Certified Reduced Basis Method for Affinely Parametric Isogeometric Analysis NURBS Approximation

note finali coverpage

(Article begins on next page)

24 April 2024

Certified Reduced Basis Method for Affinely Parametric Isogeometric Analysis NURBS Approximation

Denis Devaud [*]   Gianluigi Rozza [†]

1   Introduction and Motivation

In this work we apply reduced basis methods for parametric PDEs to an isogeometric formulation based on NURBS. The motivation for this work is an integrated and complete work pipeline from CAD to parametrization of domain geometry, then from full order to certified reduced basis solution. IsoGeometric Analysis (IGA) is a growing research theme in scientific computing and computational mechanics, as well as reduced basis methods for parametric PDEs. Their combination enhances the solution of some class of problems, especially the ones characterized by parametrized geometries we introduced in this work. For a general overview on Reduced Basis (RB) methods we recall [7, 15] and on IGA [3]. This work wants to demonstrate that it is also possible for some class of problems to deal with affine geometrical parametrization combined with a NURBS IGA formulation. This is what this work brings as original ingredients with respect to other works dealing with reduced order methods and IGA (set in a non-affine formulation, and using a POD [2] sampling without certification: see for example for potential flows [12] and for Stokes flows [17]). In this work we show a certification of accuracy and a complete integration between IGA formulation and parametric certified greedy RB formulation. Section 2 recalls the abstract setting for parametrized PDEs, Section 3 recalls IGA setting, Section 4 deals with RB formulation, and Section 5 illustrates two numerical examples in heat transfer with different parametrization.

2   Elliptic Coercive Parametrized Partial Differential Equations

In what follows, elliptic coercive parametrized partial differential equations are introduced [13, 14, 16]. We consider the following problem: given a parameter $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$s(\boldsymbol{\mu}) = l(u(\boldsymbol{\mu})), \tag{2.1}$$

where $u(\boldsymbol{\mu}) \in X$ is the solution of

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in X. \tag{2.2}$$

Here $a(\cdot, \cdot; \boldsymbol{\mu}) : X \times X \to \mathbb{R}$ is a bilinear, continuous and coercive form associated to a parametrized partial differential equation for every $\boldsymbol{\mu} \in \mathcal{D}$. The space $X := X(\Omega)$ is a Hilbert space on the computational domain $\Omega \subset \mathbb{R}^d$ endowed with the scalar product $(\cdot, \cdot)_X$ for $d = 2, 3$. Since second-order partial differential equations for scalar problems are considered, we have $H_0^1(\Omega) \subset X \subset H^1(\Omega)$, where $H^1(\Omega) := \left\{ v : \Omega \to X \,\middle|\, v \in L^2(\Omega), \nabla v \in L^2(\Omega)^d \right\}$ and $H_0^1(\Omega)$ is the space of functions in $H^1(\Omega)$ whose traces vanish on the boundary. The space $L^2(\Omega)$ denotes the set of square integrable functions. We require moreover that Ω admits a (multipatches) NURBS representation. This is explained in more details in the next

[*]Université de Neuchâtel, Institut de Statistique, Avenue de Bellevaux 51, 2000 Neuchâtel, Switzerland, `denis.devaud@unine.ch`

[†]SISSA, International School for Advanced Studies, Mathematics Area, mathLab, Via Bonomea 265, 34136 Trieste, Italy, `grozza@sissa.it`

section. The functions $f : X \to \mathbb{R}$ and $l : X \to \mathbb{R}$ are linear and continuous functionals. Finally, the set $\mathcal{D}$ denotes the parameter domain and is assumed to be finite-dimensional. More precisely, we write $\mathcal{D} := [a_1, b_1] \times \cdots \times [a_P, b_P] \subset \mathbb{R}^P$ for $a_i, b_i \in \mathbb{R}$, $i = 1, \ldots, P$. We consider here both physical and geometrical parameters. The geometrical case is further investigated in Section 3.3. For a sake of simplicity, the so-called compliant case is considered, that is (i) a is symmetric and (ii) $l = f$.

One of the crucial assumptions to apply the reduced basis method is that a admits an affine decomposition with respect to the parameter $\boldsymbol{\mu}$, that is

$$a(u, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q} \Theta^q(\boldsymbol{\mu}) a^q(u, v). \tag{2.3}$$

Here $\Theta^q : \mathcal{D} \to \mathbb{R}$ denotes a (smooth) $\boldsymbol{\mu}$-dependent function and $a^q : X \times X \to \mathbb{R}$ is a $\boldsymbol{\mu}$-independent bilinear continuous form for $q = 1, \ldots, Q$. Since the compliant case is considered, we require moreover that a^q is symmetric. We do not make any further assumption on the coercivity of a^q. Note that we have assumed that the right-hand side of equation (2.2) is parameter-independent but in practice f may depend on the parameter $\boldsymbol{\mu}$. In that case, we express $f(v; \boldsymbol{\mu})$ as a sum of Q_f products of $\boldsymbol{\mu}$-dependent functions and $\boldsymbol{\mu}$-independent linear continuous forms on X.

For the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$, we define its continuity and coercivity constants for every $\boldsymbol{\mu} \in \mathcal{D}$ as

$$\gamma(\boldsymbol{\mu}) := \sup_{v \in X} \sup_{w \in X} \frac{a(v, w; \boldsymbol{\mu})}{\|v\|_X \|w\|_X},$$

and

$$\alpha(\boldsymbol{\mu}) := \inf_{v \in X} \frac{a(v, v; \boldsymbol{\mu})}{\|v\|_X^2},$$

where $\| \cdot \|_X$ is the norm on X induced by the scalar product $(\cdot, \cdot)_X$. Since a is continuous and coercive, there exists $0 < \alpha_0 \leq \gamma_0 < \infty$ such that $\alpha_0 \leq \alpha(\boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu}) \leq \gamma_0$ for all $\boldsymbol{\mu} \in \mathcal{D}$.

In the following section, we introduce a NURBS approximation of the problem (2.1)-(2.2). Since it is computationally unaffordable to compute such solution for every input parameter, we then consider a RB approximation of it.

3   Isogeometric Analysis NURBS Approximation

In this section, we introduce non-uniform rational B-splines (NURBS) approximation for the problem (2.1)-(2.2). First, a brief survey of B-splines and NURBS functions is conducted and the proper approximation is introduced [1, 3, 8]. We then present in Section 3.3 the affine preconditioning for parameter-dependent domains. This is a necessary assumption to obtain the affine decomposition (2.3) which in turn is crucial to perform RB approximation.

3.1   B-Splines

The B-splines functions are the basis to define NURBS. We give a brief introduction to B-splines in what follows. In the context of isogeometric analysis, the notion of *patches* is very important. They play the role of subdomains and material properties are assumed to be uniform in each patch. Unlike standard finite element (FE) analysis, the B-splines and NURBS basis functions are local to patches and not elements. The FE basis functions map the reference element in the parametric domain to each element in the physical space. B-splines functions take a patch (a set of elements) in the parameter space and map it to multiple elements in the physical domain.

Let us define a *knot vector* in one dimension as a set of non-decreasing coordinates in the parameter domain denoted $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is called the ith *knot*, $i = 1, \ldots, n + p + 1$. Here, p denotes the polynomial order of the B-splines and n the number of basis functions. The B-splines are

completely defined by the knot vector Ξ, the number of basis functions n and their order p. Since this does not affect the construction of B-splines we set by convention $\xi_1 = 0$ and $\xi_{n+p+1} = 1$. Note that repetitions are allowed in the knot vector and are used to control the local regularity across each knot. A knot vector in which ξ_1 and ξ_{n+p+1} are repeated $p+1$ times is called *open* knot vectors. In what follows, we consider only open knot vectors but the construction is the same for general knot vectors. Moreover, we may refer a patch as a subdomain and an element as a knot span, i.e. an interval of the form $[\xi_i, \xi_{i+1}]$.

