\nabla \times \boldsymbol{u} = 0$. Then there exists a scalar function φ such that $\boldsymbol{u} = \nabla\varphi$. This is called the flow potential and is given as the solution of the Laplace equation

$$\mathrm{div}(\nabla\varphi(\boldsymbol{x})) = 0, \quad \text{on } \Omega(\boldsymbol{\mu}). \tag{22}$$

---

[1]For our test problem with eight parameters a function that computes the first component $[\boldsymbol{\nu_T}]_{1,1}$ using the analytical expression is 20 pages long. While the code that evaluates the same function using the de Casteljau algorithm is much shorter, it is not computationally more effective.
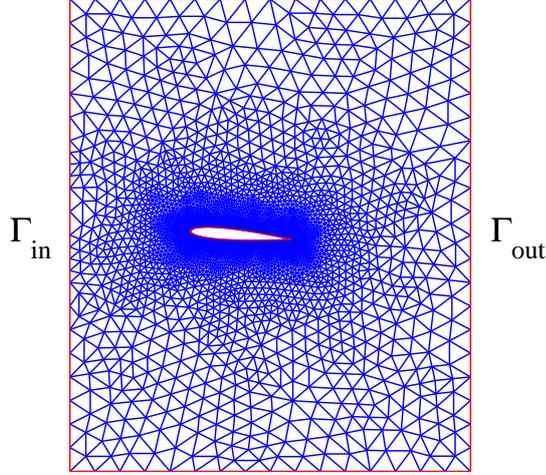
Figure 3: Initial configuration of a NACA0012 airfoil and the surrounding mesh. The mesh has $\mathcal{N} = 8043$ nodes.

An equation for the pressure is obtained using the assumptions of incompressibility and irrotationality as

$$\nabla p(\boldsymbol{x}) = -\nabla \tfrac{1}{2} |\nabla \varphi(\boldsymbol{x})|^2$$

and finally

$$p(\boldsymbol{x}) = p_0 - \tfrac{1}{2} |\nabla \varphi(\boldsymbol{x})|^2,$$

where p_0 is a reference pressure. This is the *Bernoulli equation* [28]. We have the following boundary conditions:

$$\frac{\partial \varphi}{\partial n} = 1, \text{ on } \Gamma_{in}$$

$$\varphi = 0, \text{ on } \Gamma_{out}$$

$$\frac{\partial \varphi}{\partial n} = 0, \text{ elsewhere.}$$

The exterior flow velocity is imposed with Neumann conditions, while the Dirichlet condition fixes the solution of the potential flow problem to be unique.

The inverse design problem of an airfoil involves prescribing some properties of the target airfoil, in our case the pressure distribution on the airfoil surface, and then trying to generate an airfoil that has the desired property. We performed two test cases. In Case A the initial guess was the NACA0012 airfoil and the target airfoil was the NACA4412 airfoil, which is a slightly thicker, nonsymmetric, and has increased camber. In Case B we took NACA4412 as the initial guess and NACA0012 as the target airfoil. This demonstrates going from a symmetric initial airfoil to a nonsymmetric target airfoil, and vice versa. The proof-of-concept solver was implemented in Matlab by using the symbolic toolbox for building the FFDs and constructing the empirical interpolation basis functions. The FFD configuration used was a 6×4 grid of control points
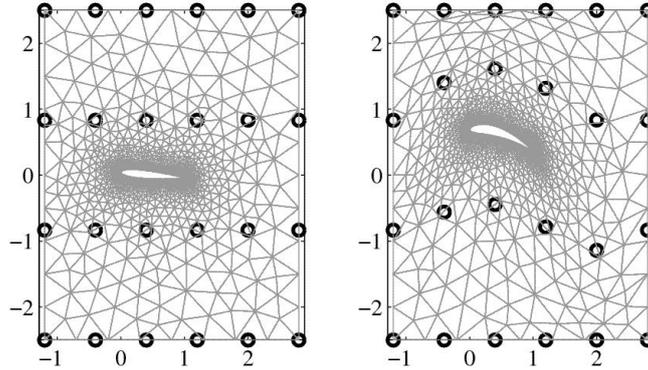
14

Figure 4: The reference configuration, the control point lattice, and a perturbed configuration in Case A with a NACA0012 initial airfoil.

