Paolo Di Stolfo*, Andreas Rademacher, and Andreas Schröder

Dual weighted residual error estimation for the finite cell method

https://doi.org/10.1515/jnma-2017-0103
Received August 18, 2017; revised September 26, 2018; accepted November 04, 2018

Abstract: The paper presents a goal-oriented error control based on the dual weighted residual method (DWR) for the finite cell method (FCM), which is characterized by an enclosing domain covering the domain of the problem. The error identity derived by the DWR method allows for a combined treatment of the discretization and quadrature error introduced by the FCM. We present an adaptive strategy with the aim to balance these two error contributions. Its performance is demonstrated for several two-dimensional examples.

Keywords: finite cell method, goal-oriented error estimation, dual-weighted residual method

Classification: 65N15, 65N85

1 Introduction

The finite cell method (FCM) is a well-established variant of the general fictitious domain approach [17, 18, 36] and was developed by Parvizian, Düster, and Rank [15, 27]. It has been applied to a vast number of both linear and nonlinear problems, including linear elasticity in 2D and 3D [15], shell problems [32], biomechanical problems [41, 42], wave propagation [21], elastoplasticity [1], and topology optimization in structural mechanics [28].

The basic idea of the FCM is to replace the possibly complicated domain of the problem by an enclosing domain of a geometrically simple shape, e.g., a (paraxial) quadrilateral in 2D or (paraxial) hexahedron in 3D. As the enclosing domain can be trivially subdivided into (paraxial) quadrilateral or hexahedral cells, mesh generation is simplified substantially. The finite element space is constructed on these cells, from which the name of the method is derived. To recover the geometry of the original problem, the integrals in the variational formulation of the problem are approximated by quadratures defined on the covering mesh of finite cells. To this end, an approximation of the original domain of sufficient quality has to be available, which is typically provided by a separate quadrature mesh. However, this approximation introduces a quadrature error which is assumed to be lower than the discretization error. A first mathematically rigorous investigation of the FCM for exact integration and certain boundary conditions as well as numerical experiments for inexact integration are provided in [12].

While it has become standard for modern finite-element techniques to include a posteriori error control and adaptivity, error estimators have neither been derived nor applied to the FCM to this day. In this work, we focus on the dual-weighted residual error (DWR) estimation method, which has become one of the most popular a posteriori techniques for standard finite elements in the last two decades. It is based on the preliminary work by Eriksson et al. [16] and was developed by Becker and Rannacher [5]. The DWR method allows for goal-oriented error estimation and, thus, supports more general, user-defined, possibly nonlinear expressions to be estimated, such as norms, point values, averages, or lift and drag coefficients, see [2] for an overview. The method relies on representing the error in terms of the solution of a dual problem, which is typical as duality arguments are the basis of many techniques in so-called goal-oriented error control [26, 29]. The DWR method has been applied to many practical problems including fluid mechanics, chemically reactive flows,

***Corresponding author: Paolo Di Stolfo,** Department of Mathematics, University of Salzburg, Hellbrunner Straße 34, 5020 Salzburg, Austria. Email: paolo.distolfo@sbg.ac.at
Andreas Rademacher, Fakultät für Mathematik (LS X), Technische Universität Dortmund, 44221 Dortmund, Germany
Andreas Schröder, Department of Mathematics, University of Salzburg, Hellbrunner Straße 34, 5020 Salzburg, Austria

and fluid–structure interaction (see, e.g., [3, 8, 19, 34, 39, 40]), as well as simplified Signorini and (frictional) contact problems (see [7, 31, 37]).

A posteriori error estimates are well-developed with respect to exact discrete solutions, i.e., solutions determined with no computational error incurred by, e.g., iterative methods or inexact integration. However, there are only a few publications dealing with a posteriori error estimates for inexact discrete solutions determined by an iterative process, such as the multigrid method [4] or Newton's method in the context of nonlinear problems [33]. A common idea of these approaches is to apply a stopping criterion that is based on balancing the discretization error with the iteration error.

In this article, we discuss the derivation and application of the DWR method in the FCM context in order to estimate both the discretization error and the quadrature error with respect to a goal functional, along with an adaptive strategy with the aim to balance these two contributions by either refining the finite-cell mesh or its associated quadrature mesh. The balancing is to be understood in the sense that, instead of striving for a very small quadrature error as it is usually the case for standard finite elements, the quadrature mesh is refined so that the quadrature error is only small enough to provide a useful discrete solution. We utilize the localization strategy for the DWR method by Braack and Ern [9] which does not require jumps over element facets and, thus, is well-suited for the FCM.

The outline of the paper is as follows. In Section 2, we give an overview of the FCM. Section 3 discusses the DWR method for goal-oriented a posteriori error estimation and provides the error identity containing terms representing the discretization and the quadrature error. An adaptive strategy realizing both discretization and quadrature mesh refinements is discussed in Section 4. In Section 5, numerical experiments for several 2D examples with different characteristics are presented. Finally, conclusions are drawn in Section 6.

2 Abstract framework for the finite cell method

In this section, we present a general nonlinear setting for the finite-cell method (FCM). For this purpose, let $\Omega \subset \mathbb{R}^d$ be a bounded domain and $\widehat{\Omega} \supseteq \Omega$ be a paraxial d-dimensional interval (i.e., a rectangle for $d = 2$ or a cuboid for $d = 3$), and let $\Gamma_D \subseteq \partial\Omega$ be the Dirichlet boundary part.

Given a Hilbert space V of functions defined on Ω with its dual space V^* and an operator $A : V \to V^*$, we aim to find a solution $u \in V$ such that

$$A(u)(\varphi) = 0 \quad \forall \varphi \in V \tag{2.1}$$

where we assume that (2.1) is uniquely solvable. Furthermore, we assume there exists a space $\widehat{V}$ of functions defined on $\widehat{\Omega}$ extending V, i.e., $\widehat{V}|_\Omega \supseteq V$. Also, we assume there exists an operator $\widehat{A} : \widehat{V} \to \widehat{V}^*$ such that

$$A(v)(\varphi) = \widehat{A}(v_0)(\varphi_0) \quad \forall v, \varphi \in V$$

where w_0 denotes the extension by zero onto $\widehat{\Omega}$ of a function w defined on Ω.

In the discrete setting, a triangulation $\mathcal{T}_h$ of $\widehat{\Omega}$ into intervals and a finite-element space $V_h \subseteq \widehat{V}$ on $\mathcal{T}_h$ can be constructed easily due to the simple form of $\widehat{\Omega}$. In the framework used in the following, we assume $V_h|_\Omega \subseteq V$, i.e., the space of restrictions is conforming. The discrete problem is to find a solution $u_h \in V_h$ such that

$$\widehat{A}(u_h)(\varphi_h) = 0 \quad \forall \varphi_h \in V_h. \tag{2.2}$$

It is assumed that the contributions $\left|\widehat{A}(u_h)(\varphi_h) - A(u_h|_\Omega)(\varphi_h|_\Omega)\right|$ are sufficiently small so that the model error can be neglected. To illustrate the spaces and operators, we consider a 2D example based on the Poisson model problem

$$A(u)(\varphi) := \int_\Omega \nabla u \cdot \nabla \varphi - \int_\Omega f\varphi = 0 \quad \forall \varphi \in V$$

where $\Omega := (0,1)^2 \cap B_1(0)$ is the quarter disk, $f \in L_2(\Omega)$, the space $V := H^1_{\Gamma_D}(\Omega) = \{v \in H^1(\Omega);\ v|_{\Gamma_D} = 0\}$, and $V^* = H^1_{\Gamma_D}(\Omega)^*$ for the Dirichlet part $\Gamma_D := [0,1] \times \{0\} \subseteq \partial\Omega$. As the operator $\widehat{A}$, we may choose

$$\widehat{A}(v)(\varphi) := \int_\Omega \nabla v \cdot \nabla \varphi + \varepsilon \int_{\widehat{\Omega}\setminus\Omega} \nabla v \cdot \nabla \varphi - \int_\Omega f\varphi \quad \forall v, \varphi \in \widehat{V}$$

where $\widehat{\Omega} := (0,1)^2$, $\varepsilon \approx 0$ (e.g., $\varepsilon = 10^{-12}$) is a positive parameter large enough to secure coercivity, and

$$\widehat{V} := \left\{v:\ v \in L_2(\widehat{\Omega}),\ v|_\Omega \in V,\ v|_{\widehat{\Omega}\setminus\Omega} \in H^1(\widehat{\Omega}\setminus\Omega)\right\}.$$

For the discrete setting, we introduce a triangulation $\mathcal{T}_h$ of $\widehat{\Omega}$ consisting of four square elements and define V_h to be the $H^1(\widehat{\Omega})$-conforming finite-element space of degree 1 on $\mathcal{T}_h$ respecting the Dirichlet boundary condition on Γ_D, implying $V_h|_\Omega \subseteq V$ and $V_h \subseteq \widehat{V}$. Since, here, the boundary of Ω matches a union of facets, Dirichlet boundary conditions can be applied in a strong manner. However, in general, the Dirichlet boundary is non-matching with $\widehat{\Omega}$, i.e., it does not equal the union of some facets in $\mathcal{T}_h$. In this case, Dirichlet boundary conditions may be applied weakly by, e.g., Nitsche's method [25, 43]. The contributions

$$\left|\widehat{A}(v_h)(\varphi_h) - A(v_h|_\Omega)(\varphi_h|_\Omega)\right| = \varepsilon \left|\int_{\widehat{\Omega}\setminus\Omega} \nabla v_h \cdot \nabla \varphi_h\right| \tag{2.3}$$

result in a model error of $O(\sqrt{\varepsilon})$ in the energy norm (see [12, p. 1047]).

