Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

# Functional approach to the error control in adaptive IgA schemes for elliptic boundary value problems

S. Matculevich

RICAM-Report 2017-24

# Functional approach to the error control in adaptive IgA schemes for elliptic boundary value problems

S. Matculevich*

July 11, 2017

**Abstract**

This work presents a numerical study of *functional type* a posteriori error estimates for *IgA approximation schemes* in the context of elliptic boundary-value problems. Along with the detailed discussion of the most crucial properties of such estimates, we present the algorithm of a reliable solution approximation together with the scheme of efficient a posteriori error bound generation that is based on solving an auxiliary problem with respect to an introduced vector-valued variable. In this approach, we take advantage of B-(THB-)spline's high smoothness for the auxiliary vector function reconstruction, which, at the same time, allows to use much coarser meshes and decrease the number of unknowns substantially. The most representative numerical results, obtained during a systematic testing of error estimates, are presented in the second part of the paper. The efficiency of the obtained error bounds is analysed from both the error estimation (indication) and the computational expenses points of view. Several examples illustrate that functional error estimates (alternatively referred to as the *majorants* and *minorants* of deviation from an exact solution) perform a much sharper error control than, for instance, residual-based error estimates. Simultaneously, assembling and solving routines for an auxiliary variable reconstruction which generate the majorant of an error can be executed several times faster than the routines for a primal unknown.

## 1 Introduction

The investigation of effective adaptive refinement procedures has recently become an active area of research in the context of fast and efficient solvers for isogeometric analysis (IgA) [24, 25]. Scheme adaptivity is naturally linked with reliable and quantitatively efficient a posteriori error estimation tools. The latter ones are expected to identify the areas of considered computational domain with relatively high discretisation errors and provide a fully automated refinement strategy in order to reach a desired accuracy level for an approximated solution.

Due to a tensor-product setting of IgA splines, mesh refinement has global effects, which include a large percentage of superfluous control points in data analysis, unwanted ripples on the surface, etc. These issues produce certain challenges at the design stage as well as complications in handling big amounts of data, and therefore naturally trigger the development of local refinement strategies for IgA. At the moment, four different IgA approaches for adaptive mesh refinement are known, i.e., T-splines, hierarchical splines, PHT-splines, and LR splines.

The localised splines of the first type, *T-splines*, were introduced in [65, 64] and analysed in [1, 4, 62, 63]. They are based on the T-junctions that allow to eliminate redundant control points from NURBS model. The thorough study confirmed that this approach generates an efficient local refinement algorithm for *analysis-suitable* T-splines [36] and avoids the excessive propagation of control points. In [3, 9], it was proposed to characterise such splines as dual-compatible T-splines, and in [43] a refinement strategy with linear complexity was described for the bivariate case.

The alternative approach that implies the local control of refinement is based on *hierarchical B-splines* (HB-splines), where in a selected refinement region basis functions are replaced with the finer ones of the same type. The procedure of designing a basis for the hierarchical spline space was suggested in [17, 30, 23] and extended in [69, 16, 61]. Such construction guarantees the linear independence of the basis and provides nested approximation spaces. However, since the partition of unity is not preserved for these splines, *truncated hierarchical B-splines* (THB-splines) have been developed (see [21]). In addition to inherited from HB-splines good stability and approximation properties [19, 66], THB-splines form a convex partition of unity, and therefore, are suitable for the application in CAD. Various usage of THB-spline for arbitrary topologies can be found in, e.g., [71, 75, 76].

*RICAM Linz, Johann Radon Institute, AT-4040 Linz, svetlana.matculevich@ricam.oeaw.ac.at

The locally defined splines of the third type, namely, *polynomial splines over hierarchical T-meshes*, are constructed for the entire space of piecewise polynomials with given smoothness on the subdivision of considered domain. Corresponding application can be found in [46, 70]. However, in this case, one must assume the reduced regularity of basis [11] or fulfil a certain constraint on admissible mesh configuration [73].

Finally, *locally refined splines* (LR-splines) rely on the idea of splitting basis functions. This technique achieves localisation but creates difficulties with linear independence [13], which has been studied in [6, 7]. The application of such type of splines has been thoroughly investigated in [13]. In [26], one can find the summary of a detailed comparison of (T)HB-splines and LR splines with respect to sparsity and condition numbers. The study concludes that even though LR splines have smaller support than THB-splines, the numerical experiments did not reveal any significant advantages of the first ones with respect to the sparsity patterns or condition numbers of mass and stiffness matrices.

The refinement tools of IgA mentioned above were combined with various a posteriori error estimation techniques. For instance, the a posteriori error estimates based on hierarchical splines were investigated in [14, 69]. In [27, 70, 8, 31], authors used the residual-based a posteriori error estimates and their modifications in order to construct mesh refinement algorithms. The latter ones, in particular, require the computation of constants related to the Clement-type interpolation operators, which are mesh-dependent and often difficult to compute for general element shapes. Moreover, these constants must be re-evaluated every time a new mesh is generated. The goal-oriented error estimators, which are rather naturally adapted to practical applications, have been lately introduced for IgA approximations and can be found in [68, 10, 32, 33].

In the current work, the terms *error estimate* and *error indicator* distinguish from each other. The first one is considered as the total upper (or lower) bound of true energy error. These are very important characteristics related to the approximate solution since they can be used to judge whether obtained data are reliable or not. In order to locate the areas of discretised domain that have the highest errors in the approximation, a quantitively sharp *error indicator* is required. The methods of a posteriori error estimation listed above are rather error indicators in this terminology and indeed were successfully used for mimicking the approximation error distribution. However, their use in the error control, i.e., a reliable estimation of the accuracy of obtained data, is rather heuristic in nature.

Below we investigate a different *functional* method providing fully guaranteed error estimates, the upper (and lower) bounds of the exact error in the various weighted norms equivalent to the global energy norm. These estimates include only global constants (independent of the mesh characteristic $h$) and are valid for any approximation from the the admissible functional space. One of the most advantageous properties of functional error estimates is their independence of the numerical method used for calculating approximate solutions. The strongest assumption about approximations is that they are conforming in the sense that they belong to a certain natural Sobolev space suited for the problem. It is important to emphasise that this is still a rather weak assumption and that no further restrictions, such as Galerkin orthogonality, are needed.

Functional error estimates were initially introduced in [58, 59] and later applied to different mathematical models summarised in monographs [45, 51, 37]. They provide guaranteed, sharp, and fully computable upper and lower bounds of errors. A pioneering study on the combination of functional type error estimates with the IgA approximations generated by tensor-product splines is presented in [29] for elliptic boundary value problems (BVP). The extensive numerical tests presented in this work confirmed that majorant produces not only good upper bounds of the error but also a quantitatively sharp error indicator. Moreover, the authors suggest the heuristic algorithm that allows to use the smoothness of B-splines for a rather efficient calculation of true error upper bound.

The current work further extends the ideas used in [29] for B-splines (NURBS) and combines the functional approach to the error control with THB-splines. Moreover, our focus is concentrated not only on the qualitative and quantitative performance of error estimates but also on the required computation time for their reconstruction. The systematic analysis of majorant's numerical properties is based on a collection of extensive tests performed on the problems of different complexity. For the error control implemented with the help of tensor-structured B-splines (NURBS) and THB-splines, we manage to obtain an impressive speed-up in majorant reconstruction by exploiting high smoothness of B-splines to our advantage. However, for the problems with sharp local changes or various singularities in the solution, the THB-splines implementation in G+Smo restricts the performance speed-up when it comes to solving the optimal system for the error majorant as well as for its element-wise evaluation. We restrict this study only to the domains modelled by a single patch, which provides at least $C^1$-continuity of the approximate solutions inside the patch. However, the application of studied majorants can be extended to a multi-patch domain, since the error estimates for stationary problems are flexible enough to handle fully non-conforming approximations (this issue has been in details addressed in [35, 67, 60]).

The error control for the problems defined on domains of complicated shapes induces another issue related to the estimation of Friedrichs' constant used by functional error estimates not only as the weight but also as the geometrical characteristic of the considered problem. When such domains are concerned, one can perform their decomposition into a collection of non-overlapping convex sub-domains, such that the global constant can be replaced by constants in local embedding inequalities (Poincaré and Poincaré-type inequalities [49, 50]). The reliable estimates of these local constants can be found in [47, 2, 44, 42]. The detxfailed derivation of functional error estimates exploiting these ideas is discussed in [51, 53] for the elliptic BVP and in [40, 39, 41] for the parabolic initial boundary value problem (I-BVP). In order to use this method, one needs to impose a crucial restriction on the multi-patch configuration, namely, each patch must be a convex sub-domain. Since in the IgA framework patches are treated as mappings from the reference domain $\widehat{\Omega} = (0,1)^d$, the estimation of local constants is reduced to the analysis of the IgA mapping and calculating the corresponding constant for $\widehat{\Omega}$.

The paper proceeds with the following structure. Section 2 formulates the general statement of the considered problem and recalls the definition of functional error estimates and their main properties in the context of reliable energy error estimation and efficient error-distribution indication. The next section serves as an overview of IgA techniques used in the current work, i.e., B-splines, NURBS, and THB-splines. In Section 4, we focus on the algorithms and details of the functional error estimates integration into the IgA framework. Last but not least, Section 5 presents the systematic selection of most relevant numerical examples and obtained results that illustrate numerical properties of studied error estimates and indicators.

## 2 Functional approach to the error control

In this section, we present a model problem, recall the well-posedness results for linear parabolic PDEs, which have been thoroughly studied in [34, 74, 72]. We also introduce a functional a posteriori error estimate for the stated model and discuss its crucial properties.

Let $\Omega \subset \mathbb{R}^d$, $d = \{2, 3\}$, be a bounded domain with Lipschitz boundary $\Gamma = \partial\Omega$. The general elliptic BVP is formulated as the system

$$-\mathrm{div}_x \boldsymbol{p} = f, \quad \text{in} \quad \Omega, \tag{1}$$

$$\boldsymbol{p} = A\nabla_x u, \quad \text{in} \quad \Omega, \tag{2}$$

$$u = 0, \quad \text{on} \quad \Gamma, \tag{3}$$

where $f \in L^2(\Omega)$. We assume that the operator $A$ is symmetric and satisfies the condition of uniform ellipticity for almost all (a.a.) $x \in \Omega$, which reads

$$\underline{\nu}_A |\xi|^2 \le A(x)\,\xi \cdot \xi \le \overline{\nu}_A |\xi|^2, \quad \text{for all} \quad \xi \in \mathbb{R}^d, \tag{4}$$

with $0 < \underline{\nu}_A \le \overline{\nu}_A < \infty$. Throughout the paper, the following notation for the norms is used:

$$\| \boldsymbol{\tau} \|^2_{A,\Omega} := (A\boldsymbol{\tau}, \boldsymbol{\tau})_\Omega, \quad \| \boldsymbol{\tau} \|^2_{A^{-1},\Omega} := (A^{-1}\boldsymbol{\tau}, \boldsymbol{\tau})_\Omega, \quad \text{for all} \quad \boldsymbol{\tau} \in [L^2(\Omega)]^d,$$

where $(A\boldsymbol{u}, \boldsymbol{v})_\Omega := \int_\Omega A\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}x$ stands for a weighted $L^2$ scalar-product for all $\boldsymbol{u}, \boldsymbol{v} \in [L^2(\Omega)]^{\mathrm{d}}$. After multiplying (1) by the test function

$$\eta \in H^1_0(\Omega) := \left\{ u \in L^2(\Omega) \;\middle|\; \nabla_x u \in L^2(\Omega),\ u|_\Gamma = 0 \right\},$$

we arrive at the standard generalised formulation of (1)–(3): find $u \in H^1_0(\Omega)$ satisfying the integral identity

$$a(u, \eta) := (A\nabla_x u, \nabla_x \eta)_\Omega = (f, \eta)_\Omega =: l(\eta), \quad \forall \eta \in H^1_0(\Omega). \tag{5}$$

According to [34], generalised problem (5) has a unique solution in $H^1_0(\Omega)$ provided that $f \in L^2(\Omega)$ and condition (4) holds.

We consider the functional error estimate, which provides a guaranteed upper bound of the distance $e := u - v$ between the generalised solution $u$ of BVP (5) and any function $v \in H^1_0(\Omega)$. It is important to emphasise that the suggested functional approach to error estimates derivation is universal for any numerical method used to discretise bilinear form (5). This fact makes it rather unique in comparison with alternative approaches, which are always tailored to the discretised version of the identity $a(u, \eta) = l(\eta)$. Later on, the considered $v$ is generated numerically, and the distance to $u$ is evaluated in terms of the total energy norm

$$|||e|||^2_\Omega := \| \nabla_x e \|^2_{A,\Omega} \tag{6}$$

as well as its element-wise contributions $\| \nabla_x e \|^2_{A,K}$, such that

$$|||e|||^2_\Omega := \sum_{K \in \mathcal{K}_h} \| \nabla_x e \|^2_{A,K}.$$

Here, $K$ represents the elements of the mesh $\mathcal{K}_h$ introduced on $\Omega$. Hence, besides being the guaranteed upper bound of total error (6), the majorant provides a quantitatively sharp indicator of local error distribution.

To derive the upper bound, we first need to transform (5) by subtracting $a(v, \eta)$ from left- (LHS) and right-hand side (RHS) and setting $\eta = e$, by that obtaining the error identity

$$|||e|||^2_\Omega = \big(f, e\big)_\Omega - \big(A\nabla_x v, \nabla_x e\big)_\Omega. \tag{7}$$

The main idea of functional approach is the introduction of an auxiliary vector-valued variable

$$\boldsymbol{y} \in H(\Omega, \mathrm{div}_x) := \Big\{ \boldsymbol{y} \in \big[L^2(\Omega)\big]^{\mathrm{d}} \;\big|\; \mathrm{div}_x \boldsymbol{y} \in L^2(\Omega) \Big\}$$

satisfying

$$(\mathrm{div}_x \boldsymbol{y}, v)_\Omega + (\boldsymbol{y}, \nabla_x v)_\Omega = 0. \tag{8}$$

In further calculations, the above-introduced variable allows additional optimisation of the majorant, whereas, for instance, the residual error estimates do not have this additional freedom in improving its values. Next, we add the identity (8) to the RHS of (7), which yields

$$|||e|||^2_\Omega = \big(f + \mathrm{div}_x \boldsymbol{y}, e\big)_\Omega + \big(\boldsymbol{y} - A\nabla_x v, \nabla_x e\big)_\Omega. \tag{9}$$

The *equilibrated* and *dual residual-functionals* obtained in the RHS of (9) mimic equations (1) and (2), respectively, and are denoted by

$$\mathbf{r}_{\mathrm{eq}}(v, \boldsymbol{y}) := f + \mathrm{div}_x \boldsymbol{y} \quad \text{and} \quad \mathbf{r}_{\mathrm{d}}(v, \boldsymbol{y}) := \boldsymbol{y} - A\nabla_x v, \tag{10}$$

respectively.

**Theorem 1** *(a) For any functions $v \in H^1_0(\Omega)$ and $\boldsymbol{y} \in H(\Omega, \mathrm{div}_x)$, we have the estimate*

$$|||e|||^2_\Omega \le \overline{\mathrm{M}}^2(v, \boldsymbol{y}; \beta) := (1+\beta) \| \mathbf{r}_{\mathrm{d}} \|^2_{A^{-1},\Omega} + (1 + \tfrac{1}{\beta}) \tfrac{C^2_{\mathrm{F}\Omega}}{\underline{\nu}_A} \big\| \mathbf{r}_{\mathrm{eq}} \big\|^2_\Omega, \tag{11}$$

*where the residuals $\mathbf{r}_{\mathrm{d}}$ and $\mathbf{r}_{\mathrm{eq}}$ are defined in (10), $\beta$ is a positive parameter, and $C_{\mathrm{F}\Omega}$ is the constant in the Friedrichs inequality [18]*

$$\|v\|_\Omega \le C_{\mathrm{F}\Omega} \|\nabla_x v\|_\Omega, \quad \forall v \in H^1_0(\Omega).$$

*(b) For $\beta > 0$, the variational problem*

$$\inf_{\substack{v \in H^1_0(\Omega) \\ \boldsymbol{y} \in H(\Omega, \mathrm{div}_x)}} \overline{\mathrm{M}}(v, \boldsymbol{y}; \beta)$$

*has a solution (with the corresponding zero-value for the functional), and its minimum is attained if and only if $v = u$ and $\boldsymbol{y} = A\nabla_x u$.*

**Proof:** For the detailed proof of this theorem we refer the reader to [52, Section 3.2]. □

Remarks below summarise several essential properties of the error estimate derived in Theorem 1.

**Remark 1** Each term on the RHS of (11) serves as the upper bound of the error that might occur in equations (1) and (2). The positive weight $\beta$ can be selected optimally in order to achieve the best value of the majorant. The constant $C_{\mathrm{F}\Omega}$ acts as a geometric characteristic for the considered domain $\Omega$ (unlike, for instance, in least-square methods, where the weights are some constants). From the author's point of view this constant is essential and cannot be excluded since it scales proportionally to the diameter of the considered $\Omega$. Moreover, in order to guarantee the reliability of $\overline{\mathrm{M}}(v, \boldsymbol{y}; \beta)$, the constant $C_{\mathrm{F}\Omega}$ must be estimated from above in a reliable way. Since in practice the term $\| \mathbf{r}_{\mathrm{eq}} \|^2_\Omega$ is rather small compared to the dominating term $\| \mathbf{r}_{\mathrm{d}} \|^2_\Omega$, the Friedrichs constant can be replaced by some penalty constant $C \ge C_{\mathrm{F}\Omega}$ (even thought it might affect the ratio of the majorant to the error). In what follows, to characterise the efficiency of (11), we use the quantity $I_{\mathrm{eff}}(\overline{\mathrm{M}}) := \overline{\mathrm{M}} / |||e|||_\Omega$ that measures the above-mentioned gap between $\overline{\mathrm{M}}(v, \boldsymbol{y}; \beta)$ and $|||e|||_\Omega$.

**Remark 2** The functional $\overline{\mathrm{M}}(v, \boldsymbol{y}; \beta)$ generates the upper bound of the error for any auxiliary $\boldsymbol{y} \in H(\Omega, \mathrm{div}_x)$ and $\beta > 0$, therefore the choice of $\boldsymbol{y}$ might vary. The first and most straightforward way to select this variable is to set $\boldsymbol{y} = G(A\,\nabla_x v)$, where $G : L^2(\Omega, \mathbb{R}^d) \to H(\Omega, \mathrm{div}_x)$ is a certain gradient-averaging operator. The advantage of using the IgA framework is that for the splines of the degree $p \geq 2$ an obtained $v$ is a $C^1$-continuous function and $\nabla_x v$ is already in $H(\Omega, \mathrm{div}_x)$, therefore no additional post-processing is needed. On the other hand, due to the quadratic structure of the majorant, it is rather obvious that the optimal error estimate value is achieved at $y = A\nabla_x u$, i.e.,

$$\|\nabla_x e\|^2_{A,\Omega} \leq \overline{\mathrm{M}}(v, \nabla_x u) = (1+\beta)\,\|\nabla_x e\|^2_{A,\Omega} + C^2_{\mathrm{F}\Omega}\,(1+\tfrac{1}{\beta})\,\|f + \mathrm{div}_x(\nabla_x u)\|^2_\Omega = (1+\beta)\,\|\nabla_x e\|^2_{A,\Omega}. \tag{12}$$

From (12), it is easy to see that if the auxiliary $\boldsymbol{y}$ is chosen optimally and $\beta$ is set to zero (in the RHS of (12)), there is no gap between $\overline{\mathrm{M}}$ and $\|\nabla_x e\|^2_{A,\Omega}$.

One of the methods providing the efficient reconstruction of both dual and primal variables is a mixed (primal or dual) method. It generates an efficient approximation of the pair $(v, \boldsymbol{y}) \in W := H^1_0(\Omega) \times H(\Omega, \mathrm{div}_x)$ that can be straightforwardly substituted into the majorant $\overline{\mathrm{M}}(v, \boldsymbol{y})$. Moreover, if the error is measured in terms of the combined norm, i.e., including the norm of the error in primal and in dual variables

$$|\!|\!|(u, \boldsymbol{p}) - (v, \boldsymbol{y})|\!|\!|_W := (\|\nabla_x(u-v)\|^2_\Omega + \|\boldsymbol{p} - \boldsymbol{y}\|^2_\Omega + \|\mathrm{div}_x(\boldsymbol{p} - \boldsymbol{y})\|^2_\Omega)^{1/2},$$

it is controlled by the residuals of the majorant in the following form

$$\tfrac{1}{\sqrt{3}}\,(\|\,\mathbf{r}_{\mathrm{d}}\,\|_{A^{-1},\Omega} + \tfrac{1}{\sqrt{\underline{\nu}_A}}\,\|\,\mathbf{r}_{\mathrm{eq}}\,\|_\Omega) \leq |\!|\!|(u, \boldsymbol{p}) - (v, \boldsymbol{y})|\!|\!|_W \leq \|\,\mathbf{r}_{\mathrm{d}}\,\|_{A^{-1},\Omega} + (1 + 2\,\tfrac{C^2_{\mathrm{F}\Omega}}{\underline{\nu}_A})^{1/2}\|\,\mathbf{r}_{\mathrm{eq}}\,\|_\Omega.$$

We note that the ratio between the majorant $\|\,\mathbf{r}_{\mathrm{d}}\,\|_{A^{-1},\Omega} + \frac{1}{\sqrt{\underline{\nu}_A}}\|\,\mathbf{r}_{\mathrm{eq}}\,\|_\Omega$ (that does not include any constants) and the error $|\!|\!|(u, \boldsymbol{p}) - (v, \boldsymbol{y})|\!|\!|_W$ is controlled by $\sqrt{3}$, which proves the robustness of such an error estimate. The series of work on this subject, e.g., [55, 57, 56], has confirmed the efficiency of such an approach.