Table 1: Approximations details

Approximation property	Value
Number of mesh nodes $\mathcal{N}$	8043
Lattice of FFD control points $\boldsymbol{P}_{\ell,k}$	6×4
Number of DOFs/design variables K	8
Error tolerance for the RB greedy ε_{tol}^{RB}	1e-4
Number of reduced basis functions N	52
Error tolerance for the EIM greedy ε_{tol}^{EIM}	2.5e-3
Total number of EIM basis functions $\widehat{M}$	165
Parameter range for free μ_k's	$[-0.5, 0.5]$

on the domain of which the eight central ones were allowed to move in the x_2'-direction, giving a total of 8 degrees of freedom. In Fig. 4 we show the lattice of control points and an example of the deformation obtained by moving the control points. In reality the mesh remains fixed at all times, thanks to the inverse transformation $\boldsymbol{T}^{-1}$ that allows us to return the potential flow problem (22) to the reference domain Ω_0. Table 1 lists the relevant dimensions of the approximation spaces for the finite element, reduced basis, and empirical interpolation approximations respectively for Case A. The results for the other case are similar and details are omitted. The tolerances for the greedy algorithm and the empirical interpolation are chosen such that the error on the interpolation is not dominating over the reduced basis convergence of the greedy algorithm (see [40]).

In this way, the number of affine terms $\widehat{M}$ and the size of the reduced basis N are chosen so that the two tolerances for the empirical interpolation ε_{tol}^{EIM} and reduced basis greedy algorithm ε_{tol}^{RB} are satisfied. We also remark that, since the inhomogeneous boundary condition $\frac{\partial \varphi}{\partial n} = 1$ was chosen on an edge perpendicular to the x_1-direction, the FFD-parameterization does not enter into the boundary term given by integration by parts and thus there was no need to perform the EIM on the right-hand side in this particular parameterization. In the *offline stage* we perform all the procedures to prepare the parameter dependent calculations. All the

15

parameter independent quantities are computed and stored. In more detail, we:

1. Compute the symbolic expression for the transformation tensor $\boldsymbol{\nu_T}$ given by (12);

2. Perform the empirical interpolation to obtain the expansion (20). The stopping tolerance for the greedy procedure was $\varepsilon_{tol}^{EIM} = 2.5\text{e-}3$ and this was satisfied with a combined total of $\widehat{M} = 165$ terms in the expansion;

3. Perform a greedy procedure for choosing the reduced basis snapshots $\boldsymbol{\mu}^n$ and compute the corresponding basis solutions ζ_n. For this part we have utilized the rbMIT package [18] in order to implement the reduced basis algorithms, i.e. for basis construction and offline-online computations, as well as the affine decompositions. The error estimates for the potential are computed according to the methodology proposed in [19, 39, 41] for coercive problems. The nonaffine parameterization and use of the EIM together with these estimates leads to error bounds that are not necessarily rigorous. However, as long as the tolerance for the EIM is chosen small enough that the empirical interpolation error does not dominate over the reduced basis approximation error, a sufficient guarantee of approximation stability is obtained in our experience. It is also possible to extend the error estimator for the nonaffine case by including a correction term coming from the error of the EIM, see e.g. [25]. For potential flows, given the error bounds on the potential solution, it is possible to get readily error bounds on the pressure and/or velocity (see [39]) obtained by the Bernoulli equation. In both cases we chose by the greedy procedure $N = 52$ basis functions, which was achieved with tolerance $\varepsilon_{tol}^{RB} = 1\text{e-}4$ for the noncorrected error bound.