While the operator $\widehat{A}$ is defined on $\widehat{\Omega}$, the domains of the involved integrals may depend on Ω. Therefore, for the computation of $\widehat{A}$ in the discrete setting, numerical integration has to be performed. In the context of the FCM, this usually involves an approximation of Ω by geometrically simple objects. These approximations result in approximate operators $\widehat{A}^{(\sim)}$ and perturbed discrete problems

$$\widehat{A}^{(\sim)}\left(u_h^{(\sim)}\right)(\varphi_h) = 0 \quad \forall \varphi_h \in V_h \tag{2.4}$$

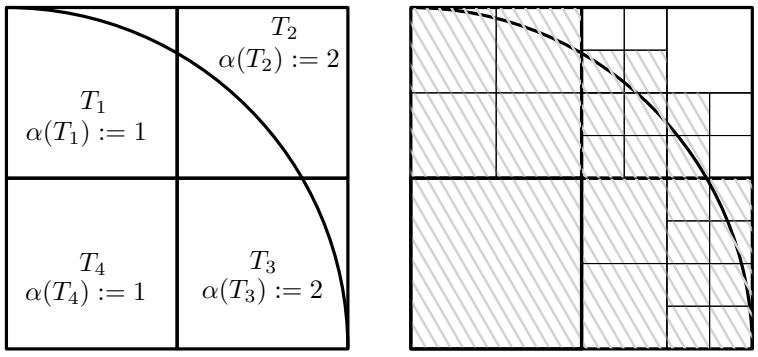
yielding perturbed discrete solutions $u_h^{(\sim)} \in V_h$. A geometrically simple replacement for Ω used in practice is the spacetree with its specializations to two and three dimensions commonly referred to as quadtree and octree, respectively [13]. Here, to each element $T \in \mathcal{T}_h$, a set of intervals Q_T is assigned via a number $\alpha(T) \in \mathbb{N}_0$ indicating the number of recursive refinements of T towards the boundary $\partial\Omega$. The set Q_T is generated by the following recursive procedure:

1. Set $i := 0$, $Q_T^{(i)} := \{T\}$.
2. If $i = \alpha(T)$, then $Q_T := Q_T^{(i)}$, exit. Otherwise, replace each interval in $Q_T^{(i)}$ that is intersected nontrivially by $\partial\Omega$ by 2^d sub-intervals, yielding $Q_T^{(i+1)}$. Increase i by 1 and go to step 2.

Finally, as an approximation $\Omega^{(\sim)}$ of the domain of integration Ω, one may use the union of all intervals in any Q_T having non-trivial intersection with Ω. The union of the remaining intervals is then an approximation to $\widehat{\Omega}\setminus\Omega$. Similarly, an approximation to $\widehat{\Omega}$ can be obtained. For the approximation of the integrals involved in the operators $\widehat{A}^{(\sim)}$, the usual quadrature rules used in the finite-element context are applied on each interval.

The result of the procedure is visualized in Fig. 1 for a finite-cell mesh for the quarter disk, where the unit square is subdivided into four equally sized elements, along with the assigned number of recursive refinements.

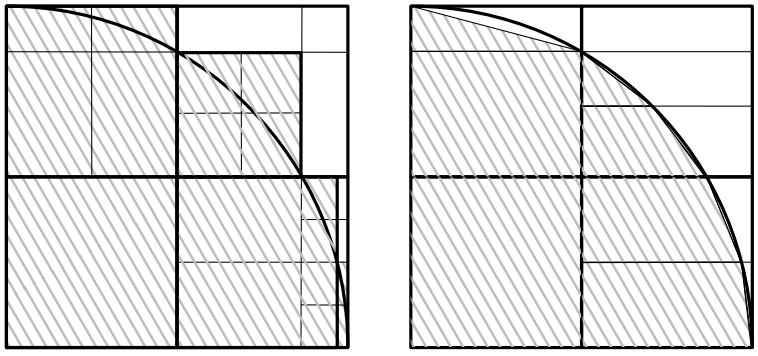
The presented procedure for establishing the space-tree only makes use of a point-in-domain test, which is typically applied to sample points of each interval (e.g., the four vertices of a rectangle). Thus, the condition whether an interval intersects $\partial\Omega$ nontrivially is checked only approximately. Due to its simplicity, the space-tree can be easily applied to complicated geometries, e.g., generated in the context of constructive solid geometry. An obvious disadvantage is the fact that it offers only a piecewise constant approximation to $\partial\Omega$. Therefore, a high number of recursive refinements may be required to approximate the domain sufficiently well. For domains with smooth boundaries, higher-order approximations of the boundary may be used. In the context of the FCM, several improvements over the space-tree have been developed (see, e.g., [11, 20, 22–24]). However, implementing these improvements is usually rather involved. Whenever the boundary is of a particular simple shape (such as a circle), one may apply the following improvements: Firstly, the quadtree



(a) A finite-cell discretization of the unit square $\widehat{\Omega}$ covering the quarter disk Ω.

(b) The resulting quadtree. A possible approximation to Ω is shaded.

Fig. 1: Visualization of the quadtree



(a) Improvement of the quadtree

(b) Piecewise linear approximation

Fig. 2: Visualization of the described improvements applicable in the case of the quarter disk

refinement on a given cell can be performed so that the intersection points of the boundary with the cell coincide with vertices of the quadtree, whenever possible (see Fig. 2a). Secondly, one may derive a piecewise linear approximation of the boundary by subdividing the elements by horizontal or vertical line segments, computing intersection points of the boundary with the segments, and replacing the boundary by line segments, so that a subdivision into triangles and quadrilaterals is produced (see Fig. 2b).

In the following, we do not confine our treatment to the space-tree, but permit a general quadrature scheme which generates an approximation $\Omega^{(\sim)}$ that allows for refinements, i. e., given an initial approximation $\Omega^{(0)} := \Omega^{(\sim)}$ for Ω, a sequence of domains $\Omega^{(n)}$, $n \geqslant 1$, may be generated fulfilling $\Omega^{(n+1)} \setminus \Omega \subset \Omega^{(n)} \setminus \Omega$, so that each approximation is an improvement of the previous one. The operators with integrals defined on these approximate domains are indicated with superscript (n), such as $A^{(n)}$. Also, we make the natural assumption that the refinement of the quadrature occurs on the level of the finite cells, i.e., there is a function $\alpha_n : \mathcal{T}_h \to \mathbb{N}_0$ indicating the number of refinements of the quadrature scheme used to approximate $T \cap \Omega$ in $\Omega^{(n)}$. An improvement $\Omega^{(n+1)}$ can be generated by assigning $\alpha_{n+1}(T) := \alpha_n(T) + 1$ for each $T \in \mathcal{T}_h$. Alternatively, one may only improve a subset $\mathcal{S} \subset \mathcal{T}_h$ by

$$\alpha_{n+1}(T) := \alpha_n(T) + 1,\ T \in \mathcal{S}, \quad \alpha_{n+1}|_{\mathcal{T}_h \setminus \mathcal{S}} := \alpha_n|_{\mathcal{T}_h \setminus \mathcal{S}}.$$

In particular, note that the theoretical considerations presented in Sect. 3 are valid for any domain approximation scheme that generates approximate operators, each of which is an improvement of the previous one.

3 The dual weighted residual method

In this section, we tailor the dual weighted residual (DWR) method for a (possibly nonlinear) problem and a (possibly nonlinear) goal functional to the FCM setting. To this end, we consider the spaces and operators introduced in Section 2 and, in addition, assume A to be three times Gateaux-differentiable. Moreover, let $J : V \to \mathbb{R}$ be a three times Gateaux-differentiable goal functional. For the kth order Gateaux derivative of a function $g : X \to Y$ for Banach spaces X, Y in a point $x \in X$, we adopt the usual identification $g^{(k)}(x)(\psi_1, \dots, \psi_k) = g^{(k)}(x)(\psi_1) \cdots (\psi_k)$, indicating that $g^{(k)}(x)$ is k-linear.