The alternative approach that provides an accurate $\boldsymbol{y}$-reconstruction follows from the minimisation problem

$$\{\boldsymbol{y}_{\min}, \beta_{\min}\} := \arg \inf_{\beta > 0} \inf_{\boldsymbol{y} \in H(\Omega, \mathrm{div}_x)} \overline{\mathrm{M}}(v, \boldsymbol{y}; \beta). \tag{13}$$

The latter one is equivalent to the variation formulation for the optimal $\boldsymbol{y}_{\min}$, i.e.,

$$\tfrac{C^2_{\mathrm{F}\Omega}}{\beta_{\min}}\,(\mathrm{div}_x \boldsymbol{y}_{\min}, \mathrm{div}_x \boldsymbol{w})_\Omega + (A^{-1}\,\boldsymbol{y}_{\min}, \boldsymbol{w})_\Omega = -\tfrac{C^2_{\mathrm{F}\Omega}}{\beta_{\min}}\big(f, \mathrm{div}_x \boldsymbol{w}\big)_\Omega + (A\nabla_x v, \boldsymbol{w})_\Omega, \quad \forall \boldsymbol{w} \in H(\Omega, \mathrm{div}_x),$$

where the optimal $\beta$ is given by $\beta_{\min} := \frac{C_{\mathrm{F}\Omega}\,\overline{\mathrm{m}}_{\mathrm{f}}}{\overline{\mathrm{m}}_{\mathrm{d}}}$ with

$$\overline{\mathrm{m}}_{\mathrm{f}} := \big\|\,\mathbf{r}_{\mathrm{eq}}\,\big\|_\Omega \quad \text{and} \quad \overline{\mathrm{m}}_{\mathrm{d}} := \|\,\mathbf{r}_{\mathrm{d}}\,\|_{A^{-1},\Omega}. \tag{14}$$

In this work, using the IgA approximation schemes' setting, we apply the second method of the efficient $\boldsymbol{y}$-reconstruction described in detail in Section 4. To compare the performance of $\overline{\mathrm{M}}$ with alternative error estimates we use the standard residual error estimator (applied, e.g., in [20])

$$\overline{\eta}^2 = \sum_{K \in \mathcal{K}_h} \overline{\eta}^2_K, \quad \overline{\eta}^2_K := h^2_K\,\|f + \mathrm{div}_x(A\nabla u_h)\|^2_{L^2(K)}, \tag{15}$$

where $h_K$ denotes the diameter of cell $K$, and $u_h$ denotes the approximation reconstructed by IgA scheme. The term measuring the jumps across the element edges, which is usually included into residual error estimates, vanishes in (15) due to the properties of $u_h$ produced by the IgA schemes. It is provided in the G+Smo package and can be accessed by using the class available from the G+Smo library [38, `stable/src/gsErrEstPoissonResidual.h`].

# 3 IgA overview: B-splines, NURBS, and THB-splines

For the consistency of exposition, we first give an overview of the general IgA framework, the definitions of B-splines, NURBS, and THB-splines, their use in the geometrical representation of the computational domain $\Omega$ and in the construction of IgA discretisation spaces.

Let $p \geq 2$ denote the degree of polynomials used for the IgA approximations, and let $n$ be the number of basis functions used to construct a $B$-spline curve. The *knot-vector* in $\mathbb{R}$ is a non-decreasing set of coordinates in

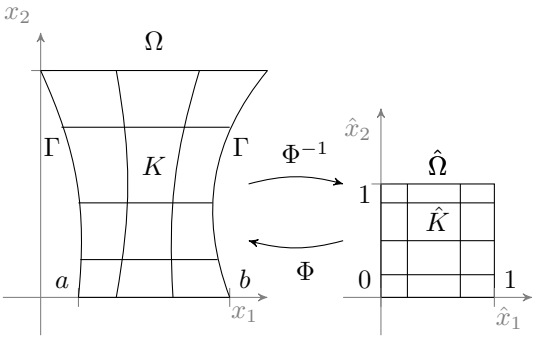



Figure 1: Mapping of $\widehat{\Omega}$ to $\Omega$.

the parameter domain, written as $\Xi = \{\xi_1, ..., \xi_{n+p+1}\}$, $\xi_i \in \mathbb{R}$, where $\xi_1 = 0$ and $\xi_{n+p+1} = 1$. The knots can be repeated, and the multiplicity of the $i$-th knot is indicated by $m_i$. Throughout the paper, we consider only open knot vectors, i.e., $m_1 = m_{n+p+1} = p + 1$. For the one-dimensional parametric domain $\widehat{\Omega} := (0, 1)$, $\widehat{\mathcal{K}}_h := \{\widehat{K}\}$ denotes a locally quasi-uniform mesh, where each element $\widehat{K} \in \widehat{\mathcal{K}}_h$ is constructed by distinct neighbouring knots. The global size of $\widehat{\mathcal{K}}_h$ is denoted by

$$\hat{h} := \max_{\widehat{K} \in \widehat{\mathcal{K}}_h} \{\hat{h}_{\widehat{K}}\}, \quad \text{where} \quad \hat{h}_{\widehat{K}} := \operatorname{diam}(\widehat{K}).$$

Henceforth, we assume locally quasi-uniform meshes, i.e., the ratio of two neighbouring elements $\widehat{K}_i$ and $\widehat{K}_j$ satisfies the inequality

$$c_1 \leq \frac{\hat{h}_{\widehat{K}_i}}{\hat{h}_{\widehat{K}_j}} \leq c_2, \quad \text{where} \quad c_1, c_2 > 0.$$

The *univariate B-spline basis functions* $\widehat{B}_{i,p} : \widehat{\Omega} \to \mathbb{R}$ are defined by means of the Cox-de Boor recursion formula

$$\widehat{B}_{i,p}(\xi) := \tfrac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \widehat{B}_{i,p-1}(\xi) + \tfrac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \widehat{B}_{i+1,p-1}(\xi), \quad \widehat{B}_{i,0}(\xi) := \begin{cases} 1 & \text{if} \quad \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{16}$$

where a division by zero is defined to be zero. The B-splines are $(p - m_i)$-times continuously differentiable across the $i$-th knot with multiplicity $m_i$. Hence, if $m_i = 1$ for inner knots, the B-splines of a degree e.o.c. are $C^{p-1}$ continuous across them.

The *multivariate B-splines* on the parameter domain $\widehat{\Omega} := (0, 1)^d$, $d = \{1, 2, 3\}$, are defined as tensor products of the corresponding univariate ones. In the multidimensional case, we define a knot-vector dependent on the coordinate direction $\Xi^\alpha = \{\xi_1^\alpha, ..., \xi_{n^\alpha + p^\alpha + 1}^\alpha\}$, $\xi_i^\alpha \in \mathbb{R}$, where $\alpha = 1, ..., d$ indicates the direction (in space or time). Furthermore, we introduce a set of multi-indices

$$\mathcal{I} = \left\{ \boldsymbol{i} = (i_1, ..., i_d) : i_\alpha = 1, ..., n_\alpha, \quad \alpha = 1, ..., d \right\}$$

and a multi-index $\boldsymbol{p} := (p_1, ..., p_d)$ indicating the order of polynomials. The tensor-product of univariate B-spline basis functions generates multivariate B-spline basis functions

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) := \prod_{\alpha=1}^{d} \widehat{B}_{i_\alpha, p_\alpha}(\xi^\alpha), \quad \text{where} \quad \boldsymbol{\xi} = (\xi^1, ..., \xi^d) \in \widehat{\Omega}. \tag{17}$$

The *univariate and multivariate NURBS basis* functions are defined in a parametric domain by means of B-spline basis functions, i.e., for a given $\boldsymbol{p}$ and any $\boldsymbol{i} \in \mathcal{I}$, NURBS basis functions are defined as $\widehat{R}_{\boldsymbol{i},\boldsymbol{p}} : \widehat{\Omega} \to \mathbb{R}$

$$\widehat{R}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) := \frac{w_i \, \widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})}{\sum_{\boldsymbol{i} \in \mathcal{I}} w_i \, \widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})}, \tag{18}$$

where $w_i \in \mathbb{R}^+$. To recall basic definitions related to THB-splines, we follow the structure outlined in [20] and consider a finite sequence of nested $d$-variate tensor-product spline spaces $\widehat{V}^0 \subset \widehat{V}^1 \subset ... \subset \widehat{V}^N$ defined on the axis aligned box-domain $\widehat{\Omega}^0 \subset \mathbb{R}^d$. To each space $V^\ell$ we assign a tensor-product B-spline basis of degree $\boldsymbol{p}$

$$\big\{\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}\big\}_{\boldsymbol{i}\in\mathcal{I}^{\ell}},\quad \mathcal{I}^{\ell}:=\{\boldsymbol{i}=(i_1,...,i_d),\quad i_k=1,...,n^{\ell}_k\quad\text{for}\quad k=1,...,d\,\big\},$$

where $\mathcal{I}^{\ell}$ is a set of multi-indices for each level, and $n^{\ell}_k$ denotes the number of univariate B-spline basis functions in the $k$-th coordinate direction. After assuming that $\mathcal{I}^{\ell}$ has a fixed ordering and rewriting the basis as $\widehat{\mathbf{B}}^{\ell}(\boldsymbol{\xi}) = (\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}))_{\boldsymbol{i}\in\mathcal{I}^{\ell}}$, it can be considered as a column-vector of basis functions. Then, a spline function $s:\widehat{\Omega}^0\to\mathbb{R}^m$ is defined by $\widehat{\mathbf{B}}^{\ell}(\boldsymbol{\xi})$ and a coefficient matrix $C^{\ell}$, i.e.,

$$s(\boldsymbol{\xi})=\sum_{\boldsymbol{i}\in\mathcal{I}^{\ell}}\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})c^{\ell}_i=\widehat{\mathbf{B}}^{\ell}(\boldsymbol{\xi})^{\mathrm{T}}\,C^{\ell},$$

where $c^{\ell}_i\in\mathbb{R}^m$ are row-coefficients of $C^{\ell}$.

Since $\widehat{V}^{\ell}\subset\widehat{V}^{\ell+1}$, the basis $\widehat{\mathbf{B}}^{\ell}$ can be represented by the linear combination of $\widehat{\mathbf{B}}^{\ell+1}$, namely,

$$s(\boldsymbol{\xi})=\widehat{\mathbf{B}}^{\ell}(\boldsymbol{\xi})^{\mathrm{T}}\,C^{\ell}=\widehat{\mathbf{B}}^{\ell+1}(\boldsymbol{\xi})^{\mathrm{T}}\,R^{\ell+1}\,C^{\ell},$$

where $R^{\ell+1}$ is a refinement matrix. Its entries can be obtained from B-splines refinement rules (see [48]). Along with nested space, a corresponding sequence of nested domains is considered

$$\widehat{\Omega}^0\supseteq\widehat{\Omega}^1\supseteq...\supseteq\widehat{\Omega}^N,\tag{19}$$

where each $\widehat{\Omega}^{\ell}\in\mathbb{R}^d$ is covered by a collection of cells with respect to the tensor-product grid of level $l$. In this work, we focus on dyadic cell refinement for the bi- and trivariate cases with uniform degrees $p_\alpha=p$ for all levels and coordinate directions, therefore, $\boldsymbol{p}=p$ in further exposition.

Let the *characteristic matrix* $X^{\ell}$ of $\widehat{\boldsymbol{B}}^{\ell}(\boldsymbol{\xi})$ w.r.t. domains $\Omega^{\ell}$ and $\Omega^{\ell+1}$ is defined as

$$X^{\ell}:=\mathrm{diag}(x^{\ell}_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{I}^{\ell}},\quad x^{\ell}_{\boldsymbol{i}}:=\begin{cases}1, & \text{if}\quad \mathrm{supp}\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}\subseteq\Omega^{\ell}\wedge\mathrm{supp}\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}\nsubseteq\Omega^{\ell+1}\\ 0, & \text{otherwise.}\end{cases}$$

Next, for each level $\ell$, the set of the indices of *active functions* can be defined with $\mathcal{I}^{\ell}_*:=\{\mathcal{I}^{\ell}:x^{\ell}_{\boldsymbol{i}}=1\}$. To store the indices of all active functions at all hierarchical levels, we define an *index set*

$$\boldsymbol{\mathcal{I}}:=\{(\ell,\boldsymbol{i}):\ell\in\{0,...,N\},\boldsymbol{i}\in\mathcal{I}^{\ell}_*\}.$$

Then, the THB-spline basis related to the hierarchical domains is defined as

$$\widehat{\mathbf{T}}(\boldsymbol{\xi})=(\mathcal{K}^{\ell}_{\boldsymbol{i}}(\boldsymbol{\xi}))_{(l,\boldsymbol{i})\in\boldsymbol{\mathcal{I}}},\quad \mathcal{K}^{\ell}_{\boldsymbol{i}}(\boldsymbol{\xi})=\mathrm{trunc}^{N}(\mathrm{trunc}^{N-1}(...\mathrm{trunc}^{\ell+1}(\widehat{B}^{\ell}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})))),$$

where the *truncation* of any function $s(\boldsymbol{\xi})\in\widehat{V}^{\ell}$ w.r.t. level $\ell+1$ is defined by

$$\mathrm{trunc}^{\ell+1}(s(\boldsymbol{\xi}))=\widehat{\mathbf{B}}^{\ell+1}(\boldsymbol{\xi})^{\mathrm{T}}\,(I^{\ell+1}-X^{\ell+1})\,R^{\ell+1}\,C^{\ell}.$$

Here, $I^{\ell+1}$ denotes an identity matrix $I^{\ell+1}$ of size $|I^{\ell+1}|\times|I^{\ell+1}|$, the multiplication of $R^{\ell+1}$ by $C^{\ell}$ represents $s(\boldsymbol{\xi})$ w.r.t. to the level $\ell+1$, and additional multiplication by $(I^{\ell+1}-X^{\ell+1})$ performs the truncation operation. For the detailed discussion of truncation operation, we refer the reader to [21, 22, 20].

The physical domain $\Omega\subset\mathbb{R}^d$ is defined by the geometrical mapping of the parametric domain $\widehat{\Omega}:=(0,1)^d$:

$$\Phi:\widehat{\Omega}\to\Omega:=\Phi(\widehat{\Omega})\subset\mathbb{R}^d,\quad \Phi(\boldsymbol{\xi}):=\sum_{\boldsymbol{i}\in\mathcal{I}}\widehat{\mathcal{B}}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi})\,c_i,\tag{20}$$

where $c_i\in\mathbb{R}^d$ are control points, and $\widehat{\mathcal{B}}$ stands for either B-splines, NURBS, or THB-basis functions . The mesh $\mathcal{K}_h$ discretizing $\Omega$ consists of elements $K\in\mathcal{K}_h$ that are the images of $\widehat{K}\in\widehat{\mathcal{K}}_h$, i.e.,

$$\mathcal{K}_h:=\big\{K=\Phi(\widehat{K}):\widehat{K}\in\widehat{\mathcal{K}}_h\big\}.$$

The global mesh size is denoted by

$$h:=\max_{K\in\mathcal{K}_h}\{\,h_K\,\},\quad h_K:=\|\nabla\Phi\|_{L^{\infty}(K)}\,\hat{h}_{\widehat{K}}.\tag{21}$$

Moreover, we assume that $\mathcal{K}_h$ is a quasi-uniform mesh, i.e., there exists a positive constant $C_u$ independent of $h$, such that $h_K\le h\le C_u\,h_K$.

# 4 Functional error estimates within the IgA framework

In this section, we present the algorithms used for general reliable computations and functional-type error estimates reconstruction. Then we proceed with commenting on the implementation of these error estimates in G+Smo and their integration into the library's structure. Finally, we present a series of examples demonstrating numerical properties of derived error majorants.

## 4.1 Reliable reconstruction of IgA approximations. Algorithms

In order to keep the presentation concise, we restrict (1)–(3) to the Dirichlet–Poisson problem

$$-\Delta_x u = f \quad \text{in} \quad \Omega := (0,1)^d \in \mathbb{R}^d,\ d = \{2,3\}, \qquad u = 0 \quad \text{on} \quad \Gamma = \partial\Omega. \tag{22}$$

Let the approximation

$$u_h \in V_{0h} := V_h \cap H_0^1(\Omega), \quad \text{where} \quad V_h \equiv \mathcal{S}_h^{p,p} := \left\{\phi_{h,i} := \hat{V}_h \circ \Phi^{-1}\right\}.$$

Here, $\hat{V}_h \equiv \hat{\mathcal{S}}_h^{p,p}$ is generated with NURBS of degree $p$, i.e., $\hat{V}_h := \operatorname{span}\left\{\widehat{\mathcal{B}}_{\boldsymbol{i},\boldsymbol{p}}\right\}_{\boldsymbol{i}\in\mathcal{I}}$. Due to the one-patch setting and restriction on the knots' multiplicity of $\hat{\mathcal{S}}_h^{p,p}$, the smoothness $u_h \in C^{p-1}$ is automatically provided. Since no numerical algorithms related to the hierarchical levels of the localised splines will be discussed below, we use the same notation for spaces generated by THB-splines. Therefore, the constructed approximation can be written as

$$u_h(x) = u_h(x_1, ..., x_d) := \sum_{i\in\mathcal{I}} \underline{\mathrm{u}}_{h,i}\, \phi_{h,i},$$

where $\underline{\mathrm{u}}_h := \left[\underline{\mathrm{u}}_{h,i}\right]_{i\in\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ is a vector of degrees of freedom (d.o.f.) defined by the system

$$\mathrm{K}_h\, \underline{\mathrm{u}}_h = \mathrm{f}_h, \quad \mathrm{K}_h := \left[(\nabla_x \phi_{h,i}, \nabla_x \phi_{h,j})_\Omega\right]_{i,j\in\mathcal{I}}, \quad \mathrm{f}_h := \left[(f, \phi_{h,i})_\Omega\right]_{i\in\mathcal{I}}. \tag{23}$$

The majorant corresponding to the problem (22) reads as

$$\overline{\mathrm{M}}(u_h, \boldsymbol{y}_h) := (1+\beta)\, \overline{\mathrm{m}}_{\mathrm{d}} + (1+\tfrac{1}{\beta})\, C_{\mathrm{F}\Omega}^2\, \overline{\mathrm{m}}_{\mathrm{eq}} = (1+\beta)\, \|\boldsymbol{y}_h - \nabla_x u_h\|_\Omega^2 + (1+\tfrac{1}{\beta})\, C_{\mathrm{F}\Omega}^2\, \|\mathrm{div}_x \boldsymbol{y}_h + f\|_\Omega^2, \tag{24}$$

where $\overline{\mathrm{m}}_{\mathrm{eq}}$ and $\overline{\mathrm{m}}_{\mathrm{d}}$ are defined in (14), $\beta > 0$ and $\boldsymbol{y}_h \in H(\Omega, \mathrm{div}_x)$. The approximation space for

$$\boldsymbol{y}_h \in Y_h \equiv \oplus^d \mathcal{S}_h^{q,q} \equiv \mathcal{S}_h^{q,q} \oplus ... \oplus \mathcal{S}_h^{q,q} := \left\{\hat{Y}_h \circ \Phi^{-1}\right\}$$

is generated by the push-forward corresponding space in the parametric domain

$$\hat{Y}_h := \oplus^d \hat{\mathcal{S}}_h^{q,q} \equiv \hat{\mathcal{S}}_h^{q,q} \oplus ... \oplus \hat{\mathcal{S}}_h^{q,q}.$$

Here, $\hat{\mathcal{S}}_h^{q,q}$ is a space of NURBS with the degree $q$ for each of $d$ components of $\boldsymbol{y}_h = (y_h^{(1)}, ..., y_h^{(d)})^{\mathrm{T}}$. The details of the numerical reconstruction of (24) were thoroughly studied in [29]. The best estimate follows from the optimisation of $\overline{\mathrm{M}}(u_h, \boldsymbol{y}_h)$ w.r.t. function

$$\boldsymbol{y}_h := \sum_{i\in\mathcal{I}} \underline{\mathbf{y}}_{h,i}\, \boldsymbol{\psi}_{h,i}.$$

The basis functions $\boldsymbol{\psi}_{h,i}$ generate the space $Y_h$, whereas $\underline{\mathbf{y}}_h := \left[\underline{\mathbf{y}}_{h,i}\right]_{i\in\mathcal{I}} \in \mathbb{R}^{d|\mathcal{I}|}$ is a vector of d.o.f of $\boldsymbol{y}_h$ defined by a system

$$\left(C_{\mathrm{F}\Omega}^2\, \mathrm{Div}_h + \beta\, \mathrm{M}_h\right)\, \underline{\mathbf{y}}_h = -C_{\mathrm{F}\Omega}^2\, \mathrm{z}_h + \beta\, \mathrm{g}_h, \tag{25}$$

where

$$\mathrm{Div}_h := \left[(\mathrm{div}_x \boldsymbol{\psi}_i, \mathrm{div}_x \boldsymbol{\psi}_j)_\Omega\right]_{i,j=1}^{d|\mathcal{I}|}, \qquad \mathrm{z}_h := \left[\left(f, \mathrm{div}_x \boldsymbol{\psi}_j\right)_\Omega\right]_{j=1}^{d|\mathcal{I}|},$$
$$\mathrm{M}_h := \left[(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)_\Omega\right]_{i,j=1}^{d|\mathcal{I}|}, \qquad \mathrm{g}_h := \left[\left(\nabla_x v, \boldsymbol{\psi}_j\right)_\Omega\right]_{j=1}^{d|\mathcal{I}|}.$$

According to the numerical results obtained in [29], the most efficient majorant reconstruction (with uniform refinement) is obtained when $q$ is set substantially higher than $p$. Let us assume that $q = p + m$, $m \in \mathbb{N}^+$. At the same time, when $u_h$ is reconstructed on the mesh $\mathcal{K}_h$, we use a coarser one $\mathcal{K}_{Mh}$, $M \in \mathbb{N}^+$ in order to recover $\boldsymbol{y}_h$. For the reader's convenience, all used notation is summarised in Table 1. The initial mesh $\mathcal{K}_h^0$ and

| | |
|---|---|
| $p$ | degree of the splines used for $u_h$ approximation |
| $S_h^{p,p}$ | approximation space for the scalar-functions generated by splines |
| $q$ | degree of the splines used for $\boldsymbol{y}_h$ approximation |
| $\oplus^d S_h^{q,q}$ | approximation space for the $d$-dimensional vector-functions generated by splines |
| $S_h^{q,q} \oplus S_h^{q,q}$ | approximation space for the two-dimensional vector-functions generated by splines |
| $m$ | $q-p$ |
| $M$ | coarsening ratio of the global size of the mesh for $u_h$ approximation to the global size of the mesh for $\boldsymbol{y}_h$ reconstruction |
| $\mathcal{K}_h$ ($\mathcal{K}_h^{u_h}$) | mesh used for $u_h$ approximation |
| $\mathcal{K}_{Mh}$ ($\mathcal{K}_h^{\boldsymbol{y}_h}$, $M=1$) | mesh used for $\boldsymbol{y}_h$ reconstruction |
| $N_{\text{ref}}$ | number of uniform or adaptive refinement steps |
| $N_{\text{ref},0}$ | number of initial refinement steps performed before testing |
| $\mathbb{M}_*(\theta)$ | marking criterion $*$ with the parameter $\theta$ |

Table 1: Table of notations.

the basis functions defined on it are assumed to be given via the geometry representation of the computational domain. The exact representation of geometry on the initial (the coarsest) level is preserved in the process of mesh refinement.