The B-spline functions are constructed recursively with respect to the polynomial order. For $p = 0$ and an open knot vector Ξ, we define

$$
N_{i,0}(x) := \begin{cases} 1 & \text{if } \xi_i \leq x \leq \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases}
$$

For $p = 1, 2, \ldots$, we define recursively the B-spline basis functions as

$$
N_{i,p}(x) := \frac{x - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(x) + \frac{\xi_{i+p+1} - x}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(x). \tag{3.1}
$$

We present in Figure 1 an example of B-spline basis functions for $n = 10$ and $p = 3$ and the knot vector $\xi = \{0, 0, 0, 0, 0.25, 0.25, 0.25, 0.5, 0.75, 0.75, 1, 1, 1, 1\}$. The equation (3.1) is called the Cox-de-Boor recursion formula [4, 5]. Note that for $p = 0, 1$, the B-spline basis functions coincide with the FE ones. The B-splines constitute a partition of the unity, that is

$$
\sum_{i=1}^{n} N_{i,p}(x) = 1, \qquad \forall x \in [0, 1].
$$

A second feature is that they are pointwise non-negative, i.e. $N_{i,p}(x) \geq 0, \forall x \in [0, 1]$. This implies that the coefficients of the mass matrix are greater or equal than zero. The support of $N_{i,p}$ is $[\xi_i, \xi_{i+p+1}]$. The basis function $N_{i,p}$ has $p - m_i$ continuous derivatives, where m_i is the multiplicity of ξ_i, i.e. the number of repetitions of ξ_i. An important remark is that the B-spline basis functions are not interpolatory at the location of knot values ξ_i unless the multiplicity of ξ_i is exactly p.
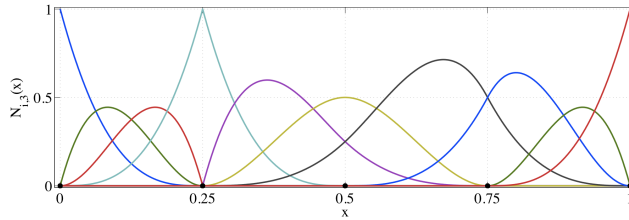

Fig. 1: Example of B-spline basis functions for $\xi = \{0, 0, 0, 0, 0.25, 0.25, 0.25, 0.5, 0.75, 0.75, 1, 1, 1, 1\}$, $n = 10$ and $p = 3$. We see that the regularity is related to the multiplicity of each ξ_i. Moreover, for $\xi_i = 0.25$ we have $m_i = p$ and we see that the basis function is interpolatory at this knot.

We are now in position to define B-spline curves, surfaces and solids in $\mathbb{R}^d$. Let us assume that we are given three sets of B-spline basis functions $\{N_{i,p}\}$, $\{M_{j,q}\}$ and $\{L_{k,r}\}$ constructed on the knot vectors $\{\xi_1, \ldots, \xi_{n+p+1}\}$, $\{\eta_1, \ldots, \eta_{m+q+1}\}$ and $\{\zeta_1, \ldots, \zeta_{l+r+1}\}$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$ and $k = 1, \ldots, l$, respectively.

The B-spline curves are obtained by considering linear combinations of B-spline basis functions. Let $C_i \in \mathbb{R}^d$ be the coefficients referred as *control points*, for $i = 1, \ldots, n$. We then define a B-spline curve as

$$
S(x) := \sum_{i=1}^{n} N_{i,p}(x) C_i.
$$

3

Such curves have at least as many continuous derivatives across an element boundary than its underlying B-spline basis function has across the corresponding knot value. A crucial property of the B-spline curves is that an affine transformation of the curve is obtained by applying the transformation to the control points. It is the so-called *affine covariance* and play an important role in the affine decomposition (2.3) when considering parameter-dependent domains. Now that the univariate B-splines have been introduced, we generalize the definition to higher dimensions by considering a tensor product structure.

Given a so-called *control net* $\{C_{i,j}\} \subset \mathbb{R}^d$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, we define a B-spline surface as

$$S(x, y) := \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(x) M_{j,q}(y) C_{i,j}.$$

Several properties of the B-spline surfaces result from their tensor product structures. For instance, the basis also forms a partition of the unity and the number of continuous partial derivatives are determined from the underlying one-dimensional knot vector and polynomial order. The local support is also deducted from the one-dimensional basis, that is the support of $N_{i,p}(x) M_{j,q}(y)$ is $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+q+1}]$.

Finally, we introduce the definition of a B-spline solid. Considering a control lattice $\{C_{i,j,k}\}$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$ and $k = 1, \ldots, l$, it is defined as

$$S(x, y, z) := \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} N_{i,p}(x) M_{j,q}(y) L_{k,r}(z) C_{i,j,k}.$$

The properties of the B-spline solids are a direct extension of those presented in the case of surfaces. In particular, the affine covariance property still holds for B-spline surfaces and solids. Note that what has been presented here is valid for a single patch. The case of multipatches geometries is introduced in the context of NURBS basis functions in the next section.

For the purpose of the analysis presented in Section 3.4, we briefly discuss the notions of h-refinement, p-refinement and k-refinement. A complete discussion can be found in [3, 8]. The notion of h-refinement in FE analysis is similar to the *knot insertion* in IGA. Let us consider a knot vector $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$ and associated control points $\{B_1, \ldots, B_n\}$. Considering a knot $\bar{\xi} \in [\xi_k, \xi_{k+1}[$, we then build the new knot vector as $\Xi = \{\xi_1, \ldots, \xi_k, \bar{\xi}, \xi_{k+1}, \ldots, \xi_{n+p+1}\}$ and the associated control points $\{\bar{B}_1, \ldots, \bar{B}_{n+1}\}$ as

$$\bar{B}_i := \alpha_i B_i + (1 - \alpha_i) B_{i-1}, \tag{3.2}$$

where

$$\alpha_i := \begin{cases} 1, & 1 \leq i \leq k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & k + 1 \leq i \leq n + p + 2. \end{cases} \tag{3.3}$$

By choosing the new control points as (3.2) and (3.3), it is possible to maintain the continuity of the original basis functions. Note that it is possible to insert repetition of already existing knot values. This will decrease the regularity of the basis functions at this knot. An important remark is that the solution spanned by the increased basis functions based contains the one spanned by the original B-splines. This allows to keep the geometry unchanged by inserting new knots.

The second concept introduced here is the FE p-refinement, which analogous is *order elevation*. It is possible to increase the polynomial order of the basis functions. To keep the regularity of the previous B-splines, it is necessary to repeat each knot value of the knot vector. As in the case of knot insertion, the new span contains the one from the original basis functions.

The last notion is the one of k-refinement which does not have an analogous in FE analysis. It is based on the principle that order elevation and knot insertion do not commute. If we insert a new knot value $\bar{\xi}$, the continuity of the basis functions at this knot will be C^{p-1}. If then we further increase the order of the basis, the multiplicity of $\bar{\xi}$ increase to keep this continuity. Instead, if we first increase the order of the basis

to q and then insert a new knot value, the continuity will be C^{q-1} at this knot. This second process is called k-refinement. It allows to control the number of new basis functions. Hence the number of degrees of freedom associated to the B-splines will also be kept under control, which in turn to keep low computational costs.

3.2 Non-Uniform Rational B-Splines

The introduction of NURBS allows us to exactly represent domains that it is not possible to describe considering polynomials. The construction of such geometries in $\mathbb{R}^d$ are obtained by projective transformations in $\mathbb{R}^{d+1}$. It is then possible to construct for instance conic sections. Such projective transformation yields rational polynomial functions.