4. (a) Assemble the finite element matrices $\boldsymbol{\mathcal{A}}_{i,j}^m \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ corresponding to the bilinear forms (21) and the right-hand sides $\boldsymbol{\mathcal{F}}^m \in \mathbb{R}^{\mathcal{N}}$;

   (b) Pre- and post-multiply the matrices with the reduced basis representation to obtain the reduced size system matrices $\boldsymbol{A}_{i,j}^m \in \mathbb{R}^{N \times N}$ such that

   $$\boldsymbol{A}_{i,j}^m = \boldsymbol{\mathcal{Z}}^T \boldsymbol{\mathcal{A}}_{i,j}^m \boldsymbol{\mathcal{Z}}$$

   and the reduced right-hand sides $\boldsymbol{f}^m \in \mathbb{R}^N$ such that

   $$\boldsymbol{f}^m = \boldsymbol{\mathcal{Z}}^T \boldsymbol{\mathcal{F}}^m,$$

   where

   $$\boldsymbol{\mathcal{Z}} = \begin{bmatrix} \zeta_1 \ \zeta_2 \ \ldots \ \zeta_n \end{bmatrix}$$

   is a $\mathcal{N} \times N$ matrix with the basis vectors ζ_n as columns.

   The offline stage is dependent on the reference domain Ω_0, the boundary conditions, and the FFD setup. Therefore for both Case A and Case B we had to perform the offline stage separately. The FFD and EIM steps are independent of the reference shape (in this case the initial shape of the airfoil), so
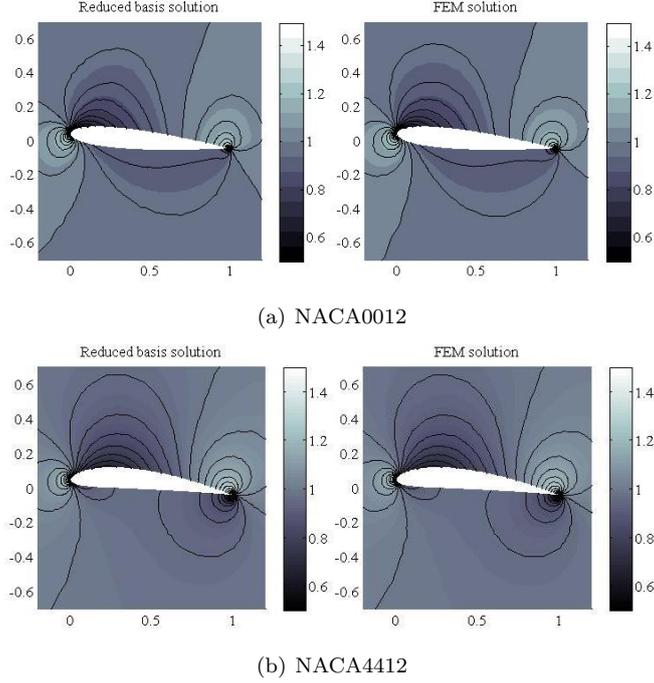
(a) NACA0012

(b) NACA4412

Figure 5: The dynamic pressure field around the airfoils for the FEM and RB solutions for $N = 52$ basis functions.

these had to be only performed once for both cases. In total the work that had to be done in each case consisted of matrix assembly and greedy algorithms for snapshot selection, together with the Successive Constraint Method for estimation of the lower bound of coercivity constant.