The classical strategy of the reliable $u_h$-approximation is summarised in Algorithm 1. Let us assume that the problem data such as $f$, $u_0$, and $\Omega$ of (1)–(3) are provided. The Input of Algorithm 1 is the initial mesh $\mathcal{K}_h$ (or the one obtained on the previous refinement step). It provides the refined version of it denoted by $\mathcal{K}_{h_{\text{ref}}}$ as an output. The process of new mesh generation can be divided into classical block-chain, i.e.,

$$\text{APPROXIMATE} \to \text{ESTIMATE} \to \text{MARK} \to \text{REFINE}.$$

On the APPROXIMATE step, we construct the system that provides the d.o.f. of $u_h$, i.e., we assemble the matrix $\mathrm{K}_h$ and RHS $\mathrm{f}_h$ defined in (23), and solve it with a direct sparse $\mathrm{LDL}^{\mathrm{T}}$ Cholesky factorisations for $d=2$ and conjugate gradient (CG) method for $d=3$. In the follow-up report, we will investigate how the selection of the initial guess enhances the performance of the iterative solver. In particular, we use the work [12, 5] that studies the so-called cascadic preconditioned conjugate gradient (CPCG) method. The latter one has an improved speed of convergence due to initial guess chosen as an interpolation of the approximation obtained on the previous refinement (hierarhical) level. It appeared that such a cascadic structure of the meshes by itself realises some kind of preconditioning. The time spent on assembling and solving sub-procedures is tracked and saved in vectors $t_{\text{as}}(u_h)$ and $t_{\text{sol}}(u_h)$, respectively. This notation is used in the upcoming examples to analyse the efficiency of Algorithm 1 and compare the computational costs for its blocks.

The next ESTIMATE step is responsible for the reconstruction of global estimate $\overline{\mathrm{M}}(u_h)$ as well as the element-wise error indicator distribution $\overline{\mathrm{m}}_{\mathrm{d}}(u_h)$ (see (14)) that follows from $\overline{\mathrm{M}}(u_h)$. The time spent for this chain-block is measured by $t_{\text{as}}(\boldsymbol{y}_h) + t_{\text{sol}}(\boldsymbol{y}_h)$. Its detailed description is presented in Algorithm 2.

In the chain-block MARK, we use a marking criterion denoted by $\mathbb{M}_*(\theta)$. It provides the algorithm for defining the threshold $\Theta_*$ for selecting those $K \in \mathcal{K}_h$ for further refinement that satisfies the criterion

$$\overline{\mathrm{m}}^2_{\mathrm{d},K} \geq \Theta_*(\mathbb{M}_*(\theta)), \quad K \in \mathcal{K}_h.$$

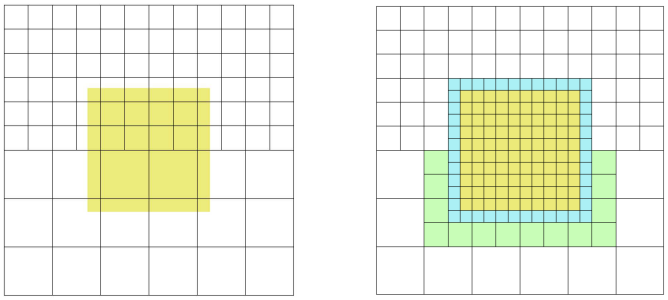
In the library [38], several marking strategies are considered. The first marking criterion defines the 'absolute threshold', and it is denoted as **GARU** (an abbreviation for 'greatest appearing residual utilisation'). The corresponding threshold reads as

$$\Theta_{\text{GARU}} := \theta \max_{K \in \mathcal{K}_h} \{\overline{\mathrm{m}}^2_{\mathrm{d},K}\}, \quad \theta \in (0,1).$$

The percentage of marked elements (dictated by this criterion) varies at each refinement step since $\Theta_{\text{GARU}}$ considers only the absolute value of the largest local error, without taking into account the element-wise distribution of the error.

The second marking criterion defining the 'relative threshold' is denoted as $\mathbb{M}_{\mathbf{PUCA}}$, where **PUCA** stands for 'percent-utilising cutoff ascertainment'. The corresponding amount of elements selected for the refinement can be approximated as follows:

$$|\{K : \overline{\mathrm{m}}^2_{\mathrm{d},K} > \Theta_{\text{PUCA}}\}_{K \in \mathcal{K}_h}| \approx (1-\theta) \cdot |\{K\}_{K \in \mathcal{K}_h}|, \quad \theta \in (0,1).$$



(a) (b)

Figure 2: Example of the box insertion in the second hierarchical level of THB-spline (the example is taken from [20]).

For instance, if we let $\theta = 0.7$, $\Theta_{\text{PUCA}}$ is chosen such that $\overline{\text{m}}^2_{\text{d},K} \geq \Theta_{\text{PUCA}}$ holds for 30% of elements.

Last and most widely used criterion is called bulk marking (also known as the Dörfler marking [15]) and is denoted as $\mathbb{M}_{\mathbf{BULK}}(\theta)$. According to this marking strategy, we select the subset of elements from the collection $\mathcal{K}_h$ that has been sorted w.r.t. element-wise contributions $\overline{\text{m}}^2_{\text{d},K}$, i.e., $\mathcal{K}'_h \underset{\text{K}}{\leftarrow} \mathcal{K}^{\text{sort}}_h := \text{sort}_{\overline{\text{m}}^2_{\text{d},K}}\{\mathcal{K}_h\}$, until we satisfy

$$\sum_{K \in \mathcal{K}'_h} \overline{\text{m}}^2_{\text{d},K} \geq \Theta_{\text{BULK}} := (1-\theta) \sum_{K \in \mathcal{K}_h} \overline{\text{m}}^2_{\text{d},K}, \quad \theta \in (0,1).$$

This way, we form a subset of elements which contains the highest indicated errors. The selection process stops when the error accumulated on previous steps exceeds the 'bulk' level (threshold) defined by $\theta$. In the case of uniform refinement, all elements of $\mathcal{K}_h$ are marked for refinement (i.e, $\theta = 0$). If the numerical IgA scheme is implemented correctly, the error is supposed to descrease at least as $O(h^p)$ (which is verified throughout the numerical tests in Section 5).

Finally, on the last REFINE step, we apply the refinement algorithm $\mathcal{R}$ to those elements that have been selected on the MARK level. Since the THB-splines are based on the subdomains of different hierarchical levels, the procedure $\mathcal{R}$ increases the level of subdomains that have been selected by $\mathbb{M}_*(\theta)$. For $\mathcal{R}$, a dyadic cell refinement is applied. To prevent the cases of refinement, when the inserted box is not aligned with the current hierarchical mesh (occurrence of the L-shaped cells), 'affected' cells of lower levels are locally subdivided to adapt to the inserted box. For that in further examples we specify the extension of the refined box by one cell (see, e.g., Figure 2). Here, Figure 2a illustrates the box insertion (yellow area) in the second hierarchical level of THB-spline. In Figure 2b, blue cells around the inserted box are the 'one-cell' extension of the yellow area. Green cells of the first level are the so-called the 'affected' cells of zero level that have been locally subdivided to adapt to the inserted box.

**Algorithm 1** Reliable reconstruction of $u_h$ (a single refinement step)

**Input:** $\mathcal{K}_h$ {discretisation of $\Omega$}
span $\{\phi_{h,i}\}$, $i = 1, ..., |\mathcal{I}|$ {$V_h$-basis}

**APPROXIMATE**:

- ASSEMBLE the matrix $\text{K}_h$ and RHS $\text{f}_h$. $:\mathbf{t}_{\text{as}}(\mathbf{u_h})$
- SOLVE $\text{K}_h\, \underline{\text{u}}_h = \text{f}_h$. $:\mathbf{t}_{\text{sol}}(\mathbf{u_h})$
- Reconstruct $u_h = \sum_{i \in \mathcal{I}} \underline{\text{u}}_i\, \phi_{h,i}(x)$.

**ESTIMATE**: Reconstruct $\overline{\text{M}}(u_h)$ and $\overline{\text{m}}_{\text{d}}(u_h)$. $:\mathbf{t}_{\text{as}}(\boldsymbol{y_h}) + \mathbf{t}_{\text{sol}}(\boldsymbol{y_h})$

**MARK**: Using the marking criteria $\mathbb{M}_*(\theta)$, select the elements $K$ of mesh $\mathcal{K}_h$ that must be refined.

**REFINE**: Execute the refinement strategy: $\mathcal{K}_{h_{\#ref.}} = \mathcal{R}(\mathcal{K}_h)$.

**Output:** $\mathcal{K}_{h_{\#ref.}}$ {refined discretisation of $\Omega$}

Let us now consider the structure of Algorithm 2, which clarifies the ESTIMATE step of Algorithm 1 in the

context of functional type error estimates. On the Input step, the algorithm receives the approximate solution $u_h$ reconstructed with the IgA scheme as the first argument. Then, since the majorant is minimised with respect to a vector-valued variable $\boldsymbol{y}_h \in Y_h$, the algorithm is provided with the collection of basis functions generating the space $Y_h := \operatorname{span}\{\boldsymbol{\psi}_{h,i}\}$, $i = 1, ..., d|\mathcal{I}|$. The last input parameter $N^{\mathrm{it}}_{\mathrm{maj}}$ defines the number of the optimisation circles executed to obtain a good enough minimiser of $\overline{\mathrm{M}}$. According to the tests performed in [54], one or two iterations are usually rather sufficient in order to achieve the reasonable accuracy of error majorant. Technically, if the ratio between $\overline{\mathrm{m}}_{\mathrm{eq}}$ and $\overline{\mathrm{m}}_{\mathrm{d}}$ is small enough, the loop can be exited even if $n < N^{\mathrm{it}}_{\mathrm{maj}}$. This condition might minimise the computational costs for the error control. However, for the consistency of exposition this is not incorporated into Algorithm 2 but only noted here as a remark.

It is crucial to emphasise that both matrices $\mathrm{Div}_h, \mathrm{M}_h$ and vectors $\mathrm{z}_h$, $\mathrm{g}_h$ are assembled only once and remain unchanged in the minimisation procedure. The loop is iterated $N^{\mathrm{it}}_{\mathrm{maj}}$ times, where on each step the optimal $\underline{\mathbf{y}}^{(n)}_h$ and $\beta^{(n)}$ are reconstructed. In our implementation, the optimality system for the flux (cf. (25)) is solved by direct sparse LDL$^{\mathrm{T}}$ Cholesky factorisations for $d = 2$ and by a conjugate gradient method for $d = 3$ (again, the initial guess is reconstructed from the approximation obtained on the earlier refinement). The time spent on ASSEMBLE and SOLVE steps w.r.t. system (25) is measured by $\mathbf{t}_{\mathrm{as}}(\boldsymbol{y}_{\mathbf{h}})$ and $\mathbf{t}_{\mathrm{sol}}(\boldsymbol{y}_{\mathbf{h}})$ respectively and compared to $\mathbf{t}_{\mathrm{as}}(u_h)$ and $\mathbf{t}_{\mathrm{sol}}(u_h)$ times in forthcoming numerical examples. Besides the computational costs related to the assembling and solving (23) and (25), we measure the time spent on element-wise (e/w) evaluation of error, majorant, and the residual error estimator denoted by $t_{\mathrm{e/w}}(\|\nabla_x e\|)$, $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$, and $t_{\mathrm{e/w}}(\overline{\eta})$, respectively.

**Algorithm 2** **ESTIMATE** step (majorant minimisation)

**Input:** $u_h$ {approximation}
$\mathcal{K}_h$ {disctretisation of $\Omega$},
$\operatorname{span}\{\boldsymbol{\psi}_{h,i}\}$, $i = 1, ..., d|\mathcal{I}|$ {$Y_h$-basis},
$N^{\mathrm{it}}_{\mathrm{maj}}$ {number of optimisation iterations}

**ASSEMBLE** $\mathrm{Div}_h, \mathrm{M}_h \in \mathbb{R}^{d|\mathcal{I}|\times d|\mathcal{I}|}$ and $\mathrm{z}_h$, $\mathrm{g}_h \in \mathbb{R}^{d|\mathcal{I}|}$. $:\mathbf{t}_{\mathrm{as}}(\boldsymbol{y}_{\mathbf{h}})$

Set $\beta^{(0)} = 1$.

**for** $n = 1$ **to** $N^{\mathrm{it}}_{\mathrm{maj}}$ **do**

**SOLVE** $\left(C^2_{\mathrm{F}\Omega}/\beta^{(n-1)}\,\mathrm{Div}_h + \mathrm{M}_h\right)\underline{\mathbf{y}}^{(n)}_h = -C^2_{\mathrm{F}\Omega}/\beta^{(n-1)}\,\mathrm{z}_h + \mathrm{g}_h$. $:t_{\mathrm{sol}}(\boldsymbol{y}_h)$

Reconstruct $\boldsymbol{y}^{(n)}_h := \sum_{i\in\mathcal{I}} \underline{\mathbf{y}}^{(n)}_{h,i}\,\boldsymbol{\psi}_{h,i}$.

Compute $\overline{\mathrm{m}}^{(n)}_{\mathrm{eq}} := \| f + \mathrm{div}_x \boldsymbol{y}^{(n)}_h \|^2_\Omega$ and $\overline{\mathrm{m}}^{(n)}_{\mathrm{d}} := \|\boldsymbol{y}^{(n)}_h - \nabla_x u_h\|^2_\Omega$.

Compute $\beta^{(n)} = \frac{C_{\mathrm{F}\Omega}\,\overline{\mathrm{m}}^{(n)}_{\mathrm{eq}}}{\overline{\mathrm{m}}^{(n)}_{\mathrm{d}}}$.

**end for**

Compute $\overline{\mathrm{M}}(u_h, \boldsymbol{y}^{(n)}_h; \beta^{(n)}) := (1+\beta^{(n)})\,\overline{\mathrm{m}}^{(n)}_{\mathrm{eq}} + (1+\frac{1}{\beta^{(n)}})\,C^2_{\mathrm{F}\Omega}\overline{\mathrm{m}}^{(n)}_{\mathrm{d}}$.

**Output:** $\overline{\mathrm{M}}$ {total error majorant on $\Omega$},
$\overline{\mathrm{m}}^{(n)}_{\mathrm{d}}$ {indicator of error distribution over $\mathcal{K}_h$}

## 4.2 Implementation of functional error estimate in G+Smo

G+Smo is an open-source object-oriented C++ library for isogeometric analysis. The library exploits object polymorphism and inheritance techniques in order to support the variety of different discretisation bases. The implementation of basis functions and geometries is dimension-independent. The main ideology of development process is producing a high quality, efficient, and easy to use code that is cross-platform compatible.

The *hierarchical splines* in G+Smo are implemented on top of NURBS module, a dimension- independent implementation of classical tensor-product B-splines and their rational counterparts. The core of THB-splines is the representation of the hierarchical domain as a binary subdivision tree data-structure generalised from the quad-tree implementation presented in [28]. The leaves of such a tree provide the partition of the domain in quadrilateral ($d = 2$) and cubical ($d = 3$) subdomains that are part of the same hierarchical level $\ell$.

For a basis compilation, the characteristics matrices $X^\ell$ are precomputed and stored in sparse format for all levels. To identify the subset of basis functions that needs to be truncated, the query is executed to perform a support overlay on the tree-structure . It is followed by the evaluation procedure, which is reduced to computing

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_\mathrm{d}$ | $\overline{\mathrm{m}}_\mathrm{eq}$ | $I_\mathrm{eff}(\overline{\mathrm{M}})$ | $I_\mathrm{eff}(\overline{\eta})$ | *e.o.c* |
|---|---|---|---|---|---|---|---|
| (a) $\boldsymbol{y}_h \in S^{5,5}_{3h} \oplus S^{5,5}_{3h}$ ($m=3$, $M=3$) | | | | | | | |
| 3 | 2.5648e-03 | 3.2806e-03 | 3.1546e-03 | 5.5974e-04 | 1.2791 | 11.0113 | 3.4565 |
| 5 | 1.5952e-04 | 1.9770e-04 | 1.9084e-04 | 3.0441e-05 | 1.2393 | 10.9580 | 2.3602 |
| 7 | 9.9673e-06 | 1.1974e-05 | 1.1921e-05 | 2.3799e-07 | 1.2013 | 10.9546 | 2.0901 |
| 9 | 6.2294e-07 | 7.4549e-07 | 7.4502e-07 | 2.0851e-09 | 1.1967 | 10.9545 | 2.0225 |
| 11 | 3.8934e-08 | 4.6571e-08 | 4.6564e-08 | 3.2185e-11 | 1.1962 | 10.9545 | 2.0056 |
| (b) $\boldsymbol{y}_h \in S^{9,9}_{7h} \oplus S^{9,9}_{7h}$ ($m=7$, $M=7$) | | | | | | | |
| 3 | 2.5648e-03 | 2.6756e-03 | 2.5800e-03 | 4.2495e-04 | 1.0432 | 11.0113 | 3.4565 |
| 5 | 1.5952e-04 | 1.7737e-04 | 1.6869e-04 | 3.8537e-05 | 1.1118 | 10.9580 | 2.3602 |
| 7 | 9.9673e-06 | 1.0215e-05 | 1.0035e-05 | 7.9975e-07 | 1.0248 | 10.9546 | 2.0901 |
| 9 | 6.2294e-07 | 6.9080e-07 | 6.3274e-07 | 2.5797e-07 | 1.1089 | 10.9545 | 2.0225 |
| 11 | 3.8934e-08 | 4.0932e-08 | 3.9140e-08 | 7.9608e-09 | 1.0513 | 10.9545 | 2.0056 |

Table 2: *Ex. 1.* Error, majorant (with dual and equilibrated terms), efficiency indices, and e.o.c. w.r.t. uniform ref. steps.

tensor-product B-splines basis functions (see details in [20] and references therein). The latter is done via the recursive definition using precomputed coefficients. The evaluation of the field at a given point is summarised in [20, Algorithm 1]. An adaptive refinement algorithm is equivalent to box insertion into the domain, i.e., for the case $d = 2$, the quadrilateral domain is inserted into the higher level of the binary tree. After changing the structure of the domain, the characteristics matrices are updated locally.

Implementation of functional error estimates is divided into two logical parts. The first one is related to the ASSEMBLE and SOLVE steps of Algorithm 2 that recover d.o.f. of the optimal $\boldsymbol{y}_h$-reconstruction for $\overline{\mathrm{M}}$. The assembling of (25) is performed by two classes `gsVisitorDivDiv.h` and `gsVisitorDualPoisson.h` with a structure similar to the library class `gsVisitorPoisson.h`. The latter one is responsible for assembling the system $\mathrm{K}_h \, \underline{\mathrm{u}}_h = \mathrm{f}_h$ corresponding to the variational formulation

$$(\nabla_x u, \nabla_x \eta)_\Omega = (f, \eta)_\Omega, \quad \forall \eta \in H^1_0(\Omega).$$

Analogously, matrix $\mathrm{Div}_h$ and RHS $\mathrm{z}_h$ are assembled by `gsVisitorDivDiv.h` class for the equation

$$-(\mathrm{div}_x \boldsymbol{y}, \mathrm{div}_x \boldsymbol{\psi})_\Omega = (f, \mathrm{div}_x \boldsymbol{\psi})_\Omega, \quad \forall \boldsymbol{\psi} \in H(\Omega, \mathrm{div}_x),$$

whereas $\mathrm{MM_h}$ and $\mathrm{g}_h$ are generated by `gsVisitorDualPoisson.h` for

$$(\boldsymbol{y}, \boldsymbol{\psi})_\Omega = (\nabla_x v, \boldsymbol{\psi})_\Omega, \quad \forall \boldsymbol{\psi} \in H(\Omega, \mathrm{div}_x).$$

The second logical step is the majorant's element-wise evaluation. Its implementation is based on the parent class `gsNorm.h` that is responsible for the (element-wise and total) norm evaluation. Therefore, by only overwriting the function that performs actual computation of $\overline{\mathrm{M}}_K$ and $\overline{m}_{\mathrm{d},K}$ on each $K \in \mathcal{K}_h$, the majorant's functionality is integrated into G+Smo library. In order to advance the performance of assembling and e/w evaluation of the majorant, we use the OpenMP technology to perform the evaluation of its independent components $\overline{m}_\mathrm{d}$ and $\overline{m}_\mathrm{eq}$.