For the unique solution $u \in V$ of the problem in (2.1), we may formulate the following trivial optimization problem which connects the problem with the goal functional:

$$u = \arg\min_{\varphi \in V} J(\varphi) \quad \text{subject to} \quad A(u)(\varphi) = 0 \ \ \forall \varphi \in V.$$

Introducing the Lagrangian $\mathcal{L} : V \times V \to \mathbb{R}$ with $\mathcal{L}(v, w) := J(v) - A(v)(w)$, we seek a Lagrangian multiplier $z \in V$ such that (u, z) is a stationary point of $\mathcal{L}$, yielding

$$\mathcal{L}'(u, z)(\varphi, \psi) = \big(J'(u)(\varphi) - A'(u)(\varphi, z), -A(u)(\psi)\big) = 0 \quad \forall \varphi, \psi \in V.$$

Thus, in addition to seeking a solution u in (2.1), we seek a function z that solves the dual problem

$$A'(u)(\varphi, z) = J'(u)(\varphi) \quad \forall \varphi \in V. \tag{3.1}$$

In the FCM setting, the nonconformity $V_h \not\subseteq V$ and the approximation of operators and functionals have to be taken into account in the computation of the discrete solutions. Let $\Omega^{(n)}$ be an approximation for Ω that allows for refinement. Similar to the case of A and $\widehat{A}^{(n)}$ described in Section 2, we assume that approximations $J^{(n)}$, $J'^{(n)}$ for J, J' exist, where all integrals on Ω occurring in the definition are replaced by a quadrature rule on $\Omega^{(n)}$. Instead of the discrete dual problem, its perturbation

$$\widehat{A}'^{(n)}\Big(u_h^{(n)}\Big)\Big(\varphi_h, z_h^{(n)}\Big) = J'^{(n)}\Big(u_h^{(n)}|_\Omega\Big)(\varphi_h|_\Omega) \quad \forall \varphi_h \in V_h \tag{3.2}$$

is solved. We are interested in a representation of the exact error

$$\mathrm{err}_{\mathrm{ex}} := J(u) - J\Big(u_h^{(n)}|_\Omega\Big). \tag{3.3}$$

In the next section, we derive an error representation for $\mathrm{err}_{\mathrm{ex}}$. It includes non-computable terms, which we assume to be negligibly small, and higher-order approximations to continuous solutions. The question of how to compute these approximations is addressed in Section 3.2. Also, localization techniques for the discretization and the quadrature error are discussed in Section 3.3.

3.1 Error representation

The error $\mathrm{err}_{\mathrm{ex}}$ from equation (3.3) ought to be computable except for minor perturbations: We assume that there exists a sufficiently precise approximation $J^{(n+k)}$ of J for a fixed $k \in \mathbb{N}$, such that the resulting perturbation error is negligibly small. Moreover, the representation should allow for a separation of two error sources, i.e., the discretization and the quadrature. We derive such a representation by adapting the proof stated for the standard FEM case in [33, Prop. 3.1], where a representation of the error with respect to any perturbation $v_h \in V_h$ of the discrete solution is provided. In the FCM setting, we show that the error is composed of the sum of a discretization-related error term, a quadrature-related error term, and some terms which are assumed to be negligibly small (e.g., of higher order). The representation requires computable replacements u^+, z^+ for the unknown solutions u, z, as well as improvements $G^{(n+k)}$ of the approximations $G^{(n)}$ for each $G \in \{A, A', \widehat{A}, \widehat{A}', J, J'\}$ with the property that $\big|G^{(n+k)}(\cdot) - G(\cdot)\big|$ is negligibly small. Recall that we assume that a quadrature scheme allowing for these improvements is available. For instance, when the space-tree is

used, the approximations with index $n+k$ may be obtained from approximations with index n by refining the space-tree k times.

Before stating the representation, we briefly explain the occurring terms. The residuals ρ, ρ^* are defined as

$$\begin{aligned}\rho(\cdot) &:= -\widehat{A}^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)(\cdot)\\ \rho^*(\cdot) &:= J'^{(n+k)}\Big(u_h^{(n)}|_\Omega\Big)(\cdot|_\Omega) - \widehat{A}'^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\Big(\cdot, \Big(z_h^{(n)}\Big)_0\Big).\end{aligned}$$

Here, we abbreviate by v_0 the extension of $v|_\Omega$ by zero onto $\widehat{\Omega}$. The residuals ρ, ρ^* occur in the definitions of the terms $e_D^{(n+k)}(u^+|_\Omega, z^+|_\Omega)$, $e_{NS,D}(u^+|_\Omega, z^+|_\Omega)$ related to the discretization error. These terms are defined for $v, w \in V$ by

$$\begin{aligned}e_D^{(n+k)}(v,w) &:= \frac{1}{2}\Big(\rho\Big(w_0 - \Big(z_h^{(n)}\Big)_0\Big) + \rho^*\Big(v_0 - \Big(u_h^{(n)}\Big)_0\Big)\Big)\\ e_{NS,D}(v,w) &:= \frac{1}{2}(\rho(z_0 - w_0) + \rho^*(u_0 - v_0)).\end{aligned} \tag{3.4}$$

We assume the approximations u^+ and z^+ to be such that the residuals in the errors $u_0 - u_0^+$ and $z_0 - z_0^+$ are of higher order compared to those in $u_0 - (u_h^{(n)})_0$ and $z_0 - (z_h^{(n)})_0$.

Abbreviating the errors $e := u - u_h^{(n)}|_\Omega$ and $e^* := z - z_h^{(n)}|_\Omega$, we obtain the quadrature-related error term $e_Q^{(n+k)}$ and additional contributions $e_{NS,Q}$, $e_{NS,J}$, and $e_{NS,\mathcal{L}}$:

$$\begin{aligned}e_Q^{(n+k)} &:= -\widehat{A}^{(n+k)}\Big(u_h^{(n)}\Big)\Big(z_h^{(n)}\Big)\\ e_{NS,Q} &:= \widehat{A}^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\Big(\Big(z_h^{(n)}\Big)_0\Big) - A\Big(u_h^{(n)}|_\Omega\Big)\Big(z_h^{(n)}|_\Omega\Big)\\ e_{NS,J} &:= \Big(J - J^{(n+k)}\Big)\Big(u_h^{(n)}|_\Omega\Big)\\ e_{NS,\mathcal{L}} &:= \frac{1}{2}\Big(\Big(J' - J'^{(n+k)}\Big)\Big(u_h^{(n)}|_\Omega\Big)(e) - \Big(A'\Big(u_h^{(n)}|_\Omega\Big)\Big(e, z_h^{(n)}|_\Omega\Big) - \widehat{A}'^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\Big(e_0, \Big(z_h^{(n)}\Big)_0\Big)\Big)\\ &\qquad - \Big(A\Big(u_h^{(n)}|_\Omega\Big)(e^*) - \widehat{A}^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)((e^*)_0)\Big)\Big).\end{aligned} \tag{3.5}$$

These additional terms are assumed to be negligibly small for k sufficiently large due to the fact that they consist of evaluations of differences $G - G^{(n+k)}$. To see that $e_Q^{(n+k)}$ can be regarded as a quadrature error term, assume that exact integration is available, i.e., $\widehat{A}^{(n+k)} = \widehat{A}$. Then, we may abbreviate $u_h := u_h^{(n)}$, $z_h := z_h^{(n)}$, and

$$e_Q := e_Q^{(n+k)} = -\widehat{A}(u_h)(z_h) = 0.$$

Therefore, the term $e_Q^{(n+k)}$ vanishes if all operators are exact. If only approximations to the operators are available or a perturbed solution is inserted, the term $e_Q^{(n+k)}$ will be nonzero in general and, thus, may be regarded as a perturbation error. This error may be caused by, e.g., numerical quadrature or by an iterative method as in [33].

Finally, the term e_ε describes the error incurred by the FCM approximation in $\widehat{\Omega} \setminus \Omega$, which is assumed to be negligibly small for ε sufficiently small. It is defined as

$$e_\varepsilon := \widehat{A}^{(n+k)}\Big(u_h^{(n)}\Big)\Big(z_h^{(n)}\Big) - \widehat{A}^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\Big(\Big(z_h^{(n)}\Big)_0\Big). \tag{3.6}$$

Typically, the value of e_ε is $\mathcal{O}(\varepsilon)$, see also (2.3).

