GUARANTEED AND SHARP A POSTERIORI ERROR ESTIMATES
IN ISOGEOMETRIC ANALYSIS

STEFAN K. KLEISS AND SATYENDRA K. TOMAR

ABSTRACT. We present functional-type a posteriori error estimates in isogeometric analysis. These estimates, derived on functional grounds, provide guaranteed and sharp upper bounds of the exact error in the energy norm. Moreover, since these estimates do not contain any unknown/generic constants, they are fully computable, and thus provide quantitative information on the error. By exploiting the properties of non-uniform rational B-splines, we present efficient computation of these error estimates. The numerical realization and the quality of the computed error distribution are addressed. The potential and the limitations of the proposed approach are illustrated using several computational examples.

1. INTRODUCTION

The geometry representations in finite element methods (FEM) and computer aided design (CAD) have been developed independent of each other, and are optimized for the purposes within their respective fields. As a consequence, the representations are different from each other, and a transfer of geometry information from CAD to FEM programmes (and vice versa) requires a transformation of geometry data. These transformations are, in general, not only costly, but also prone to approximation errors, and may require manual input.

Isogeometric analysis (IGA), introduced by Hughes et al. [23], see also [12], aims at closing this gap between FEM and CAD. The key observation is that it is a widespread standard in CAD to use geometry representations based on non-uniform rational B-splines (NURBS), and that these NURBS basis functions have properties which make them suitable as basis functions for FEM. Instead of transforming the geometry data to a conventional FEM representation, the original geometry description is used directly, and the underlying NURBS functions are used as basis for the discrete solution. This way, the geometry is represented exactly in the sense that the geometry obtained from CAD is not changed. Thus, the need for data transformation is eliminated, and furthermore, the exact representation from the coarsest mesh is preserved throughout the refinement process. IGA has been thoroughly studied and analyzed (see, e.g., [3, 7, 13, 24, 40]), and its potential has been shown by successful applications to a wide range of problems (see, e.g., [5, 6, 11, 19, 31]).

As mentioned above, the most widely used spline representations in CAD are based on NURBS. The straightforward definition of NURBS basis functions leads to a tensor-product structure of the basis functions, and thus of the discretization. Since naive mesh refinement in a tensor-product setting has global effects, the development of local refinement strategies for isogeometric analysis is a subject of current active research. Such local refinement techniques include, for example, T-splines [4, 30, 37, 38, 39], truncated hierarchical B-splines (THB-splines) [21, 22], polynomial splines over hierarchical T-meshes (PHT-splines) [15, 43], and locally-refineable splines (LR-splines) [16, 26].

The issue of adaptive, local refinement is closely linked to the question of efficient a posteriori error estimation (see, e.g., [1, 36] for a general overview on error estimators). In the light of adaptive refinement, an error estimator has to identify the areas where further refinement is needed due to the local error being significantly larger than in the rest of the domain. Hence, an accurate indication of the error distribution is essential. Another important objective in computing a posteriori error estimates is to address the quality assurance, i.e., to quantify the error in the computed solution with certain degree of guarantee. However, a posteriori error estimation in isogeometric analysis is still in an infancy stage. To the best of the authors’ knowledge, the only published results are [14, 17, 25, 28, 42, 43, 44, 45, 41].
A posteriori error estimates based on hierarchical bases, proposed by Bank and Smith [2], have been used in [17, 42]. The reliability and efficiency of this approach is subjected to the saturation assumption on the (enlarged) underlying space and the constants in the strengthened Cauchy inequality. As the authors remarked, the first assumption is critical and its validity depends on the considered example. Moreover, an accurate estimation of constants in the strengthened Cauchy inequality requires the solution of generalized minimum eigenvalue problem. As noted in [25, Page 41], this approach delivers less than satisfactory results.

Residual-based a posteriori error estimates have been used in [25, 43, 44, 45]. This approach requires the computation of constants in Clement-type interpolation operators. Such constants are mesh (element) dependent, often generic/unknown or incomputable for general element shape; and the global constant often over-estimates the local constants, and thus the exact error. This fact has been explicitly stated by the authors in [25, Pages 42-43] and in [43, Remark 1].

Goal-oriented error estimation approach has been studied in [14, 28, 41]. The results presented in these studies show that neither the estimates of this approach are guaranteed to be an upper bound, nor the efficiency indices of the estimates are sharp. Moreover, this approach also requires the solution of an adjoint problem, the cost of which can not be entirely neglected.

The approach of Zienkiewicz-Zhu type a posteriori error estimates is based on post-processing of approximate solutions, and depend on the superconvergence properties of the underlying basis. To the best of authors’ knowledge, superconvergence properties for B-splines (NURBS) functions are not yet known.

Summarily, in general situations, the reliability and efficiency of these methods often depend on undetermined constants, which is not suitable for quality assurance purposes. In this paper, we present functional-type a posteriori error estimates for isogeometric discretizations. These error estimates, which were introduced in [33, 34, 35] and have been studied for various fields (see [36] and the references therein), provide guaranteed, sharp and fully computable bounds (without any generic undetermined constants). These estimates are derived on purely functional grounds (based on integral identities or functional analysis) and are thus applicable to any conforming approximation in the respective space. For elliptic problems with the weak solution \( u \in H^1_0(\Omega) \), these error bounds involve computing an auxiliary function \( y \in H(\Omega, \text{div}) \). In order to get a sharp estimate, this function \( y \) is computed by solving a global problem. This could be perceived as a drawback when compared to error estimation techniques which rely on local computations and are thus apparently cheaper. However, as briefly explained above, our emphasis is not only on adaptivity, but also on quantifying the error in the computed solution (and thus guaranteeing the quality of the computed solution). Therefore, the associated cost should be weighed against the stated objectives. To the best of authors’ knowledge, there is no other, particularly cheaper, method available which can fulfill these objectives in general situations. In this paper, we will elaborate how such estimates can be computed efficiently by a proper set-up of the global problem.

Two aspects motivate the application of functional-type error estimates in IGA. Firstly, unlike the standard Lagrange basis functions, NURBS basis functions of degree \( p \) are, in general, globally \( C^{p-1} \)-continuous. Hence, NURBS basis functions of degree \( p \geq 2 \) are, in general, at least \( C^1 \)-continuous, and therefore, their gradients are automatically in \( H(\Omega, \text{div}) \). Thereby, we avoid constructing complicated functions in \( H(\Omega, \text{div}) \), in particular for higher degrees (see, e.g., [9, 10, 20]). Secondly, since the considered problem is solved in an isogeometric setting, an efficient implementation of NURBS basis functions is readily available, which can be used to construct the above mentioned function \( y \). Hence, applying the technique of functional-type a posteriori error estimation in a setting that relies only on the use of already available NURBS basis functions is greatly appealing.

The remainder of this paper is organized as follows. In Section 2, we define the model problem, and recall the definition and some important properties of B-spline and NURBS basis functions. In Section 3, we first recall functional-type a posteriori error estimates and known implementation issues. Then, we derive a quality criterion and the local error indicator. In Section 4, we discuss a cost-efficient realization of the proposed error estimator using an illustrative numerical example. Further numerical examples are presented in Section 5, and finally, conclusions are drawn in Section 6.

2. Preliminaries

In order to fix notation and to provide an overview, we define the model problem and recall the definition and some aspects of isogeometric analysis in this section.
2.1. Model Problem. Let \( \Omega \subset \mathbb{R}^2 \) be an open, bounded and connected Lipschitz domain with boundary \( \partial \Omega \). We shall consider the following model problem:

Find the scalar function \( u : \overline{\Omega} \to \mathbb{R} \) such that

\[
- \text{div}(A \nabla u) = f \quad \text{in} \, \Omega, \\
u = u_D \quad \text{on} \, \Gamma_D = \partial \Omega,
\]

where \( A, f \) and \( u_D \) are given data. We assume that \( A \) is a symmetric positive definite matrix and has a positive inverse \( A^{-1} \), and that there exist constants \( c_1, c_2 > 0 \) such that

\[
c_1 |\xi|^2 \leq A\xi \cdot \xi \leq c_2 |\xi|^2, \quad \forall \xi \in \mathbb{R}^2.
\]

Then, the norms

\[
\|v\|^2_A = \int_{\Omega} A v \cdot v \, dx, \quad \|v\|^2_{A^{-1}} = \int_{\Omega} A^{-1} v \cdot v \, dx,
\]

are equivalent to the \( L^2 \)-norm \( \|v\|^2 = \int_{\Omega} v \cdot v \, dx \). The weak form of problem (1) can be written as follows:

Find \( u \in V_g \), such that

\[
a(u, v) = f(v), \quad \forall v \in V_0,
\]

where \( V_0 \subset H^1(\Omega) \) contains the functions which vanish on \( \Gamma_D \), and \( V_g \subset H^1(\Omega) \) contains the functions satisfying the Dirichlet boundary conditions \( u = u_D \) on \( \Gamma_D \). We assume that the problem data \( A, f \) and \( u_D \) are given such that the bilinear form \( a(\cdot, \cdot) \) is bounded, symmetric and positive definite, and that \( f(\cdot) \) is a bounded linear functional. The energy norm of a function \( v \) is given by \( \|\nabla v\|_A = \sqrt{a(v, v)} \).

Note that we have considered the Dirichlet problem only for the sake of simplicity. Functional-type error estimates can be easily generalized to problems with mixed boundary conditions, see, e.g., [29, 36].

We discretize the problem (4) in the standard way by choosing a finite-dimensional manifold \( V_h \subset V_g \) and looking for a discrete solution \( u_h \in V_h \). This leads to a linear system of equations of the form

\[
K_h u_h = f_h,
\]

where \( K_h \) is the stiffness matrix induced by the bilinear form \( a(\cdot, \cdot) \), \( f_h \) is the load vector, and \( u_h \) is the coefficient vector of the discrete solution \( u_h \).

2.2. B-Splines, NURBS and Isogeometric Analysis. We briefly recall the definition of B-spline basis functions and NURBS mappings. We only provide the basic definitions and properties relevant for the scope of this paper. For detailed discussions of NURBS basis functions, geometry mappings and their properties, we refer to, e.g., [12, 13, 23, 32] and the references therein. The following standard definitions and statements can also be found there.