# 5 Numerical examples

In the current section, we present a series of examples demonstrating the numerical properties of the error majorants discussed above. We start with relatively simple examples, in which we aim to introduce the main properties of majorant and, at the same time, familiarise the reader with the structure of performed numerical tests. This approach is intended to bring the focus to analysis in more complicated examples discussed further.

**Example 1** First, we consider a basic example with

$$u = (1 - x_1)\, x_1^2\, (1 - x_2)\, x_2, \quad f = -\big(2\,(1 - 3\,x_1)\,(1 - x_2)\,x_2 - 2\,(1 - x_1)\,x_1^2\big) \quad \text{in} \quad \Omega$$

and homogenous Dirichlet boundary condition (BC).

Let the primal variable be approximated by the splines of degree $p = 2$, i.e., the discretisation space $S_h^{p,p}$. For the uniform refinement (unif. ref.), we first test the idea introduced in [29] and compare two different settings for spaces approximating auxiliary dual variable $\boldsymbol{y}_h \in S^{q,q}_{Mh}$:

$$(a)\ q = 5,\ m = 3,\ M = 3, \quad \text{and} \quad (b)\ q = 9,\ m = 7,\ M = 7. \tag{26}$$

| # ref | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\rm as}(u_h)$ | $t_{\rm as}(\boldsymbol{y}_h)$ | $t_{\rm sol}(u_h)$ | $t_{\rm sol}(\boldsymbol{y}_h)$ | $t_{\rm e/w}(\|\nabla_x e\|)$ | $t_{\rm e/w}(\overline{\rm M})$ | $t_{\rm e/w}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| (a) $\boldsymbol{y}_h \in S^{5,5}_{3h} \oplus S^{5,5}_{3h}$ ($q=5$, $m=3$, $M=3$) | | | | | | | | | |
| 1 | 9 | 36 | 0.0013 | 0.0023 | 0.0001 | 0.0017 | 0.0000 | 0.0008 | 0.0006 |
| 3 | 36 | 36 | 0.0010 | 0.0025 | 0.0001 | 0.0018 | 0.0005 | 0.0025 | 0.0022 |
| 5 | 324 | 81 | 0.0094 | 0.0216 | 0.0008 | 0.0094 | 0.0087 | 0.0159 | 0.0276 |
| 7 | 4356 | 441 | 0.0729 | 0.2506 | 0.0439 | 0.0730 | 0.1830 | 0.1571 | 0.2858 |
| 9 | 66564 | 4761 | 1.2661 | 4.0725 | 3.6962 | 8.3926 | 2.5329 | 2.8220 | 4.1740 |
| 11 | 1052676 | 68121 | 22.4621 | 68.2723 | 211.1700 | 570.4293 | 37.8862 | 37.9696 | 65.6940 |
| (b) $\boldsymbol{y}_h \in S^{9,9}_{7h} \oplus S^{9,9}_{7h}$ ($q=9$, $m=7$, $M=7$) | | | | | | | | | |
| 1 | 9 | 100 | 0.0008 | 0.0234 | 0.0001 | 0.0122 | 0.0002 | 0.0025 | 0.0004 |
| 3 | 36 | 100 | 0.0006 | 0.0167 | 0.0001 | 0.0132 | 0.0003 | 0.0038 | 0.0012 |
| 5 | 324 | 100 | 0.0089 | 0.0258 | 0.0010 | 0.0057 | 0.0048 | 0.0269 | 0.0167 |
| 7 | 4356 | 100 | 0.0750 | 0.0140 | 0.0401 | 0.0093 | 0.1564 | 0.5749 | 0.3073 |
| 9 | 66564 | 169 | 1.1129 | 0.1967 | 3.2580 | 0.0763 | 2.5923 | 6.2473 | 4.2985 |
| 11 | 1052676 | 625 | 17.6219 | 3.9372 | 196.0170 | 1.2941 | 35.1466 | 99.9845 | 61.1072 |

Table 3: *Ex. 1.* Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$, time of e/w evaluation of error, majorant, and residual error estimator w.r.t. uniform ref. steps.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\rm M}$ | $\overline{\rm m}_{\rm d}$ | $\overline{\rm m}_{\rm eq}$ | $I_{\rm eff}(\overline{\rm M})$ | $I_{\rm eff}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 2.5648e-03 | 3.0154e-03 | 3.0000e-03 | 6.8225e-05 | 1.1757 | 11.0115 | 3.4566 |
| 5 | 1.5952e-04 | 1.7571e-04 | 1.6338e-04 | 5.4779e-05 | 1.1015 | 10.9580 | 2.3602 |
| 7 | 9.9672e-06 | 1.1959e-05 | 1.0675e-05 | 5.7051e-06 | 1.1998 | 10.9547 | 2.0901 |
| 9 | 6.2294e-07 | 6.4308e-07 | 6.3365e-07 | 4.1905e-08 | 1.0323 | 10.9545 | 2.0225 |
| 11 | 3.8934e-08 | 4.0029e-08 | 3.9480e-08 | 2.4369e-09 | 1.0281 | 10.9545 | 2.0056 |

Table 4: *Ex. 1.* Error, majorant (with dual and reliability terms), efficiency indices, and e.o.c. for $\boldsymbol{y}_h \in S^{3,3}_{8h} \oplus S^{3,3}_{8h}$ w.r.t. uniform ref. steps.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\rm as}(u_h)$ | $t_{\rm as}(\boldsymbol{y}_h)$ | $t_{\rm sol}(u_h)$ | $t_{\rm sol}(\boldsymbol{y}_h)$ | $t_{\rm e/w}(\|\nabla_x e\|)$ | $t_{\rm e/w}(\overline{\rm M})$ | $t_{\rm e/w}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 16 | 0.0009 | 0.0015 | 0.0000 | 0.0001 | 0.0001 | 0.0003 | 0.0003 |
| 3 | 36 | 16 | 0.0008 | 0.0005 | 0.0000 | 0.0001 | 0.0006 | 0.0007 | 0.0018 |
| 5 | 324 | 16 | 0.0081 | 0.0006 | 0.0005 | 0.0001 | 0.0184 | 0.0112 | 0.0285 |
| 7 | 4356 | 16 | 0.0753 | 0.0004 | 0.0173 | 0.0001 | 0.1391 | 0.1071 | 0.2534 |
| 9 | 66564 | 25 | 1.1899 | 0.0009 | 1.3832 | 0.0001 | 2.2776 | 1.6354 | 4.0632 |
| 11 | 1052676 | 121 | 19.9547 | 0.0114 | 107.0756 | 0.0020 | 36.0268 | 26.0721 | 63.6307 |

Table 5: *Ex. 1.* Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$, time of e/w evaluation of error, majorant, and residual error estimator for $\boldsymbol{y}_h \in S^{3,3}_{8h} \oplus S^{3,3}_{8h}$ w.r.t. uniform ref. steps.

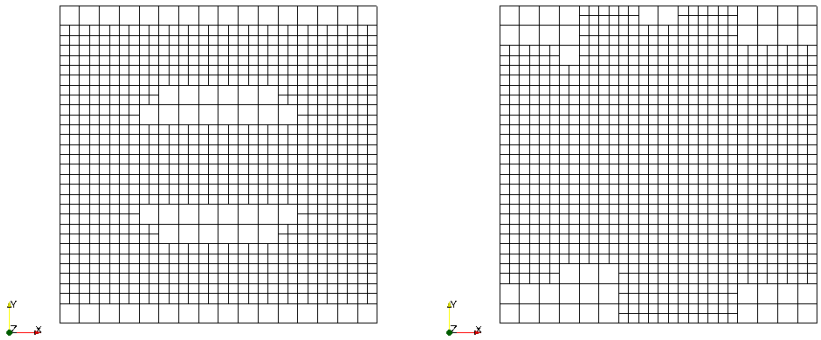
We perform $N_{\rm ref} = 11$ uniform refinement steps (ref. steps), and the obtained numerical results are presented in Tables 2–3 (such that the upper and the lower parts of them correspond to the cases (a) and (b), respectively). The efficiency of functional error majorant is confirmed by corresponding indices, i.e., $I_{\rm eff}(\overline{\rm M}) = 1.1961$ for the case (a) and $I_{\rm eff}(\overline{\rm M}) = 1.0024$ for the case (b) (see the shaded column of Table 2). The expected error order of convergence (e.o.c.) $p = 2$ is confirmed by the last column of Table 2.

When the computational costs are considered, the time spent on the reconstruction of $\boldsymbol{y}_h$ (i.e., $t_{\rm as}(\boldsymbol{y}_h) + t_{\rm sol}(\boldsymbol{y}_h)$) is about $2-3$ times greater than the time $t_{\rm as}(u_h) + t_{\rm sol}(u_h)$ in the setting (a). However, for the case (b), the assembling time of systems $\mathrm{Div}_h$ and $\mathrm{M}_h$ denoted by $t_{\rm as}(\boldsymbol{y}_h)$ takes approximately 1/4-th of the assembling time for $\mathrm{K_h}$ denoted by $t_{\rm as}(u_h)$. Similarly, solving the system (25) denoted by $t_{\rm sol}(\boldsymbol{y}_h)$ requires only 1/150-th of time spent on solving (23), i.e., $t_{\rm sol}(u_h)$.

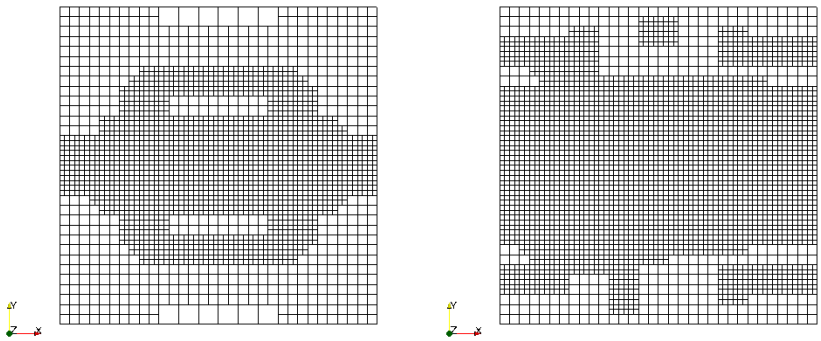
Due to the smoothness of the exact solution in this example, we can even use splines of lower degree for the flux approximation, e.g., $q = 3$, but at the same time reconstruct it on a much coarser mesh than for $u_h$, e.g., $M = 8$. The resulting efficiency indices are illustrated in Table 5, and corresponding times spent on the reconstruction of $u_h$ and $\boldsymbol{y}_h$ (e.i., $\overline{\rm M}(u_h, \boldsymbol{y}_h)$ and $\overline{\rm m}_{\rm d}(u_h, \boldsymbol{y}_h)$) are presented in Table 4. By looking at Table 5, one can see the considerable speed-up in time required for reconstruction of $\boldsymbol{y}_h$ in comparison to $u_h$:

$$\frac{t_{\rm as}(u_h)}{t_{\rm as}(\boldsymbol{y}_h)} \approx \frac{19.9547}{0.0114} \approx 1750 \quad \text{and} \quad \frac{t_{\rm sol}(u_h)}{t_{\rm sol}(\boldsymbol{y}_h)} \approx \frac{107.0756}{0.0020} \approx 53538.$$

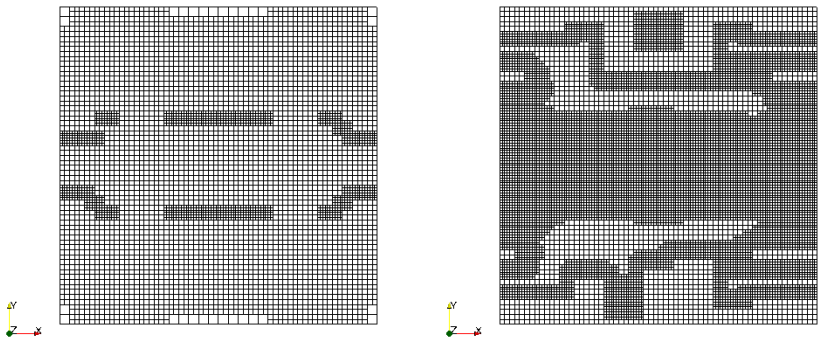
For an adaptive refinement strategy, we combine the THB-splines [30, 69, 21], which support local refinement, and functional error estimate (24). We use bulk marking with parameter $\theta = 0.4$. Let us start with the following setting: $u_h \in S^{2,2}_h$, where $S^{2,2}_h$ is generated by THB-splines, and $\boldsymbol{y}_h \in S^{3,3}_{8h} \oplus S^{3,3}_{8h}$, where $S^{3,3}_{8h} \oplus S^{3,3}_{8h}$ is generated



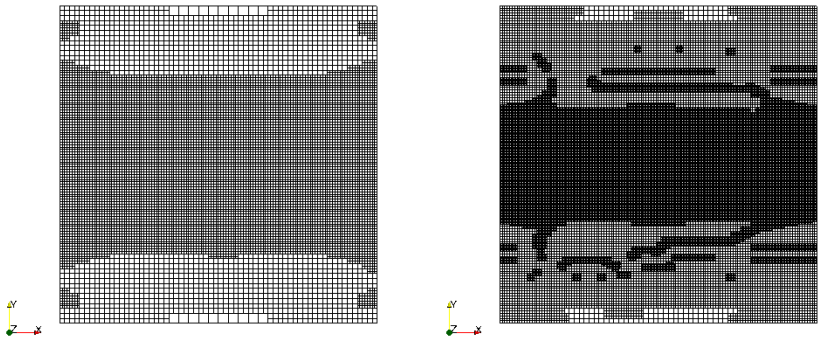
(a) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$ (b) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



(c) ref. # 6: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$ (d) ref. # 6: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



(e) ref. # 7: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$ (f) ref. # 7: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



(g) ref. # 8: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$ (h) ref. # 8: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$

Figure 3: *Ex. 1.* Evolution of adaptive meshes obtained with the marking criteria $\mathbb{M}_{\mathbf{BULK}}(0.4)$ and $\mathbb{M}_{\mathbf{BULK}}(0.2)$ w.r.t. adaptive ref. steps.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 7.9113e-03 | 8.9838e-03 | 8.1878e-03 | 3.5362e-03 | 1.1356 | 8.7164 | 0.9252 |
| 5 | 5.5188e-04 | 7.5068e-04 | 7.3580e-04 | 6.6131e-05 | 1.3602 | 9.9802 | 2.9781 |
| 7 | 4.8373e-05 | 6.1113e-05 | 5.9155e-05 | 8.7003e-06 | 1.2634 | 10.0270 | 2.4156 |
| 9 | 6.1176e-06 | 8.6725e-06 | 8.4757e-06 | 8.7451e-07 | 1.4176 | 10.3944 | 1.6446 |
| 11 | 6.1657e-07 | 6.3268e-07 | 6.2654e-07 | 2.7268e-08 | 1.0261 | 10.7801 | 2.5543 |

Table 6: *Ex. 1.* Error, majorant (with dual and reliability terms), efficiency indices, and e.o.c. w.r.t. adaptive ref. steps with the marking $\mathbb{M}_{\mathrm{BULK}}(0.2)$.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 16 | 0.0023 | 0.0031 | 0.0001 | 0.0003 | 0.0008 | 0.0038 | 0.0024 |
| 3 | 36 | 16 | 0.0055 | 0.0021 | 0.0001 | 0.0002 | 0.0032 | 0.0146 | 0.0170 |
| 5 | 305 | 16 | 0.0839 | 0.0023 | 0.0008 | 0.0002 | 0.1223 | 0.1920 | 0.2382 |
| 7 | 3224 | 16 | 1.4683 | 0.0035 | 0.0581 | 0.0002 | 1.4490 | 2.8517 | 2.8234 |
| 9 | 38276 | 16 | 27.1005 | 0.0021 | 1.9923 | 0.0002 | 22.0243 | 30.7559 | 38.1258 |
| 11 | 396360 | 49 | 3153.3647 | 0.0495 | 73.2963 | 0.0017 | 218.8799 | 328.0449 | 410.5585 |

Table 7: *Ex. 1.* Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$, time of e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps with the marking $\mathbb{M}_{\mathrm{BULK}}(0.2)$.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 2 | 5.5286e-02 | 6.3291e-02 | 5.7322e-02 | 2.6518e-02 | 1.1448 | 10.3894 | 3.9940 |
| 4 | 3.2077e-03 | 4.0140e-03 | 3.4919e-03 | 2.3195e-03 | 1.2514 | 10.9176 | 2.3839 |
| 5 | 7.9894e-04 | 1.4534e-03 | 1.4273e-03 | 1.1597e-04 | 1.8191 | 10.9451 | 2.1856 |
| 6 | 1.9955e-04 | 1.2390e-03 | 1.1931e-03 | 2.0405e-04 | 6.2091 | 10.9521 | 2.0914 |
| 8 | 1.2468e-05 | 9.8611e-05 | 3.7673e-05 | 2.7074e-04 | 7.9091 | 10.9543 | 2.0226 |
| 10 | 7.7924e-07 | 8.4668e-07 | 7.7970e-07 | 2.9758e-07 | 1.0865 | 10.9544 | 2.0056 |

Table 8: *Ex. 2*, $k_1 = k_2 = 1$. Error, majorant (with dual and reliability terms), efficiency indices, and e.o.c. w.r.t. uniform ref. steps.

by the basis of THB-splines as well. Overall, $N_{\mathrm{ref}} = 11$ refinements are executed to obtain the error illustrated in Table 6. The time spent on that generate $u_h$ and $\boldsymbol{y}_h$ for corresponding error estimates is illustrated in Table 7. By using a mesh that is up to 8 times coarser than the one for $u_h$, we manage to spare computational time for reconstructing the optimal $\boldsymbol{y}_h$ and speed up the overall reconstruction of majorant. In the current configuration, we obtain the following ratios:

$$\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx \frac{3153.3647}{0.0495} \approx 63704 \quad \text{and} \quad \frac{t_{\mathrm{sol}}(u_h)}{t_{\mathrm{sol}}(\boldsymbol{y}_h)} \approx \frac{73.2963}{0.0017} \approx 43115.$$

The comparison of meshes obtained while refining with different parameters can be found on Figure 3, i.e., $\theta = 0.4$ (left column) and $\theta = 0.2$ (right column). It is obvious from the plots that the smaller bulk parameter $\theta$ is, the higher the percentage of refined elements in the mesh is.

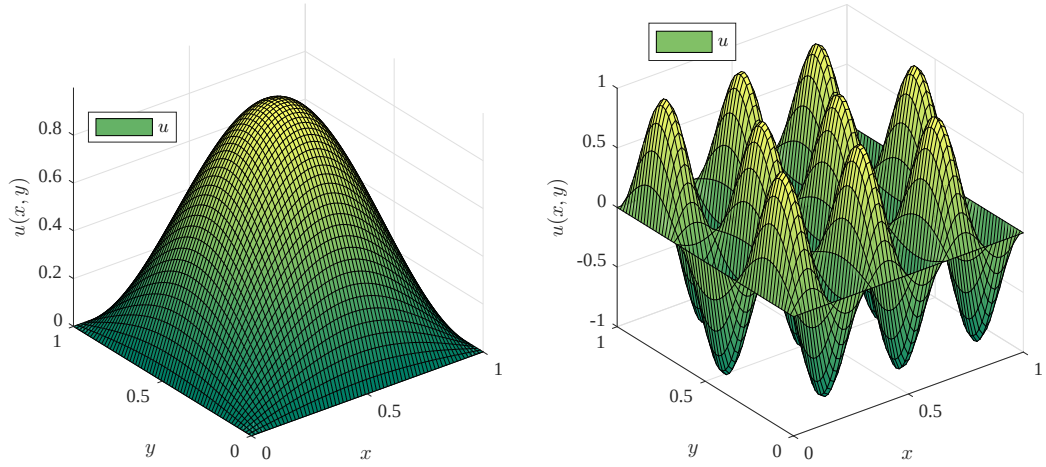
**Example 2** Next, we consider an example with a parametrised exact solution. By choosing different parameters, we study the properties of the majorant on the subdomains of $\Omega$, where $u$ has fast-growing gradients. Namely, we let $\Omega$ be a unit square, and let the exact solution and RHS be chosen as follows:

$$\begin{aligned} u &= \sin(k_1\,\pi\,x_1)\,\sin(k_2\,\pi\,x_2) && \text{in} \quad \Omega, \\ f &= (k_1^2 + k_2^2)\,\pi^2\,\sin(k_1\,\pi\,x_1)\,\sin(k_2\,\pi\,x_2) && \text{in} \quad \Omega, \\ u_D &= 0 && \text{on} \quad \Gamma. \end{aligned}$$

First, let $k_1 = k_2 = 1$. For such parameters, the exact solution is illustrated in Figure 4a. The function $u_h$ is approximated by $S_h^{2,2}$ whereas $\boldsymbol{y}_h$ by $S_{6h}^{5,5} \oplus S_{6h}^{5,5}$, and $N_{\mathrm{ref}} = 11$ uniform ref. steps are considered. The resulting performance of majorant is presented in Table 8. At the same time, the computational effort spent on $\boldsymbol{y}_h$-reconstruction is several times lower than for $u_h$, i.e.,

$$\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx \frac{20.2540}{0.2224} \approx 91 \quad \text{and} \quad \frac{t_{\mathrm{sol}}(u_h)}{t_{\mathrm{sol}}(\boldsymbol{y}_h)} \approx \frac{166.5165}{0.1367} \approx 1218,$$

which can be observed from Table 9.