Proposition 3.1. Let u resp. z be the solution of the primal resp. dual problem in (2.1) resp. (3.1) with approximations $u^+, z^+ \in V^+$, where $\widehat{V} \supseteq V^+ \supseteq V_h$. Then, for the perturbed discrete solutions $u_h^{(n)}$ resp. $z_h^{(n)}$ of (2.4) resp. (3.2), it holds that

$$\begin{aligned}J(u) - J^{(n+k)}\Big(u_h^{(n)}|_\Omega\Big) = e_D^{(n+k)}(u^+|_\Omega, z^+|_\Omega) + e_Q^{(n+k)} + e_{NS,Q} + e_{NS,J} + e_{NS,\mathcal{L}}\\ + e_{NS,D}(u^+|_\Omega, z^+|_\Omega) + e_\varepsilon + \mathcal{R}_h^{(3)}\end{aligned} \tag{3.7}$$

where the higher-order remainder $\mathcal{R}_h^{(3)}$ is given by

$$\mathcal{R}_h^{(3)} := \int_0^1 \Big(J'''\Big(u_h^{(n)}|_\Omega + te\Big)(e,e,e) - A'''(u_h|_\Omega + te)\Big(e,e,e,z_h^{(n)}|_\Omega + te^*\Big) \\ -3A''\Big(u_h^{(n)}|_\Omega + te\Big)(e,e,e^*)\Big)\, t(t-1)\,\mathrm{d}t.$$

Proof. Let $\ell : \mathbb{R} \to \mathbb{R}$ be defined as

$$\ell(t) := \mathcal{L}(\gamma_u(t), \gamma_z(t)) = \mathcal{L}\Big(\Big(u_h^{(n)}|_\Omega, z_h^{(n)}|_\Omega\Big) + t(e,e^*)\Big)$$

where $\gamma_u(t) := u_h^{(n)}|_\Omega + te$, $\gamma_z(t) := z_h^{(n)}|_\Omega + te^*$. Note that $\gamma_u'(t) = e$, $\gamma_z'(t) = e^*$. The derivative of ℓ is

$$\ell'(t) = \mathcal{L}'(\gamma_u(t), \gamma_z(t))(e,e^*)\cdot(1,1)^T.$$

Applying the definition of $\mathcal{L}$, we get

$$\ell'(t) = J'(\gamma_u(t))(e) - A'(\gamma_u(t))(e,\gamma_z(t)) - A(\gamma_u(t))(e^*).$$

Applying differentiation twice more yields

$$\ell'''(t) = J'''(\gamma_u(t))(e,e,e) - A'''(\gamma_u(t))(e,e,e,\gamma_z(t)) - 3A''(\gamma_u(t))(e,e,e^*). \tag{3.8}$$

The error using the exact functional J can be written in the following form:

$$\begin{aligned} J(u) - J\Big(u_h^{(n)}|_\Omega\Big) &= \mathcal{L}(u,z) + A(u)(z) - \mathcal{L}\Big(u_h^{(n)}|_\Omega, z_h^{(n)}|_\Omega\Big) - A\Big(u_h^{(n)}|_\Omega\Big)\Big(z_h^{(n)}|_\Omega\Big) \\ &= \mathcal{L}(u,z) - \mathcal{L}\Big(u_h^{(n)}|_\Omega, z_h^{(n)}|_\Omega\Big) - A\Big(u_h^{(n)}|_\Omega\Big)\Big(z_h^{(n)}|_\Omega\Big) \\ &= \ell(1) - \ell(0) - A\Big(u_h^{(n)}|_\Omega\Big)\Big(z_h^{(n)}|_\Omega\Big) \\ &= \int_0^1 \ell'(t)\,\mathrm{d}t - A\Big(u_h^{(n)}|_\Omega\Big)\Big(z_h^{(n)}|_\Omega\Big). \end{aligned}$$

It follows that

$$J(u) - J^{(n+k)}\Big(u_h^{(n)}|_\Omega\Big) = \int_0^1 \ell'(t)\,\mathrm{d}t + e_Q^{(n+k)} + e_\varepsilon + e_{NS,Q} + e_{NS,J}.$$

We use the error representation of the trapezoidal rule to obtain

$$\int_0^1 \ell'(t)\,\mathrm{d}t = \frac{1}{2}\big(\ell'(0) + \ell'(1)\big) + \frac{1}{2}\int_0^1 \ell'''(t)t(t-1)\,\mathrm{d}t.$$

Since $\ell'''(t)$ has been determined in (3.8), it remains to inspect the terms $\ell'(0)$ and $\ell'(1)$. We use that $\gamma_u(0) = u_h^{(n)}|_\Omega$, $\gamma_z(0) = z_h^{(n)}|_\Omega$, and $\gamma_u(1) = u$, $\gamma_z(1) = z$, so that

$$\ell'(1) = J'(u)(e) - A'(u)(e,z) - A(u)(e^*) = 0$$

since (u,z) is a stationary point of $\mathcal{L}$. We see that

$$\begin{aligned} \ell'(0) &= J'\Big(u_h^{(n)}|_\Omega\Big)(e) - A'\Big(u_h^{(n)}|_\Omega\Big)\Big(e, z_h^{(n)}|_\Omega\Big) - A\Big(u_h^{(n)}|_\Omega\Big)(e^*) \\ &= \rho^*(e_0) + \rho((e^*)_0) + 2e_{NS,\mathcal{L}}. \end{aligned}$$

Thus, $\ell'(0) = 2e_D^{(n+k)}(u,z) + 2e_{NS,\mathcal{L}}$. The definition $\mathcal{R}_h^{(3)} = \frac{1}{2}\int_0^1 \ell'''(t)t(t-1)\,\mathrm{d}t$ and the calculation

$$\begin{aligned} e_D^{(n+k)}(u,z) &= \frac{1}{2}\Big(\rho\Big(z_0 - \Big(z_h^{(n)}\Big)_0\Big) + \rho^*\Big(u_0 - \Big(u_h^{(n)}\Big)_0\Big)\Big) \\ &= \frac{1}{2}(\rho(z_0 - (z^+)_0) + \rho^*(u_0 - (u^+)_0)) + \frac{1}{2}\Big(\rho\Big((z^+)_0 - \Big(z_h^{(n)}\Big)_0\Big) + \rho^*\Big((u^+)_0 - \Big(u_h^{(n)}\Big)_0\Big)\Big) \\ &= e_{NS,D}(u^+|_\Omega, z^+|_\Omega) + e_D^{(n+k)}(u^+|_\Omega, z^+|_\Omega) \end{aligned}$$

imply the proposed error identity. □

We assume that several error terms in (3.4), (3.5), and (3.6) are negligibly small. In Section 5, we will see that the adaptive algorithm presented in the next section can be configured in such a way that these terms are indeed negligibly small, which justifies our assumption. Ignoring these terms, we arrive at the following approximate error representation.

Corollary 3.1. Omitting all error terms which can be ensured to be negligibly small, we have the approximate error representation

$$\mathrm{err}_{\mathrm{ex}} \approx \mathrm{comp}^{(n+k)} := J(u) - J^{(n+k)}\left(u_h^{(n)}|_\Omega\right) \approx e_D^{(n+k)}(u^+|_\Omega, z^+|_\Omega) + e_Q^{(n+k)} =: \eta^{(n+k)}. \tag{3.9}$$

The effectivity index (or overestimation index) is defined as

$$\mathrm{eff}^{(n+k)} := \frac{\eta^{(n+k)}}{\mathrm{comp}^{(n+k)}}. \tag{3.10}$$

3.2 Approximation of the solutions of the continuous problems

The unknown quantities u and z are approximated by computable functions u^+ and z^+. To this end, several methods have been proposed in the literature. The first method computes approximations by solving the discrete problems in a finite-element space of higher polynomial degree, e.g., by doubling each local polynomial degree [6]. However, this is too expensive except for simple test problems. Usually, patched meshes are employed, i.e., for an element of $\mathcal{T}_h$ with its parent element P in the mesh refinement history, all 2^d children of P are elements of $\mathcal{T}_h$. This implies that whenever an element is refined, all its siblings are refined as well. In this case, the patches of the mesh can be joined to form a finite-element space $V_{2h,2p}$ of double mesh width and double polynomial degree. The computation of u^+, z^+ is then approximately as expensive as the computation of $u_h^{(n)}$, $z_h^{(n)}$.

Another method requiring patched meshes uses local higher-order interpolation to compute more accurate approximations by, again, viewing each patch as a single element of a coarser mesh and doubling the polynomial degree [2]. This eliminates the need of computing additional discrete solutions. Under certain regularity assumptions, it can be shown that the error incurred is of higher order (see [2, Section 5.2]). However, one has to take care that the resulting functions are elements of V by ensuring continuity of the interpolation on the boundary of the patches, which is difficult when hanging nodes are present [30].

3.3 Localization

To perform the finite-cell mesh adaptation, the discretization error $e_D^{(n+k)}$ has to be localized to nonnegative elementwise contributions $\eta_{D,T}^{(n+k)}$. To this end, several methods are available, for all of which good effectivity indices have been demonstrated for many practical problems (see, e.g., the references in the introduction). A first method applies elementwise partial integration leading to an inner residual and a boundary residual involving integrals over the boundary of each element. An obvious disadvantage of this method is the possibly costly computation of strong residuals and jump terms. Also, the strong adjoint residual formulation may not even be available [35]. In the context of the FCM, another disadvantage consists in the necessity of determining intersections between the boundary of the domain and the element boundaries, as these intersections are not required for the application of the FCM. Also, these intersections have to be determined with great precision in order not to introduce additional errors. A second method, which uses the variational formulation directly, has been proposed by Richter and Wick [35]. Here, a partition of unity for the nodes of the finite-element mesh is inserted into the error representation.