The *online stage* is driven by the the optimization algorithm that involves an order of 10^2 evaluations of the cost function and the underlying state equations, and consists of using the stored matrices $\boldsymbol{A}_{i,j}^m$ and vectors $\boldsymbol{f}^m$ to assemble and solve the system (7). For the flow potential computation of the snapshot solutions with the FEM we used linear triangular elements and then performed patch averaging [55] to post-process the solution and recover the solution gradients in order to compute the dynamic pressure. In Fig. 5(a) we display the dynamic pressure fields for the reference airfoil ($\boldsymbol{\mu} = \boldsymbol{0}$) for both the FEM and RB solutions in Case A, and in Fig. 5(b) for Case B. The error between the "truth approximation" FEM solution and its RB approximation is $||\varphi^{\mathcal{N}} - \varphi_N^{\mathcal{N}}||_X = 1.12\text{e-}5$ at this parameter value for Case A (relative error 5.56e-7) and $||\varphi^{\mathcal{N}} - \varphi_N^{\mathcal{N}}||_X = 8.9263\text{e-}6$ for Case B (relative error 4.44e-7).

To enable the comparison of pressure distributions on the surfaces of two different airfoils, define $\gamma_{\boldsymbol{\mu}} : [0,1] \to \Gamma_{airfoil}(\boldsymbol{\mu})$ to be a closed curve with counterclockwise parameterization such that $\gamma_{\boldsymbol{\mu}}' \equiv$ constant almost everywhere. The scaled pressure distribution on a parametric airfoil $\Omega(\boldsymbol{\mu})$ is then defined as

$$\widetilde{p}(r, \boldsymbol{\mu}) := p_0 - \tfrac{1}{2}|\nabla\varphi(\gamma_{\boldsymbol{\mu}}(r), \boldsymbol{\mu})|^2 \text{ for } r \in [0,1]. \tag{23}$$

The pressure field is recovered in the online step by first recovering the gradient of the reduced basis solution

of the flow potential and then using the Bernoulli equation. Using the scaled pressure we can write the L^2-defect functional for the pressure distribution

$$J(\boldsymbol{\mu}) = \frac{\int_0^1 |\widetilde{p}(r, \boldsymbol{\mu}) - \widetilde{p}_{target}(r)|^2 \, dr}{\int_0^1 |\widetilde{p}_{target}(r)|^2 \, dr} + \eta \left[\alpha(\boldsymbol{\mu}) - 5° \right]^2,$$

where $\widetilde{p}(r, \boldsymbol{\mu})$ is the scaled pressure distribution on the perturbed airfoil surface, $\widetilde{p}_{target}$ is the scaled pressure distribution on the target airfoil surface given by the finite element method, and $\alpha(\boldsymbol{\mu})$ is the desired angle of attack of the parametric airfoil. The penalty term was added to ensure that the optimized airfoil has the desired angle of attack. We used a fixed value of $\eta = 10^2$ for all tests. The penalty approach works well for this problem, but is limited to local optimization where the optimal design is close to the initial design. We also introduce the reduced defect functional

$$J_N(\boldsymbol{\mu}) = \frac{\int_0^1 |\widetilde{p}_N(r, \boldsymbol{\mu}) - \widetilde{p}_{target}(r)|^2 \, dr}{\int_0^1 |\widetilde{p}_{target}(r)|^2 \, dr} + \eta \left[\alpha(\boldsymbol{\mu}) - 5° \right]^2, \tag{24}$$

where the reduced pressure $\widetilde{p}_N$ is computed from (23) by replacing φ with $\varphi_N^{\mathcal{N}}$ the reduced basis approximation of the flow potential for the purpose of evaluating the cost functional (24) in a highly effective fashion. The classical inverse design optimization problem in the parametric setting is then to find $\boldsymbol{\mu}^*$ the solution of

$$\min_{\boldsymbol{\mu} \in \mathcal{D}} \quad J(\boldsymbol{\mu}), \tag{25}$$

a highly nonlinear programming problem but with relatively few parameters. The reduced version of this optimization problem uses instead the reduced functional and is to find $\boldsymbol{\mu}_N^*$ the solution of

$$\min_{\boldsymbol{\mu} \in \mathcal{D}} \quad J_N(\boldsymbol{\mu}), \tag{26}$$