(a) $k_1 = k_2 = 1$ (b) $k_1 = 6, k_2 = 3$

Figure 4: *Ex. 2.* Exact solution $u = \sin(k_1\,\pi\,x_1)\,\sin(k_2\,\pi\,x_2)$.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\text{as}}(u_h)$ | $t_{\text{as}}(\boldsymbol{y}_h)$ | $t_{\text{sol,dir}}(u_h)$ | $t_{\text{sol,dir}}(\boldsymbol{y}_h)$ | $t_{\text{sol,iter}}(u_h)$ | $t_{\text{sol,iter}}(\boldsymbol{y}_h)$ |
|---|---|---|---|---|---|---|---|---|
| 2 | 36 | 36 | 0.0007 | 0.0013 | 0.0001 | 0.0014 | 0.0000 | 0.0010 |
| 4 | 324 | 36 | 0.0091 | 0.0016 | 0.0013 | 0.0010 | 0.0001 | 0.0007 |
| 5 | 1156 | 36 | 0.0289 | 0.0015 | 0.0057 | 0.0008 | 0.0598 | 0.1015 |
| 6 | 4356 | 36 | 0.0723 | 0.0017 | 0.0342 | 0.0007 | 0.0067 | 0.0003 |
| 8 | 66564 | 81 | 1.5561 | 0.0141 | 3.1404 | 0.0036 | 0.6299 | 0.0045 |
| 10 | 1052676 | 441 | 20.2540 | 0.2224 | 166.5165 | 0.1367 | 33.6298 | 0.1121 |

Table 9: *Ex. 2*, $k_1 = k_2 = 1$. Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$ (with direct and iterative solvers).

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\text{M}}$ | $\overline{\text{m}}_{\text{d}}$ | $\overline{\text{m}}_{\text{eq}}$ | $I_{\text{eff}}(\overline{\text{M}})$ | $I_{\text{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 3.1030e-02 | 3.1818e-02 | 3.1057e-02 | 3.3805e-03 | 1.0254 | 10.8671 | 2.1371 |
| 5 | 1.9203e-03 | 1.9909e-03 | 1.9303e-03 | 2.6939e-04 | 1.0367 | 10.9490 | 2.0254 |
| 7 | 1.1995e-04 | 1.8194e-04 | 1.2028e-04 | 2.7397e-04 | 1.5168 | 10.9541 | 2.0058 |

Table 10: *Ex. 2*, $k_1 = 6, k_2 = 3$. Error, majorant (with dual and reliability terms), efficiency indices, and e.o.c. w.r.t. uniform ref. steps.

Let us consider now a more complicated case with $k_1 = 6$ and $k_2 = 3$ (see Figure 4b). For an efficient flux reconstruction, we apply the same strategy as discussed in Ex. 1, i.e., we increase the degree of B-splines used for the space approximating $\boldsymbol{y}_h$, but at the same time, we use $M = 8$ times coarser mesh, i.e., $S^{9,9}_{8h} \oplus S^{9,9}_{8h}$. First, we analyse the results obtained by global refinement; they are presented in Tables 10 and 11. We consider $N_{\text{ref}} = 8$ uniform ref. steps (starting from a rather fine initial mesh generated by $N_{\text{ref},0} = 4$ initial ref. steps of original geometry and the basis assigned for it). In column 6 of Table 10, one can see that $I_{\text{eff}}$ takes values up to 6.2091 but decreases back to 1.0865 once we start refining the basis for the variable $\boldsymbol{y}_h$ as well. In particular, at the refinements steps 5 and 6, the initial mesh of 36 d.o.f. or $\boldsymbol{y}_h$ becomes relatively coarse in comparison to the basis for $u_h$ and must be refined in order to obtain efficient values of $\overline{\text{M}}$. Concerning the time spent on assembling and solving the systems in (23) and (25), we obtain the following ratios taken from Table 11, namely,

$$\frac{t_{\text{as}}(u_h)}{t_{\text{as}}(\boldsymbol{y}_h)} \approx \frac{17.3623}{3.2302} \approx 2 \quad \text{and} \quad \frac{t_{\text{sol}}(u_h)}{t_{\text{sol}}(\boldsymbol{y}_h)} \approx \frac{144.7056}{1.3482} \approx 107.$$

In the case of adaptive refinement, we also use the space $S^{9,9}_{7h} \oplus S^{9,9}_{7h}$ generated by THB-splines. Let the bulk threshold be defined by parameter $\theta = 0.4$, which causes the refinement of approximately 60% of all elements for the primal variable $u_h$. The obtained numerical results are presented in Tables 12–13.

Let us compare the performance of majorant in the uniform refinement and adaptive refinement strategies. Due to the implementation of THB-splines evaluation on G+Smo [38], the assembling of matrices both for $\boldsymbol{y}_h$ and $u_h$ is slower w.r.t. B-splines (compare the third and fourth columns of Table 13 to the third and fourth columns of Table 11). For d.o.f.($u_h$) $\approx 4000$, in the first case we spend $t_{\text{as}}(u_h) = 0.0813$ secs (second row highlighted with grey background in Table 11) in comparison to $t_{\text{as}}(u_h) = 3.1285$ secs for the THB-splines

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\text{as}}(u_h)$ | $t_{\text{as}}(\boldsymbol{y}_h)$ | $t_{\text{sol}}(u_h)$ | $t_{\text{sol}}(\boldsymbol{y}_h)$ | $t_{\text{e/w}}(\|\nabla_x e\|)$ | $t_{\text{e/w}}(\overline{\text{M}})$ | $t_{\text{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 324 | 625 | 0.0053 | 2.9646 | 0.0007 | 0.2622 | 0.0047 | 0.1177 | 0.0158 |
| 3 | 4356 | 625 | 0.0831 | 3.4472 | 0.0396 | 0.6094 | 0.2142 | 0.4602 | 0.3388 |
| 5 | 66564 | 625 | 1.1135 | 2.9721 | 2.3809 | 1.5243 | 2.4025 | 6.4769 | 3.9421 |
| 7 | 1052676 | 625 | 17.3623 | 3.2302 | 144.7056 | 1.3482 | 45.4342 | 102.9160 | 71.1602 |

Table 11: *Ex. 2*, $k_1 = 6, k_2 = 3$. Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. uniform ref. steps.

| $N_{\text{ref}}$ | $\|\nabla_x e\|_\Omega$ | $\overline{\text{M}}$ | $\overline{\text{m}}_{\text{d}}$ | $\overline{\text{m}}_{\text{eq}}$ | $I_{\text{eff}}(\overline{\text{M}})$ | $I_{\text{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 4.2892e-02 | 4.3740e-02 | 4.2910e-02 | 3.6902e-03 | 1.0198 | 9.6688 | 2.1266 |
| 5 | 5.3723e-03 | 5.5714e-03 | 5.4777e-03 | 4.1637e-04 | 1.0371 | 9.9389 | 1.8082 |
| 7 | 6.4564e-04 | 7.2116e-04 | 6.5719e-04 | 2.8420e-04 | 1.1170 | 10.5034 | 2.3521 |

Table 12: *Ex. 2*, $k_1 = 6, k_2 = 3$. Error, majorant (with dual and reliability terms), efficiency indices w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\text{as}}(u_h)$ | pics/ $t_{\text{as}}(\boldsymbol{y}_h)$ | $t_{\text{sol}}(u_h)$ | $t_{\text{sol}}(\boldsymbol{y}_h)$ | $t_{\text{e/w}}(\|\nabla_x e\|)$ | $t_{\text{e/w}}(\overline{\text{M}})$ | $t_{\text{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 324 | 625 | 0.1125 | 23.3092 | 0.0010 | 0.2418 | 0.0918 | 7.0919 | 0.1938 |
| 3 | 3468 | 625 | 1.2291 | 22.0679 | 0.0313 | 0.6313 | 1.5198 | 12.0444 | 2.9476 |
| 5 | 31640 | 625 | 44.5650 | 22.1864 | 0.9953 | 1.2794 | 18.0067 | 107.7677 | 32.6181 |
| 7 | 205060 | 625 | 1135.8096 | 21.3158 | 17.4630 | 1.3050 | 105.0698 | 583.0759 | 190.6939 |

Table 13: *Ex. 2*, $k_1 = 6, k_2 = 3$. Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\text{M}}$ | $\overline{\text{m}}_{\text{d}}$ | $\overline{\text{m}}_{\text{eq}}$ | $I_{\text{eff}}(\overline{\text{M}})$ | $I_{\text{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 2 | 5.5665e-03 | 1.7635e-01 | 5.6126e-02 | 4.2226e-01 | 31.6798 | 8.5283 | 2.9617 |
| 4 | 2.6655e-04 | 1.1913e-02 | 4.0094e-03 | 2.7762e-02 | 44.6942 | 10.6943 | 2.1266 |
| 6 | 1.6374e-05 | 3.0360e-05 | 1.7856e-05 | 4.3919e-05 | 1.8541 | 10.8469 | 2.0163 |
| 8 | 1.0223e-06 | 1.1654e-06 | 1.1146e-06 | 1.7861e-07 | 1.1400 | 10.8565 | 2.0031 |

Table 14: *Ex. 3.* Error, majorant (with dual and reliability terms), efficiency indices, and e.o.c. w.r.t. uniform ref. steps.

(highlighted with grey background row in Table 13 respectively), which is about 45 times slower. Moreover, these ratios grow as d.o.f.($u_h$) increases. For the auxiliary variable $\boldsymbol{y}_h$, the assembling time for THB-splines is 4–5 times slower than when using B-splines. A similar increase in time can be observed for the element-wise evaluation of the error, majorant, and residual error estimator illustrated in the last three columns of Table 13 (in comparison to Table 11). This slowdown can be explained by a certain bottleneck that is present in the evaluation of THB-splines in G+Smo library.

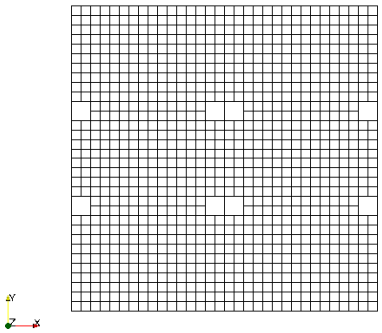
Analogously to the previous example, we demonstrate the evaluation of adaptive meshes for different marking criteria, i.e., marking $\mathbb{M}_{\mathbf{BULK}}(0.4)$ (left column of Figure 5) and $\mathbb{M}_{\mathbf{BULK}}(0.6)$ (right column of Figure 5). It resembles the patterns obtained in [29, Example 1], however, in the current case, due to the local structure of THB-splines, many superfluous d.o.f. are eliminated.

**Example 3** Next, we consider an example with a sharp local jump in the exact solution. Let $\Omega := (0, 2) \times (0, 1)$,

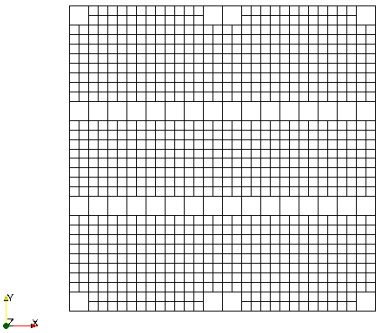
$$u = (x_1^2 - 2\,x_1)\,(x_2^2 - x_2)\,e^{-100\,|(x_1,x_2)-(1.4,0.95)|} \quad \text{in} \quad \Omega,$$

where the jump is located in the point $(x_1, x_2) = (1.4, 0.95)$ (see Figure 6), $f$ is calculated by substituting $u$ into (22), and the Dirichlet BC are homogenous. First, we run the test with the uniform refinement strategy. The obtained results are summarised in Tables 14–15. Several systematically performed tests showed that in order to perform a reliable estimation of the error in $u_h \in S_h^{2,2}$, it is optimal to take $\boldsymbol{y}_h \in S_{3h}^{4,4} \oplus S_{3h}^{4,4}$, i.e., we obtain efficient error bounds with the minimal computation effort spent on assembling and solving (25).

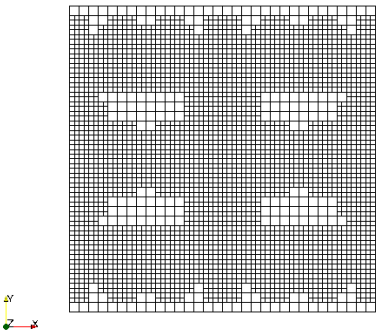
Still, the most interesting test-case is the one that checks the performance of the majorant in an adaptive algorithm. A series of tests showed that the optimal setting (in terms of quality of the error bounds and computational time spent on its reconstruction) is the approximation of $\boldsymbol{y}_h$ with THB-basis functions of degree 4, i.e., $\boldsymbol{y}_h \in S_h^{4,4} \oplus S_h^{4,4}$. At the same time, we consider the same $\mathcal{K}_h$ that is used for approximation of $u_h$. The obtained decrease of error and majorant with the marking criteria $\mathbb{M}_{\mathbf{BULK}}(0.4)$ and $\mathbb{M}_{\mathbf{BULK}}(0.6)$ is illustrated



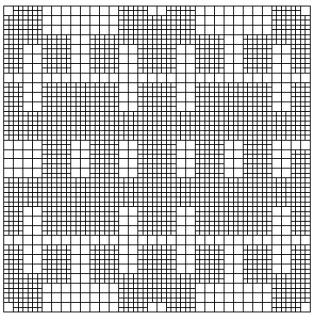
(a) ref. # 2: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



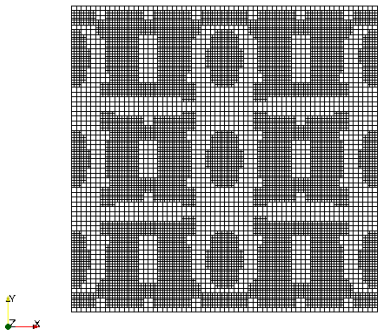
(b) ref. # 2: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.6))$



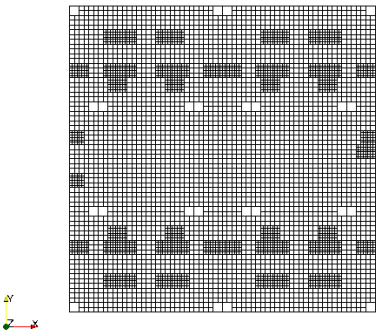
(c) ref. # 3: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



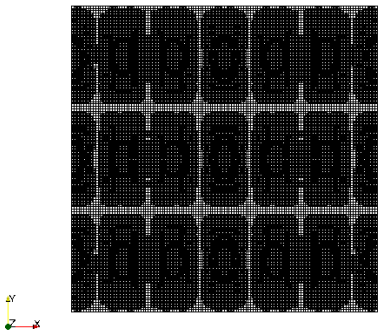
(d) ref. # 3: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.6))$



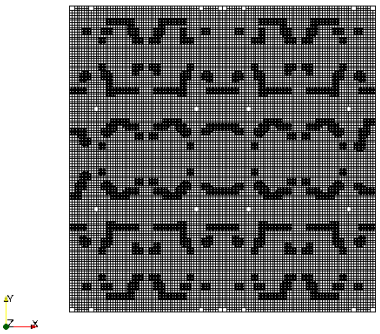
(e) ref. # 4: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



(f) ref. # 4: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.6))$



(g) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



(h) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.6))$

Figure 5: *Ex. 2*, $k_1 = 6, k_2 = 3$. Evolution of adaptive meshes obtained with the marking criteria $\mathbb{M}_{\mathbf{BULK}}(0.4)$ (left) and $\mathbb{M}_{\mathbf{BULK}}(0.6)$ (right) w.r.t. adaptive ref. steps.

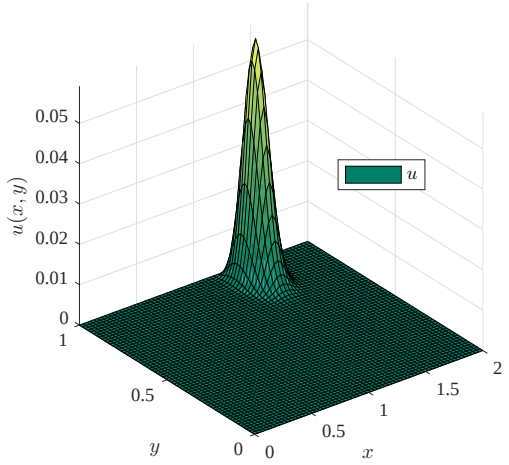



Figure 6: *Ex. 3.* Exact solution $u = (x_1^2 - 2\,x_1)\,(x_2^2 - x_2)\,e^{-100\,|(x_1,x_2)-(1.4,0.95)|}$.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\text{as}}(u_h)$ | $t_{\text{as}}(\boldsymbol{y}_h)$ | $t_{\text{sol}}(u_h)$ | $t_{\text{sol}}(\boldsymbol{y}_h)$ | $t_{\text{e/w}}(\|\nabla_x e\|)$ | $t_{\text{e/w}}(\overline{\text{M}})$ | $t_{\text{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1156 | 400 | 0.0299 | 0.1359 | 0.0058 | 0.0254 | 0.0705 | 0.0665 | 0.1066 |
| 4 | 16900 | 1296 | 0.5347 | 0.5352 | 0.4255 | 0.1728 | 0.9429 | 0.6846 | 1.4733 |
| 6 | 264196 | 17424 | 7.3424 | 7.9156 | 23.5576 | 16.7765 | 14.1433 | 10.1504 | 22.3347 |
| 8 | 4202500 | 266256 | 107.9652 | 121.7985 | 1516.5229 | 970.6061 | 238.5717 | 155.4827 | 370.6186 |

Table 15: *Ex. 3.* Time for assembling and solving the systems that generate $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. uniform ref. steps.

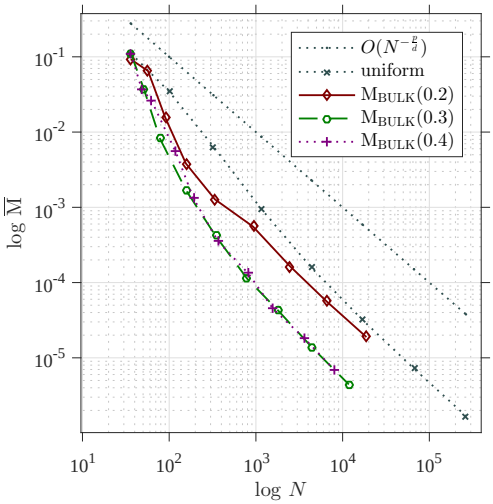



Figure 7: Convergence of the majorant for different marking criteria.

in Table 16, the corresponding time expenses are summarised in Table 17. The most efficient error decrease is obtained for the bulk parameter $\theta = 0.4$, which can be detected from Figure 7.

Figure 8 presents the evolution of physical meshes obtained during the refinement steps with different marking criteria $\mathbb{M}_{\mathbf{BULK}}(0.4)$ and $\mathbb{M}_{\mathbf{BULK}}(0.2)$. Again, it is easy to observe from the graphics that the percentage of the refined elements on the right is higher than the percentage of such elements on the left.

When the exact solution contains large local changes in the gradient (such as the one in current example), the assembling and solving the system (25) becomes harder than respective procedures for (23). This can be explained by the size of generated optimal system (25) providing the reconstruction of vector-valued $\boldsymbol{y}_h$. This drawback can be possibly eliminated by introducing multi-threading techniques (e.g., OpenMP, MPI) into the implementation of THB-splines. However, this matter stays beyond the focus of current paper and will be addressed in the upcoming technical report.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| (a) $\theta = 0.4$ | | | | | | | |
| 2 | 4.0502e-02 | 2.8670e-01 | 1.0725e-01 | 6.3030e-01 | 7.0786 | 5.5395 | 3.8159 |
| 4 | 5.2223e-03 | 3.7058e-02 | 1.5518e-02 | 7.5657e-02 | 7.0960 | 8.9835 | 4.7759 |
| 6 | 8.7993e-04 | 2.1154e-03 | 9.8705e-04 | 3.9631e-03 | 2.4040 | 8.6122 | 3.3554 |
| 8 | 1.1156e-04 | 1.9665e-04 | 1.1451e-04 | 2.8852e-04 | 1.7627 | 9.5564 | 2.8809 |
| (b) $\theta = 0.2$ | | | | | | | |
| 2 | 3.6612e-02 | 2.0753e-01 | 8.3292e-02 | 4.3637e-01 | 5.6683 | 5.7646 | 3.0397 |
| 4 | 1.3527e-03 | 4.0090e-03 | 1.6115e-03 | 8.4210e-03 | 2.9637 | 9.3503 | 5.0006 |
| 6 | 1.5416e-04 | 3.1033e-04 | 1.6976e-04 | 4.9375e-04 | 2.0130 | 10.0080 | 2.0152 |
| 8 | 1.7351e-05 | 2.2095e-05 | 1.7587e-05 | 1.5832e-05 | 1.2734 | 10.4611 | 2.1194 |

Table 16: *Ex. 3.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| (a) $\theta = 0.4$ | | | | | | | | | |
| 2 | 124 | 145 | 0.0547 | 0.4156 | 0.0002 | 0.0025 | 0.0532 | 0.1738 | 0.1418 |
| 4 | 243 | 245 | 0.2481 | 2.6372 | 0.0008 | 0.0063 | 0.3529 | 0.9892 | 0.9102 |
| 6 | 736 | 633 | 0.7903 | 10.7018 | 0.0052 | 0.0393 | 0.9605 | 3.5833 | 2.2969 |
| 8 | 2460 | 2231 | 2.3106 | 33.6222 | 0.0349 | 0.4035 | 2.9163 | 11.5602 | 4.4633 |
| (b) $\theta = 0.2$ | | | | | | | | | |
| 2 | 140 | 160 | 0.0449 | 0.3684 | 0.0002 | 0.0028 | 0.0651 | 0.1682 | 0.1111 |
| 4 | 366 | 366 | 0.2435 | 2.7071 | 0.0015 | 0.0124 | 0.3104 | 1.0138 | 0.5248 |
| 6 | 2043 | 1883 | 1.8328 | 25.3480 | 0.0246 | 0.3000 | 2.0493 | 8.0682 | 3.4391 |
| 8 | 15373 | 13974 | 17.4622 | 264.3344 | 0.4683 | 8.7014 | 15.9445 | 58.1277 | 25.9898 |

Table 17: *Ex. 3.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.