A third method known as the algebraic filtering approach has been described by Braack and Ern [9] which is also based on the variational formulation. The method relies on patched meshes and the associated canonical finite-element spaces $V_{2h,p}$ and $V_{2h,2p}$ formed by treating each patch as an element of degree p and $2p$, respectively. In order to reconstruct higher-order solutions, interpolation and filtering operators are defined,

which we briefly describe in the case of meshes without hanging nodes similar to [35]: As the spaces $V_{h,p}$ and $V_{2h,2p}$ have the same numbers of unknowns in the same Lagrange points, one can define an interpolation operator $i^* : V_{h,p} \to V_{2h,2p}$ by assigning to $v_h \in V_{h,p}$ the element $i^* v_h$ of $V_{2h,2p}$ uniquely determined by the values of v_h in those Lagrange points. The filtering operator is defined by $\pi_{2h} := \mathrm{id} - i_{2h}$ for a finite-element interpolation operator $i_{2h} : V \to V_{2h,p}$. The name of the method stems from the observation that $\pi_{2h} v_h$ is a strictly local algebraic process acting on the coefficient vector $\mathbf{v} \in \mathbb{R}^N$ of $v_h \in V_{h,p}$, since $\pi_{2h} v_h = v_h - i_{2h} v_h = \sum_{j=1}^N \mathbf{v}_j (\varphi_{h,j} - i_{2h}\varphi_{h,j}) =: \sum_j (\pi_{2h}\mathbf{v})_j \varphi_{h,j}$ for the basis $(\varphi_{h,j})_{j=1}^N$ of $V_{h,p}$ (see [35, eq. (40)]). Finally, the localization of $e_D^{(n+k)}$ to nodal contributions is as follows:

$$\begin{aligned} e_D^{(n+k)} \approx \sum_{j=1}^N \frac{1}{2} \Big(&- \widehat{A}^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\big((i^* - \mathrm{id})(\varphi_{h,j})_0 (\pi_{2h}\mathbf{z})_j\big) + J'^{(n+k)}\Big(u_h^{(n)}|_\Omega\Big)\big(\big((i^* - \mathrm{id})(\varphi_{h,j})_0 (\pi_{2h}\mathbf{u})_j\big)|_\Omega\big) \\ &- \widehat{A}'^{(n+k)}\Big(\Big(u_h^{(n)}\Big)_0\Big)\big((i^* - \mathrm{id})(\varphi_{h,j})_0 (\pi_{2h}\mathbf{u})_j, \Big(z_h^{(n)}\Big)_0\big)\Big) =: \sum_{j=1}^N \tilde{\eta}_{D,j} \end{aligned} \tag{3.11}$$

The node-wise error contributions $\eta_{D,j}^{(n+k)} := |\tilde{\eta}_{D,j}^{(n+k)}|$ are then used for the marking step in the adaptive procedure, e.g., by refining all elements touching node j or by explicitly assigning an elementwise indicator $\eta_{D,T}^{(n+k)}$ based on the node-wise contributions and performing the usual elementwise refinement.

To measure the quality of the localization, we define the indicator index

$$\mathrm{ind}_D^{(n+k)} := \frac{\sum_{T \in \mathcal{T}_h} \eta_{D,T}^{(n+k)}}{|e_D^{(n+k)}|} \tag{3.12}$$

similar to [35, eq. (26)] which takes into account the overestimation of the discretization error caused by taking the absolute value of possibly negative local values $\tilde{\eta}_{D,T}^{(n+k)}$.

The quadrature error estimator value $e_Q^{(n+k)}$ is used to check if the quadrature mesh resolves the computational domain Ω sufficiently well in order to obtain a meaningful discrete solution. For its localization, we split it canonically into element-wise contributions

$$e_Q^{(n+k)} = \sum_{T \in \mathcal{T}_h} \eta_{Q,T}^{(n+k)}$$

and utilize the same localization techniques as for the discretization error.

4 Refinement strategy

As identity (3.7) allows for a separation of the error into a term representing the discretization error $e_D^{(n+k)}$ and a term representing the quadrature error $e_Q^{(n+k)}$, we may perform finite-cell and quadrature mesh adaptation using these terms.

In the case of finite elements with exact quadrature, each step in the Solve–Estimate–Mark–Refine (SEMR) loop for adaptivity is well-examined at least for linear problems [10, 14]. However, for the finite-cell method, there are no strategies available for choosing the accuracy of the quadrature mesh. In practice, when e.g. the space-tree is used, a fixed number of recursive refinements throughout the computation is chosen. However, the chosen depth might be too high and a coarser integration mesh might be sufficient.

A possible heuristic strategy for a quadrature scheme allowing for refinements is the following: Considering the function $\alpha : \mathcal{T}_h \to \mathbb{N}_0$ from Section 2 assigning to each finite cell the number of refinements in the quadrature scheme, we set $\alpha \equiv d$ initially for some small $d \in \mathbb{N}_0$. This produces an initial quadrature mesh that is still coarse. During each iteration of the SEMR loop, it is checked whether the overall precision of the quadrature is sufficient for the current finite-cell computation. If this is not the case, then α is increased by 1 on elements with high quadrature error contribution and the computation is repeated. The involved check whether the overall precision of the quadrature is sufficient aims at balancing the two error contributions, the

discretization and the quadrature error, in particular, the quadrature mesh is refined if the quadrature error exceeds a certain multiple of the discretization error. Recall that, in the standard FEM setting, convergence of the discretization error to zero as $h \to 0$ is typically shown under the assumption that the quadrature error incurred by the standard quadrature rules is of the same order of magnitude as the discretization error. However, in the FCM setting, more complicated quadrature rules allowing for adaptive refinement (such as the space-tree) have to be utilized, for which theoretical justifications are still lacking. Numerical results suggest that the estimated quadrature error has to be below the discretization error in order to obtain meaningful discrete solutions and convergence. Therefore, we cannot expect an optimal balancing in the usual sense that the error contributions are approximately equal: The contributions are balanced such that the quadrature error is only as small as necessary to deliver reliable discrete solutions, compared to the usual approaches of over-integration up to a very high accuracy or even machine precision. Thus, we expect the ratio of quadrature error to discretization error not to be close to 1, but to be moderately small (e.g., 0.01).

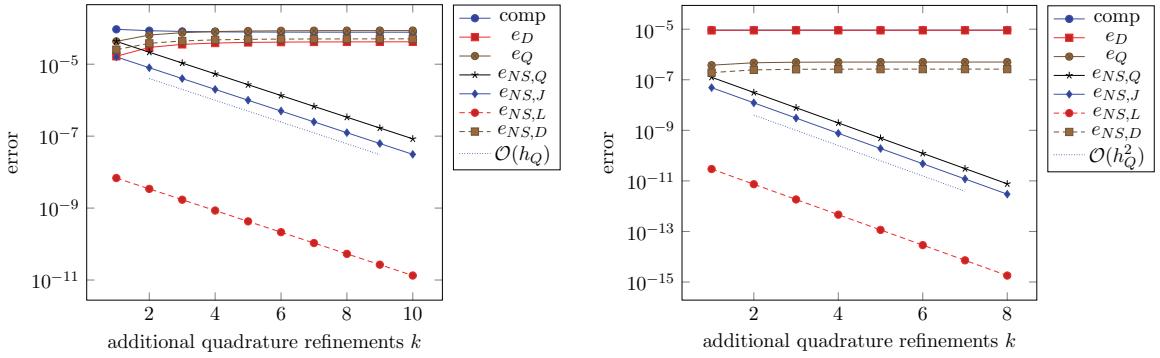
An issue that requires attention is the fact that, whenever an element T of the finite-cell mesh is refined and $\alpha(T) = 0$, the quadrature mesh is refined in T as well, so that the quadrature error might decrease when only a decrease in the discretization error is intended. Hence, if $\alpha(T) > 0$, it is reasonable to set $\alpha(T') := \alpha(T) - 1$ for any child T' of T in order not to introduce an additional improvement of the quadrature mesh where only an improvement in the finite-cell mesh is indicated. Furthermore, it is useful to introduce a lower bound $l \in \mathbb{N}_0$ on α, e.g., $\alpha \geqslant l := 0$, when derefinements are not supported.

The adaptive strategy is summarized in the following steps. We emphasize that the error terms from (3.9) as well as the finite-cell mesh and the number of quadrature mesh refinements α now depend on the iteration index i of the SEMR loop. Also, to indicate the dependence of the approximate terms on α when the space-tree is used, we replace the generic number n indicating a sequence of approximations to the exact operators and functionals by the number of quadrature mesh refinements per element, given by the function α. Hence, we write $\alpha + k$ instead of $n + k$ indicating that the quadrature mesh defined by α is refined k times globally.