The process to construct NURBS basis functions is presented here and follows mainly [3, 8]. Let us consider a knot vector Ξ, a number of basis functions n, a polynomial order p and a set of control points $\{B_i^w\}$ in $\mathbb{R}^{d+1}$ defining a B-spline curve. Such points are called *projective* control points for the associated NURBS curve. We then define the control points of the NURBS curve as follows

$$
\begin{aligned}
w_i &:= (B_i^w)_{d+1}, & i &= 1, \dots, d, \\
(B_i)_j &:= (B_i^w)_j / w_i, & i, j &= 1, \dots, d,
\end{aligned}
$$

where $(B_i)_j$ is the jth component of the vector B_i. The scalars w_i are called weights. Let $\{N_{i,p}\}$ be the B-spline basis functions associated to Ξ, n and p. Based on the definition of the control points, we can introduce the NURBS basis functions defined as

$$
R_i^p(x) := \frac{N_{i,p}(x)w_i}{\sum_{i'=1}^n N_{i',p}(x)w_j}. \tag{3.4}
$$

The associated NURBS curve is then defined as

$$
C(x) := \sum_{i=1}^n R_i^p(x)B_i.
$$

Considering the basis functions defined by (3.4), we define NURBS surfaces and solids in the same manner. To do this, we define rational basis functions for surfaces and solids. Let $\{M_{j,q}\}$ and $\{L_{k,r}\}$ be B-spline basis functions for $1 \leq j \leq m$ and $1 \leq k \leq l$. Moreover, consider projective control nets and lattices $\{B_{i,j}^w\}$ and $\{B_{i,j,k}^w\}$ in $\mathbb{R}^{d+1}$, respectively. The weights to construct the NURBS basis functions are given by

$$
\begin{aligned}
w_{i,j} &:= \left(B_{i,j}^w\right)_{d+1}, & i, j &= 1, \dots, d, \\
w_{i,j,k} &:= \left(B_{i,j,k}^w\right)_{d+1}, & i, j, k &= 1, \dots, d.
\end{aligned}
$$

We then define NURBS basis functions for surfaces and solids as

$$
\begin{aligned}
R_{i,j}^{p,q}(x,y) &:= \frac{N_{i,p}(x)M_{j,q}(y)w_{i,j}}{\sum_{i'=1}^n \sum_{j'=1}^m N_{i',p}(x)M_{j',q}(y)w_{i',j'}}, & i, j &= 1\dots, d, \\
R_{i,j,k}^{p,q,r}(x,y,z) &:= \frac{N_{i,p}(x)M_{j,q}(y)L_{k,r}(z)w_{i,j,k}}{\sum_{i'=1}^n \sum_{j'=1}^m \sum_{k'=1}^l N_{i',p}(x)M_{j',q}(y)L_{k',r}(z)w_{i',j',k'}}, & i, j, k &= 1\dots, d.
\end{aligned}
$$

The properties stated for the B-spline basis functions also hold for the NURBS. In particular, they form a partition of the unity and their continuity and support are the same as the underlying B-splines. The affine covariance property also holds for NURBS functions. Moreover, the basis functions are interpolatory at knot values where the multiplicity is equal to the order. The notions of h-, p- and k-refinement generalize to NURBS functions. Note that if all the weights are equal, the NURBS coincide with the underlying B-splines due to the partition of the unity property. In nearly all the practical applications, it is necessary to have

multiple patches to describe the domain with NURBS functions. This also allows to have different material properties, each associated to a different patch. The only feature to pay attention to is the regularity of the basis across the patches interfaces. Usually, C^0 is the only regularity guaranteed, but techniques can be used to increase it [3]. In Figure 2, we present several examples of NURBS solids obtained considering multipatches representations.

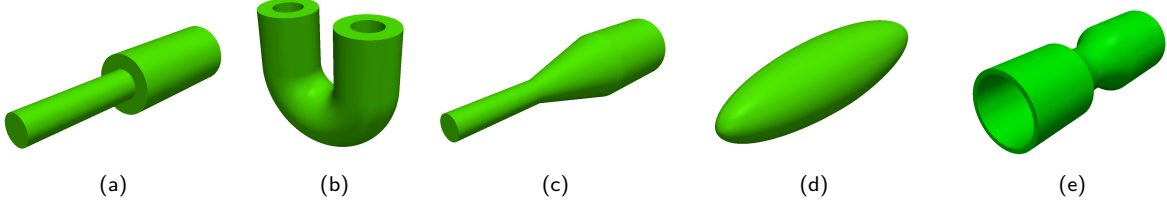<div align="center">(a)   (b)   (c)   (d)   (e)</div>

Fig. 2: Examples of NURBS solids obtained considering multiple patches. The number of patches used for each example are 3 (a), 3 (e), 3 (c), 4 (d) and 4 (b), respectively.

To simplify the notations, we denote by $R_{i,p}$, $1 \leq i \leq n$ the NURBS basis functions and $\{B_i\}$ the associated control points for curves, surfaces and solids. We also use the notation Ξ for the associated knot vectors. Note that it is a slight abuse of notation because in the case of surfaces and solids, p and Ξ are vectors and matrices, respectively.

For the purpose of our analysis, we require that the computational domain Ω can be obtained through a NURBS parametrization. To introduce the notations, we impose that Ω is parameter-independent. The parameter-dependent case is treated in the next section. Let us consider the following decomposition of the domain

$$\overline{\Omega} = \bigcup_{k=1}^{P_{\mathrm{dom}}} \overline{\Omega}^k, \tag{3.5}$$

where $\Omega^j \cap \Omega^k = \emptyset$ for $1 \leq j < k \leq P_{\mathrm{dom}}$. We require that for every subdomain Ω^k, there exist p_k, n_k, Ξ_k, NURBS basis functions $\{R_{i,p}^k\}$ and associated control points $\mathcal{B}^k := \{B_i^k\}$ such that for every $y \in \Omega^k$, there exists $x \in \mathcal{H}^d$ satisfying

$$y = F^k(x) := \sum_{i=1}^{n_k} R_{i,p_k}^k(x) B_i^k. \tag{3.6}$$

Here $\mathcal{H}^d = [0,1]^d$ denotes the unit hypercube in d-dimension and $F^k : (0,1)^d \to \Omega^k$. Considering for every $1 \leq k \leq P_{\mathrm{dom}}$ the function F^k defined above, we construct a global mapping $F : (0,1)^d \to \Omega$ which describes the whole computational domain. We assume that F is smooth and invertible. In that case, we say that Ω admits a NURBS representation through F. So far, we have only considered parameter-independent geometries. In the next section, we introduce the affine preconditionning conditions for parameter-dependent domains.

3.3   Affine Preconditionning for Parameter-Dependent Domains

In many applications, it is of great interest to consider parameter-dependent geometries. We introduce here the conditions that need to be fulfilled in that case to be able to perform the RB method presented in this paper. In particular, it is important that the affine decomposition (2.3) of the bilinear form a still holds. Let us consider the domain splitting introduced in (3.5). The computational domain for an input parameter $\boldsymbol{\mu} \in \mathcal{D}$ is denoted $\Omega_o(\boldsymbol{\mu})$. Here, the subscript o stands for the original domain.