**Example 4** One of the classical benchmark examples (containing a singularity in the exact solution) is the problem with a $L$-shaped domain $\Omega := (\text{-}1,1) \times (\text{-}1,1) \backslash [0,1) \times [0,1)$. The Dirichlet BC are defined on $\Gamma$ by the load $u_D = r^{1/3} \sin(\theta)$, where

$$r = (x_1^2 + x_2^2) \quad \text{and} \quad \theta = \begin{cases} \frac{1}{3}\,(2\,\mathrm{atan2}(x_2, x_1) - \pi), & x_2 > 0, \\ \frac{1}{3}\,(2\,\mathrm{atan2}(x_2, x_1) + 3\,\pi), & x_2 \leq 0. \end{cases}$$

The corresponding exact solution $u = u_D$ has the singularity in the point $(r, \theta) = (0, 0)$ (see also Figure 9a). The initial geometry data and the mesh are defined by coefficients

$$C = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & -1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{27}$$

marked with red marker on Figure 9b, and knots-vectors $\kappa = \{0, 0, 0.5, 1, 1\}$ and $s = \{0, 0, 1, 1\}$. In order to provide the required regularity for $u_h$ and $\boldsymbol{y}_h$, we perform the degree elevation such that the mesh $\mathcal{K}_h^{u_h}$ for $u_h \in S_h^{p,p}$ $(p = 2)$ is based on knot-vectors

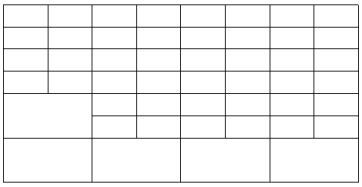
$$\kappa = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\} \quad \text{and} \quad s = \{0, 0, 0, 1, 1, 1\},$$

whereas the mesh $\mathcal{K}_h^{\boldsymbol{y}_h}$ for $\boldsymbol{y}_h \in S_h^{q,q} \oplus S_h^{q,q}$ $(q = 3)$ is based on

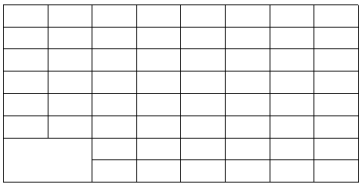
$$\kappa = \{0, 0, 0, 0, 0.5, 0.5, 0.5, 0.5, 1, 1, 1, 1\} \quad \text{and} \quad s = \{0, 0, 0, 0, 1, 1, 1, 1\} \tag{28}$$

for each component $y_h^{(i)}$, $i = 1, 2$, of vector $\boldsymbol{y}_h$.

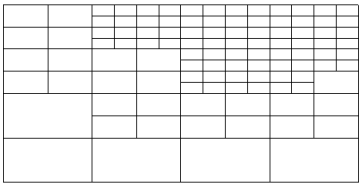
The performance of the error majorant is compared to the performance of the residual error indicator in Table 18, where the first one is constructed with the help of fluxes of different smoothness, i.e., $\boldsymbol{y}_h \in S_h^{3,3} \oplus S_h^{3,3}$ (case (a)) and $\boldsymbol{y}_h \in S_h^{5,5} \oplus S_h^{5,5}$ (case (b)). The results in this table demonstrate that by increasing the degree of splines that approximate $\boldsymbol{y}_h$, we reconstruct a sharper $\overline{\mathrm{M}}$. The residual error estimate (dependent only on $u_h$ and local $h_K$) stays always on the same 'accuracy level', i.e., $I_{\mathrm{eff}}(\overline{\eta}) \approx 13.98$. However, the time required for



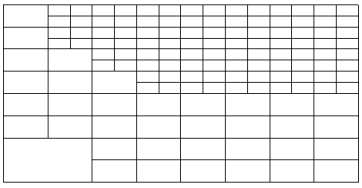
(a) ref. # 3: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



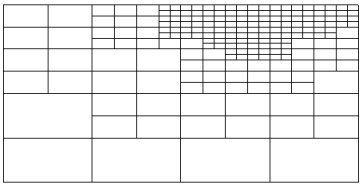
(b) ref. # 3: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



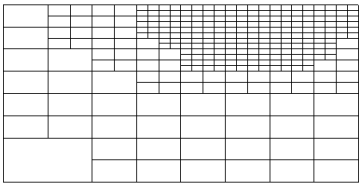
(c) ref. # 4: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



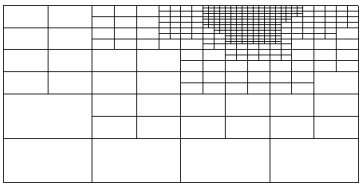
(d) ref. # 4: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



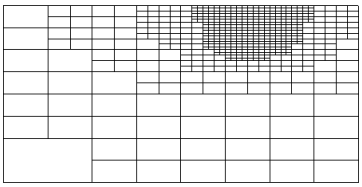
(e) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



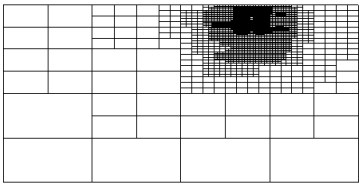
(f) ref. # 5: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



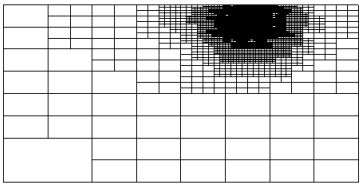
(g) ref. #6: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



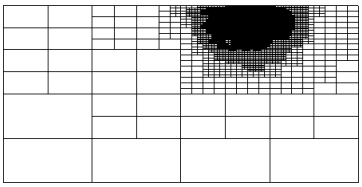
(h) ref. # 6: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$



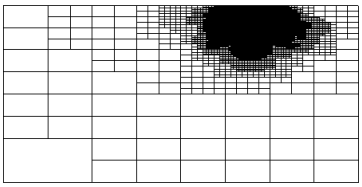
(i) ref. # 8: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



(j) ref. # 8: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$

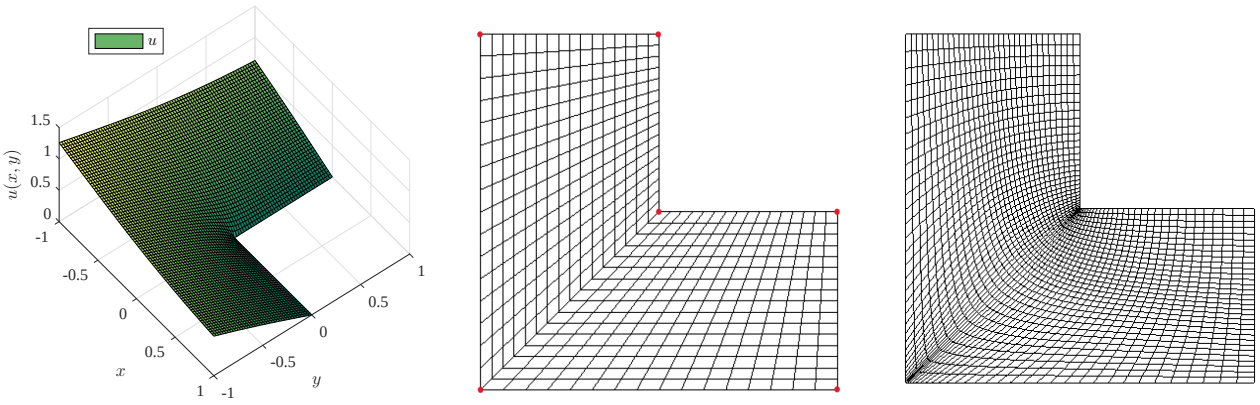


(k) ref. # 9: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.4))$



(l) ref. # 9: $\mathcal{K}_h(\mathbb{M}_{\mathbf{BULK}}(0.2))$

Figure 8: *Ex. 3.* Adaptive meshes obtained for the bulk parameters $\theta = 0.2$ (left) and $\theta = 0.4$ (right).




(a) $u(x_1, x_2)$ (b) $\mathcal{K}_h$ defined on $\Omega$ (c) $\mathcal{K}_h$ defined on $\Omega$

Figure 9: *Ex. 4.* (a) Exact solution $u = r^{1/3}\sin(\theta)$. (b) Initial geometry data with Greville's points (27) and knots (28) as well as a corresponding mesh generated with $C^0$-continuous geometrical mapping. (c) Initial geometry data with Greville's points with double control points at the corners and a corresponding mesh generated with $C^1$-continuous geometrical mapping.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| | | | (a) $\boldsymbol{y}_h \in S_h^{3,3} \oplus S_h^{3,3}$ | | | | |
| 2 | 2.5892e-02 | 4.7272e-02 | 3.9162e-02 | 1.8016e-02 | 1.8258 | 13.9232 | 18.6654 |
| 4 | 1.0407e-02 | 2.2730e-02 | 1.7637e-02 | 1.1316e-02 | 2.1842 | 13.9855 | 6.2019 |
| 6 | 4.1604e-03 | 1.0524e-02 | 8.0432e-03 | 5.5107e-03 | 2.5295 | 14.0017 | 3.2355 |
| 8 | 1.6662e-03 | 4.8518e-03 | 3.7789e-03 | 2.3835e-03 | 2.9119 | 13.9991 | 2.1746 |
| | | | (b) $\boldsymbol{y}_h \in S_h^{5,5} \oplus S_h^{5,5}$ | | | | |
| 2 | 2.5925e-02 | 3.5626e-02 | 3.2016e-02 | 8.0202e-03 | 1.3742 | 13.9366 | 21.3045 |
| 4 | 1.0447e-02 | 1.5141e-02 | 1.3362e-02 | 3.9514e-03 | 1.4493 | 13.9585 | 6.0421 |
| 6 | 4.1479e-03 | 6.2243e-03 | 5.5552e-03 | 1.4864e-03 | 1.5006 | 13.9891 | 3.1337 |
| 8 | 1.6439e-03 | 2.6393e-03 | 2.3495e-03 | 6.4376e-04 | 1.6055 | 13.9757 | 1.8647 |

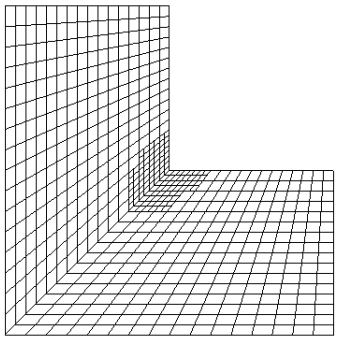
Table 18: *Ex. 4.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.$(u_h)$ | # d.o.f.$(\boldsymbol{y}_h)$ | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| | | | | (a) $\boldsymbol{y}_h \in S_h^{3,3} \oplus S_h^{3,3}$ | | | | | |
| 2 | 661 | 734 | 0.1490 | 1.2537 | 0.0017 | 0.0180 | 0.2305 | 0.5557 | 0.4514 |
| 4 | 829 | 893 | 0.3026 | 2.8881 | 0.0029 | 0.0308 | 0.4250 | 1.1521 | 0.7439 |
| 6 | 1371 | 1385 | 0.7033 | 7.1900 | 0.0078 | 0.0505 | 1.0406 | 2.8226 | 1.7063 |
| 8 | 3218 | 3119 | 1.9178 | 26.0401 | 0.0422 | 0.1966 | 2.3883 | 7.5138 | 3.9977 |
| | | | | (b) $\boldsymbol{y}_h \in S_h^{5,5} \oplus S_h^{5,5}$ | | | | | |
| 2 | 657 | 888 | 0.1796 | 8.0567 | 0.0028 | 0.0931 | 0.2689 | 3.4699 | 0.5032 |
| 4 | 812 | 1031 | 0.3555 | 23.4200 | 0.0043 | 0.0821 | 0.5937 | 6.3704 | 0.9918 |
| 6 | 1410 | 1523 | 0.7036 | 84.8119 | 0.0089 | 0.3560 | 0.9859 | 26.1258 | 1.6313 |
| 8 | 3496 | 3251 | 2.0176 | 189.9995 | 0.0451 | 0.7057 | 2.8277 | 64.8042 | 4.6431 |

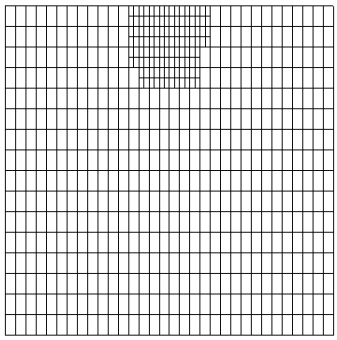
Table 19: *Ex. 4.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.

reconstruction of $\overline{\mathrm{M}}$, increases as well (see Table 19). Hence, the selection of space for the dual variable $\boldsymbol{y}_h$ is dictated by the smoothness of exact solution (or RHS) and by possible restrictions on the allocated time for the a posteriori error estimates control.

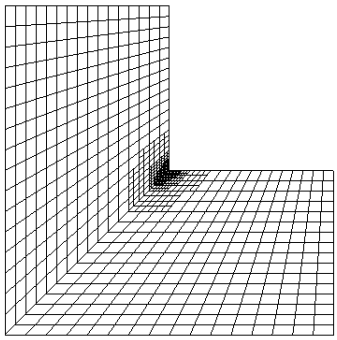
In Figure 10, we illustrate the evolution of the adaptive meshes discretising physical domain $\Omega$ (left) and corresponding to them meshes discretising $\widehat{\Omega}$. From presented graphics one can see that the refinement is localised in the area close to the singularity point and no superfluous refinement is performed. We also perform the test, where, starting with the same initial mesh Figure 9b, we compare $\mathcal{K}_h$ generated by the refinement based on the majorant (error indicator $\overline{\mathrm{m}}^2_{\mathrm{d},K}$) and by the refinement based on true error distribution $\|\nabla_x e\|^2_K$. These meshes



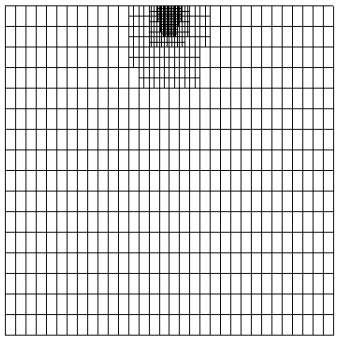
(a) ref. # 2: $\Omega$ and $\mathcal{K}_h$



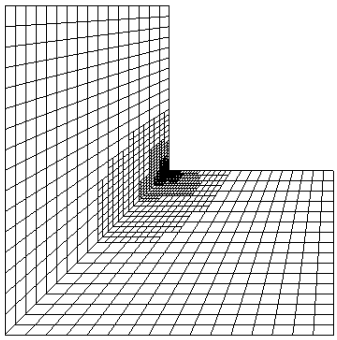
(b) ref. # 2: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$



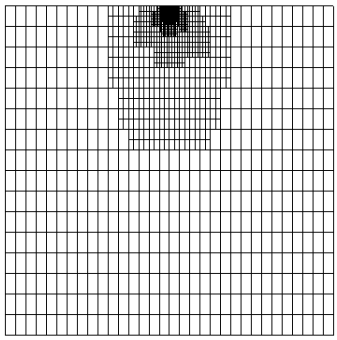
(c) ref. # 4: $\Omega$ and $\mathcal{K}_h$



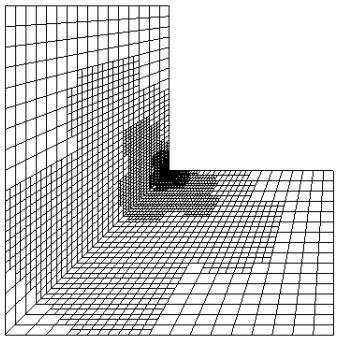
(d) ref. # 4: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$



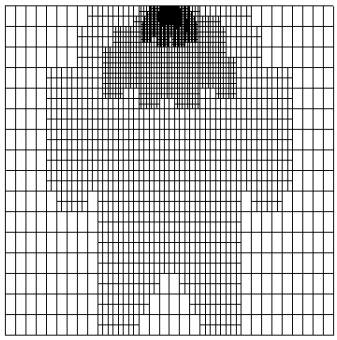
(e) ref. # 6: $\Omega$ and $\mathcal{K}_h$



(f) ref. # 6: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$

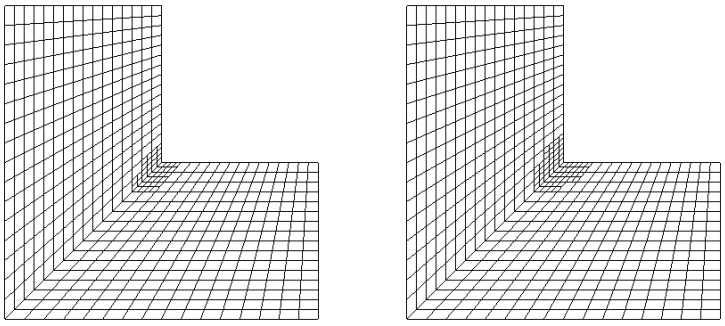


(g) ref. # 8: $\Omega$ and $\mathcal{K}_h$

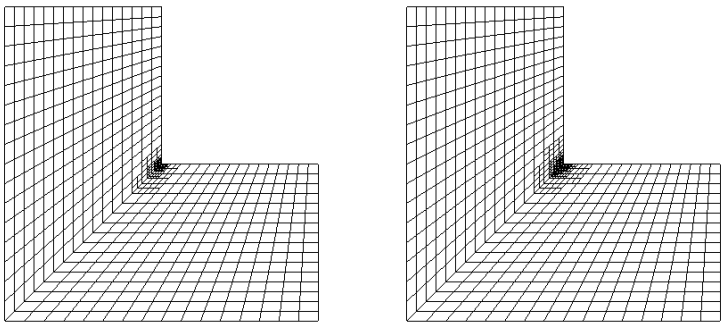


(h) ref. # 8: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$

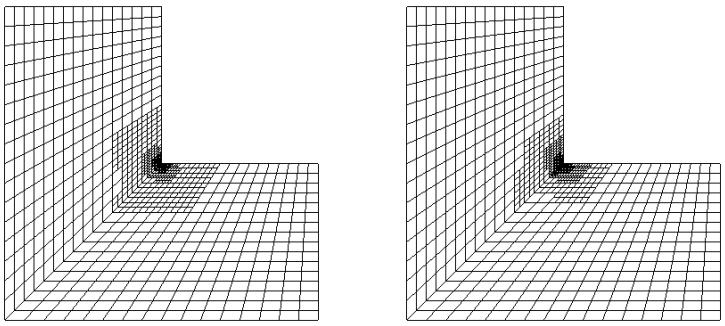
Figure 10: *Ex. 4*. Comparison of meshes on the physical and parametrical domains w.r.t. adaptive ref. steps.



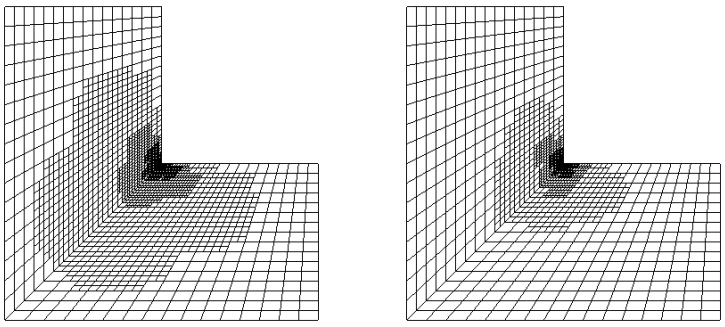
(a) ref. # 2: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (b) ref. # 2: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(c) ref. # 4: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (d) ref. # 4: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(e) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (f) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(g) ref. # 8: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (h) ref. # 8: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$

Figure 11: Comparison of meshes generated by two refinement strategies, i.e., the true error (left) and error indicator provided by the majorant (right), with the marking criterion $\mathbb{M}_{\mathbf{BULK}}(0.6)$.