1. Set $i := 0$. Initialize the finite-cell mesh $\mathcal{T}_i$. Choose an initial depth $d \in \mathbb{N}_0$ and set the number of refinements of the quadrature scheme $\alpha_i(T) := d$ for each $T \in \mathcal{T}_i$. Set l to be the minimum possible depth. Choose $0 < \rho \leqslant 1$, e. g., $\rho := 0.01$. Choose a stopping criterion, e.g., stop if the maximum number of degrees of freedom is reached or if a prescribed error tolerance is met.
2. Construct the quadrature mesh for $\mathcal{T}_i$ associated to α_i.
3. Solve: Compute solutions $u_{h,i}^{(\alpha_i)}, z_{h,i}^{(\alpha_i)}$ of the perturbed discrete problems from (2.4), (3.2). Compute approximations u_i^+, z_i^+ to u, z.
4. Estimate: Choose $k \in \mathbb{N}$ and construct a quadrature mesh on $\mathcal{T}_i$ associated to $\alpha_i + k$ to compute estimators $e_{D,i}^{(\alpha_i+k)}, e_{Q,i}^{(\alpha_i+k)}$ and indicators $\eta_{D,T,i}^{(\alpha_i+k)}, \eta_{Q,T,i}^{(\alpha_i+k)}$ for each $T \in \mathcal{T}_i$. If the stopping criterion is fulfilled, stop. If $|e_{Q,i}^{(\alpha_i+k)}| \geqslant \rho\, |e_{D,i}^{(\alpha_i+k)}|$, localize the quadrature error contributions, mark the elements T with highest error contributions using a marking strategy (see step 5), set $\alpha_i(T) := \alpha_i(T) + 1$ for such elements, and go to step 2.
5. Mark: Choose an appropriate marking strategy, such as fixed-fraction marking or maximum marking [5], and mark elements with respect to the local discretization error $\eta_{D,T,i}^{(n+k)}$ for finite-cell mesh refinement.
6. Refine: Refine each marked element in $\mathcal{T}_i$ to obtain $\mathcal{T}_{i+1}$. For the quadrature mesh, let $\alpha_{i+1} : \mathcal{T}_{i+1} \to \mathbb{N}_0$ and set $\alpha_{i+1}|_{\mathcal{T}_i \cap \mathcal{T}_{i+1}} := \alpha_i$ except for children T' of any marked element T, where $\alpha_{i+1}(T') := \alpha_i(T) - 1$ unless $\alpha_i(T) - 1 < l$.
7. Increase i by 1 and go to step 2.

5 Numerical results

In this section, we study the properties of the a posteriori error estimator and the adaptive algorithm by means of three examples which are characterized by some nonlinearities. In the first example, the primal and the dual solutions are known and smooth. Then we consider an example with a known but non-smooth solu-



(a) Piecewise constant boundary approximation with $n = 4$. (b) Piecewise linear boundary approximation with $n = 1$.

Fig. 3: Single contributions to the error identity for increasing number of additional quadrature refinements.

tion. In the third example, the setting is more complex with respect to the geometry as well as the problem formulation. In this section, we omit indices such as $(n + k)$ to improve the readability.

5.1 Quarter disk

The first domain is the quarter disk $\Omega := B_1(0) \cap (0, 1)^2$ with Dirichlet boundary part $\Gamma_D := ([0, 1] \times \{0\}) \cup (\{0\} \times [0, 1])$ and Neumann boundary part $\Gamma_N := \partial\Omega \backslash \Gamma_D$. We face the difficulty that the circular domain cannot be represented exactly by quadrilateral finite elements. Thus, we embed Ω into the rectangle $\widehat{\Omega} := (0, 1)^2$. We solve the non-linear diffusion–reaction equation $-\Delta u + u^3 = f$, $u|_{\Gamma_D} = 0$, $\partial_n u|_{\Gamma_N} = g_N$. Its weak form reads

$$a(u)(v) = \int_\Omega \nabla u \cdot \nabla v + u^3 v = F(v) = \int_\Omega fv + \int_{\Gamma_N} gv$$

with Gateaux derivative

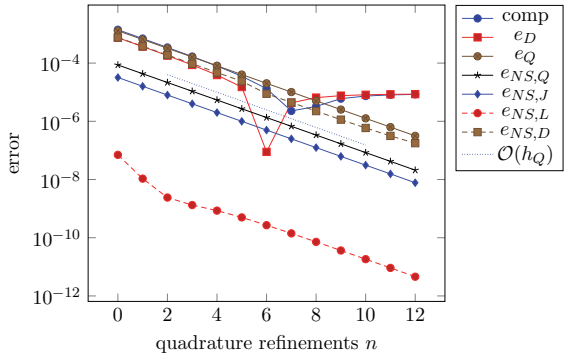
$$a'(u)(v, w) = \int_\Omega \nabla v \cdot \nabla w + 3u^2 vw.$$

The functions f and g_N are chosen such that the solution $u(x, y) := \sin(\pi x) \sin(\pi y)$ is obtained. The quantity of interest J is selected according to the analytic dual solution

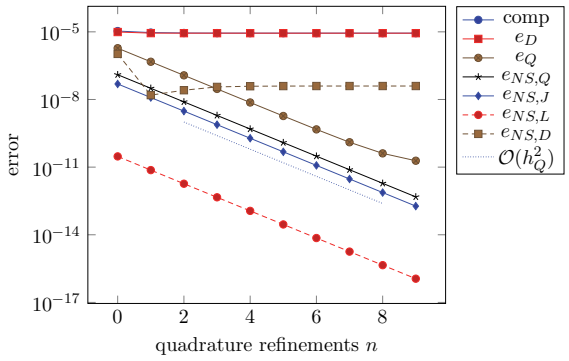
$$z(r, \varphi) := \left(-r^2 + 2r\right)\left(-\frac{16}{\pi^2}\varphi^2 + \frac{8}{\pi}\varphi\right)$$

given in polar coordinates.

First, we investigate the dependence of the a posteriori error estimator on the number of additional quadrature refinements denoted by k. We fix $\varepsilon = 10^{-10}$ and the number of mesh elements in $\mathcal{T}$ to 262 144 and use a constant quadrature refinement of $\alpha(T) := n$ for each $T \in \mathcal{T}$. In Fig. 3, the different contributions to the error identity (3.7) are plotted. We use a quadrature with a piecewise quadratic boundary approximation to evaluate the integrals almost exactly, if needed. We consider the piecewise constant and piecewise linear boundary approximation in the numerical quadrature described in Section 2 and set $n := 4$ for the piecewise constant and $n := 1$ for the piecewise linear scheme. For both quadrature techniques, we generally obtain the same results. The terms comp, e_D, e_Q, and $e_{NS,D}$ are approximately constant as expected, if k is sufficiently large. For a piecewise constant boundary approximation, we need $k \geqslant 4$ and in the linear case $k \geqslant 1$. The terms $e_{NS,Q}$, $e_{NS,J}$, and $e_{NS,\mathcal{L}}$, which we want to assume as negligibly small, are decreasing with increasing k. We denote the mesh width of the quadrature mesh by h_Q and note that an incrementation of k by 1 results in halving h_Q. For the piecewise constant boundary approximation the terms $e_{NS,Q}$, $e_{NS,J}$, and $e_{NS,\mathcal{L}}$ are of order $\mathcal{O}(h_Q)$ and for the piecewise linear one we find $\mathcal{O}(h_Q^2)$.



(a) Piecewise constant boundary approximation.



(b) Piecewise linear boundary approximation.

Fig. 4: Single contributions to the error identity for increasing number of quadrature refinements.

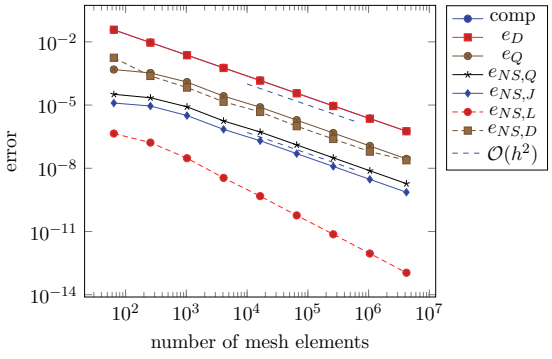


Fig. 5: Single contributions to the error identity for varying mesh size using a linear boundary approximation with $n = k = 2$.