Let \( p \) be a non-negative degree and let \( s = (s_1, \ldots, s_m) \) be a knot vector with \( s_i \leq s_{i+1} \) for all \( i \). We consider only open knot vectors, i.e., knot vectors \( s \) where the multiplicity of a knot is at most \( p \), except for the first and last knot which have multiplicity \( p + 1 \). For simplicity, we assume that \( s_1 = 0 \) and \( s_m = 1 \), which can be easily achieved by a suitable scaling. The \( n = m - p - 1 \) univariate B-spline basis functions \( B_{i,p}^s : (0,1) \to \mathbb{R}, i = 1, \ldots, n \), are defined recursively as follows:

\[
B_{i,0}^s(\xi) = \begin{cases} 
1 & \text{for } s_i \leq \xi < s_{i+1} \\
0 & \text{else}
\end{cases}
\]

\[
B_{i,p}^s(\xi) = \frac{\xi - s_i}{s_{i+p} - s_i} B_{i,p-1}^s(\xi) + \frac{s_{i+p+1} - \xi}{s_{i+p+1} - s_{i+1}} B_{i+1,p-1}^s(\xi).
\]

Whenever a zero denominator appears in the definition above, the corresponding function \( B_{i,p}^s \) is zero, and the whole term is considered to be zero. For open knot vectors, the first and last basis function are interpolatory at the first and the last knot, respectively. The derivatives of B-spline basis functions are given by the following formula:

\[
\partial^k B_{i,p}^s(\xi) = \frac{p}{s_{i+p} - s_i} B_{i,p-1}^s(\xi) - \frac{p}{s_{i+p+1} - s_{i+1}} B_{i+1,p-1}^s(\xi).
\]

B-spline basis functions of degree \( p \) are, in general, globally \( C^{p-1} \)-continuous. In the presence of repeated knots, the continuity reduces according to the multiplicity, i.e., if a knot appears \( k \) times, the continuity of a B-spline basis function of degree \( p \) at that knot is \( C^{p-k} \).
Let \( \{B_{i,j}^p\}_{i,j=1}^{n_1} \) and \( \{B_{i,j}^q\}_{i,j=1}^{n_2} \) be two families of B-spline basis functions defined by the degrees \( p \) and \( q \), and the open knot vectors
\[
s = (s_1, \ldots, s_{n_1+p+1}), \quad t = (t_1, \ldots, t_{n_2+q+1}),
\]
respectively. We denote the set of all double-indices \((i,j)\) by
\[
\mathcal{I}_R = \{(i,j) : \ i \in \{1, \ldots, n_1\}, j \in \{1, \ldots, n_2\}\}.
\]
Let \( w_{i,j}, (i,j) \in \mathcal{I}_R \), be positive weights. The bivariate \textit{NURBS} basis functions \( R_{(i,j)}(\xi_1, \xi_2), (i,j) \in \mathcal{I}_R \) are defined as follows:
\[
R_{(i,j)}(\xi_1, \xi_2) = \frac{w_{i,j} B_{i,p}^s(\xi_1) B_{j,q}^t(\xi_2)}{\sum_{(k,l) \in \mathcal{I}_R} w_{k,l} B_{k,p}^s(\xi_1) B_{l,q}^t(\xi_2)}.
\]

The continuity of the B-spline basis functions is inherited by the NURBS basis functions. Note that B-splines can be seen as a special case of NURBS with all weights being equal to one. Hence, we will not distinguish between these two and we will only use the term \textit{NURBS} in the remainder of the paper.

The set of functions
\[
\hat{V}_h = \text{span}\{R_{(i,j)} : (i,j) \in \mathcal{I}_R\},
\]
associated with the parameter domain \( \hat{\Omega} = (0,1)^2 \), is uniquely determined by the degrees \( p \) and \( q \), the knot vectors \( s \) and \( t \), and the weights \( w \). To reflect the associated polynomial degrees in respective dimensions, we will also use the notation \( S_{p,q}^h \) for \( \hat{V}_h \), which denotes the NURBS function of degree \( p \) and \( C^{p-1} \)-continuity in the first coordinate, degree \( q \) and \( C^{q-1} \)-continuity in the second coordinate, and where the parameter \( h \) is the characteristic cell size (non-vanishing knot-span) of the mesh for \( \hat{V}_h \).

Given the set of functions \( \hat{V}_h \) and a control net of control points \( P_{(i,j)} \in \mathbb{R}^2 \), where \((i,j) \in \mathcal{I}_R\), the two-dimensional \textit{NURBS-surface} \( G : \hat{\Omega} \to \Omega \) is defined by
\[
G(\xi_1, \xi_2) = \sum_{(i,j) \in \mathcal{I}_R} R_{(i,j)}(\xi_1, \xi_2) P_{(i,j)}.
\]

We refer to \( \Omega = G(\hat{\Omega}) \) as the physical domain. We assume that the geometry mapping is continuous and bijective (i.e., not self-penetrating), which are natural assumptions for CAD-applications.

In isogeometric analysis, the isoparametric principle is applied by using the same basis functions for the discrete solution \( u_h \) which are used for representing the geometry. For detailed discussion, we refer the reader to, e.g., [12, 13, 23]. The discrete solution \( u_h \) on the physical domain \( \Omega \) is represented as follows:
\[
u_h(x) = \sum_{(i,j) \in \mathcal{I}_R} u_{(i,j)} \left( R_{(i,j)} \circ G^{-1}\right)(x),
\]
where \( u_{(i,j)} \in \mathbb{R} \) are real-valued coefficients which form the coefficient vector \( u_h \). The discrete functions space is thus defined by
\[
V_h = \text{span}\{R_{(i,j)} \circ G^{-1} : (i,j) \in \mathcal{I}_R\}.
\]

The initial mesh, and thereby the basis functions on this initial mesh, are assumed to be given via the geometry representation of the computational domain, i.e., the initial discretization is already determined by the problem domain. The exact representation of the geometry on the initial (coarsest) level is preserved in the process of mesh refinement.

As mentioned in the introduction, the straightforward definition of NURBS basis functions, leads to a tensor-product structure of the discretization, which is the focus of this paper. Nevertheless, the error estimator presented herein is also applicable to local refinement techniques (e.g., T-splines, THB-splines, PHT-splines, LR-splines, see Section 1) since it is derived purely on functional grounds.

3. Functional-type a Posteriori Error Estimates

In the first two parts of this section, we will discuss the well-known theoretical upper bound for the error in the energy norm (see, e.g., [33, 34, 35, 36]), and we recall how to minimize this upper bound in order to get a sharp error estimate (see, e.g., [27, 29]). Thereafter, in Section 3.3, we will derive a quality criterion from the discussed theory. We will comment on the realization in the isogeometric context in Section 4.
3.1. Guaranteed Upper Bound for the Error. The starting point for the proposed method is the following main result, which gives an upper bound for the error in the energy norm. It can be found, e.g., in [34, 35, 36].

**Theorem 3.1.** Let \( C_Ω \) be the constant in the Friedrich’s type inequality \( \| v \| \leq C_Ω \| \nabla v \|_A, \forall v \in V_0 \). Let \( u \) be the exact solution of the problem (4), and let \( u_h \in V_h \) be an approximate solution. Then, the following estimate holds:

\[
\| \nabla u - \nabla u_h \|_A \leq \| A \nabla u - y \|_A + C_Ω \| \div y + f \|,
\]

where \( y \) is an arbitrary vector-valued function in \( H(Ω, \div) \), and the norms are as defined in (3).

The constant \( C_Ω \) depends only on the domain \( Ω \) and the coefficient matrix \( A \) (but not on the underlying mesh), see, e.g., [29, 36]. Note that \( C_Ω \) can be computed either numerically or, if one can find a domain \( Ω_\Box \supset Ω \), where \( Ω_\Box \) is a square domain with side-length \( ℓ \), then \( C_Ω \leq c_2 \| \frac{ℓ}{πd} \|, \) where \( d \) is the dimension and \( c_2 \) is the constant in (2).

Note that, if we choose \( y \) via the (unknown) exact solution \( y = A \nabla u \), both sides of (8) coincide. Hence, the estimate is sharp in the sense that, for any fixed \( u_h \), we can find a function \( y \) such that the upper bound is as close to the exact error as desired. The estimate given in Theorem 3.1 is a guaranteed and fully computable upper bound for any conforming approximation \( u_h \in V_h \).

In the following, we describe some approaches to construct the function \( y \) and discuss their relative merits. For this reason, we consider a numerical example, referred to as Example 1 in the remainder, whose solution is a smoothly varying function in both directions.

**Example 1. Sinus function in a unit square:** In this numerical example, the computational domain is the unit square \( Ω = (0, 1)^2 \) and \( u_h \in \mathcal{S}_h^{2,2} \), i.e., a piecewise quadratic function in both directions. The coefficient matrix is the identity matrix, i.e., \( A = I \), and the exact solution is given by

\[
u = \sin(6πx) \sin(3πy)\]

The right-hand-side \( f \) and the (homogeneous) boundary conditions \( u_D \) are determined by the prescribed exact solution \( u \).

3.1.1. Post-processing of \( u_h \). It is possible to obtain good error indicators by constructing a function \( y \) by some post-processing of the discrete solution \( u_h \), see [29, 36] and the references therein. Since \( u_h \in C^{p-1}_h \), we have \( \nabla u_h \in (C^{p-2}_h)^2 \subset H(Ω, \div) \) for \( p \geq 2 \). Choosing \( y = \nabla u_h \) will thus result in

\[
\| \nabla u - \nabla u_h \| \leq C_Ω \| \Delta u_h + f \|.
\]

Once we have calculated \( \eta_Q := \| \Delta u_h + f \|_Q \) for each cell \( Q \) of the mesh, we can compare the local errors and choose a criterion for selecting cells which will be marked for further refinement. Typically, one chooses a threshold \( Θ \) and marks all cells \( Q \) for refinement, where the local error is above this threshold. There are several possibilities for determining \( Θ \), e.g., the bulk-criterion proposed in [18]. For simplicity, we choose a percentage \( ψ \) and mark a cell \( Q \) for refinement, if

\[
η_Q > Θ, \text{ where } Θ = (100 - ψ)-\text{percentile of } \{η_Q\}_Q.
\]

The \( α \)-percentile of a set \( \mathcal{A} = \{a_1, \ldots, a_N\} \) denotes the value \( \bar{a} \) below which \( α \) percent of all values \( a_i \) fall. For example, if we choose \( ψ = 20\% \) in (10), then \( Θ \) is chosen such that \( n_Q > Θ \) holds for 20\% of all cells \( Q \).