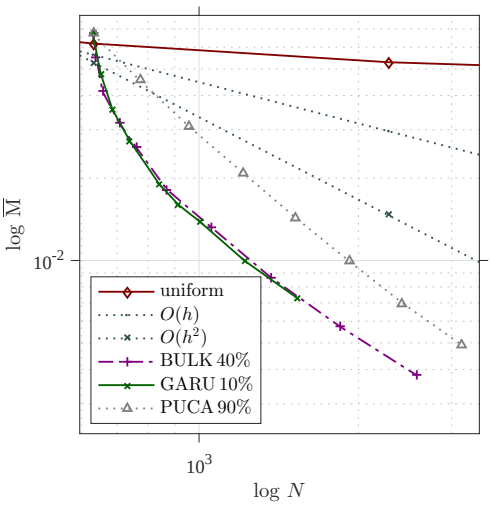



Figure 12: Convergence of majorant.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 2 | 1.3359e-02 | 3.6140e-02 | 2.7595e-02 | 1.8983e-02 | 2.7053 | 42.6770 | 5.5339 |
| 3 | 8.7388e-03 | 2.3661e-02 | 1.8061e-02 | 1.2442e-02 | 2.7076 | 55.7909 | 6.0339 |
| 4 | 5.7148e-03 | 1.5862e-02 | 1.1998e-02 | 8.5837e-03 | 2.7755 | 74.0819 | 4.5481 |
| 5 | 3.7337e-03 | 1.0751e-02 | 8.0370e-03 | 6.0286e-03 | 2.8794 | 99.4303 | 2.4711 |
| 6 | 2.4386e-03 | 7.3896e-03 | 5.4326e-03 | 4.3474e-03 | 3.0303 | 134.3125 | 1.7454 |
| 7 | 1.5774e-03 | 4.8596e-03 | 3.6979e-03 | 2.5806e-03 | 3.0807 | 183.3358 | 1.3410 |

Table 20: *Ex. 4.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps, $\boldsymbol{y}_h \in S_h^{3,3} \oplus S_h^{3,3}$.

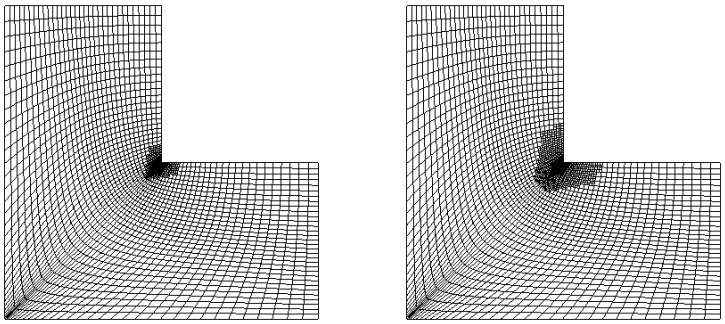
are illustrated in Figure 11, and it is obvious that the majorant provides an adequate strategy for the adaptive refinement. The efficiency of the studied error bounds is also confirmed by comparing of majorant decrease for different marking criteria with respect to the # d.o.f.$(u_h)$. In particular, Figure 12 shows that using majorant in combination with the marking criterion $\mathbb{M}_{\mathbf{BULK}}(0.4)$ does not only improve e.o.c. (in comparison to the one provided by the uniform refinement) but also provides an even better one than $p = 2$.

The above considered geometrical configuration exploited $C^0$-continuous mapping $\Phi$ between $\widehat{\Omega}$ and $\Omega$. One can also consider a mesh generated by a different $C^1$-continuous geometrical mapping (illustrated on Figure 9c). The initial geometry data and the mesh are defined by Greville's points with double control points in the corners. For this setting in Figure 13, we verify the efficiency of the error majorant by comparing $\mathcal{K}_h$ generated during the refinement based on the error indicator $\overline{\mathrm{m}}^2_{\mathrm{d},K}$ and the refinement based on true error distribution $\|\nabla_x e\|^2_K$. The advantage of using the error majorant instead of residual-based error estimates for such a problem can be observed from Table 20. It demonstrates that the latter one overestimates the error 183 times and the efficiency index continues to grow.

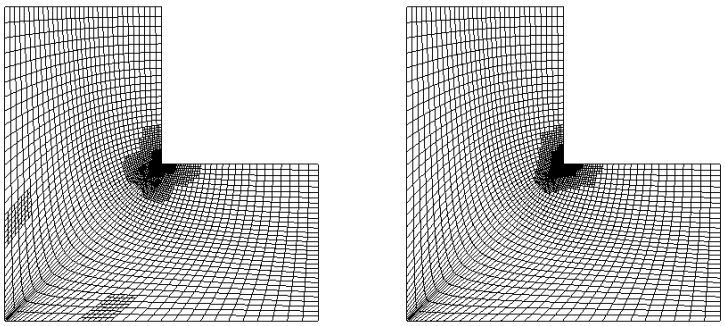
**Example 5** The final example with a two-dimensional domain is defined on a quoter-annulus with the following exact solution and RHS:

$$\begin{aligned} u &= \cos x_1\, e^{x_2} \quad \text{in} \quad \Omega, \\ f &= 0 \quad \text{in} \quad \Omega, \\ u_D &= \cos x_1\, e^{x_2} \quad \text{on} \quad \Gamma. \end{aligned}$$

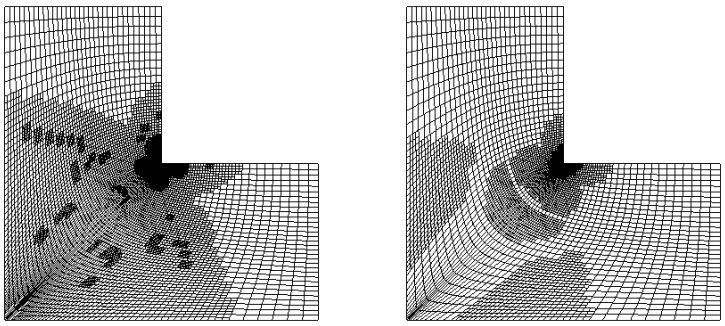
Due to the IgA framework, the Dirichlet BC for the approximation $u_h$ are fully satisfied, therefore, functional error estimates can be applied for the domains with curved boundaries providing a fully reliable (not heuristic) error control. The results obtained for the adaptive refinement for $u_h \in S_h^{2,2}$, $\boldsymbol{y} \in S_{4h}^{4,4} \oplus S_{4h}^{4,4}$, and bulk parameter $\theta = 0.4$, are illustrated in Tables 21–22. Again, even for such specific geometry, the generation of the optimal $\boldsymbol{y}_h$ (assembling and solving the (25)) requires several times less computational effort than $u_h$: $\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx \frac{38.5360}{19.1958} \approx 2$ and $\frac{t_{\mathrm{sol}}(u_h)}{t_{\mathrm{sol}}(\boldsymbol{y}_h)} \approx \frac{3.4323}{0.4727} \approx 7$. Moreover, in Figure 15, we illustrate the evolution of the meshes $\widehat{\mathcal{K}}_h$ and $\mathcal{K}_h$ on the parametric $\widehat{\Omega}$ and physical $\Omega$ domains, respectively.



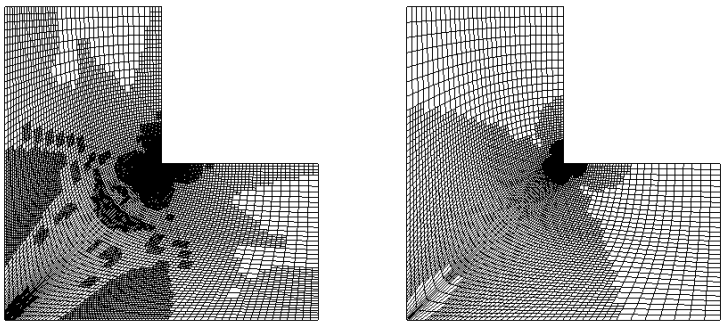
(a) ref. # 2: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (b) ref. # 2: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(c) ref. # 4: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (d) ref. # 4: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(e) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (f) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$



(g) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \|\nabla_x e\|_K^2$ (h) ref. # 6: $\widehat{\mathcal{K}}_h \leftarrow \overline{\mathrm{m}}_{\mathrm{d},K}^2$

Figure 13: Comparison of the meshes generated by two refinement strategies, i.e., the true error (left) and error indicator provided by the majorant (right), with the marking criterion $\mathbb{M}_{\mathbf{BULK}}(0.2)$.

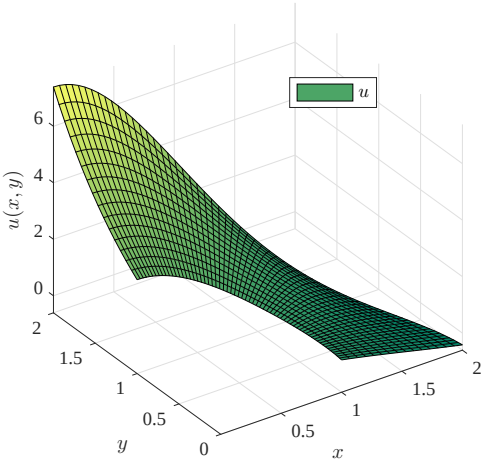



Figure 14: *Ex. 5.* Exact solution $cos(x_1)\, e^{x_2}$.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 1.9377e-03 | 2.2682e-03 | 2.1466e-03 | 2.7011e-04 | 1.1706 | 8.0838 | 1.8290 |
| 5 | 2.7731e-04 | 3.9296e-04 | 3.7669e-04 | 3.6143e-05 | 1.4171 | 8.2886 | 2.1330 |
| 7 | 4.7689e-05 | 6.6456e-05 | 5.5448e-05 | 2.4453e-05 | 1.3935 | 8.3850 | 2.1653 |

Table 21: *Ex. 5.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 324 | 400 | 0.1059 | 1.7198 | 0.0010 | 0.0245 | 0.1592 | 0.7331 | 0.3351 |
| 3 | 1389 | 400 | 0.4475 | 1.0655 | 0.0081 | 0.0166 | 0.5914 | 0.9428 | 1.1633 |
| 5 | 9125 | 400 | 5.0846 | 1.1225 | 0.1925 | 0.0363 | 5.1422 | 9.5749 | 9.1553 |
| 7 | 50291 | 1623 | 38.5360 | 19.1958 | 3.4323 | 0.4727 | 21.0657 | 96.3344 | 41.5038 |

Table 22: *Ex. 5.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.

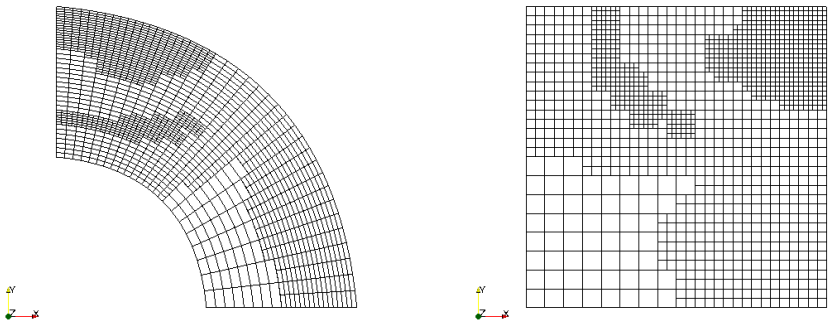
| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 2 | 9.7268e-04 | 1.1328e-03 | 1.0104e-03 | 6.6641e-04 | 1.1646 | 14.2071 | 4.9421 |
| 4 | 5.7843e-05 | 7.9238e-05 | 7.8966e-05 | 1.4807e-06 | 1.3699 | 13.4654 | 2.7348 |
| 6 | 3.6029e-06 | 4.6376e-06 | 4.5126e-06 | 6.8010e-07 | 1.2872 | 13.4195 | 2.1808 |
| 8 | 2.2513e-07 | 2.8772e-07 | 2.7822e-07 | 5.1726e-08 | 1.2780 | 13.4166 | 2.0451 |

Table 23: *Ex. 6.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. uniform ref. steps.

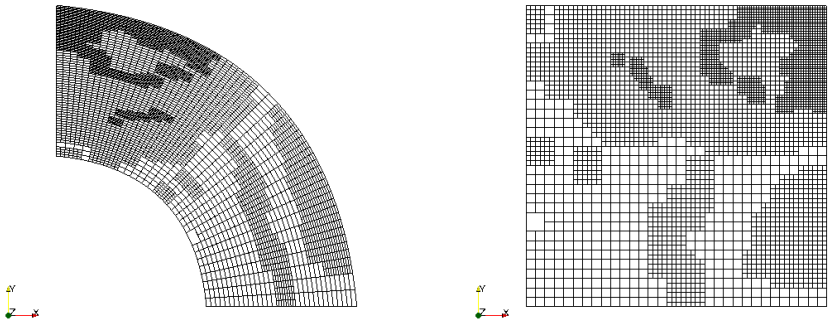
**Example 6** Last two examples are dedicated to three-dimensional problems. Let $\Omega = (0,1)^3 \in \mathbb{R}^3$,

$$\begin{aligned} u &= (1-x_1)\, x_1^2\, (1-x_2)\, x_2^2\, (1-x_3)\, x_3^2 \quad \text{in} \quad \Omega, \\ u_D &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{on} \quad \Gamma. \end{aligned}$$

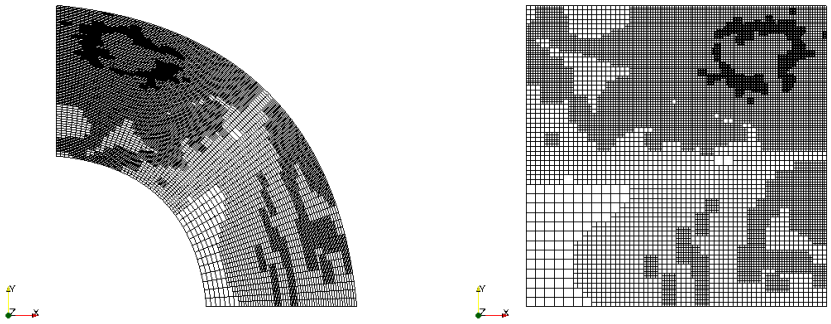
The uniform refinement strategy assuming that $u_h \in S_h^{2,2}$ and $\boldsymbol{y}_h \in S_{6h}^{3,3} \oplus S_{6h}^{3,3}$ provides numerical results illustrated in Tables 23 and 24. Comparison of the majorant performance to the accuracy of residual error estimates, i.e., $I_{\mathrm{eff}}(\overline{\mathrm{M}}) = 1.2378$ and $I_{\mathrm{eff}}(\overline{\eta}) = 13.4166$, confirms that the latter one always overestimates the error (even for such a smooth exact solution). Moreover, the computational costs of majorant generation is a hundred times less than the computational time for a primal variable, namely, $\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx 578$, and $\frac{t_{\mathrm{sol}}(u_h)}{t_{\mathrm{sol}}(\boldsymbol{y}_h)} \approx 136$. It is important to note that values of the last three columns of Table 24 illustrate sub-optimal time for the element-wise evaluation. As mentioned earlier, this issue is related to the implementation of element-wise iterator currently used in G+Smo and will be addressed in the follow-up reports. The results obtained while performing an adaptive refinement strategy with the bulk marking criterion $\mathbb{M}_{\mathbf{BULK}}(0.4)$ are summarised in Tables 25–26. The obtained ratios between majorants and the true error as well as the computational time it requires to be



(a) ref. # 3: $\Omega$ and $\mathcal{K}_h$ (b) ref. # 3: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$



(c) ref. # 4: $\Omega$ and $\mathcal{K}_h$ (d) ref. # 4: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$



(e) ref. # 5: $\Omega$ and $\mathcal{K}_h$ (f) ref. # 5: $\widehat{\Omega}$ and $\widehat{\mathcal{K}}_h$

Figure 15: *Ex. 5.* Comparison of meshes on the physical and parametrical domains w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.($u_h$) | # d.o.f.($\boldsymbol{y}_h$) | $t_{\text{as}}(u_h)$ | $t_{\text{as}}(\boldsymbol{y}_h)$ | $t_{\text{sol}}(u_h)$ | $t_{\text{sol}}(\boldsymbol{y}_h)$ | $t_{\text{e/w}}(\|\nabla_x e\|)$ | $t_{\text{e/w}}(\overline{\text{M}})$ | $t_{\text{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 64 | 64 | 0.0121 | 0.0292 | 0.0000 | 0.0131 | 0.0013 | 0.0079 | 0.0065 |
| 4 | 1000 | 64 | 0.1535 | 0.0477 | 0.0005 | 0.0084 | 0.1913 | 0.2480 | 0.2962 |
| 6 | 39304 | 64 | 6.2179 | 0.0529 | 0.2208 | 0.0106 | 11.7921 | 15.9026 | 17.5110 |
| 8 | 2197000 | 343 | 324.7514 | 0.5641 | 54.6425 | 0.1760 | 560.9245 | 452.5727 | 865.4832 |

Table 24: *Ex. 6.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. uniform ref. steps.

generated, i.e.,

$$\frac{t_{\text{as}}(u_h)}{t_{\text{as}}(\boldsymbol{y}_h)} \approx \frac{324.75145}{0.5641} \approx 575 \quad \text{and} \quad \frac{t_{\text{sol}}(u_h)}{t_{\text{sol}}(\boldsymbol{y}_h)} \approx \frac{54.6425}{0.1760} \approx 310,$$

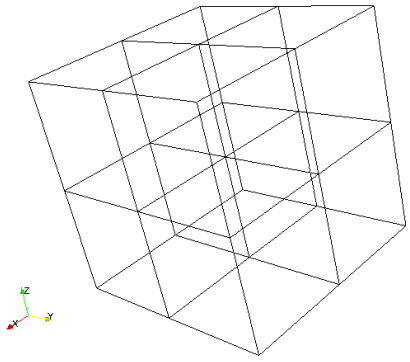
confirm the efficiency of the functional approach to the error control. The mesh evolution for the considered adaptive refinement strategy is illustrated in Figure 16. One can see that the main refinement is performed closer to the centre of the computational domain $\Omega$, which is similar to the results obtained for two-dimensional case.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 2.3387e-04 | 2.7678e-04 | 2.7288e-04 | 2.1210e-05 | 1.1835 | 13.6135 | 3.5152 |
| 5 | 1.9918e-05 | 2.7263e-05 | 2.3917e-05 | 1.8207e-05 | 1.3687 | 11.7620 | 2.2298 |
| 7 | 2.2272e-06 | 3.1275e-06 | 2.9019e-06 | 1.2276e-06 | 1.4042 | 12.4614 | 2.0710 |

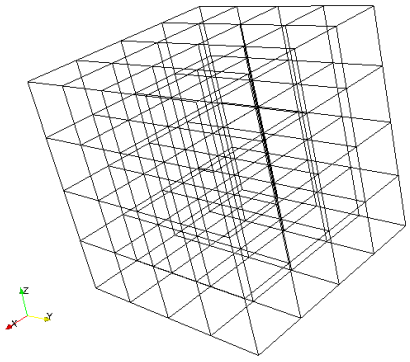
Table 25: *Ex. 6.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.$(u_h)$ | # d.o.f.$(\boldsymbol{y}_h)$ | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 27 | 64 | 0.0146 | 0.0718 | 0.0000 | 0.0049 | 0.0066 | 0.0445 | 0.0213 |
| 3 | 216 | 64 | 0.3379 | 0.0773 | 0.0001 | 0.0107 | 0.4165 | 0.8699 | 1.0376 |
| 5 | 4074 | 64 | 21.5658 | 0.0837 | 0.0420 | 0.0074 | 26.7396 | 39.1294 | 50.8124 |
| 7 | 127265 | 64 | 3054.9337 | 0.0838 | 3.2532 | 0.0087 | 1415.2231 | 1720.7416 | 2433.3058 |

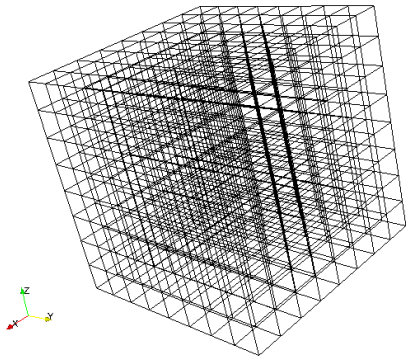
Table 26: *Ex. 6.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.



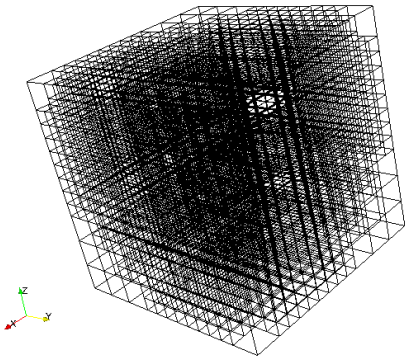
(a) ref. # 1: $\mathcal{K}_h$ with # d.o.f.$(u_h) = 64$



(b) ref. # 2: $\mathcal{K}_h$ with # d.o.f.$(u_h) = 216$



(c) ref. # 3: $\mathcal{K}_h$ with # d.o.f.$(u_h) = 963$, $\mathcal{K}_h$



(d) ref. # 4: $\mathcal{K}_h$ with # d.o.f.$(u_h) = 4074$

Figure 16: *Ex. 6.* Evolution of meshes on the physical unit cube w.r.t. adaptive ref. steps with bulk marking criterion $\mathbb{M}_{\mathrm{BULK}}(0.4)$.