The domain $\Omega_o(\boldsymbol{\mu})$ needs to be represented as the image of a reference domain through an affine mapping. Let us choose $\boldsymbol{\mu}_{\mathrm{ref}} \in \mathcal{D}$ as a parameter that represents our reference domain, i.e. $\Omega = \Omega_o(\boldsymbol{\mu}_{\mathrm{ref}})$. Moreover,

we denote $\Omega^k = \Omega_o^k(\boldsymbol{\mu}_{\mathrm{ref}})$ while considering the decomposition (3.5). We need that for every $1 \leq k \leq P_{\mathrm{dom}}$, there exists an affine mapping $\mathcal{T}^k(\cdot; \boldsymbol{\mu}) : \Omega^k \to \Omega_o^k(\boldsymbol{\mu})$ such that

$$\overline{\Omega}_o^k(\boldsymbol{\mu}) = \mathcal{T}^k(\overline{\Omega}^k; \boldsymbol{\mu}).$$

The mappings $\mathcal{T}^k(\cdot; \boldsymbol{\mu})$ have to be bijective and collectively continuous, that is

$$\mathcal{T}^k(x; \boldsymbol{\mu}) = \mathcal{T}^l(x; \boldsymbol{\mu}), \qquad \forall x \in \overline{\Omega}^k \cap \overline{\Omega}^l, \ 1 \leq k < l \leq P_{\mathrm{dom}}. \tag{3.7}$$

Due to the affine covariance property of the NURBS functions, we only need to require that the control points can be obtained as the image of reference control points through an affine mapping. More formally, let us denote by $\{B_i^k(\boldsymbol{\mu})\}$ the control points associated with the subdomains $\Omega_o^k(\boldsymbol{\mu})$. We then require that for every $1 \leq k \leq P_{\mathrm{dom}}$, there exists an affine mapping $\mathcal{T}^k(\cdot; \boldsymbol{\mu}) : \Omega^k \to \Omega_o^k(\boldsymbol{\mu})$ such that

$$B_i^k(\boldsymbol{\mu}) = \mathcal{T}^k(B_i^k; \boldsymbol{\mu}),$$

where $\{B_i^k\}$ are the control points associated to the reference subdomains Ω^k. Turning to the condition (3.7), we require that

$$\mathcal{T}^k(B_i^k; \boldsymbol{\mu}) = \mathcal{T}^l(B_i^k; \boldsymbol{\mu}), \qquad \forall B_i^k \in \mathcal{B}^k \cap \mathcal{B}^l, \ 1 \leq k < l \leq P_{\mathrm{dom}}.$$

In other words, we only need to ensure continuity of the mappings through the control points defining the interfaces of patches to obtain the continuity on the whole interface. More explicitly, we define the affine mappings $\mathcal{T}^k$ for every $x \in \overline{\Omega}^k$ and $\boldsymbol{\mu} \in \mathcal{D}$ as

$$\mathcal{T}^k(x; \boldsymbol{\mu}) := C^k(\boldsymbol{\mu}) + G^k(\boldsymbol{\mu})x,$$

where $C^k : \mathcal{D} \to \mathbb{R}^d$ and $G^k : \mathcal{D} \to \mathbb{R}^{d \times d}$ for every $1 \leq k \leq P_{\mathrm{dom}}$. To define the affine decomposition of the bilinear form a, we need to define the Jacobians and inverse of the transformations as

$$J^k(\boldsymbol{\mu}) \quad := \quad |\det(G^k(\boldsymbol{\mu}))|, \tag{3.8}$$

$$D^k(\boldsymbol{\mu}) \quad := \quad \left(G^k(\boldsymbol{\mu})\right)^{-1}, \tag{3.9}$$

for $1 \leq k \leq P_{\mathrm{dom}}$. Based on the $\mathcal{T}^k$ transformations, we can define a global affine mapping $\mathcal{T} : \Omega \to \Omega_o(\boldsymbol{\mu})$ as

$$\mathcal{T}(x; \boldsymbol{\mu}) := \mathcal{T}^k(x; \boldsymbol{\mu}), \qquad k = \min\left\{ 1 \leq l \leq P_{\mathrm{dom}} \, \middle| \, x \in \overline{\Omega}^l \right\}.$$

The mapping $\mathcal{T}$ is globally bijective and piecewise affine. The choice of the minimum is arbitrary and could be chosen differently.

In what follows, we give an example of an affine transformation applied to a 3-dimensional toroidal solid. It is built on four patches, which yields $P_{\mathrm{dom}} = 4$, and to every patch are associated 27 control points. Based on that, it is possible to uniquely determine C^k and G^k for $1 \leq k \leq P_{\mathrm{dom}}$. In that case, the transformations are given by

$$C^k = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad G^k = \begin{pmatrix} \mu & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad 1 \leq k \leq 4, \tag{3.10}$$

where we have considered the single parameter μ that controls the semi-axis x. In Figure 3, the original domain and the transformed one for $\mu = 1.5$ are depicted together with their lattices of control points. We see that the transformation is exactly applied to the control points.

Now that our affine preconditionning assumption has been stated, we need to express our bilinear form on the reference domain. The problem (2.1)-(2.2) is defined on the original domain $\Omega_o(\boldsymbol{\mu})$. To be able to obtain the affine expansion (2.3) for the bilinear form a arising from the weak formulation of a second-order
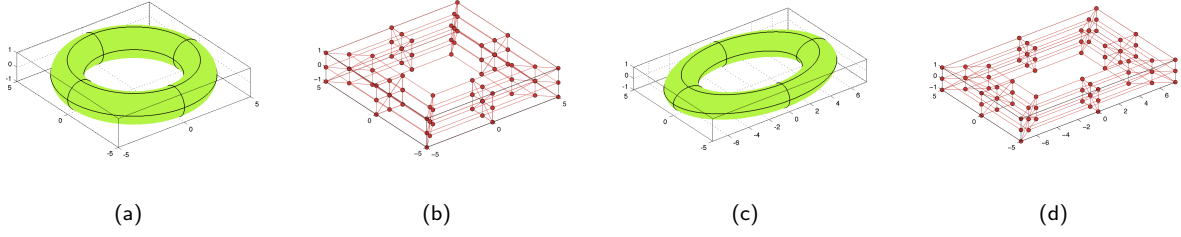
Fig. 3: Example of the affine transformation (3.10) applied to a torus. The original domain and its lattice of control points are presented in (a) and (b), respectively. The torus after transformation is depicted in (c) while its lattice is presented in (d).

PDE, we need that the underlying integrals are defined on the reference domain. This is presented in details in what follows for two-dimensional problems but the case $d = 3$ is treated analogously. On the original domain, the problem is the following one: given a parameter $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$s_o(\boldsymbol{\mu}) = l(u_o(\boldsymbol{\mu})),$$

where $u_o(\boldsymbol{\mu}) \in X_o$ is the solution of

$$a_o(u_o(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_o(v), \qquad \forall v \in X.$$

Since we are considering second-order partial differential, we require that a_o can be written as

$$a_o(v, w; \boldsymbol{\mu}) = \sum_{k=1}^{P_{\mathrm{dom}}} \int_{\Omega^k(\boldsymbol{\mu})} \left[\begin{array}{ccc} \frac{\partial v}{\partial x_1} & \frac{\partial v}{\partial x_2} & v \end{array} \right] A_o^k(\boldsymbol{\mu}) \left[\begin{array}{c} \frac{\partial w}{\partial x_1} \\ \frac{\partial w}{\partial x_2} \\ v \end{array} \right],$$

where $A_o^k : \mathcal{D} \to \mathbb{R}^{3 \times 3}$ is a symmetric positive semi-definite matrix. We express the right-hand side f_o in the same way, that is

$$f_o(v) = \sum_{k=1}^{P_{\mathrm{dom}}} \int_{\Omega^k(\boldsymbol{\mu})} f_o^k v, \tag{3.11}$$

where $f_o^k \in \mathbb{R}$. Note that it is possible to have a parameter-dependent right-hand side by simply replacing f_o^k by $f_o^k(\boldsymbol{\mu})$. As already discussed, we need $\boldsymbol{\mu}$-independent integrals to be able to fulfill the affine assumption (2.3) for the bilinear form a. We then consider the problem (2.1)-(2.2) with the bilinear form a expressed as

$$a(v, w; \boldsymbol{\mu}) = \sum_{k=1}^{P_{\mathrm{dom}}} \int_{\Omega^k} \left[\begin{array}{ccc} \frac{\partial v}{\partial x_1} & \frac{\partial v}{\partial x_2} & v \end{array} \right] A^k(\boldsymbol{\mu}) \left[\begin{array}{c} \frac{\partial w}{\partial x_1} \\ \frac{\partial w}{\partial x_2} \\ v \end{array} \right], \tag{3.12}$$

where the $A^k : \mathcal{D} \to \mathbb{R}^{3 \times 3}$ are defined for $\boldsymbol{\mu} \in \mathcal{D}$ as

$$A^k(\boldsymbol{\mu}) = J^k(\boldsymbol{\mu}) \mathcal{G}^k(\boldsymbol{\mu}) A_o^k(\boldsymbol{\mu}) \left(\mathcal{G}^k(\boldsymbol{\mu}) \right)^T, \qquad 1 \le k \le P_{\mathrm{dom}}.$$