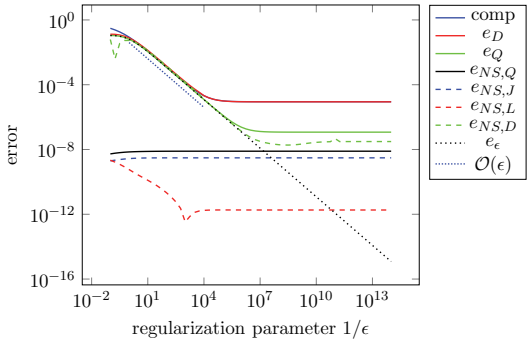


Fig. 6: Single contributions to the error identity for decreasing ε

Second, we examine the quadrature level n. To this end, we choose $\varepsilon = 10^{-10}$ and the number of mesh elements in $\mathcal{T}$ again as 262 144. Furthermore, we set $k = 4$ for the piecewise constant boundary approximation and $k = 2$ for the linear one. The results are presented in Fig. 4. The terms comp and e_D are approximately constant from the beginning in the case of linear boundary approximation. The term $e_{NS,D}$ becomes constant for $n \geqslant 4$ as expected. Also, all other terms are of order $\mathcal{O}(h_Q^2)$. In the case of a piecewise constant boundary approximation in the quadrature, we find that the terms comp and e_D become approximately constant for $n \geqslant 10$, while we do not see this for $e_{NS,D}$ here. The reason is that the values of $e_{NS,D}$ are polluted by the quadrature error even for $n = 12$. All other terms are of order $\mathcal{O}(h_Q)$.

Third, we consider varying mesh sizes. The results are depicted in Fig. 5 where $\varepsilon = 10^{-10}$ and a piecewise linear boundary approximation is used. We observe the optimal convergence in comp of order $\mathcal{O}(h^2)$. The boundary approximation is accurate enough such that no further refinements of the quadrature are needed. We see higher order convergence for $e_{NS,L}$, while the terms e_Q, $e_{NS,J}$, and $e_{NS,Q}$ are of order $\mathcal{O}(h^2)$ but are essentially smaller than the error in J. The term in which we are mainly interested here is $e_{NS,D}$ measuring the error with respect to u^+ and z^+. This term is also of order $\mathcal{O}(h^2)$ and is considerably smaller.

Fourth, we take a look at the dependence of the a posteriori error estimator on the regularization parameter ε. We fix the number of mesh elements in $\mathcal{T}$ to 262 144 and approximate the boundary in the quadrature by piecewise linear functions with $n = k = 2$. The results are illustrated in Fig. 6. We observe that for $\varepsilon < 10^{-4}$ the error is approximately constant as well as all other terms except for e_Q, $e_{NS,D}$ and e_ε. The numerical error estimator becomes constant for $\varepsilon < 10^{-6}$, which illustrates its stronger dependence on ε. This is also observed for $e_{NS,D}$, which is constant for $\varepsilon < 10^{-7}$. The term e_ε is of order $\mathcal{O}(\varepsilon)$ as expected. For large ε, we find

Tab. 1: Quarter disk: Adaptive iteration, number of degrees of freedom, range of quadrature levels, number of quadrature points and associated computational error, estimated discretization and quadrature error, and effectivity and indicator indices for a piecewise constant boundary approximation.

r	DOF	α	QP	|comp|	$\lvert e_D \rvert$	$\lvert e_Q \rvert$	eff	ind_D
1	81	0	398	$6.98 \cdot 10^{-3}$	$3.16 \cdot 10^{-3}$	$1.00 \cdot 10^{-2}$	1.89	1.03
4	81	0 – 3	818	$1.97 \cdot 10^{-3}$	$2.70 \cdot 10^{-3}$	$1.74 \cdot 10^{-3}$	0.48	1.02
5	253	1 – 3	1 562	$5.83 \cdot 10^{-4}$	$5.00 \cdot 10^{-5}$	$1.51 \cdot 10^{-3}$	2.67	1.05
7	253	1 – 5	3 170	$6.58 \cdot 10^{-4}$	$7.28 \cdot 10^{-4}$	$3.73 \cdot 10^{-4}$	0.54	1.02
8	821	1 – 4	5 198	$1.41 \cdot 10^{-4}$	$1.87 \cdot 10^{-5}$	$4.00 \cdot 10^{-4}$	2.70	1.04
10	821	2 – 6	10 886	$1.93 \cdot 10^{-4}$	$2.23 \cdot 10^{-4}$	$9.52 \cdot 10^{-5}$	0.66	1.04
11	2 833	1 – 5	19 334	$2.52 \cdot 10^{-5}$	$1.67 \cdot 10^{-5}$	$9.48 \cdot 10^{-5}$	3.10	1.07
13	2 833	3 – 7	41 678	$5.17 \cdot 10^{-5}$	$6.15 \cdot 10^{-5}$	$2.47 \cdot 10^{-5}$	0.71	1.05
14	3 241	2 – 7	41 982	$3.94 \cdot 10^{-5}$	$5.04 \cdot 10^{-5}$	$2.37 \cdot 10^{-5}$	0.68	1.03
15	11 993	1 – 6	76 248	$1.06 \cdot 10^{-5}$	$1.84 \cdot 10^{-6}$	$2.48 \cdot 10^{-5}$	2.16	1.05
17	11 993	3 – 8	151 968	$7.14 \cdot 10^{-6}$	$1.12 \cdot 10^{-5}$	$8.72 \cdot 10^{-6}$	0.35	1.04
18	46 189	2 – 7	281 830	$5.71 \cdot 10^{-6}$	$1.24 \cdot 10^{-6}$	$9.00 \cdot 10^{-6}$	1.79	1.05
21	46 189	5 – 10	890 566	$2.22 \cdot 10^{-6}$	$3.29 \cdot 10^{-6}$	$1.82 \cdot 10^{-6}$	0.66	1.04
22	48 205	5 – 10	904 478	$2.18 \cdot 10^{-6}$	$2.99 \cdot 10^{-6}$	$1.71 \cdot 10^{-6}$	0.59	1.02
23	185 533	4 – 9	1 422 858	$7.87 \cdot 10^{-7}$	$1.55 \cdot 10^{-8}$	$1.64 \cdot 10^{-6}$	2.07	1.05
25	185 533	6 – 11	3 116 970	$4.40 \cdot 10^{-7}$	$7.50 \cdot 10^{-7}$	$5.32 \cdot 10^{-7}$	0.50	1.04
26	190 989	5 – 11	3 100 030	$4.37 \cdot 10^{-7}$	$6.90 \cdot 10^{-7}$	$5.22 \cdot 10^{-7}$	0.38	1.03
27	728 291	4 – 10	5 155 976	$2.98 \cdot 10^{-7}$	$4.78 \cdot 10^{-8}$	$5.05 \cdot 10^{-7}$	1.85	1.06
29	728 291	6 – 12	11 001 704	$9.24 \cdot 10^{-8}$	$1.83 \cdot 10^{-7}$	$1.51 \cdot 10^{-7}$	0.34	1.06
30	744 899	5 – 12	10 984 642	$9.22 \cdot 10^{-8}$	$1.66 \cdot 10^{-7}$	$1.48 \cdot 10^{-7}$	0.19	1.02

Tab. 2: Quarter disk: Adaptive iteration, range of quadrature levels, number of quadrature points and associated computational error, estimated discretization and quadrature error, and effectivity and indicator indices for a piecewise constant boundary approximation for a fixed uniformly refined mesh with 16 641 degrees of freedom.