**Example 7** Finally, in the last example, we test derived error estimates on a complex three-dimensional geometry of a *G*-shape. The exact solution, RHS, and Dirichlet BC are defined as follows:

$$\begin{aligned} u &= 10\,\cos x_1\, e^{x_2}\, x_3 && \text{in} \quad \Omega, \\ f &= 0 && \text{in} \quad \Omega, \\ u_D &= 10\,\cos x_1\, e^{x_2}\, x_3 && \text{on} \quad \Gamma. \end{aligned}$$

Let us consider $N_{\mathrm{ref},0} = 1$ initial uniform refinement of the mesh assigned to the geometry (illustrated in Figure 17). The numerical results of 5 adaptive ref. steps are summarised in Tables 27–28. From Table 27 analysing the efficiency of the error estimates, it is obvious that $\overline{\mathrm{M}}$ is at least 40 times sharper than the residual

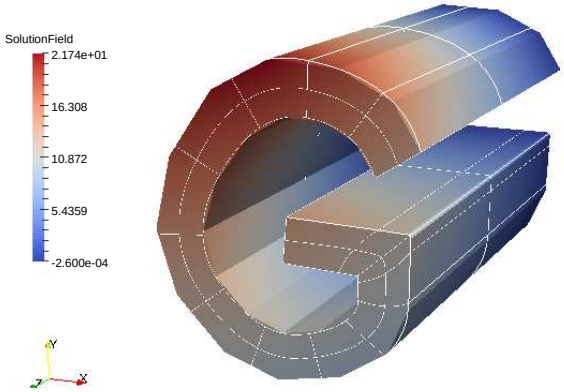



Figure 17: *Ex. 7.* Exact solution $u = 10 \cos x_1 \, e^{x_2} \, x_3$.

| # ref. | $\|\nabla_x e\|_\Omega$ | $\overline{\mathrm{M}}$ | $\overline{\mathrm{m}}_{\mathrm{d}}$ | $\overline{\mathrm{m}}_{\mathrm{eq}}$ | $I_{\mathrm{eff}}(\overline{\mathrm{M}})$ | $I_{\mathrm{eff}}(\overline{\eta})$ | e.o.c. |
|---|---|---|---|---|---|---|---|
| 3 | 1.5495e-02 | 1.6552e-02 | 1.6016e-02 | 2.9195e-03 | 1.0682 | 42.1400 | 3.1601 |
| 4 | 5.5530e-03 | 6.7468e-03 | 6.0466e-03 | 3.8103e-03 | 1.2150 | 59.4362 | 3.0093 |
| 5 | 2.3080e-03 | 3.6335e-03 | 2.7857e-03 | 4.6130e-03 | 1.5743 | 47.1796 | 2.5110 |

Table 27: *Ex. 7.* Error, majorant (with dual and reliability terms), efficiency indices, error, e.o.c. w.r.t. adaptive ref. steps.

| # ref. | # d.o.f.$(u_h)$ | # d.o.f.$(\boldsymbol{y}_h)$ | $t_{\mathrm{as}}(u_h)$ | $t_{\mathrm{as}}(\boldsymbol{y}_h)$ | $t_{\mathrm{sol}}(u_h)$ | $t_{\mathrm{sol}}(\boldsymbol{y}_h)$ | $t_{\mathrm{e/w}}(\|\nabla_x e\|)$ | $t_{\mathrm{e/w}}(\overline{\mathrm{M}})$ | $t_{\mathrm{e/w}}(\overline{\eta})$ |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 1108 | 575 | 8.5398 | 3.4636 | 0.0056 | 3.2101 | 11.8510 | 7.5239 | 21.5933 |
| 4 | 3082 | 575 | 31.4262 | 2.9348 | 0.0686 | 3.7503 | 38.9502 | 26.7215 | 77.4521 |
| 5 | 8798 | 575 | 97.6563 | 7.1416 | 0.2659 | 3.1107 | 122.7187 | 101.5395 | 229.4474 |

Table 28: *Ex. 7.* Time for assembling and solving the systems generating d.o.f. of $u_h$ and $\boldsymbol{y}_h$ as well as the time spent on e/w evaluation of error, majorant, and residual error estimator w.r.t. adaptive ref. steps.

error indicator. If one analyses the computation costs, it is easy to notice that the assembling time of the system (25) is better than the assembling time of (23), i.e., $\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx 14$. However, the costs for solving the system (25) is higher, i.e., $\frac{t_{\mathrm{as}}(u_h)}{t_{\mathrm{as}}(\boldsymbol{y}_h)} \approx \frac{1}{12}$. The element-wise evaluation of $\|\nabla_x e\|$, $\overline{\mathrm{M}}$, and $\overline{\eta}$ must take at least as long as $t_{\mathrm{as}}(u_h)$ (since both are performed with $u_h$ element-wise), which is confirmed in the last three columns of Table 28.

The evolution of the meshes on both physical and parametric domains is illustrated in Figure 18. The final solution and meshes obtained on the final step are presented in Figure 19. The latter plots are presented from the axis $O_z$ view-point so that the reader can clearly see the finer parts of the physical and parametric meshes.

**Acknowledgments** The research is supported by the Austrian Science Fund (FWF) through the NFN S117-03 project. Implementation was carried out using the open-source C++ library G+Smo [38].

# References

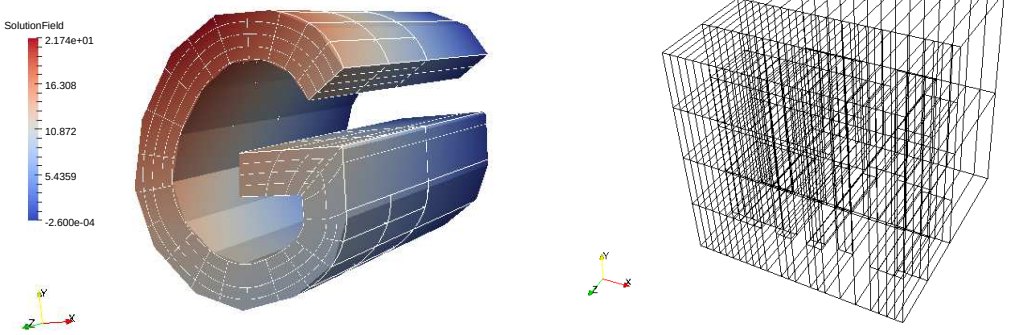
[1] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):229–263, 2010.

[2] M. Bebendorf. A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen*, 22(4):751–756, 2003.

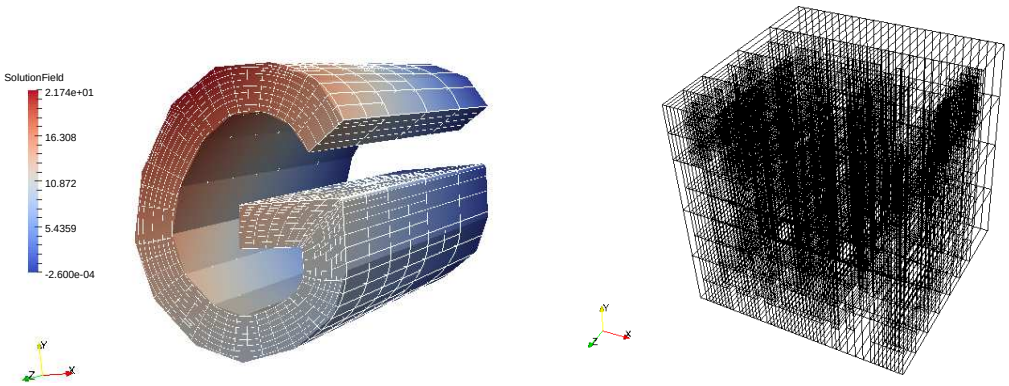
[3] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-suitable T-splines are dual-compatible. *Comput. Methods Appl. Mech. Engrg.*, 249/252:42–51, 2012.

[4] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. IsoGeometric analysis using T-splines on two-patch geometries. *Comput. Methods Appl. Mech. Engrg.*, 200(21-22):1787–1803, 2011.

[5] F. A. Bornemann and P. Deuflhard. The cascadic multigrid method for elliptic problems. *Numer. Math.*, 75(2):135–152, 1996.

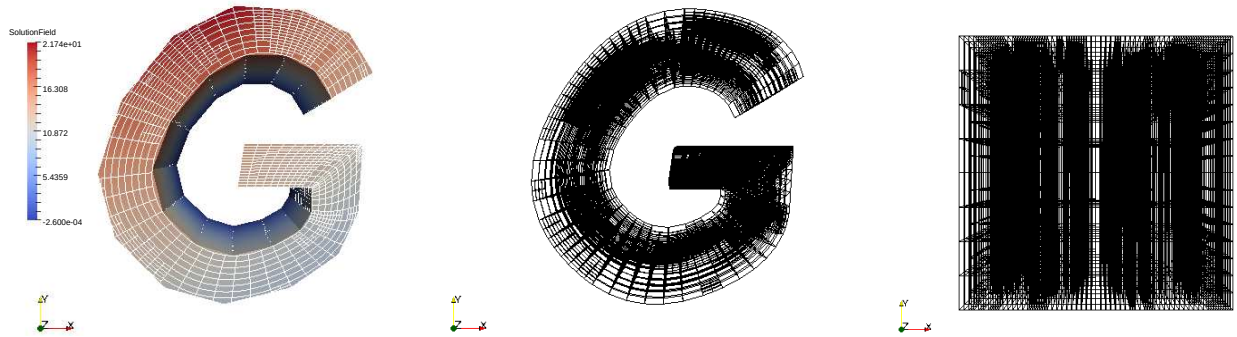



(a) ref. # 1: $u_h$ on $\Omega$ with mesh $\mathcal{K}_h$ (b) ref. # 1: $\widehat{\mathcal{K}}_h$




(c) ref. # 3: $u_h$ on $\Omega$ with mesh $\mathcal{K}_h$ (d) ref. # 3: $\widehat{\mathcal{K}}_h$

Figure 18: *Ex. 7.* Comparison of meshes on the physical and parametrical domains w.r.t. adaptive ref. steps.




(a) ref. # 4: $u_h$ on $\Omega$ with mesh $\mathcal{K}_h$ (b) ref. # 4: $t\mathcal{K}_h$ (c) ref. # 4: $\widehat{\mathcal{K}}_h$

Figure 19: *Ex. 7.* Solution on the domain $\Omega$ and corresponding meshes on physical and parametrical domains, view from the axis $0_z$.

[6] A. Bressan. Some properties of LR-splines. *Comput. Aided Geom. Design*, 30(8):778–794, 2013.

[7] A. Bressan and B. Jüttler. A hierarchical construction of LR meshes in 2D. *Comput. Aided Geom. Design*, 37:9–24, 2015.

[8] A. Buffa and C. Giannelli. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. Technical Report arxiv: 1502.00565, arxiv:math.NA, 2015.

[9] L. Beirão Da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Analysis-suitable T-splines of arbitrary degree: definition, linear independence and approximation properties. *Math. Models Methods Appl. Sci.*, 23(11):1979–2003, 2013.

[10] L. Dedè and H. A. F. A. Santos. B-spline goal-oriented error estimators for geometrically nonlinear rods. *Comput. Mech.*, 49(1):35–52, 2012.

[11] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical t-meshes. *Graph. Models*, 70:76–86, 2008.

[12] P. Deuflhard. Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results. In *Domain decomposition methods in scientific and engineering computing (University Park, PA, 1993)*, volume 180 of *Contemp. Math.*, pages 29–42. Amer. Math. Soc., Providence, RI, 1994.

[13] T. Dokken, T. Lyche, and K. F. Pettersen. Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Design*, 30(3):331–356, 2013.

[14] M. R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local $h$-refinement with T-splines. *Comput. Methods Appl. Mech. Engrg.*, 199(5-8):264–275, 2010.

[15] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[16] E. J. Evans, M. A. Scott, X. Li, and D. C. Thomas. Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 284:1–20, 2015.

[17] D.R. Forsey and R.H. Bartels. Hierarchical b-spline refinement. *Comput. Graphics*, 22:205–212, 1988.

[18] K. Friedrichs. On certain inequalities and characteristic value problems for analytic functions and for functions of two variables. *Trans. Amer. Math. Soc.*, 41(3):321–364, 1937.

[19] C. Giannelli and B. Jüttler. Bases and dimensions of bivariate hierarchical tensor-product splines. *J. Comput. Appl. Math.*, 239:162–178, 2013.

[20] C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, and J. Speh. THB-splines: an effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 299:337–365, 2016.

[21] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: the truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29(7):485–498, 2012.

[22] C. Giannelli, B. Juttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.*, 40(2):459–490, 2014.

[23] G. Greiner and K. Hormann. Interpolating and approximating scattered 3D data with hierarchical tensor product $B$-splines. In *Surface fitting and multiresolution methods (Chamonix–Mont-Blanc, 1996)*, pages 163–172. Vanderbilt Univ. Press, Nashville, TN, 1997.

[24] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.*, 194(39-41):4135–4195, 2005.

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

[26] K. A. Johannessen, F. Remonato, and T. Kvamsdal. On the similarities and differences between classical hierarchical, truncated hierarchical and LR B-splines. *Comput. Methods Appl. Mech. Engrg.*, 291:64–101, 2015.

[27] K.A. Johannessen. *An adaptive isogeometric finite element analysis.* PhD thesis, Master's thesis, Norwegian University of Science and Technology, 2009.

[28] G. Kiss, Carlotta Giannelli, and B. Jüttler. Algorithms and data structures for truncated hierarchical B-splines. In *Mathematical methods for curves and surfaces*, volume 8177 of *Lecture Notes in Comput. Sci.*, pages 304–323. Springer, Heidelberg, 2014.

[29] S. K. Kleiss and S. K. Tomar. Guaranteed and sharp a posteriori error estimates in isogeometric analysis. *Comput. Math. Appl.*, 70(3):167–190, 2015.

[30] R. Kraft. Adaptive and linearly independent multilevel $B$-splines. In *Surface fitting and multiresolution methods (Chamonix–Mont-Blanc, 1996)*, pages 209–218. Vanderbilt Univ. Press, Nashville, TN, 1997.

[31] M. Kumar, T. Kvamsdal, and K. A. Johannessen. Simple a posteriori error estimators in adaptive isogeometric analysis. *Comput. Math. Appl.*, 70(7):1555–1582, 2015.

[32] G. Kuru. *Goal-Adaptive Isogeometric Analysis with Hierarchical Splines.* PhD thesis, Master's thesis, Mechanical Engineering, Eindhoven University of Technology, 2013.

[33] G. Kuru, C. V. Verhoosel, K. G. van der Zee, and E. H. van Brummelen. Goal-adaptive isogeometric analysis with hierarchical splines. *Comput. Methods Appl. Mech. Engrg.*, 270:270–292, 2014.

[34] O. A. Ladyzhenskaya. *The boundary value problems of mathematical physics.* Springer, New York, 1985.

[35] R. Lazarov, S. Repin, and S. K. Tomar. Functional a posteriori error estimates for discontinuous Galerkin approximations of elliptic problems. *Numer. Methods Partial Differential Equations*, 25(4):952–971, 2009.

[36] X. Li, J. Zheng, T. W. Sederberg, Th. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Comput. Aided Geom. Design*, 29(1):63–76, 2012.

[37] O. Mali, P. Neittaanmäki, and S. Repin. *Accuracy verification methods*, volume 32 of *Computational Methods in Applied Sciences*. Springer, Dordrecht, 2014. Theory and algorithms.

[38] A. Mantzaflaris, ..., and others (see website). G+smo (geometry plus simulation modules) v0.8.1. http://gs.jku.at/gismo, 2015.

[39] S. Matculevich, P. Neittaanmäki, and S. Repin. A posteriori error estimates for time-dependent reaction-diffusion problems based on the Payne–Weinberger inequality. *AIMS*, 35(6):2659–2677, 2015.

[40] S. Matculevich and S. Repin. Computable bounds of the distance to the exact solution of parabolic problems based on Poincaré type inequalities. *Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov (POMI)*, 425(1):7–34, 2014.

[41] S. Matculevich and S. Repin. Estimates for the difference between exact and approximate solutions of parabolic equations on the basis of Poincaré inequalities for traces of functions on the boundary. *Differential Equations*, 52(10):1355–1365, 2016.

[42] S. Matculevich and S. Repin. Explicit constants in poincaré-type inequalities for simplicial domains. *Computational Methods in Applied Mathematics - CMAM*, 16(2):277–298, 2016.

[43] P. Morgenstern and D. Peterseim. Analysis-suitable adaptive T-mesh refinement with linear complexity. *Comput. Aided Geom. Design*, 34:50–66, 2015.

[44] A. I. Nazarov and S. I. Repin. Exact constants in Poincare type inequalities for functions with zero mean boundary traces. *Mathematical Methods in the Applied Sciences*, 2014. Puplished in arXiv.org in 2012, math/1211.2224.

[45] P. Neittaanmäki and S. Repin. *Reliable methods for computer simulation*, volume 33 of *Studies in Mathematics and its Applications*. Elsevier Science B.V., Amsterdam, 2004. Error control and a posteriori estimates.

[46] N. Nguyen-Thanh, H. Nguyen-Xuan, S. P. A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Comput. Methods Appl. Mech. Engrg.*, 200(21-22):1892–1908, 2011.

[47] L. E. Payne and H. F. Weinberger. An optimal Poincaré inequality for convex domains. *Arch. Rational Mech. Anal.*, 5:286–292 (1960), 1960.

[48] L. Piegl and W. Tiller. *The NURBS book*. Springer Berlin Heidelberg, 1997.

[49] H. Poincare. Sur les Equations aux Derivees Partielles de la Physique Mathematique. *Amer. J. Math.*, 12(3):211–294, 1890.

[50] H. Poincare. Sur les Equations de la Physique Mathematique. *Rend. Circ. Mat. Palermo*, 8:57–156, 1894.

[51] S. Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin, 2008.

[52] S. Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin, 2008.

[53] S. Repin. Estimates of constants in boundary-mean trace inequalities and applications to error analysis. In *Numerical Mathematics and Advanced Applications - ENUMATH 2013*, volume 103 of *Lecture Notes in Computational Science and Engineering*, pages 215–223. Springer, Switzerland, 2015.

[54] S. Repin, S. Sauter, and A. Smolianski. A posteriori error estimation for the Dirichlet problem with account of the error in the approximation of boundary conditions. *Computing*, 70(3):205–233, 2003.

[55] S. Repin, S. Sauter, and A. Smolianski. A posteriori error estimation for the Poisson equation with mixed Dirichlet/Neumann boundary conditions. In *Proceedings of the 10th International Congress on Computational and Applied Mathematics (ICCAM-2002)*, volume 164/165, pages 601–612, 2004.

[56] S. Repin, S. Sauter, and A. Smolianski. Two-sided a posteriori error estimates for mixed formulations of elliptic problems. *SIAM J. Numer. Anal.*, 45(3):928–945, 2007.

[57] S. Repin and A. Smolianski. Functional-type a posteriori error estimates for mixed finite element methods. *Russian J. Numer. Anal. Math. Modelling*, 20(4):365—382, 2005.

[58] S. I. Repin. A posteriori error estimation for nonlinear variational problems by duality theory. *Zapiski Nauchnych Seminarov POMIs*, 243:201–214,, 1997.

[59] S. I. Repin. A posteriori error estimates for approximate solutions to variational problems with strongly convex functionals. *Journal of Mathematical Sciences*, 97:4311–4328, 1999.

[60] Sergey I. Repin and Satyendra K. Tomar. Guaranteed and robust error bounds for nonconforming approximations of elliptic problems. *IMA J. Numer. Anal.*, 31(2):597–615, 2011.

[61] D. Schillinger, L. Dedé, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and T.J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Methods Appl. Mech. Engrg.*, 249/252:116–150, 2012.

[62] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *Internat. J. Numer. Methods Engrg.*, 88(2):126–156, 2011.

[63] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Comput. Methods Appl. Mech. Engrg.*, 213/216:206–222, 2012.

[64] T. W. Sederberg, D. C. Cardon, G. T. Finnigan, N. N. North, J. Zheng, and T. Lyche. T-splines simplification and local refinement. *ACM Trans. Graphics*, 23(3):276–283, 2004.

[65] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. *ACM Trans. Graphics*, 22(3):477–484, 2003.

[66] H. Speleers and C. Manni. Effortless quasi-interpolation in hierarchical spaces. *Numer. Math.*, 132(1):155–184, 2016.

[67] S. K. Tomar and S. I. Repin. Efficient computable error bounds for discontinuous Galerkin approximations of elliptic problems. *J. Comput. Appl. Math.*, 226(2):358–369, 2009.

[68] K. G. van der Zee and C. V. Verhoosel. Isogeometric analysis-based goal-oriented error estimation for free-boundary problems. *Finite Elem. Anal. Des.*, 47(6):600–609, 2011.

[69] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.*, 200(49-52):3554–3567, 2011.

[70] P. Wang, J. Xu, J. Deng, and F. Chen. Adaptive isogeometric analysis using rational pht-splines. *Computer-Aided Design*, 43(11):1438–1448, 2011.

[71] X. Wei, Y. Zhang, T. J. R. Hughes, and M. A. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Comput. Methods Appl. Mech. Engrg.*, 291:1–20, 2015.

[72] J. Wloka. *Partial Differential Equations.* Cambridge University Press, 1987.

[73] M. Wu, J. Xu, R. Wang, and Z. Yang. Hierarchical bases of spline spaces with highest order smoothness over hierarchical T-subdivisions. *Comput. Aided Geom. Design*, 29(7):499–509, 2012.

[74] E. Zeidler. *Nonlinear functional analysis and its applications. II/A*. Springer-Verlag, New York, 1990.

[75] U. Zore and B. Jüttler. Adaptively refined multilevel spline spaces from generating systems. *Comput. Aided Geom. Design*, 31(7-8):545–566, 2014.

[76] U. Zore, B. Jüttler, and J. Kosinka. On the linear independence of truncated hierarchical generating systems. *J. Comput. Appl. Math.*, 306:200–216, 2016.