Here the matrices $\mathcal{G}^k(\boldsymbol{\mu})$ are defined as

$$\mathcal{G}^k(\boldsymbol{\mu}) := \left(\begin{array}{ccc} D^k(\boldsymbol{\mu}) & & 0 \\ & & 0 \\ 0 & 0 & 1 \end{array} \right), \qquad 1 \le k \le P_{\mathrm{dom}},$$

8

where $J^k(\boldsymbol{\mu})$ and $D^k(\boldsymbol{\mu})$ are defined by (3.8) and (3.9), respectively. Note that this holds under the assumptions presented at the begin of this section. In the same manner, the right-hand side is expressed as

$$f(v) = \sum_{k=1}^{P_{\text{dom}}} \int_{\Omega^k} f^k(\boldsymbol{\mu})v,$$

where $f^k : \mathcal{D} \to \mathbb{R}$ is defined by

$$f^k(\boldsymbol{\mu}) = J^k(\boldsymbol{\mu})f_o^k, \qquad 1 \le k \le P_{\text{dom}}.$$

We can then explicitly expand (3.12) to obtain the affine decomposition (2.3) for the bilinear form a. In the development presented here, the $A^k(\boldsymbol{\mu})$ and $f^k(\boldsymbol{\mu})$ are local to patches and may represent different material properties and geometry variations.

3.4 Isogeometric Analysis NURBS Approximation of Elliptic Coercive Parametrized PDEs

We present in this section the isogeometrical analysis NURBS approximation of the problem (2.1)-(2.2). In this context, the isoparametric concept is considered, that is the solution is represented in the same space as the geometry. In that case, the mesh of the NURBS is defined as the product of the knot vectors and the elements are the knot spans. The degrees of freedom associated with the basis functions are called *control variables*.

Let us assume that Ω admits a NURBS parametrization through F as defined in (3.6). To simplify the notations, we consider a single set of indices $\{1, \ldots, \mathcal{N}\}$ for the degrees of freedom and we write

$$F(x) = \sum_{i=1}^{\mathcal{N}} \tilde{R}_{i,p}(x)B_i, \qquad x \in (0,1)^d, \tag{3.13}$$

for NURBS basis functions $\{\tilde{R}_{i,p}\}$ and associated control points $\{B_i\}$. To represent our solution in a finite dimensional space, we need to define the basis functions

$$R_{i,p} := \tilde{R}_{i,p} \circ F^{-1}, \tag{3.14}$$

where F is the invertible mapping defined by (3.13). Based on that representation, we construct the NURBS approximation space

$$X^{\mathcal{N}} := \text{span}\,\{R_{i,p}\}_{1 \le i \le \mathcal{N}} \subset X. \tag{3.15}$$

As already discussed in Section 3.1, the process of knot insertion does not change the underlying geometry. In that setting, increasing $\mathcal{N}$ does not change the shape of the parametrized domain and so we keep the exact parametrization while refining the mesh. For approximation properties of NURBS approximation spaces, we refer the reader to [1]. We approximate the solution of (2.1)-(2.2) by an element of $X^{\mathcal{N}}$. The approximate problem is the following one: given a parameter $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$s^{\mathcal{N}}(\boldsymbol{\mu}) = l(u^{\mathcal{N}}(\boldsymbol{\mu})), \tag{3.16}$$

where $u^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ is the solution of

$$a(u^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in X^{\mathcal{N}}. \tag{3.17}$$

Considering the basis $\{R_{i,p}\}$ for $X^{\mathcal{N}}$ defined by (3.14), we extend the NURBS solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \mathcal{D}$ as

$$u^{\mathcal{N}}(x, \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} u_i^{\mathcal{N}}(\boldsymbol{\mu})R_{i,p}(x), \qquad x \in \Omega,$$

9

where the coefficients $u_i^{\mathcal{N}}(\boldsymbol{\mu})$ are called control variables. The regularity of $u^{\mathcal{N}}(\boldsymbol{\mu})$ follows from that of the NURBS basis. For instance, the continuity of the solution across element boundaries depends on the continuity of the underlying basis functions across the associated knot span.

Our goal then becomes to solve the problem (3.16)-(3.17) with high precision. However, for real-time context and many query problems, it would be computationally unaffordable to approximate the solution for each input parameter. For that reason, we introduce in the next section a method to approximate such solution with reduced computation costs.

4 Reduced Basis Method for Isogeometric Analysis NURBS Approximation

As it has already been pointed out, it is computationally unaffordable to compute a new NURBS solution for every input parameter $\boldsymbol{\mu}$. The goal of the RB method is then to approximate the NURBS solution $u^{\mathcal{N}}(\boldsymbol{\mu})$ with reduced computational costs. Considering $\mathcal{N}$ sufficiently large, we have that $u^{\mathcal{N}}(\boldsymbol{\mu})$ is close enough to $u(\boldsymbol{\mu})$ in a certain norm so that the NURBS approximation can be viewed as the "truth" solution.

Given a positive integer $N_{\max} \ll \mathcal{N}$, we construct a sequence of approximation spaces

$$X_1^{\mathcal{N}} \subset X_2^{\mathcal{N}} \subset \cdots \subset X_{N_{\max}}^{\mathcal{N}} \subset X^{\mathcal{N}}. \tag{4.1}$$

Those spaces are obtained considering a Greedy algorithm presented more in details in Section 4.1. The hierarchical hypothesis (4.1) is important to ensure the efficiency of the method. Several spaces can be considered to construct such sequence, but they all focus on the smooth parametric manifold $\mathcal{M}^{\mathcal{N}} := \left\{ u^{\mathcal{N}}(\boldsymbol{\mu}) \middle| \ \boldsymbol{\mu} \in \mathcal{D} \right\}$. If it is smooth enough, we can expect it to be well approximated by low-dimensional spaces. In what follows, we consider the special case of Lagrange reduced basis spaces built using a master set of parameter points $\boldsymbol{\mu}^n \in \mathcal{D}$, $1 \leq n \leq N_{\max}$. Other examples such as the POD spaces [16] could be considered. For $1 \leq N \leq N_{\max}$, we define $S^N := \left\{ \boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N \right\}$ and the associated Lagrange RB spaces

$$X_N^{\mathcal{N}} := \mathrm{span} \left\{ u^{\mathcal{N}}(\boldsymbol{\mu}^n) \middle| \ 1 \leq n \leq N \right\}.$$