Both minimization problems were solved using Matlab's nonlinear programming routines. For problem (26) the minimization algorithm converged to an optimality tolerance of 1e-6 with 19 SQP iterations using numerical Jacobians and a total of 192 reduced basis solutions in Case A, and with 20 SQP iterations and 199 reduced basis solutions in Case B. The final value of the reduced cost functional was $J_N(\boldsymbol{\mu}_N^*) = 4.7\text{e-}3$ in Case A (from an initial value $J_N(\boldsymbol{\mu}_N^0) = 0.53$) and $J_N(\boldsymbol{\mu}_N^*) = 1.3\text{e-}3$ in Case B (from an initial value $J_N(\boldsymbol{\mu}_N^0) = 0.76$). When computing with the full cost functional we obtained the final cost $J(\boldsymbol{\mu}_N^*) = 5.28\text{e-}3$ in Case A, and $J(\boldsymbol{\mu}_N^*) = 1.25\text{e-}3$ in Case B. Thus it can be observed that for these test problems the reduced optimization problem (26) was successfully solved and, while the obtained solutions are suboptimal, they are reasonable also approximations for the solution of the full optimization problem (25). In Table 2 we give the initial and final values of the reduced cost functional, and measure the suboptimality of the inverse design by looking at the relative error

$$\frac{J(\boldsymbol{\mu}_N^*) - J(\boldsymbol{\mu}^*)}{J(\boldsymbol{\mu}^*)}$$

Table 2: Comparison of the reduced optimal solution $\boldsymbol{\mu}_N^*$ and the true optimal solution $\boldsymbol{\mu}^*$

		Case A	Case B
Initial value	$J_N(\boldsymbol{\mu}_N^0)$	5.32e-1	7.60e-1
Reduced optimum	$J_N(\boldsymbol{\mu}_N^*)$	4.71e-3	1.30e-3
True cost at $\boldsymbol{\mu}_N^*$	$J(\boldsymbol{\mu}_N^*)$	5.28e-3	1.25e-3
True optimum	$J(\boldsymbol{\mu}^*)$	4.63e-3	1.19e-3
Suboptimality	$\frac{J(\boldsymbol{\mu}_N^*)-J(\boldsymbol{\mu}^*)}{J(\boldsymbol{\mu}^*)}$	14.0%	5.0%

between the true optimal value and the value of J at the reduced optimum. As we can observe, the reduced optimization solutions are suboptimal, but only by less than 15%.

The pressure distributions on the surface of the designed airfoil compared to the target are displayed in Fig. 6(a) and Fig. 6(b). The dynamic pressure fields around the target airfoil and the final inverse design obtained from the optimization procedure are shown in Fig. 7(a). Both solutions were visualized with the FEM. Already with eight degrees of freedom the pressure field around the airfoil is qualitatively near the true one, and the airfoil shape is close to the target airfoil chosen to test the method.

The error of the reduced basis solution for the inverse design was $||\varphi^{\mathcal{N}} - \varphi_N^{\mathcal{N}}||_X = 3.78\text{e-}4$ in Case A (relative error 2.03e-5) and $||\varphi^{\mathcal{N}} - \varphi_N^{\mathcal{N}}||_X = 6.44\text{e-}5$ in Case B (relative error 3.35e-6). While the a posteriori estimators used to control the greedy basis selection are not strictly rigorous for this nonaffinely parameterized problem, the obtained approximations are below the specified error tolerance at least for these parameter points.