![Cells marked by exact error with ψ = 20% in Example 1](image)

**Figure 1.** Cells marked by exact error with \( ψ = 20\% \) in Example 1, \( V_h = \mathcal{S}_h^{2,2} \).
To show the efficiency of the estimator (9), in Figure 1, we present the cells marked for refinement by the exact error. The cells marked for refinement by the majorant given in (9) are presented in Figure 2. We see that starting from the mesh $32 \times 32$, the majorant is able to nicely capture the refinement pattern of exact error. However, from a closer look at the convergence of the exact error and the majorant, see Figure 3, we find that though such an estimate is a guaranteed upper bound and very cheap to compute, it over-estimates the exact error, and its convergence is slower than the exact error (due to a lack of proper scaling, different operators acting on $u_h$ on both sides).

3.1.2. Global minimization. In order to obtain a sharp estimate (and not just an indicator), therefore, one has to find a function $y$ which minimizes the right-hand-side of (8). For minimizing the estimate (8) numerically, we first rewrite the estimate in the following form

$$\|\nabla u - \nabla u_h\|^2_A \leq (1 + \beta)\|A\nabla u_h - y\|^2_A + (1 + \frac{1}{\beta})C^2\|\text{div } y + f\|^2 =: M_\beta^2(y, \beta),$$

where $\beta > 0$ is a free parameter [29, 36]. Note that the upper bound in (11) holds true for any fixed $y \in H(\Omega, \text{div})$ and $\beta > 0$. Hereinafter, for simplicity, we will refer to $M_\beta^2(y, \beta)$ as the majorant. Introducing

$$a_1 = 1 + \beta, \quad a_2 = (1 + \frac{1}{\beta})C^2, \quad B_1 = \|A\nabla u_h - y\|^2_A, \quad B_2 = \|\text{div } y + f\|^2,$$

we can briefly write the majorant as

$$M_\beta^2(y, \beta) = a_1B_1 + a_2B_2.$$
The efficiency index, defined by
\[ I_{\text{eff}} = \frac{M_{\oplus}(y, \beta)}{\|\nabla u - \nabla u_h\|_A}, \]
indicates how close the calculated majorant is to the exact error. The closer \( I_{\text{eff}} \) is to 1, the better the estimate. Therefore, obtaining a sharp estimate requires to find \( y \in H(\Omega, \text{div}) \) and \( \beta > 0 \) as solutions to the global minimization problem
\[ \min_{y \in H(\Omega, \text{div}), \, \beta > 0} M_{\oplus}^2(y, \beta). \]

The technique for finding such minimizing parameters \( y \) and \( \beta \) will be discussed in Sections 3.2 and 4.2. Before proceeding further, we give the following Lemma 3.3, which can be found in [36, Prop. 3.10]. It provides an analytical result on the sharpness of the bound \( M_{\oplus}^2(y, \beta) \). For later reference, we also sketch the proof.

**Definition 3.2.** A sequence of finite-dimensional subspaces \( \{Y_j\}_{j=1}^\infty \) of a Banach-space \( Y \) is called limit dense in \( Y \), if for any \( \varepsilon > 0 \) and any \( v \in Y \), there exists an index \( j_\varepsilon \), such that \( \inf_{p_k \in Y_k} \|p_k - v\| < \varepsilon \) for all \( k > j_\varepsilon \).

**Lemma 3.3.** Let the spaces \( \{Y_j\}_{j=1}^\infty \) be limit dense in \( H(\Omega, \text{div}) \). Then
\[ \lim_{j \to \infty} \inf_{y_j \in Y_j, \beta > 0} M_{\oplus}^2(y_j, \beta) = \|\nabla u - \nabla u_h\|_A^2. \]

**Proof.** Recall that the \( H(\Omega, \text{div}) \)-norm \( \| \cdot \|_{\text{div}} \) is defined by \( \|v\|_{\text{div}}^2 = \|v\|^2 + \|\text{div } v\|^2 \). Let \( \varepsilon > 0 \) be arbitrarily small, but fixed. Let \( j_\varepsilon \) be the index such that, for all \( k > j_\varepsilon \), there exists a \( p_k \in Y_k \) with \( \|\nabla u - p_k\|_{\text{div}} < \varepsilon \). Then,
\[ \inf_{y_j \in Y_j, \beta > 0} M_{\oplus}^2(y_j, \beta) \leq M_{\oplus}^2(p_k, \varepsilon) = (1 + \varepsilon)\|\nabla u_h - p_k\|_A^2 + (1 + \frac{1}{\varepsilon})C_{\Omega}^2\|f + \text{div } p_k\|^2. \]

Since \( \|Av\|_A = \|v\|_A \), we can write
\[ \|\nabla u_h - p_k\|_A \leq \|\nabla u_h - \nabla u_h\|_A + \|\text{div } u - p_k\|_A = \|\nabla u_h - u_h\|_A + \|\text{div } u - p_k\|_A. \]
The norm \( \| \cdot \|_A \) is equivalent to the \( L^2 \)-norm, so there exists a constant \( c_A \), such that the second term in the right-hand side can be bounded by
\[ \|\text{div } u - p_k\|_A \leq c_A\|A\nabla u - p_k\| \leq c_A\|A\nabla u - p_k\|_{\text{div}} \leq c_A\varepsilon. \]
Hence, we obtain the following estimate for the first term in (16):
\[ \|\nabla u_h - p_k\|_A \leq c_A\|\nabla u - p_k\|_{\text{div}} \leq c_A\|A\nabla u - p_k\|_{\text{div}} \leq c_A\varepsilon. \]

Since \( f = -\text{div } A\nabla u \), we can bound the second term in (16) as follows:
\[ \|\text{div } p_k + f\| = \|\text{div } p_k - \text{div } A\nabla u\| \leq \|p_k - A\nabla u\|_{\text{div}} \leq \varepsilon. \]

With (17) and (18), we can rewrite (16) as
\[ M_{\oplus}^2(p_k, \varepsilon) \leq (1 + \varepsilon)(\|\nabla u - \nabla u_h\|_A^2 + O(\varepsilon)) + (1 + \frac{1}{\varepsilon})C_{\Omega}^2\varepsilon^2 = \|\nabla u - \nabla u_h\|_A^2 + O(\varepsilon). \]
Hence, the bound \( M_{\oplus}^2(p_k, \varepsilon) \) tends to \( \|\nabla u - \nabla u_h\|_A^2 \) as \( \varepsilon \to 0 \).

**3.2. Steps Involved in Minimizing \( M_{\oplus}^2(y, \beta) \).** As mentioned above, we need to find parameters \( y \) and \( \beta \) which minimize the majorant. To do this, we apply an interleaved iteration process in which we alternate fixing one of the variables and minimizing with respect to the other. This process, which we summarize in the following, has been described, e.g., in [27, 29].

**Step 1:** Minimization with respect to \( y \): Assume that \( \beta > 0 \) is given and fixed, either by an initial guess or as a result of Step 2 below. We view the majorant \( M_{\oplus}^2(y) \) as a quadratic function of \( y \) and calculate its Gateaux-derivative \( M_{\oplus}^2(y)' \) with respect to \( y \) in direction \( \hat{y} \). Setting \( M_{\oplus}^2(y)' = 0 \), we obtain
\[ a_1 \int_\Omega A^{-1}y \cdot \hat{y} \, dx + a_2 \int_\Omega \text{div } y \, \text{div } \hat{y} \, dx = a_1 \int_\Omega \nabla u_h \cdot \hat{y} \, dx - a_2 \int_\Omega f \, \text{div } \hat{y} \, dx, \]
where \( a_1 = 1 + \beta \) and \( a_2 = (1 + \frac{1}{\beta})C_{\Omega}^2 \), as defined in (12). In order to solve (19), we choose a finite-dimensional subspace \( Y_h \subset H(\Omega, \text{div}) \) and search for a solution \( y_h \in Y_h \). Testing in all directions \( \tilde{y} \in Y_h \) leads to a linear system of equations which we write as

\[
L_h y_h = r_h.
\]

Here, \( L_h \) and \( r_h \) are the matrix and the vector induced by the left hand side and the right hand side of equation (19), respectively. By solving (20), we obtain the coefficient vector \( y_h \) for the discrete function \( y_h \), minimizing \( M_{\beta}^2(y_h) \) in \( Y_h \subset H(\Omega, \text{div}) \). Note that this process requires non-negligible cost as we need to assemble \( L_h \) and \( r_h \) and solve the system (20).

**Step 2:** Minimization with respect to \( \beta \): Assume that \( y_h \) is given from Step 1. By direct calculation, we see that \( M_{\beta}^2(\beta) \) is minimized with respect to \( \beta \) by setting

\[
\beta = C_{\Omega} \sqrt{\frac{B_2}{B_1}},
\]

where \( B_1 \) and \( B_2 \) are as defined in (12). Note that the evaluation of \( B_1 \) and \( B_2 \) (and thus \( \beta \)) requires only the evaluation of integrals, and thus involves negligible cost.

Steps 1 and 2 are repeated iteratively. We will refer to one loop of applying Step 1 and Step 2 as one *interleaved iteration*. Once we have computed minimizers \( y_h \) and \( \beta \), the computation of the majorant \( M_{\beta}(y_h, \beta) \) is straightforward as it requires only the evaluation of the integrals.

Note that the matrix \( L_h \) can be written as

\[
L_h = a_1 L_h^1 + a_2 L_h^2,
\]

where \( L_h^1 \) and \( L_h^2 \) correspond to the terms \( \int_{\Omega} A^{-1} y \cdot \tilde{y} \, dx \) and \( \int_{\Omega} \text{div} \, y \, \text{div} \, \tilde{y} \, dx \) in (19), respectively. Since the matrices \( L_h^1 \) and \( L_h^2 \) in (22) do not change in the interleaved iteration process, they need to be assembled only once. Analogously to (22), we can write \( r_h \) as

\[
r_h = a_1 r_h^1 - a_2 r_h^2,
\]

where \( r_h^1 \) and \( r_h^2 \) correspond to the terms \( \int_{\Omega} \nabla u_h \cdot \tilde{y}_h \, dx \) and \( \int_{\Omega} f \, \text{div} \, \tilde{y}_h \, dx \) in (19), respectively. The terms \( r_h^1 \) and \( r_h^2 \) also need to be assembled only once since they also do not change in the interleaved iteration process. The full matrix \( L_h \) and vector \( r_h \), however, do change in each iteration, because of the change in \( \beta \) and \( y_h \). Based on past numerical studies, see, e.g., [27, 29], and the results presented in Sections 4 and 5, it has been found that for linear problems, one or two such interleaved iterations are enough for obtaining a sufficiently accurate result.