The selection of the snapshots $u^{\mathcal{N}}(\boldsymbol{\mu}^n)$ is one of the crucial points of the RB method and is further investigated in the next section. We apply the Gram-Schmidt process in the $(\cdot, \cdot)_X$ inner product to the snapshots $u^{\mathcal{N}}(\boldsymbol{\mu}^n)$ in order to obtain mutually orthonormal functions $\zeta_n^{\mathcal{N}}$. In that case, we have $X_N^{\mathcal{N}} = \mathrm{span} \left\{ \zeta_n^{\mathcal{N}} \middle| \ 1 \leq n \leq N \right\}$. Since colinearities are avoided using the Gram-Schmidt process, we are ensured that the N obtained is minimal. The RB approximation of the problem (3.16)-(3.17) is obtained considering Galerkin projection: given $\boldsymbol{\mu} \in \mathcal{D}$, evaluate

$$s_N^{\mathcal{N}}(\boldsymbol{\mu}) = f(u_N^{\mathcal{N}}(\boldsymbol{\mu})), \tag{4.2}$$

where $u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}}$ is the solution of

$$a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v), \qquad \forall v \in X_N^{\mathcal{N}}. \tag{4.3}$$

Since the particular compliant case is considered, we obtain

$$s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu}) = \|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2, \tag{4.4}$$

where $\|\cdot\|_{\boldsymbol{\mu}}$ is the energy norm induced by the inner product $a(\cdot, \cdot; \boldsymbol{\mu})$. In Section 4.2, we present an example of an inexpensive and efficient a posteriori error estimator $\Delta_N(\boldsymbol{\mu})$ for $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}$ on which the Greedy algorithm is based. Due to the relation (4.4), it is possible to ensure that the error arising from the RB approximation on our output of interest is bounded by a prescribed tolerance.

Since $u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X_N^{\mathcal{N}} = \mathrm{span} \left\{ \zeta_n^{\mathcal{N}} \middle| \ 1 \leq n \leq N \right\}$, we expand it as

$$u_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_{N,m}^{\mathcal{N}}(\boldsymbol{\mu}) \zeta_m^{\mathcal{N}}. \tag{4.5}$$

The unknowns then become the coefficients $u_{N,m}^{\mathcal{N}}(\boldsymbol{\mu})$. Inserting (4.5) in (4.2) and (4.3) and using the hypothesis that f is linear and a bilinear, we obtain

$$s_N^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{m=1}^{N} u_{N,m}^{\mathcal{N}}(\boldsymbol{\mu}) f(\zeta_m^{\mathcal{N}}),$$

and

$$\sum_{m=1}^{N} u_{N,m}^{\mathcal{N}}(\boldsymbol{\mu}) a(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) = f(\zeta_n^{\mathcal{N}}), \qquad 1 \le n \le N. \tag{4.6}$$

The stiffness matrix associated to the system (4.6) is of size $N \times N$ with $N \le N_{\max} \ll \mathcal{N}$. It yields a considerably smaller computational effort than to solve the system associated to (3.17), which matrix is of size $\mathcal{N} \times \mathcal{N}$. However, the formation of the stiffness matrix involves the computation of the $\zeta_m^{\mathcal{N}}$ associated with the $\mathcal{N}$-dimensional NURBS space.

This drawback is avoided by constructing an Offline-Online procedure taking advantage of the affine decomposition (2.3). In the Offline stage, the Greedy algorithm is used to construct the set of parameters S^N. Then, the $u^{\mathcal{N}}(\boldsymbol{\mu}^n)$ and the $\zeta_n^{\mathcal{N}}$ are built for $1 \le n \le N$. The $f(\zeta_n^{\mathcal{N}})$ and $a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}})$ are also formed and stored. Note the importance here of the affine decomposition. It implies that the vector and matrices stored are independent of the input parameter $\boldsymbol{\mu}$.

In the Online part, the stiffness matrix associated to (4.6) is assembled considering the affine decomposition (2.3). This yields

$$a(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(\zeta_m^{\mathcal{N}}, \zeta_n^{\mathcal{N}}), \qquad 1 \le m, n \le N.$$

The same process is applied to the right-hand side f. The $N \times N$ system (4.3) is then solved to obtain $u_{N,m}^{\mathcal{N}}(\boldsymbol{\mu})$, $1 \le m \le N$. Finally, the output of interest (4.2) is computed considering the coefficients obtained.

As already discussed, one of the main feature of the RB method is that we have a posteriori error estimators $\Delta_N(\boldsymbol{\mu})$ for $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2 = s^{\mathcal{N}}(\boldsymbol{\mu}) - s_N^{\mathcal{N}}(\boldsymbol{\mu})$ whose computation costs are independent of $\mathcal{N}$. It allows us to certify our method and make it reliable. A discussion on such estimators is presented in Section 4.2.

4.1 Greedy Algorithm for the Snapshots Selection

One of the most important step taking place in the Offline stage is the selection of the parameters $\boldsymbol{\mu}^n$, $1 \le n \le N$. Several algorithms are available in the literature [16] but we introduce here a greedy procedure for completeness. The general idea of this procedure is to retain at iteration N the snapshot $u^{\mathcal{N}}(\boldsymbol{\mu}^N)$ which approximation by $X_{N-1}^{\mathcal{N}}$ is the worst. Let us assume that we are given a finite sample of points $\Xi \subset \mathcal{D}$ and pick randomly a first parameter $\boldsymbol{\mu}^1 \in \Xi$. Then for $N = 2, \ldots, N_{\max}$, compute

$$\boldsymbol{\mu}^N := \arg \max_{\boldsymbol{\mu} \in \Xi} \Delta_{N-1}(\boldsymbol{\mu}),$$

where $\Delta_N(\boldsymbol{\mu})$ is a sharp and inexpensive a posteriori error estimator for $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\|_{H_0^1(\Omega)}$ or $\|u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}$. The algorithm is typically stopped when $\Delta_N(\boldsymbol{\mu})$ is smaller than a prescribed tolerance for every $\boldsymbol{\mu} \in \Xi$. It is clear that the precision of the approximation spaces obtain increase with the size of the sample considered.

Since $X_{N-1}^{\mathcal{N}} \subset X_N^{\mathcal{N}}$, we expect to have $\Delta_N(\boldsymbol{\mu}) \le \Delta_{N-1}(\boldsymbol{\mu})$, which ensures that $N_{\max} < \infty$. Even if this procedure has not been proven to convergence, it is widely used and many examples have been presented to illustrate its convergence. The derivation of $\Delta_N(\boldsymbol{\mu})$ is crucial for the Greedy and we introduce an example of such estimator in the next section.

4.2 A posteriori error estimators for elliptic coercive partial differential equations

The main ingredient of the Greedy algorithm procedure is the computation of the error estimator, which has to be independent of $\mathcal{N}$. In fact, it is used online to certify that the error of our RB approximation with respect to the truth solution is under control. For completeness, the derivation of such estimator is presented here when the so-called compliant case is considered, i.e. a is symmetric and $f = l$. See e.g. [16] for the non-compliant case. Let us introduce the error $e^{\mathcal{N}}(\boldsymbol{\mu}) = u^{\mathcal{N}}(\boldsymbol{\mu}) - u_N^{\mathcal{N}}(\boldsymbol{\mu}) \in X^{\mathcal{N}}$ which satisfies the following equation

$$a(e^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) - a(u_N^{\mathcal{N}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}) =: r(v; \boldsymbol{\mu}), \ \forall v \in X^{\mathcal{N}} \tag{4.7}$$

where $r(\cdot; \boldsymbol{\mu}) \in (X^{\mathcal{N}})'$ is the residual and $(X^{\mathcal{N}})'$ denotes the dual space of $X^{\mathcal{N}}$. To define our a posteriori error estimator, we need to have a lower bound $\alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu})$ of $\alpha^{\mathcal{N}}(\boldsymbol{\mu})$ such that $0 < \alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu}) \leq \alpha^{\mathcal{N}}(\boldsymbol{\mu}) \ \forall \boldsymbol{\mu} \in \mathcal{D}$ and the online costs to compute $\alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu})$ are independent of $\mathcal{N}$. We then define the following a posteriori error estimator

$$\Delta_N(\boldsymbol{\mu}) := \frac{\|r(\cdot; \boldsymbol{\mu})\|_{(X^{\mathcal{N}})'}}{\alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu})}.$$

To compute $\|r(\cdot; \boldsymbol{\mu})\|_{(X^{\mathcal{N}})'}$, the main ingredients are to use the affine assumption (2.3) on a and the expansion (4.5) of $u_N^{\mathcal{N}}(\boldsymbol{\mu})$ in the space $X_N^{\mathcal{N}}$. Then, using the definition (4.7) of the residual, this leads to a system depending only on N for every $\boldsymbol{\mu}$, which makes the computation independent of $\mathcal{N}$.