A few words about the computational cost are in order. The offline stage of the reduced basis is expensive, and took several hours of computation time to perform for this single problem. The online stage is much more effective, taking only 0.06 s for each matrix assembly and solution of the system (18). We estimate this to be about 300 times faster than assembling and solving the same equations with the FEM. This means that solving the reduced inverse design problem (26), which typically requires an order of 10^2 function evaluations during the nonlinear programming loop, takes only 50 s on a typical desktop computer. In comparison, solving the inverse design problem (25) with the full FE solutions used to evaluate the pressure takes about 1.5 hours on the same setup. The reduced optimization is roughly 100 times more efficient (and not 300 times as expected) due to the fact that in addition to cost functional evaluations we also have to perform post-processing to recover the pressure and the complexity of this depends on $\mathcal{N}$. The offline stage is also independent of the target pressure distribution chosen and even the cost functional used. We could therefore opt to solve other CFD optimization problems, such as lift-drag optimization, without incurring any additional costs in the offline stage.
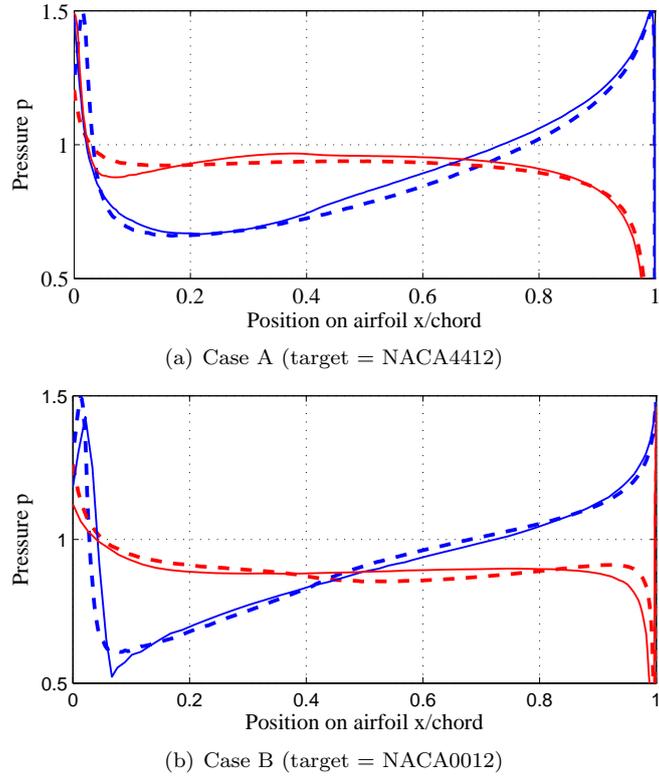
(a) Case A (target = NACA4412)

(b) Case B (target = NACA0012)

Figure 6: Pressure distributions on the airfoil top and bottom surfaces on the target airfoil (solid line) and the inverse design (dashed)

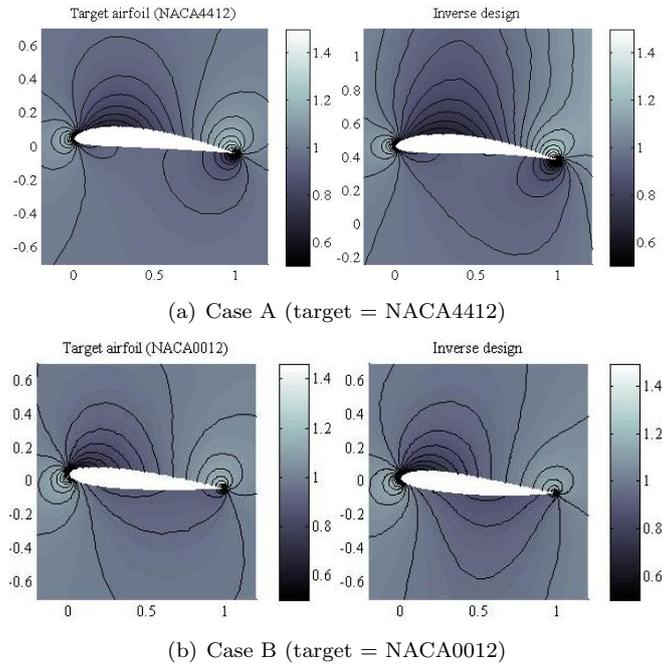(a) Case A (target = NACA4412)

(b) Case B (target = NACA0012)

Figure 7: The dynamic pressure field around the target airfoil (NACA4412) and the inverse design. Both visualizations were obtained using the FEM.