To recapitulate, we summarize the steps for computing the majorant in Algorithm 1.

**Algorithm 1** Computation of the majorant \( M_{\beta} \)

**Require:** \( u_h, f, C_{\Omega}, Y_h \)

**Ensure:** \( M_{\beta} \)

\[
\beta := \text{initial guess}
\]

Assemble and store \( L_h^1, L_h^2, r_h^1, r_h^2 \)

while convergence is not achieved or maximum number of interleaved iterations is not reached do

\[
L_h := (1 + \beta)L_h^1 + (1 + \frac{1}{\beta})C_{\Omega}^2 L_h^2
\]

\[
r_h := (1 + \beta)r_h^1 - (1 + \frac{1}{\beta})C_{\Omega}^2 r_h^2
\]

Solve \( L_h y_h = r_h \) for \( y_h \)

\[
B_1 := \| A \nabla u_h - y_h \|^2_A
\]

\[
B_2 := \| \text{div} \, y_h + f \|^2_A
\]

\[
\beta := C_{\Omega} \sqrt{B_2 / B_1}
\]

end while

\[
M_{\beta}(y, \beta) := \sqrt{(1 + \beta)B_1 + (1 + \frac{1}{\beta})C_{\Omega}^2 B_2}
\]

**Remark 3.4.** Note that the space \( H(\Omega, \text{div}) \), where the auxiliary quantity \( y \) is sought, is a global space, and for a general complicated problem, it is not immediately clear how to locally compute \( y \) without global effect. That being said, a local version of our estimator can be devised for specific problems.
and data (like equilibration of flux approach), however, that will restrict its generality, which is not very appealing to us. Therefore, in the remainder of the paper, we will focus on computing the majorant from the global minimization problem.

3.3. Quality Indicator and Local Error Indicator. So far, we have defined the majorant and discussed how we minimize (numerically) the majorant over \(Y_a\). Another important question, especially in the light of adaptive, local refinement, is whether a calculated majorant does correctly capture the error distribution. From the proof of Lemma 3.3, we recall the following observation:

\[
\begin{align*}
    a_1 B_1 &\rightarrow \|\nabla u - \nabla u_h\|_A^2 \\
    a_2 B_2 &\rightarrow 0, \text{ as } y_h \in H(\Omega, \text{div}) \rightarrow A\nabla u.
\end{align*}
\]

From this, we deduce the following quality indicator.

**Proposition 3.5.** The distribution of the exact error is captured correctly, if

\[
a_1 B_1 > C_B a_2 B_2
\]

with some constant \(C_B > 1\).

This criterion is easy to check, since the terms appearing in (25) are evaluated in the process of minimizing \(M^p(y, \beta)\). It was found from extensive numerical studies (see examples presented in Sections 4 and 5) that an accurate distribution of the error is obtained for \(C_B \geq 5\).

**Remark 3.6.** For the choice of \(C_B \geq 5\), we have \(a_2 B_2 < a_1 B_1 / 5\), and therefore, \(\|\nabla u - \nabla u_h\|_A \leq \sqrt{1.2 a_1 B_1} \). One can see from all the tables in Sections 4 and 5, that whenever this criterion is satisfied, we have \(I_{eff} \leq 1.2 \) (the ratio of \(\sqrt{a_1 B_1} / \|\nabla u - \nabla u_h\|_A\) appears to be of the same magnitude as \(\sqrt{1+1/C_B}\)). Note that this criterion does not require \(a_2 B_2\) to be close to zero, but just less than \(a_1 B_1 / 5\). Since these approximations (of the original problem and the auxiliary problem in \(H(\Omega, \text{div})\) are monotonically convergent, the approximation at any level will only improve at the next refinement level, and this is why the results get better for any further refinement. Clearly, all the terms are fully computable, and thus, usable in an algorithm.

We define the local error indicator \(\eta_Q\) on a cell \(Q\) as the restriction of the first component of the majorant to the cell \(Q\), i.e., by

\[
\eta_Q^2(y_h) = \int_Q (\nabla u_h - A^{-1} y_h)(A \nabla u_h - y_h) \, dx.
\]

The factor \((1+\beta)\) is omitted, since this scalar factor is the same for all cells of the domain. As remarked in the observation (24), the first component will converge to the exact error, thus providing a good indicator for the error distribution. A more detailed discussion of this indicator can be found in [36, Sec. 3.6.4]. For refinement based on \(\eta_Q\), we again use the criterion (10).

4. Efficiency and Computational Cost of the Proposed Estimator in the Isogeometric Context

We now discuss the efficiency and the computational cost of the proposed estimator based on the global minimization steps presented in Section 3.2. Through out this Section, we again consider Example 1 from Section 3.1. All the computations for this example and the examples presented in Section 5 are performed in MATLAB® on an HP workstation Z420 with Intel Xeon CPU E5-1650, 3.2 GHz, 12 Cores and 16 GB RAM, and the linear systems (5) and (20) are solved using the in-built direct solver. The right-hand-side \(f\) and the boundary conditions \(u_D\) are determined by the prescribed exact solution \(u\).

We study the efficiency of the majorant based on straight forward computational procedure, as discussed in Section 4.1, and based on cost-efficient procedure, as discussed in Section 4.2, which coarsens the mesh and increases the polynomial degree simultaneously. This alternative cost-efficient procedure will then be used in Section 5 for further numerical examples. In all the numerical results of Example 1 in this Section, the initial guess for \(\beta\) is 0.01.

In the tables, we indicate the mesh-size by the number of interior knot spans of the knot vectors \(s\) and \(t\), respectively. By this, we mean the number of knot spans without counting the vanishing knot spans at
the beginning and the end of the open knot vectors. For example, if

\[ s = (0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1), \]
\[ t = (0, 0, 0, 0.5, 0.5, 1, 1, 1, 1), \]

then the mesh-size is \( 4 \times 3 \), since the empty knot span \((0.5, 0.5)\) in \( t \) is also counted as an interior knot span.

We compare the timings for assembling and for solving the linear systems (5) and (20), as well as the total time for assembling and solving. In the presented tables, these timings are shown in the columns labeled assembling-time, solving-time, and sum, respectively. The label \( pde \) indicates that the column corresponds to solving the partial differential equation (5), i.e., to assembling \( K_h \) and solving (5) for \( u_h \). The label \( est \) indicates that the timings correspond to the estimator, i.e., assembling \( L_h \) and solving (20) for \( y_h \). In the column labeled \( \frac{est}{pde} \) we present the ratio of these timings. Note that these ratios were computed before rounding the numbers, i.e., taking the ratios of the reported numbers may result in slightly different values.

The computed efficiency indices \( I_{\text{eff}} \) (see (14)) are presented in tables. In order to check the quality criterion discussed in Section 3.3, we present the values of \( a_1 B_1 \) and \( a_2 B_2 \) and see whether the inequality (25) is fulfilled or not. To indicate the quality of the error distribution captured by the majorant, we plot which cells are marked for refinement based on the exact local error and the criterion (10) (plotted in black), and compare this to the refinement marking based on the criterion (10) applied to the computed error estimate (plotted in magenta).

4.1. Straightforward Procedure.

**Case 0. (Straightforward Procedure)** For the first choice for \( \hat{Y}_h \), we use the same mesh as for \( \hat{V}_h \), and choose

\[ \hat{Y}_h = \mathcal{S}^{p+1,p}_{h} \otimes \mathcal{S}^{p+1}_{h}. \]

The function space \( \hat{Y}_h \) is then defined by the well known Piola transformation [8].

We consider the same setting as presented in Example 1 in Section 3.1. In Table 1, we present the computed efficiency indices obtained with this choice of \( \hat{Y}_h \), which show that upper bound approaches 1 (representing exact error) as the mesh is refined. The dashed line in Table 1 indicates that the criterion (25) is fulfilled with \( C_{\text{up}} = 5 \) (actually 4.94) starting from the mesh \( 64 \times 64 \). The cells marked by the error estimator are shown in Figure 4. When comparing these plots to those presented in Figure 1, we see that the error distribution is captured accurately starting from the mesh \( 32 \times 32 \).

The timings presented in Table 2, however, show that the computation of the error estimate is costlier (about 4.5 times) than assembling and solving the original problem. This is not surprising, since, when \( N_u \) denotes the number of degrees of freedom (DOF) of \( u_h \), the number of DOF of \( y_h \), which is vector-valued, is asymptotically \( 2N_u \). This results in higher assembly time and the solution time for the linear system (where a direct solver is used). Clearly, this straightforward approach is not cost-efficient. In the next section, therefore, we discuss some cost-efficient approaches for computing \( y_h \).

| mesh-size | \( I_{\text{eff}} \) | \( a_1 B_1 \) | \( a_2 B_2 \) |
|-----------|----------------|-------------|-------------|
| \( 8 \times 8 \) | 3.43           | 2.62e+01   | 1.17e+02   |
| \( 16 \times 16 \) | 1.92           | 6.07e-01   | 6.19e-01   |
| \( 32 \times 32 \) | 1.41           | 2.29e-02   | 9.71e-03   |
| \( 64 \times 64 \) | 1.20           | 1.15e-03   | 2.35e-04   |
| \( 128 \times 128 \) | 1.10           | 6.51e-05   | 6.54e-06   |
| \( 256 \times 256 \) | 1.05           | 3.87e-06   | 1.95e-07   |
| \( 512 \times 512 \) | 1.03           | 2.36e-07   | 5.94e-09   |

**Table 1.** Efficiency index and components of the majorant in Example 1, Case 0, \( \hat{V}_h = \mathcal{S}_h^{2,2}, \hat{Y}_h = \mathcal{S}_h^{3,2} \otimes \mathcal{S}_h^{2,3} \).
4.2. Alternative Cost-Efficient Procedure. Recall that the cost of Step 1 of the algorithm presented in Section 3.2 depends on the choice of $Y_h \subset H(\Omega, \text{div})$. As shown in Lemma 3.3, we can make the estimate as sharp as we desire by choosing a suitably large space $Y_h$. However, the larger $Y_h$ is chosen, the more costly setting up and solving the system (20) becomes. Clearly, it is highly desirable to keep the cost for error estimation below the cost for solving the original problem.