The procedure used to compute the coercivity lower bound $\alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu})$ is the so-called successive constraint method (SCM) [10]. Considering sets based on parameter samples and the terms Θ_q^a of the affine decomposition (2.3), it is possible to reduce this problem to a linear optimization problem. This method works by taking into account neighbour informations for the parameters and its precision increases with the size of the neighbourhood considered. The SCM also creates a coercivity upper bound $\alpha_{\mathrm{UB}}^{\mathcal{N}}(\boldsymbol{\mu})$ of $\alpha^{\mathcal{N}}(\boldsymbol{\mu})$ in the same manner. The algorithm is stopped when $\max_{\boldsymbol{\mu} \in \Xi} \left(\alpha_{\mathrm{UB}}^{\mathcal{N}}(\boldsymbol{\mu}) - \alpha_{\mathrm{LB}}^{\mathcal{N}}(\boldsymbol{\mu})/\alpha_{\mathrm{UB}}^{\mathcal{N}}(\boldsymbol{\mu}) \right)$ is smaller than a prescribed tolerance ε.

We emphasize on the fact that it is very important that the costs associated to the computation of Δ_N are independent of $\mathcal{N}$. That allows us to develop the Offline-Online procedure discussed in Section 4, which is a crucial ingredient for the reduced basis method.

5 Numerical Illustrations

In this section, we present several numerical illustrations of the method introduced in this paper. The first example considered is a case of heat conduction involving only physical parameters, i.e. we use different conductivity coefficients in regions of the domain. Then a case containing geometrical parameters is introduced. The aim of the first two illustrations is to present the possibilities that are allowed while using NURBS basis functions. For this reason, both are computed over curvy three dimensional domains. In particular, the second example illustrates the theory developed in Section 3.3 to treat parameter dependent geometries. All computations have been performed using the `Matlab` [11] packages `GeoPDEs` [6] and `rbMIT` [9] for the NURBS and RB approximations, respectively.

Our goal in this section is to present standard examples to show that the method under consideration yields indeed good results. For this reason, all cases involve simple elliptic equations of the form

$$\begin{aligned} -\nabla \left(a(\boldsymbol{\mu}, x) \nabla u \right) &= f, &&\text{in } \Omega(\boldsymbol{\mu}), \\ u &= g, &&\text{on } \Gamma_D(\boldsymbol{\mu}), \\ a(\boldsymbol{\mu}, x) \frac{\partial u}{\partial n} &= h, &&\text{on } \Gamma_N(\boldsymbol{\mu}), \end{aligned} \tag{5.1}$$

with $\Gamma_N \cap \Gamma_D = \emptyset$ and $\partial\Omega = \Gamma_N \cup \Gamma_D$. In particular, note that f, g and h are parameter independent. Moreover, we only deal with piecewise constant conductivity coefficients $a(\boldsymbol{\mu}, x)$. For all examples, the prescribed tolerance for the greedy algorithm is 10^{-6}.

5.1   Physical Parameters for Heat Conduction in a Pipeline

In this first example, we consider heat conduction in a pipeline. The domain under consideration is depicted in Figure 4. It is built on 5 different patches, one for every straight part and one for each of the curvy ones. The domain is parameter independent and we consider three parameters $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in [1,5]^3$, each one being the conductivity coefficient in one of the straight portion. More precisely,

$$a(\boldsymbol{\mu}, x) := \mu_1 \chi_{\Omega_1} + \chi_{\Omega_2} + \mu_2 \chi_{\Omega_3} + \chi_{\Omega_4} + \mu_3 \chi_{\Omega_5},$$

where χ_{Ω_i} is the characteristic function over the ith patch, $1 \leq i \leq 5$. Let us denote the input boundary by Γ_{in}, the output by Γ_{out} and the inner and outer circular ones by Γ_{curve}. The functions are given by $f = 0$, $g = 0$ and $h = \chi_{\Gamma_{\text{out}}}$ and the associated boundaries are given by $\Gamma_D := \Gamma_{\text{in}}$ and $\Gamma_N := \Gamma_{\text{curve}} \cup \Gamma_{\text{out}}$. This simulation can be interpreted as heat conduction in a metal pipe where different metals constitute the structure and an imposed temperature is considered on one of the flat faces.
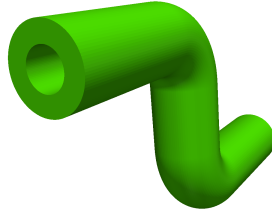

Fig. 4: Computation domain for the pipeline test case. Five patches were necessary to build the structure, one for each straight part and one for each angle.

The number of degrees of freedom for the NURBS approximation is $\mathcal{N} = 16650$ while the size of the RB space is 17, which yields a big reduction of the computational costs. The computation time to perform the offline step is 27 minutes and the average evaluation time for the RB approximation is $5 \cdot 10^{-4}$ seconds. We present in Figure 5 the convergence of the greedy algorithm.
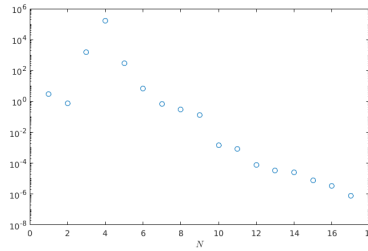

Fig. 5: Convergence of the greedy algorithm (see Section 4.1) for the pipeline test case of Section 5.1.

Finally, in figure 6, we show the solution on the whole domain for different values of the parameters. Note that the solution is not completely smooth at the interfaces of the patches. This comes from the fact that we have C^1 continuity in each of the patch while we only have C^0 continuity at the interfaces. Methods exist to obtain more regularity at the interfaces (see e.g. [3]).

5.2   Geometrical Parameters for Heat Conduction in a Cylinder

We present here the case of a parameter dependent geometry. The reference domain is a cylinder of radius 2 and height 1 oriented in the z direction, i.e. $\Omega(\boldsymbol{\mu}_{\text{ref}}) := \{(r, \theta, z) \mid r \in [0,2], \theta \in [0,2\pi], z \in [0,1]\}$ where

13

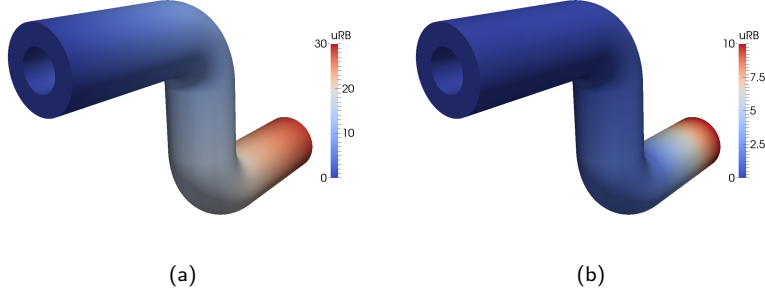(a)   (b)

Fig. 6: Reduced basis approximation of the pipeline test case for the physical parameters (a) $\boldsymbol{\mu} = (1, 1, 1)$ and (b) $\boldsymbol{\mu} = (3, 2, 5)$. Note that the scale is not the same in both cases. The first case gives rise to a perfect linear approximation, which is the expected behavior. The second one displays a lack of smoothness at the interfaces of the patches. This is due to the fact that we have C^1 continuity inside each patch while only continuity is guaranteed at the interfaces.