7. Conclusions

We have presented a method for efficient shape design of PDE-modelled systems. We combined free-form deformations to generate the parametric geometry and reduced basis methods to solve the resulting parameterized PDEs in an inverse problem framework of shape optimization. To demonstrate the applicability of the proposed approach we considered an inverse airfoil design problem in 2-d potential flow. It was confirmed that using free-form deformations to generate the parametric domain deformations a small number of design variables is sufficient to solve already quite challenging inverse airfoil design problems. The reduced basis methods can then be used to obtain an efficient online procedure for shape optimization, where several iterative evaluations of the state equations are needed and computational efficiency is appreciated. The free-form deformations are independent of the mesh, geometry, and even the PDE model. Thus we believe that in the future it will be possible to extend the proposed methods to more realistic and interesting problems using different viscous flow models.

References

[1] I.H. Abbott and A.E. von Doenhoff. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Courier Dover Publications, 1959.

[2] V. Agoshkov, A. Quarteroni, and G. Rozza. Shape design in aorto-coronaric bypass anastomoses using perturbation theory. *SIAM J. Numer. Anal.*, 44(1):367–384, 2006.

[3] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA J.*, 16:525–528, 1978.

[4] E.I. Amoiralis and I.K. Nikolos. Freeform deformation versus B-spline representation in inverse airfoil design. *J. Comput. Inf. Sci. Engrg.*, 8, 2008.

[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] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear programming - theory and algorithms (2nd Ed.)*. John Wiley & Sons, New York, 1993.

[7] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods (2nd Ed.)*. Springer-Verlag, New York, 2002.

[8] C. de Boor. *A practical guide to splines*. Springer-Verlag, New York, 2001.

[9] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier– Stokes equations by the natural norm approach. *SIAM J. Num. Anal.*, 46(4):2039–2067, 2008.

[10] G. Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Morgan Kaufmann, 2001.

[11] J.P. Fink and W.C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *Z. Angew. Math. Mech.*, 63(1):21–28, 1983.

[12] K.C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress Aerospace Sci.*, 38:43–76, 2002.

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

[14] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 39(1):157–181, 2005.

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

[16] B. Haasdonk, M. Ohlberger, and G. Rozza. A reduced basis method for evolution schemes with parameter-dependent explicit operators. Technical Report 09/07 - N, FB 10, University of Münster, 2007. Accepted by ETNA.

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

[18] D.B.P. Huynh, N.C. Nguyen, A.T. Patera, and G. Rozza. http://augustine.mit.edu, Rapid reliable solution of the parametrized partial differential equations of continuum mechanics and transport (web site).

[19] 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 costants. *C. R. Acad. Sci. Paris. Sér. I Math.*, 345:473–478, 2007.

[20] H.J. Lamousin and W.N. Waggenspack. NURBS-based free-form deformations. *IEEE Comput. Graph. Appl.*, 14(6):59–65, 1994.

[21] P.A. LeGresley and J.J. Alonso. Airfoil design optimization using reduced order models based on proper orthogonal decomposition. *Proc. Fluids 2000 Conf. Exhibit*, 2000.

[22] Y. Maday, N.C. Nguyen, A.T. Patera, and G.S.H. Pau. A general multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.*, 8(1), 2009.

[23] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids.* Oxford University Press, Oxford, 2001.

[24] A.M. Morris, C.B. Allen, and T.C.S. Rendall. CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation. *Int. J. Numer. Methods Fluids*, 58:827–860, 2008.

[25] N.C. Nguyen. A posteriori error estimation and basis adaptivity for reduced-basis approximation of nonaffine-parametrized linear elliptic partial differential equations. *J. Comp. Phys.*, 227:983–1006, 2007.