As discussed above, choosing $\hat{Y}_h$ as in (27) does not result in a cost-efficient method. Apart from the fact that $y_h$ is vector-valued while $u_h$ is scalar, another aspect contributes to the high cost for the procedure presented in Section 4.1. Recall that, by choosing $\hat{Y}_h$ as in (27), we have

$$
y_1 \in S_h^{p+1, p},$$

$$
y_2 \in S_h^{p, p+1},$$

i.e., the components of $y_h$ are in different spline spaces. Hence, we have to compute different basis functions for $y_1$ and $y_2$ (note that this can be a costly procedure for higher polynomial degrees). Furthermore, when assembling, for example, the matrix $L_1$, we need to compute integrals over products of basis functions of the form

$$
\int_{\Omega} R_i R_j \, dx.
$$

With $\hat{Y}_h$ as in (27), the product $R_i R_j$ of basis functions of $y_1$ is different to the product of basis functions of $y_2$, hence, the integrals have to be evaluated independently for $y_1$ and $y_2$.

**Case 1.** In the light of these observations, and since $(C^{p-2})^d \subset H(\Omega, \text{div}), \forall p \geq 2$, we study the following alternative choice for $\hat{Y}_h$.

$$
(28) \quad \hat{Y}_h = S_h^{p+1, p+1} \otimes S_h^{p+1, p+1}.
$$

Thereby, we choose a function space $\hat{Y}_h$ on the parameter domain and, analogously to the relation of $\hat{V}_h$ and $V_h$, we define the function space $Y_h$ by the push-forward

$$
Y_h = \hat{Y}_h \circ G^{-1}.
$$
We refer to this setting as Case 1 in the remainder of the paper. With this choice, \( y_1 \) and \( y_2 \) are contained in the same spline spaces. Hence, the basis functions need to be computed only once, and any computed function values can be used for both components of \( y_h \).

The computed efficiency indices are presented in Table 3, which show that we obtain even better (i.e., sharper) upper bounds for the exact error with \( \hat{Y}_h \) as in (28) than with the choice (27). When we compare the plots of the cells marked by the error estimator in Figure 5 to the plots in Figure 1, we see that the error distribution is again captured accurately starting from the mesh \( 32 \times 32 \). The dashed line in Table 3 indicates that the criterion (25) is fulfilled with \( C_{\Theta} = 5 \) starting from the mesh \( 64 \times 64 \).

The timings obtained with this method are presented in Table 4. This approach reduced the total time needed for computing the majorant from a factor of about 4.5 to a factor of approximately 3 compared to the time for assembling and solving the original problem. Nevertheless, a factor of 3 in the timings is still not very appealing, and demands further reduction in the cost.

| mesh-size | \( I_{\text{eff}} \) | \( a_1 B_1 \) | \( a_2 B_2 \) |
|-----------|-----------------|--------------|--------------|
| 8 × 8     | 2.77            | 8.08e+01     | 1.24e+01     |
| 16 × 16   | 1.71            | 5.75e-01     | 3.96e-01     |
| 32 × 32   | 1.32            | 2.14e-02     | 7.05e-03     |
| 64 × 64   | 1.16            | 7.11e-03     | 1.78e-04     |
| 128 × 128 | 1.08            | 6.39e-05     | 5.08e-06     |
| 256 × 256 | 1.04            | 3.83e-06     | 1.53e-07     |
| 512 × 512 | 1.02            | 2.35e-07     | 4.69e-09     |

Table 3. Efficiency index and components of the majorant in Example 1, Case 1, \( \hat{V}_h = S_{h}^{2,2}, \hat{Y}_h = S_{h}^{3,3} \otimes S_{h}^{3,3} \).

![Cells marked by error estimator with \( \psi = 20\% \) in Example 1, Case 1, \( \hat{V}_h = S_{h}^{2,2}, \hat{Y}_h = S_{h}^{3,3} \otimes S_{h}^{3,3} \).](image)

Remark 4.1. Note that the use of equal degree polynomials for both the components of \( \hat{Y}_h \) is only possible because of extra continuity readily available from NURBS basis functions. A counter-part is not possible in FEM case simply because the derivatives of FEM basis functions (with \( C^0 \)-continuity)
is only in $L^2$, and hence, one cannot avoid using proper subspaces of $H(\Omega, \text{div})$, e.g., Raviart-Thomas space (with unequal degree polynomials in both the dimensions for both the components). It is further important to note from a close inspection of Tables 1 and 3 that the results from equal degree components of vector-valued quantity outperformed the results from unequal degree case.

In order to further reduce the computational cost, we reduce the number of DOFs of $y_h$ by coarsening the mesh by a factor $K$ in each dimension. The number of DOFs of $y_h$ is thus reduced to $2N_u/K^2$ (asymptotically). The larger $K$ is chosen, the greater the reduction of DOFs will be. At the same time, if the coarsening is done too aggressively, sharp features might not be detected properly on coarse meshes. We counter the reduction in accuracy due to mesh-coarsening by increasing the polynomial degree of $y_h$ by some positive integer $k$, i.e., we choose
\[ \hat{Y}_h = S_{K}^{p+k,p+k} \otimes S_{K}^{p+k,p+k}. \]

Note that, if desired, one could also choose different factors $K_1$ and $K_2$ and different degree increases $k_1$ and $k_2$ for the first and second component, respectively.

**Remark 4.2.** With the choices of $\hat{Y}_h$ as in (29), we take advantage of the following specific property of univariate NURBS basis functions. For $C^{p-1}$-continuity, increasing the polynomial degree by $k$ only adds a total of $k$ additional basis functions. In other words, the global smoothness can be increased at the cost of only a few additional DOFs. Coarsening the mesh by a factor $K$, however, will also reduce the number of DOFs by the same factor $K$ (asymptotically).

Moreover, as we will see from the three cases of Example 1, asymptotically we get better efficiency indices with higher degree $p$ and coarser meshes as compared to lower degree $p$ and finer meshes. This phenomenon is similar to the $p$ finite element discretization for problems with smooth solutions, where increasing the polynomial degree for a fixed mesh size $h$ is much more advantageous than decreasing the mesh size $h$ for a fixed (low) polynomial degree. Nevertheless, such a low cost construction for higher degree $p$ is not possible in FEM discretizations.

Note that Case 1 discussed above fits into this framework, since Case 1 corresponds to the choice $K = k = 1$.

**Case 2.** For the next setting, we apply moderate mesh-coarsening by choosing $K = k = 2$ (i.e., $\hat{Y}_h = S_{2h}^{p+2,p+2} \otimes S_{2h}^{p+2,p+2}$).

Similar to Case 1, the function space $\hat{Y}_h$ is defined by the push-forward
\[ Y_h = \hat{Y}_h \circ G^{-1}. \]

This setting will be referred to as Case 2 in the remainder of the paper. The computed efficiency indices along with the magnitudes of the terms $a_1B_1$ and $a_2B_2$ for Case 2 are presented in Table 5, and the marked cells are plotted in Figure 6. The dashed line indicates that criterion (25) is fulfilled with $C_{\text{db}} = 5$, and that a good upper bound of the error is computed and the correct error distribution is captured on meshes starting from $64 \times 64$. On coarse meshes, however, the efficiency index is larger than in Case 1, which is due to the boundary effects. The timings presented in Table 6 show that, even though Case 2 is faster than Case 1, this approach still costs roughly as much as solving the original problem. This is due to the costlier evaluation of the higher degree basis functions, as well as the increased support and overlap of the basis functions, which results in more non-zero entries in $L_h$ than in $K_h$.

**Case 3.** To further improve the timings, we coarsen the mesh more aggressively by a factor of 4 and, at the same time, increase the polynomial degree of $y_h$ by 4, as compared to $u_h$, i.e.,
\[ K = k = 4 \] (i.e., $\hat{Y}_h = S_{4h}^{p+4,p+4} \otimes S_{4h}^{p+4,p+4}$).

Again, similar to Case 1, the function space $Y_h$ is defined by the push-forward
\[ Y_h = \hat{Y}_h \circ G^{-1}. \]

We refer to this setting as Case 3 in the remainder of the paper. This aggressive coarsening notably affects the efficiency index on coarse meshes, see Table 7. On fine meshes, however, the efficiency indices are close to 1 in all presented cases. The number of DOFs of $y_h$ in Case 3 is only $N_u/8$ (asymptotically). The timings presented in Table 8 show that this setting results in a method which can be performed
### Table 5. Efficiency index and components of the majorant in Example 1, Case 2, $\hat{V}_h = S_2^{2,2}, \hat{Y}_h = S_2^{4,4} \otimes S_2^{4,4}$.

| mesh-size | $I_{\text{eff}}$ | $a_1B_1$ | $a_2B_2$ |
|-----------|------------------|-----------|-----------|
| $8 \times 8$ | 14.19 | 1.59e+03 | 8.53e+02 |
| $16 \times 16$ | 8.49 | 1.97e+01 | 4.32e+00 |
| $32 \times 32$ | 1.82 | 3.05e-02 | 2.41e-02 |
| $64 \times 64$ | 1.16 | 1.12e-03 | 1.76e-04 |
| $128 \times 128$ | 1.04 | 6.14e+00 | 4.08e+00 |
| $256 \times 256$ | 1.01 | 3.72e-02 | 3.32e-08 |
| $512 \times 512$ | 1.00 | 2.31e-07 | 5.13e-10 |

Significantly faster (at almost half of the cost) than solving the original problem. The more aggressive reduction of DOF outweighs the additional costs mentioned above, even though the polynomial degree is now increased by 4.