(r, θ, z) denote the cylindrical coordinates in $\mathbb{R}^3$. The reference cylinder is depicted in Figure 7a. To build it, four patches were necessary.

The transformations under consideration are scaling with respect to the y and z axis. More precisely, three parameters $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3) \in [1, 5]^3$ are considered, where μ_1 scales the portion of the domain for which $y > 0$, μ_2 the one where $y < 0$, and μ_3 scales in the z direction. In other words, the transformation in the part of the domain where $y > 0$ is given by

$$
C^k = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad G^k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix},
$$

while in the region $y < 0$ it reads

$$
C^k = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \qquad G^k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix}.
$$

In Figure 7, we present the domain after application of the affine transformation for different values of the parameters.

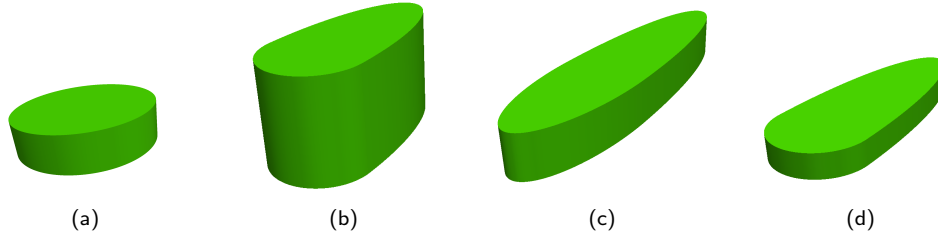(a)   (b)   (c)   (d)

Fig. 7: Computational domain for the cylinder test case for different values of the parameters. The original domain is presented in (a). Four patches were necessary to build the structure, one for each quarter of the cylinder. The affine transformation from Section 5.2 were considered for (b) $\boldsymbol{\mu} = (1, 3, 4)$, (c) $\boldsymbol{\mu} = (3, 5, 2)$ and (d) $\boldsymbol{\mu} = (1, 4, 1)$.

14

Considering the conductivity coefficient in (5.1), we have $a(\boldsymbol{\mu}, x) = 1$. In order to describe the boundary conditions, let us denote by Γ_{bot}, Γ_{top} and Γ_{curve} the bottom, top and curvy boundaries, respectively. We impose homogeneous Dirichlet boundary conditions on $\Gamma_D := \Gamma_{\text{curve}}$ and unitary Neumann conditions on $\Gamma_N := \Gamma_{\text{top}} \cup \Gamma_{\text{bot}}$. Finally the right-hand side function is $f = 10\chi_B$, where B is the ball of radius 0.2 centered at $(0, 0, 0.5)$.

Turning to the computational costs, the number of degrees of freedom for the NURBS approximation is $\mathcal{N} = 3240$ and the one of the RB is $N = 57$. The whole offline procedure took 30 minutes while the average RB evaluation takes $5 \cdot 10^{-4}$ seconds. In Figure 8, we show the convergence of the greedy algorithm for this case.
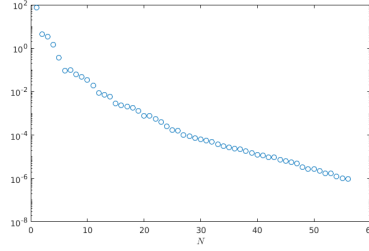

Fig. 8: Convergence of the greedy algorithm (see Section 4.1) for the cylinder test case of Section 5.2.

The solution for several values is depicted in Figure 9. We present the RB approximation on the whole domain as well as its evaluation on the plane $\{(x, y, z) \in \Omega \mid y = 0\}$.
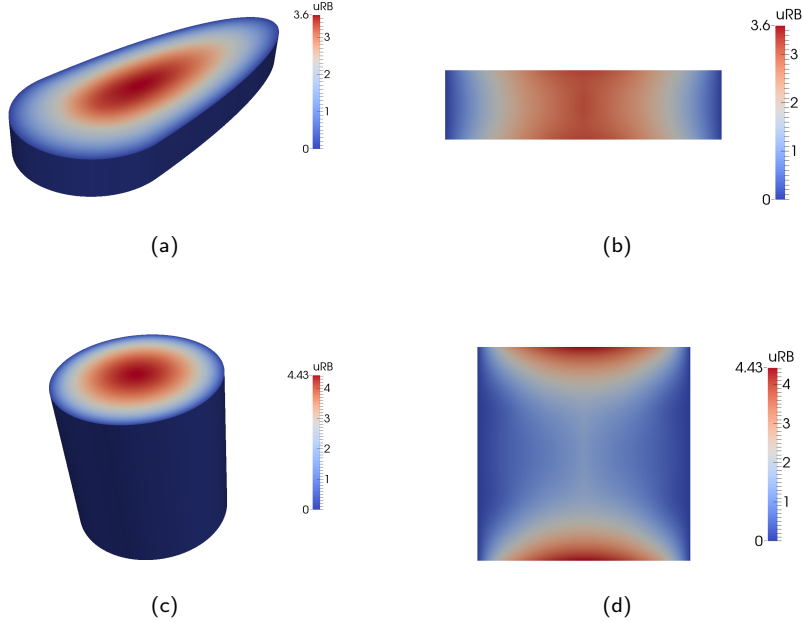

Fig. 9: Reduced basis approximation of the cylinder test case for (a) $\boldsymbol{\mu} = (1, 4, 1)$ and (c) $\boldsymbol{\mu} = (1, 1, 4)$. The pictures (b) and (d) represent the value of the field on the plane $\{(x, y, z) \in \Omega \mid y = 0\}$ for the values considered in (a) and (c), respectively. Note that different scales have been used for the different values of the parameters.

References

[1] Y. Bazilevs, L. Beirao da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[2] F. Chinesta, A. Huerta, G. Rozza, and K. Willcox. Model Order Reduction. Encyclopedia of Computational Mechanics 2016, Elsevier.

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

[4] M.G. Cox. The numerical evaluation of b-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.

[5] C. De Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[6] C. De Falco, A. Reali, and R. Vázquez. GeoPDEs: a research tool for isogeometric analysis of PDEs. *Advances in Engineering Software*, 42(12):1020–1034, 2011.

[7] J.S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics, 2015.

[8] T.J.R. 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.

[9] D.B.P. Huynh, N.C. Nguyen, G. Rozza, and A.T. Patera. rbMIT software: Copyright MIT. *Technology Licensing Office*, 2006-07. Web address to download: http://augustine.mit.edu/.

[10] 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.

[11] The MathWorks Inc. Matlab. *version 8.1.0.604 (R2013a)*, 2013.

[12] A. Manzoni, F. Salmoiraghi, and L. Heltai. Reduced Basis Isogeometric Methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils. *Computer Methods in Applied Mechanics and Engineering*, 284:1147–1180, 2015.

[13] A.T. Patera and G. Rozza. *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*. MIT, 2007. Web address to download: http://augustine.mit.edu/.

[14] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):1–49, 2011.

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

[16] 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. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[17] F. Salmoiraghi, F. Ballarin, L. Heltai, and G. Rozza. Isogeometric analysis-based reduced order modelling for incompressible linear viscous flows in parametrized shapes. *Advanced Modelling and Simulation in Engineering Sciences*, 3:21, 2016.