[26] N.C. Nguyen, A.T. Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Methods Engrg.*, 73:521–543, 2008.

[27] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA J.*, 18(4):455–462, 1980.

[28] R.L. Panton. *Incompressible Flow (3rd Ed).* Wiley & Sons, Chichester, 2005.

[29] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equation.* Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering.

[30] L.A. Piegl and W. Tiller. *The NURBS Book.* Springer-Verlag, New York, 1997.

[31] T.A. Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Math. Comput.*, 45(172):487–496, 1985.

[32] C. Praveen and R. Duvigneau. Low cost PSO using metamodels and inexact pre-evaluation: Application to aerodynamic shape design. *Comput. Methods Appl. Mech. Engrg.*, 198:1087–1096, 2009.

[33] C. Prud'homme, D.V. Rovas, K. Veroy, and A.T. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 36(5):747–771, 2002.

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

[35] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numer. Methods Partial Differential Equations*, 23(4):923–948, 2007.

[36] A. Quarteroni, G. Rozza, L. Dedé, and A. Quaini. Numerical approximation of a control problem for advection-diffusion processes. In *System Modeling and Optimization, Ceragioli et al.(Eds.), Proc. 22nd IFIP TC7 Conf., Turin, 2005.* Springer-Verlag, New York, 2006.

[37] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problem. In *Advances in Numerical Mathematics, W. Fitzgibbon, R. Hoppe, J. Periaux, O. Pironneau, and Y. Vassilevski (Eds.),* pages 193–216, 2007.

[38] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucker. Surrogate-based analysis and optimization. *Prog. Aerospace Sci.*, 41:1–28, 2005.

[39] G. Rozza. Reduced basis approximation and a posteriori error bounds for potential flows in parametrized geometries. *Submitted*, 2009.

[40] G. Rozza. Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. *Comput. Vis. Sci.*, 12(1):23–35, 2009.

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

[42] G. Rozza and K. Veroy. On the stability of Reduced Basis methods for Stokes Equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244–1260, 2007.

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

[44] J.A. Samareh. Novel multidisciplinary shape parameterization approach. *J. Aircraft*, 38(6), 2001.

[45] J.A. Samareh. Aerodynamic shape optimization based on free-form deformation. In *Proc. 10th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, volume 6, pages 3672–3683, 2004.

[46] S.S. Sarakinos, E. Amoiralis, and I.K. Nikolos. Exploring freeform deformation capabilities in aerodynamic shape parameterization. *Proc. Int. Conf. Comput. Tool*, 1:535–538, 2005.

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

[48] J. Sokolowski and J.-P. Zolésio. *Introduction to shape optimization: shape sensitivity analysis.* Springer-Verlag, Berlin, 2003.

[49] W. Song and A.J. Keane. A study of shape parameterisation methods for airfoil optimization. In *Proc. 10th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, volume 6, 2004.

[50] N. Thiyagarajan and R.V. Grandhi. 3d preform shape optimization in forging using reduced basis techniques. *Engrg. Optim.*, 37(8):797–811, 2005.

[51] T. Tonn and K. Urban. A reduced-basis method for solving parameter-dependent convection-diffusion problems around rigid bodies. In *Proc. ECCOMAS CFD*, 2006.

[52] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Meth. Fluids*, 47(8-9):773–788, 2005.

[53] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations (AIAA Paper 2003-3847). In *Proc. 16th AIAA Comput. Fluid Dynamics Conf., June 2003*, 2003.

[54] O.C. Zienkiewicz, R.T. Taylor, J.Z. Zhu, and P. Nithiarasu. *The Finite Element Method*. Elsevier Butterworth Heinemann, Oxford, 2005.

[55] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and à posteriori estimates. Part 1: The recovery technique. *Int. J. Numer. Methods Engrg.*, 33(7):1331–1364, 1992.