### Remark 4.3.
In all Cases for Example 1, criterion (25) is fulfilled with $C_{\oplus} = 5$ on meshes of size $64 \times 64$ and finer. This is indicated by the dashed lines in Tables 1, 3, 5, and 7, and is clear from Figures 4-7. Therefore, Example 1 and the examples discussed in Section 5 show that $C_{\oplus} = 5$ is a good choice for checking criterion (25) numerically, even though this choice may be conservative in some cases.

We now comment on the interleaved iterations. The results in the Tables 1-7 were obtained by applying only two interleaved iterations, as described in Section 3.2. As mentioned there, a sufficiently accurate result can be obtained already after the first such iteration. To illustrate this, we present the efficiency indices for Case 3 in Table 9, which were obtained after one, two, and four interleaved iterations, respectively. The efficiency index does vary notably on the coarser meshes, but since all of these values greatly overestimate the exact error, they do not correctly capture the error distribution. On meshes, where the criterion (25) is fulfilled with $C_{\oplus} = 5$, and thus the error distribution is correctly recovered, the differences due to more interleaved iterations are insignificant.
TABLE 7. Efficiency index and components of the majorant in Example 1, Case 3, \( \tilde{V}_h = S_{h}^{2,2}, \tilde{Y}_h = S_{4h}^{6,6} \otimes S_{4h}^{6,6} \).

| mesh-size | \( I_{eff} \) | \( a_1B_1 \) | \( a_2B_2 \) |
|-----------|-------------|-------------|-------------|
| 8 \times 8 | 11.28       | 5.38e+02   | 1.01e+03    |
| 16 \times 16 | 36.43     | 2.83e+02   | 1.60e+02    |
| 32 \times 32 | 12.63     | 2.04e+00   | 5.81e-01    |
| 64 \times 64 | 1.17      | 1.13e-03   | 1.88e-04    |
| 128 \times 128 | 1.01     | 5.98e-05   | 3.79e-07    |
| 256 \times 256 | 1.00     | 3.70e-06   | 1.24e-09    |
| 512 \times 512 | 1.00     | 2.31e-07   | 5.32e-12    |

TABLE 8. Number of DOF and timings in Example 1, Case 3, \( \tilde{V}_h = S_{h}^{2,2}, \tilde{Y}_h = S_{4h}^{6,6} \otimes S_{4h}^{6,6} \).

| mesh-size | \#DOF | assembling-time | solving-time | sum |
|-----------|-------|-----------------|--------------|-----|
|           | \( u_h \) | \( y_h \) | pde | est | est | pde | est | est | pde | est | est | pde | est | est | pde | est | est | pde | est | est | pde | est | est | pde | est | est | pde | est | est |
| 8 \times 8 | 100   | 128             | 0.04         | 0.03 | 0.76 | <0.01 | <0.01 | 1.09 | 0.04 | 0.03 | 0.76 |
| 16 \times 16 | 324   | 200             | 0.14         | 0.10 | 0.69 | <0.01 | <0.01 | 0.61 | 0.14 | 0.10 | 0.69 |
| 32 \times 32 | 1156  | 392             | 0.54         | 0.31 | 0.57 | <0.01 | <0.01 | 0.34 | 0.55 | 0.31 | 0.57 |
| 64 \times 64 | 4356  | 968             | 1.90         | 1.19 | 0.63 | 0.04 | 0.01 | 0.26 | 1.94 | 1.20 | 0.62 |
| 128 \times 128 | 16900 | 2888            | 7.49         | 4.86 | 0.65 | 0.16 | 0.14 | 0.84 | 7.66 | 4.99 | 0.65 |
| 256 \times 256 | 66564 | 9800            | 33.90        | 20.15 | 0.59 | 0.91 | 0.82 | 0.91 | 34.81 | 20.98 | 0.60 |
| 512 \times 512 | 264196 | 35912          | 194.25       | 84.70 | 0.44 | 4.10 | 5.45 | 1.33 | 198.35 | 90.15 | 0.45 |

FIGURE 7. Cells marked by error estimator with \( \psi = 20\% \) in Example 1, Case 3, \( \tilde{V}_h = S_{h}^{2,2}, \tilde{Y}_h = S_{4h}^{6,6} \otimes S_{4h}^{6,6} \).

TABLE 9. Comparison of \( I_{eff} \) for different numbers of interleaved iterations, Example 1, Case 3, \( \tilde{V}_h = S_{h}^{2,2}, \tilde{Y}_h = S_{4h}^{6,6} \otimes S_{4h}^{6,6} \).

| mesh-size | interleaved iterations |
|-----------|------------------------|
|           | 1         | 2         | 4         |
| 8 \times 8 | 11.84     | 11.28     | 11.25     |
| 16 \times 16 | 80.31    | 36.43     | 33.78     |
| 32 \times 32 | 17.36    | 12.63     | 10.11     |
| 64 \times 64 | 1.00     | 1.00      | 1.00      |
| 128 \times 128 | 1.01    | 1.01      | 1.01      |
| 256 \times 256 | 1.00    | 1.00      | 1.00      |
| 512 \times 512 | 1.00    | 1.00      | 1.00      |

Remark 4.4. The observations discussed above illustrate that one has to balance the sharpness of the majorant on the one hand, and the required computational effort on the other hand. Note that in typical practical applications, the exact solution (and thus the sharpness of the majorant) is not known. Therefore, to address the balance between sharpness and required computational effort, we propose the
following strategy. If the mesh is coarse and the total computational cost for the error estimate is moderate, we apply no (or only moderate) coarsening. When the original mesh is fine (problem size being large), we coarsen the mesh aggressively, and thereby, profit from the fast computation of the estimate. While exercising this strategy it is important to enforce the criterion (25) with $C_{\oplus} \geq 5$.

5. Numerical Examples

In this section, we present further numerical examples which illustrate the potential of the proposed a posteriori error estimator. We will present the results corresponding to the three settings discussed in Section 4, namely Case 1, Case 2 and Case 3 with the choices $K = k = 1$, $K = k = 2$, and $K = k = 4$, respectively. As in Example 1, the initial guess for $\beta$ is 0.01.

As discussed in Section 2.2, the parameter domain in all presented examples is the unit square $\hat{\Omega} = (0, 1)^2$. The mesh-sizes in the two coordinate directions, which will be presented in the tables, are determined by the respective initial meshes, which in turn, are determined by the geometry mappings. The figures plotted in black represent the computations based on the exact error, and the figures plotted in magenta represent the computations based on the majorant. The data presented in the tables is as described in the beginning of Section 4.

We first consider an example with reduced continuity $C^{p-m}$, $m > 1$.

Example 2. Sinus function in a unit square with $p = q = 4$ and $C^1$-continuity: We consider the same exact solution and the same physical domain as in Example 1, i.e.,

$$u = \sin(6\pi x) \sin(3\pi y), \quad \Omega = (0, 1)^2.$$ 

However, we now use B-splines of degree $p = q = 4$ to represent $\Omega$, and we add a triple knot at the coordinates $x = 0.5$ and $y = 0.5$. The initial knot vectors are thus given by $s = t = (0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 1, 1)$, and the geometry mapping is only $C^1$-continuous at the coordinate 0.5.

| mesh-size | $I_{\text{eff}}$ | $a_1B_1$ | $a_2B_2$ |
|-----------|-----------------|----------|----------|
| Case 1    |                 |          |          |
| 18 $\times$ 18 | 1.84        | 1.04e-03 | 9.00e-04 |
| 34 $\times$ 34 | 1.40        | 1.78e-06 | 7.23e-07 |
| 66 $\times$ 66 | 1.20        | 5.09e-09 | 1.00e-09 |
| 130 $\times$ 130 | 1.10       | 1.77e-11 | 1.74e-12 |
| 258 $\times$ 258 | 1.05       | 6.61e-14 | 3.25e-15 |
| Case 2    |                 |          |          |
| 18 $\times$ 18 | 15.43       | 7.95e-02 | 5.75e-02 |
| 34 $\times$ 34 | 6.04        | 1.14e-05 | 3.53e-05 |
| 66 $\times$ 66 | 1.76        | 7.52e-09 | 5.69e-09 |
| 130 $\times$ 130 | 1.16       | 1.87e-11 | 3.01e-12 |
| 258 $\times$ 258 | 1.04       | 6.54e-14 | 2.49e-15 |
| Case 3    |                 |          |          |
| 18 $\times$ 18 | 132.77      | 7.38e+00 | 2.76e+00 |
| 34 $\times$ 34 | 148.41      | 1.86e-02 | 9.53e-03 |
| 66 $\times$ 66 | 6.42        | 5.49e-08 | 1.21e-07 |
| 130 $\times$ 130 | 1.13       | 1.83e-11 | 2.39e-12 |
| 258 $\times$ 258 | 1.01       | 6.34e-14 | 3.78e-16 |

Table 10. Efficiency index and components of the majorant in Example 2, $\hat{V}_h = S^{4,4}_h$ with $C^1$-continuity.

The computed efficiency indices are presented in Table 10. The dashed lines, which correspond to criterion (25) being fulfilled with $C_{\oplus} = 5$, again show that more aggressive mesh-coarsening requires a finer initial mesh. By this criterion, we get a good quality of the estimate and the indicated error distribution starting from the mesh $66 \times 66$ in Case 1, and from $130 \times 130$ in Cases 2 and 3.
Figure 8. Cells marked by exact error with $\psi = 20\%$ in Example 2, $\hat{V}_h = S_h^{1,4}$ with $C^1$-continuity.

Figure 9. Cells marked by error estimator with $\psi = 20\%$ in Example 2, $\hat{V}_h = S_h^{1,4}$ with $C^1$-continuity.

We present the cells marked for refinement by the exact error in Figure 8, and the cells marked by the error estimator in Figure 9. Figure 9(a) shows that the error distribution is already captured on the mesh $18 \times 18$ in Case 1. In Case 2, we obtain a good indication of the error distribution from the mesh $66 \times 66$, i.e., before criterion (25) with $C_{\oplus} = 5$ is fulfilled. Once the error distribution is captured correctly on a certain mesh, it is also captured on all finer meshes (as in Example 1). Hence, we do not show all plots for all meshes and cases, but only the first meshes, on which the error distribution is captured correctly. Also, we omit the presentation of the timings, since the overall behavior is as in Example 1.

In the next example, we consider the case of non-trivial PDE coefficient matrix $A$.

Example 3. Let the matrix $A$ be of the form of

$$
\begin{pmatrix}
 e^{b_{11}x+b_{12}y} & 0 \\
 0 & e^{b_{21}x+b_{22}y}
\end{pmatrix},
$$

which is positive definite for $b_{ij} \in \mathbb{R}^+$, $i, j = \{1, 2\}$. This will result in the PDE operator to be of the form of

$$
e^{b_{11}x+b_{12}y} \frac{\partial^2}{\partial x^2} + e^{b_{21}x+b_{22}y} \frac{\partial^2}{\partial y^2} + b_{11}e^{b_{11}x+b_{12}y} \frac{\partial}{\partial x} + b_{22}e^{b_{21}x+b_{22}y} \frac{\partial}{\partial y}.
$$

To have this PDE operator with full generality, we take $b_{11} = 0.1$, $b_{12} = 0.8$, $b_{21} = 0.4$, $b_{22} = 0.7$. With this generality, to have a good comparison of the efficiency indices with the examples considered so far, we again choose the exact solution to be $u = \sin(6\pi x) \sin(3\pi y)$. The right hand side function is accordingly calculated and the solution has homogeneous Dirichlet boundary values.

Note that in this case the constant $C_{\Omega}$ has to be accordingly modified. For the unit square domain, and the matrix $A$ given above, its value is taken as $\frac{c_2}{\pi \sqrt{2}}$, where

$$
c_2 = \max\{e^{b_{11}x+b_{12}y}, e^{b_{21}x+b_{22}y}\}.
$$

The computed efficiency indices are presented in Table 11. The dashed lines correspond to criterion (25) being fulfilled with $C_{\oplus} = 5$. We see that the proposed estimator is robust with respect to the non-trivial PDE coefficient matrix $A$, and its performance is asymptotically similar to the case with the matrix $A$.
Cas 1
\begin{array}{|c|c|c|c|}
\hline
\text{mesh-size} & I_{\text{eff}} & a_1B_1 & a_2B_2 \\
\hline
8 \times 8 & 3.64 & 2.02e+02 & 5.73e+01 \\
16 \times 16 & 6.00 & 1.16e+01 & 7.75e+00 \\
32 \times 32 & 2.50 & 6.82e-02 & 9.65e-02 \\
64 \times 64 & 1.74 & 2.70e-03 & 1.98e-03 \\
128 \times 128 & 1.37 & 1.31e-04 & 4.83e-05 \\
256 \times 256 & - & 7.04e-06 & 1.31e-06 \\
512 \times 512 & 1.09 & 4.05e-07 & 3.77e-08 \\
\hline
\end{array}

Case 2
\begin{array}{|c|c|c|c|}
\hline
\text{mesh-size} & I_{\text{eff}} & a_1B_1 & a_2B_2 \\
\hline
8 \times 8 & 38.29 & 2.34e+04 & 5.36e+03 \\
16 \times 16 & 14.09 & 9.07e+01 & 1.61e+01 \\
32 \times 32 & 4.64 & 2.02e-01 & 3.67e-01 \\
64 \times 64 & 1.62 & 2.53e-03 & 1.55e-03 \\
128 \times 128 & 1.15 & 1.10e-04 & 1.57e-05 \\
256 \times 256 & 1.04 & 6.20e-06 & 2.16e-07 \\
512 \times 512 & 1.01 & 3.77e-07 & 3.27e-09 \\
\hline
\end{array}

Case 3
\begin{array}{|c|c|c|c|}
\hline
\text{mesh-size} & I_{\text{eff}} & a_1B_1 & a_2B_2 \\
\hline
8 \times 8 & 32.36 & 2.81e+03 & 1.77e+04 \\
16 \times 16 & 122.17 & 6.73e+03 & 1.30e+03 \\
32 \times 32 & 23.20 & 1.15e+01 & 2.71e+00 \\
64 \times 64 & 1.64 & 2.58e-03 & 1.61e-03 \\
128 \times 128 & 1.03 & 9.79e-05 & 2.41e-06 \\
256 \times 256 & 1.00 & 5.95e-06 & 7.76e-09 \\
512 \times 512 & out of memory & & & \\
\hline
\end{array}

Table 11. Efficiency index and components of the majorant in Example 3, \( V_h = S_h^{2/2} \).

\( A \) being identity \(^{†}\). Also, the presentation of the timings is again omitted since the overall behavior is as in Example 1.

In the next example, we consider a domain with a curved boundary (requiring a NURBS mapping for exact representation) and a problem whose solution has sharp peaks.

Example 4. \textbf{Domain with curved boundary:} Consider the domain of a quarter annulus. In polar coordinates, \( \Omega \) is defined by \((r, \phi) \in (1, 2) \times (0, \pi/2)\). The circular parts of the domain boundary are represented exactly by the NURBS geometry mapping of degree 2, i.e., we have \( p = q = 2 \). We set \( A = I \), and we prescribe the exact solution
\[ u = (r - 1)(r - 2)\phi(\phi - \pi/2)e^{-\alpha(r \cos \phi - 1)^2}. \]

We test our method with two values of \( \alpha \), namely,

\text{Example 4.a:} \quad \alpha = 20, \quad \text{Example 4.b:} \quad \alpha = 50.

In both examples, this function has zero Dirichlet boundary values and a peak at \( x = 1 \), the sharpness of which is determined by the value of \( \alpha \). The exact solutions are depicted in Figure 10.

In Tables 12 and 13, the efficiency index \( I_{\text{eff}} \) and the magnitudes of \( a_1B_1 \) and \( a_2B_2 \) are presented for both the cases of \( \alpha \). The dashed lines indicate the mesh-size after which criterion (25) with \( C_{\oplus} = 5 \) is fulfilled. The distribution of the marked cells is depicted in Figures 11 and 12. As before, we observe that the error distribution is represented correctly if the criterion (25) is fulfilled with \( C_{\oplus} = 5 \). When comparing Tables 12 and 13, as well as Figures 11 and 12, we notice that the more aggressive the mesh coarsening, and sharper the peak, the more refinements are needed before criterion (25) is fulfilled and the error distribution is captured correctly. Since the timings in Example 4.a and Example 4.b show the same behavior as in Example 1, both regarding assembling-time and solving-time, we omit the presentation of these numbers. Clearly, Case 3 outperforms Cases 1 and 2 in terms of cost-efficiency.

\(^{†}\)Some deviation could be attributed to the fact that we used same number of quadrature points for the evaluation of the matrices in both the cases, which is not sufficient when the PDE coefficients are of exponential form.
The computational domain is again the unit square $\Omega = (0,1)^2$, and is represented by B-splines of degree $p = q = 2$. The function $u$, which is illustrated in Figure 13, has zero Dirichlet boundary values and has two peaks at the coordinates $(0.8,0.05)$ and $(0.8,0.95)$.

In this example, we test a very basic adaptive refinement procedure using tensor-product B-splines. The discussion of isogeometric local refinement schemes is out of the scope of this paper (see Section 1 for an overview on local refinement methods).

We apply adaptive refinement based on a marking with $\psi = 25\%$, starting on an initial mesh $16 \times 16$. On the first four steps, we apply Case 1, then Case 2 on the next three steps, and thereafter Case 3.

In Figure 15, the meshes and the marked cells are shown for steps 4, 7, and 9. Clearly, the correct areas of the domain are identified and marked.
for refinement. Since the solution of the problem is sufficiently regular, the error plots in Figure 14 show that the adaptive refinement converges with the same rate as the uniform refinement, but with a better constant. However, due to the tensor-product structure of the mesh, many superfluous DOFs are inserted outside of the marked areas, which worsens the rate of convergence for given total DOFs.

Next, we consider a classical example for a posteriori error estimation and adaptive refinement studies.

Table 13. Efficiency index and components of the majorant in Example 4.b (α = 50), $\hat{V}_h = S_{h}^{2,2}$.
Figure 12. Marked cells with $\psi = 20\%$ in Example 4.b ($\alpha = 50$), $\hat{V}_h = S_h^{2.2}$.

Figure 13. Exact solution, Example 5.

| mesh-size | $l_{\text{eff}}$ | $a_1B_1$  | $a_2B_2$  | Case |
|-----------|-----------------|------------|------------|------|
| 16 × 16   | 3.77            | 9.39e-05   | 3.49e-05   | 1    |
| 25 × 26   | 2.06            | 8.62e-07   | 8.11e-07   | 1    |
| 38 × 44   | 1.69            | 4.30e-08   | 2.35e-08   | 1    |
| 64 × 74   | 1.47            | 2.79e-09   | 1.19e-09   | 1    |
| 92 × 136  | 2.82            | 8.19e-10   | 4.87e-10   | 2    |
| 184 × 256 | 1.30            | 2.05e-11   | 4.55e-12   | 2    |
| 341 × 492 | 1.11            | 1.45e-12   | 1.47e-13   | 2    |
| 652 × 934 | 1.84            | 2.55e-13   | 1.07e-13   | 3    |
| 1304 × 1868 | 1.09   | 7.40e-15   | 3.63e-16   | 3    |

Table 14. Efficiency index, components of the majorant and applied cases in Example 5, $\hat{V}_h = S_h^{2.2}$.

**Example 6. L-shaped domain:** We consider the Laplace equation

$$\Delta u = 0$$
with Dirichlet boundary conditions on the L-shaped domain $\Omega = (−1, 1)^2 \setminus [0, 1]^2$. In this example, we use a bilinear geometry mapping, i.e., $p = q = 1$. The function

$$u(r, \phi) = r^\frac{2}{3} \sin((2\phi - \pi) / 3)$$

solves (30) and is used to prescribe Dirichlet boundary conditions. The solution has a singularity at the re-entrant corner at $(0, 0)$.

We compare uniform refinement and adaptive refinement in the tensor-product setting. In this example we set $\psi = 10\%$, and to avoid the pollution near the singularity, we only use Case 1 in the majorant computations.

| mesh-size | $I_{\text{eff}}$ | $a_1B_1$ | $a_2B_2$ |
|-----------|-----------------|----------|----------|
| $16 \times 8$ | 1.1785 | 5.67e-02 | 1.71e-03 |
| $32 \times 16$ | 1.1401 | 3.44e-02 | 8.98e-04 |
| $64 \times 32$ | 1.1116 | 2.09e-02 | 4.72e-04 |
| $128 \times 64$ | 1.0898 | 1.28e-02 | 2.49e-04 |
| $256 \times 128$ | 1.0729 | 7.87e-03 | 1.32e-04 |
| $512 \times 256$ | 1.0593 | 4.86e-03 | 7.01e-05 |
| $1024 \times 512$ | 1.0485 | 3.01e-03 | 3.73e-05 |

Table 15. Efficiency index and components of the majorant in Example 6, Case 1, uniform refinement, $\hat{V}_h = S_h^{1,1}$, $\hat{Y}_h = S_h^{2,2} \otimes S_h^{2,2}$.

| mesh-size | $I_{\text{eff}}$ | $a_1B_1$ | $a_2B_2$ |
|-----------|-----------------|----------|----------|
| $16 \times 8$ | 1.1785 | 5.67e-02 | 1.71e-03 |
| $22 \times 11$ | 1.1839 | 2.68e-02 | 8.66e-04 |
| $30 \times 16$ | 1.1749 | 1.37e-02 | 4.32e-04 |
| $39 \times 23$ | 1.1622 | 7.22e-03 | 2.20e-04 |
| $55 \times 37$ | 1.1625 | 3.52e-03 | 1.05e-04 |
| $87 \times 60$ | 1.1635 | 1.74e-03 | 5.41e-05 |
| $133 \times 101$ | 1.1525 | 9.25e-04 | 2.69e-05 |

Table 16. Efficiency index and components of the majorant in Example 6, Case 1, adaptive refinement, $\hat{V}_h = S_h^{1,1}$, $\hat{Y}_h = S_h^{2,2} \otimes S_h^{2,2}$. 
The magnitudes of the components $a_1 B_1$ and $a_2 B_2$, which are presented in Table 15 for uniform refinement, and in Table 16 for adaptive refinement, show that the criterion (25) with $C_{id} = 5$ is fulfilled on all the considered meshes.

The error plots presented in Figure 16 show the expected faster convergence on the adaptively refined mesh, even though we are only using tensor-product splines. In Figure 17, meshes and marked cells are shown for steps 2 and 6, again indicating that the error indicator correctly identifies the corner singularity.

In our final example, we consider an advection dominated advection diffusion equation to see the performance of the estimator for sharp boundary layers.

**Example 7. Advection dominated advection diffusion equation:** We consider the advection diffusion equation with Dirichlet boundary conditions on the unit square $\Omega = (0,1)^2$, with $p = q = 2$, i.e.,

$$-\kappa \Delta u + b \cdot \nabla u = 0 \quad \text{in } \Omega, \quad u = u_D \quad \text{on } \partial \Omega,$$

where

$$\kappa = 10^{-6}, \quad b = (\cos \frac{\pi}{3}, \sin \frac{\pi}{3})^T, \quad u_D = \begin{cases} 1, \text{ if } y = 0 \\ 0, \text{ else} \end{cases}.$$

FIGURE 15. Meshes and marked cells in Example 5. Cases as in Table 14, $\psi = 25\%$, $\hat{V}_h = S_h^{2,2}$. 

23
We use the standard streamline upwind Petrov-Galerkin (SUPG) scheme for the stabilization. The stabilization parameter $\tau$ is set to $\tau(Q) = h_b(Q)/2|b|$, where $h_b(Q)$ is the diameter of the cell $Q$ in direction of the flow $b$, and $|b|$ is the magnitude of the vector $b$. For advection diffusion problems, we have to adapt the majorant. Since the principle method is the same, we refer the reader to [36, Section 4.3.1] for a detailed discussion. In this special case, where $A = \kappa I$ with $\kappa \ll |b|$, and with constant velocity vector $b$, the majorant $M^2_{\text{adv}}$ for the advection diffusion problem is given by

$$M^2_{\text{adv}} = (1 + \beta)\|A\nabla u_h - y\|^2_A + (1 + \frac{1}{\beta})C^2_{\text{adv}} \|\text{div} y + f - b \cdot \nabla u_h\|^2.$$
The strong advection and the discontinuous boundary conditions result in sharp layers. In Figure 18(a), the expected positions of the layers are indicated by dashed lines.

| mesh-size | \(a_1B_1\) | \(a_2B_2\) |
|-----------|-------------|-------------|
| 16 \(\times\) 16 | 1.98e-07 | 3.18e-10 |
| 64 \(\times\) 64 | 6.45e-07 | 1.15e-09 |
| 256 \(\times\) 256 | 2.28e-06 | 4.33e-09 |

Case 1

| mesh-size | \(a_1B_1\) | \(a_2B_2\) |
|-----------|-------------|-------------|
| 16 \(\times\) 16 | 1.83e-06 | 9.66e-10 |
| 64 \(\times\) 64 | 6.50e-06 | 3.65e-09 |
| 256 \(\times\) 256 | 1.86e-05 | 1.24e-08 |

Case 2

| mesh-size | \(a_1B_1\) | \(a_2B_2\) |
|-----------|-------------|-------------|
| 16 \(\times\) 16 | 3.24e-06 | 1.29e-09 |
| 64 \(\times\) 64 | 2.07e-05 | 6.52e-09 |
| 256 \(\times\) 256 | 6.86e-05 | 2.38e-08 |

Case 3

| mesh-size | \(a_1B_1\) | \(a_2B_2\) |
|-----------|-------------|-------------|
| 16 \(\times\) 16 | 1.83e-06 | 9.66e-10 |
| 64 \(\times\) 64 | 6.45e-07 | 1.15e-09 |
| 256 \(\times\) 256 | 2.28e-06 | 4.33e-09 |

TABLE 17. Comparison of terms \(a_1B_1\) and \(a_2B_2\) in Example 7, \(\hat{V}_h = S_h^{2,2}\).

The magnitudes of \(a_1B_1\) and \(a_2B_2\) presented in Table 17 indicate that the criterion (25) with \(C^\Theta = 5\) is fulfilled on all the considered meshes. The distribution of the marked cells presented in Figures 18(b)
and 18(c) provides the visual indication that the expected layers are accurately detected by the error estimator.

For this example, the timings presented in Table 18 show that, unlike the previous examples, assembling and solving the system for the estimator is faster than for the original problem not only in Case 3 (less than 1/2 of the original cost), but also in Case 2 (about 2/3 of the original cost). This is due to the SUPG stabilization which is costlier than computing the additional term $b \cdot \nabla u_h$ in the majorant $M_{\Omega, \text{adv}}^2$.

6. Conclusion

We have proposed a method for cost-efficient computation of guaranteed and sharp posteriori error estimates in IGA. This method relies only on the use of NURBS basis functions, without the need for constructing complicated basis functions in $H(\Omega, \text{div})$. We have discussed different settings which allow the user to balance the sharpness of the bound and accurate error distribution on the one hand, and the required computational cost of the error estimator on the other hand (see Remark 4.4). For the presented settings, we have derived a quality criterion, which is easy to check numerically and which indicates whether the computed estimate is sharp or not (see Remark 4.3). Two properties of NURBS basis functions are exploited. Firstly, the basis functions are, in general, automatically in $H(\Omega, \text{div})$ due to their high smoothness. Without this property, we could not use NURBS of equal degree for both components as basis functions for the minimizing function $v_h$. Secondly, increasing the polynomial degree of NURBS basis functions adds only few DOFs. This fact is necessary for keeping the computational cost of the majorant as low as possible (see Remark 4.2). It is important to note that none of these properties are possible in FEM discretizations based on $C^0$ basis functions.

Apart from the topical interest of a posteriori error estimation and adaptivity, the presented method should also be of interest in parametrization of computational domain. For example, for $r$- refinement in IGA, i.e., to optimize the placement of inner control points, the proposed estimator can be used to accurately detect the regions with large error and then use the optimization algorithm to reposition the control points. Such a problem of $r$- refinement has been studied in [44, 45].

Finally, in this paper, we have only considered tensor-product NURBS discretizations. While the extension of this method to locally refined isogeometric discretizations and also to three dimensions is, in theory, straightforward, the actual performance and efficiency of the error estimator on such methods and meshes is the subject of further studies.

Acknowledgements

The authors are grateful to Prof. Sergey I. Repin, V.A. Steklov Institute of Mathematics, St. Petersburg, for helpful discussions. Authors are also thankful to unknown referees for their helpful comments. The support from the Austrian Science Fund (FWF) through the project P21516-N18 is gratefully acknowledged.

References

[1] M. Ainsworth and J.T. Oden. A Posteriori Error Estimation in Finite Element Analysis. Wiley Interscience, 2000.
[2] R.E. Bank and R.K. Smith. A posteriori error estimates based on hierarchical bases. SIAM J. Numer. Anal., 30(4):921–935, 1993.
[3] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for $h$-refined meshes. Math. Models Methods Appl. Sci., 16(7):1031–1090, 2006.
[4] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. Comput. Methods Appl. Mech. Engrg., 199(5-8):229–263, 2010.
[5] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput. Methods Appl. Mech. Engrg., 197(1-4):173–201, 2007.
[6] Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. Comput. Mech., 43:3–37, 2008.
[7] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for $h$- $p$- $k$- refinement in isogeometric analysis. Numer. Math., 118:271–305, 2011.
[8] D. Boffi, F. Brezzi, and M. Fortin. Mixed Finite Element Methods and Applications. Springer, Berlin, 2013.
[9] A. Buffa, C. de Falco, and G. Sangalli. IsoGeometric Analysis: Stable elements for the 2D Stokes equation. Int. J. Numer. Meth. Fluids, 65(11-12):1407–1422, 2011.
[10] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. Isogeometric discrete differential forms in three dimensions. SIAM J. Numer. Anal., 49(2):818–844, 2011.
[44] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Comput. Methods Appl. Mech. Engrg.*, 200(23-24):2021–2031, 2011.

[45] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design*, 45:812–821, 2013.

(Stefan K. Kleiss) JOHANNES KEPLER UNIVERSITY, ALTENBERGER STRASSE 69, A-4040 LINZ, AUSTRIA
E-mail address: stefan.kleiss@jku.at

(Satyendra K. Tomar) DORNACHER STRASSE 6/21, A-4040 LINZ, AUSTRIA
E-mail address, Corresponding author: tomar.sk.prof@gmail.com