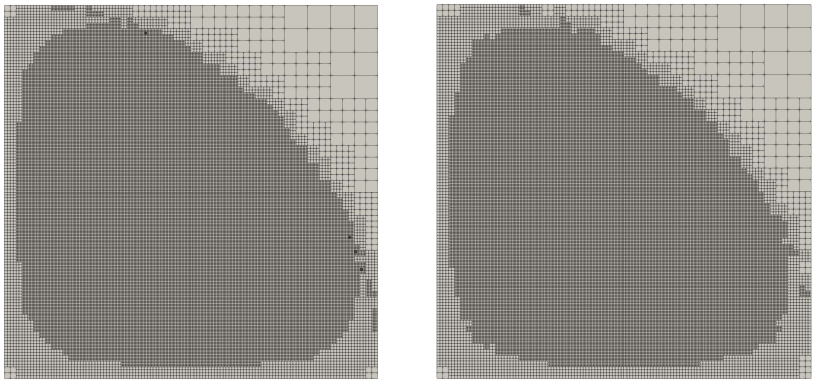
r	α	QP	|comp|	$\lvert e_D \rvert$	$\lvert e_Q \rvert$	eff	ind_D
1	0	68 078	$7.57 \cdot 10^{-4}$	$4.45 \cdot 10^{-4}$	$7.04 \cdot 10^{-4}$	1.5192	1.32
2	0 – 1	70 382	$3.70 \cdot 10^{-4}$	$2.12 \cdot 10^{-4}$	$3.51 \cdot 10^{-4}$	1.5223	1.20
3	0 – 2	70 442	$1.77 \cdot 10^{-4}$	$9.64 \cdot 10^{-5}$	$1.75 \cdot 10^{-4}$	1.5332	1.12
4	0 – 3	74 834	$8.60 \cdot 10^{-5}$	$4.20 \cdot 10^{-5}$	$9.21 \cdot 10^{-5}$	1.5592	1.08
5	0 – 4	83 678	$4.22 \cdot 10^{-5}$	$1.58 \cdot 10^{-5}$	$5.22 \cdot 10^{-5}$	1.6133	1.06
6	1 – 5	101 474	$1.66 \cdot 10^{-5}$	$1.84 \cdot 10^{-6}$	$2.89 \cdot 10^{-5}$	1.8545	1.04
7	2 – 6	135 458	$2.68 \cdot 10^{-6}$	$5.59 \cdot 10^{-6}$	$1.64 \cdot 10^{-5}$	4.0138	1.02
8	3 – 7	204 794	$5.95 \cdot 10^{-6}$	$1.03 \cdot 10^{-5}$	$8.55 \cdot 10^{-6}$	0.2850	1.02
9	4 – 8	342 746	$1.05 \cdot 10^{-5}$	$1.32 \cdot 10^{-5}$	$4.48 \cdot 10^{-6}$	0.8349	1.02
10	5 – 9	618 938	$1.29 \cdot 10^{-5}$	$1.43 \cdot 10^{-5}$	$2.32 \cdot 10^{-6}$	0.9297	1.02
11	6 – 10	1 167 098	$1.40 \cdot 10^{-5}$	$1.48 \cdot 10^{-5}$	$1.26 \cdot 10^{-6}$	0.9621	1.02
12	7 – 11	2 259 194	$1.46 \cdot 10^{-5}$	$1.50 \cdot 10^{-5}$	$7.03 \cdot 10^{-7}$	0.9782	1.02
13	8 – 12	4 447 226	$1.50 \cdot 10^{-5}$	$1.52 \cdot 10^{-5}$	$3.73 \cdot 10^{-7}$	0.9880	1.02
14	9 – 13	8 823 290	$1.52 \cdot 10^{-5}$	$1.53 \cdot 10^{-5}$	$1.86 \cdot 10^{-7}$	0.9933	1.02
15	10 – 14	17 556 986	$1.53 \cdot 10^{-5}$	$1.53 \cdot 10^{-5}$	$9.32 \cdot 10^{-8}$	0.9957	1.02
16	11 – 15	35 024 378	$1.54 \cdot 10^{-5}$	$1.54 \cdot 10^{-5}$	$4.66 \cdot 10^{-8}$	0.9969	1.02

the same behaviour also for the error and error estimator. Consequently, choosing $\varepsilon = 10^{-10}$ is a sufficiently small value to ensure that the error due to the regularization is negligibly small.

Finally, we test the adaptive algorithm in this example, where we set the minimum quadrature refinement $l = 1$ and start with $\alpha = 0$ everywhere. We work with $k = 4$ for the piecewise constant boundary approximation and with $k = 2$ for the piecewise linear one. We expect uniform mesh refinements in the original domain because of the smoothness of the primal and dual solutions. Outside of the original domain, additional refinements should not occur. This expected structure of the adaptive meshes is produced by the adaptive algorithm, cf. Fig. 7. We find the optimal convergence order for adaptive and uniform refinement as

Tab. 3: Quarter disk: Adaptive iteration, number of degrees of freedom, range of quadrature levels, number of quadrature points and associated computational error, estimated discretization and quadrature error, and effectivity and indicator indices for a piecewise linear boundary approximation.

r	DOF	α	QP	$\lvert\text{comp}\rvert$	$\lvert e_D\rvert$	$\lvert e_Q\rvert$	eff	ind_D
1	81	0	401	$5.08 \cdot 10^{-3}$	$4.65 \cdot 10^{-3}$	$1.09 \cdot 10^{-3}$	1.13	1.03
2	253	0 – 1	1 323	$1.84 \cdot 10^{-3}$	$1.45 \cdot 10^{-3}$	$7.03 \cdot 10^{-4}$	1.17	1.01
3	821	1	3 641	$3.44 \cdot 10^{-4}$	$3.12 \cdot 10^{-4}$	$4.21 \cdot 10^{-5}$	1.03	1.03
4	2 833	1	12 109	$9.48 \cdot 10^{-5}$	$8.54 \cdot 10^{-5}$	$1.43 \cdot 10^{-5}$	1.05	1.06
5	3 229	1	14 177	$7.55 \cdot 10^{-5}$	$7.06 \cdot 10^{-5}$	$8.66 \cdot 10^{-6}$	1.05	1.02
6	11 917	1	49 667	$2.00 \cdot 10^{-5}$	$1.85 \cdot 10^{-5}$	$2.89 \cdot 10^{-6}$	1.07	1.04
7	12 389	1	52 261	$1.82 \cdot 10^{-5}$	$1.73 \cdot 10^{-5}$	$1.76 \cdot 10^{-6}$	1.05	1.01
8	46 997	1	192 451	$4.73 \cdot 10^{-6}$	$4.48 \cdot 10^{-6}$	$5.49 \cdot 10^{-7}$	1.06	1.04
9	48 533	1	200 271	$4.39 \cdot 10^{-6}$	$4.22 \cdot 10^{-6}$	$3.17 \cdot 10^{-7}$	1.03	1.02
10	184 549	1	748 611	$1.12 \cdot 10^{-6}$	$1.08 \cdot 10^{-6}$	$8.07 \cdot 10^{-8}$	1.04	1.05
11	189 957	1	774 619	$1.06 \cdot 10^{-6}$	$1.04 \cdot 10^{-6}$	$4.06 \cdot 10^{-8}$	1.01	1.02
12	729 845	1	2 943 023	$2.72 \cdot 10^{-7}$	$2.67 \cdot 10^{-7}$	$1.19 \cdot 10^{-8}$	1.03	1.06
13	747 537	1	3 024 991	$2.63 \cdot 10^{-7}$	$2.60 \cdot 10^{-7}$	$7.48 \cdot 10^{-9}$	1.02	1.02



(a) Mesh in the 22nd adaptive iteration with piecewise constant boundary approximation.

(b) Mesh in the 8th adaptive iteration with piecewise linear boundary approximation.

Fig. 7: Adaptive meshes for the quarter disk example with $\rho = 1$.

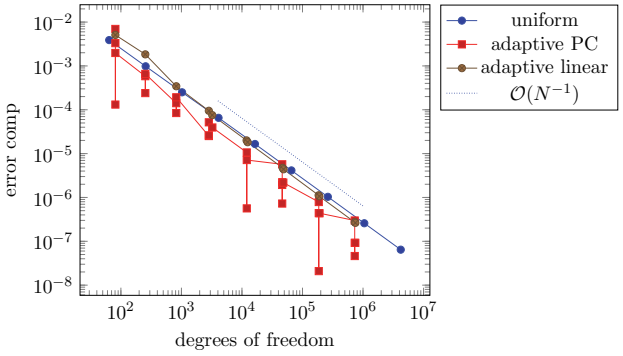


Fig. 8: Convergence results for the quarter disk example with $\rho = 1$.

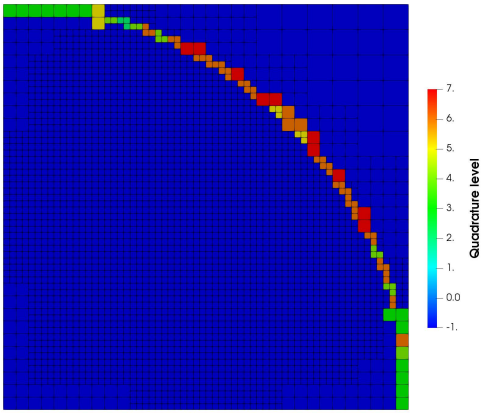
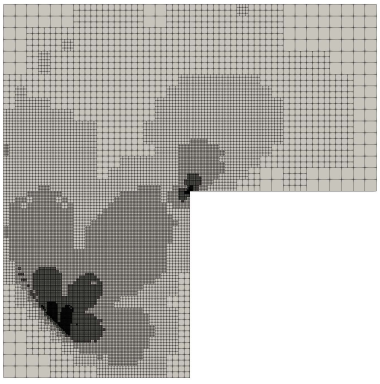
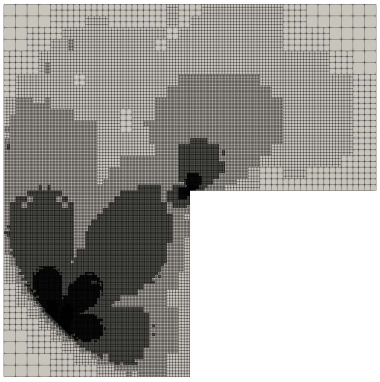


Fig. 9: Quarter disk: Distribution of the quadrature level in the 14th iteration of adaptive algorithm for the piecewise constant boundary approximation.



(a) Mesh in the 28^{th} adaptive iteration with piecewise constant boundary approximation.



(b) Mesh in the 19^{th} adaptive iteration with piecewise linear boundary approximation.

Fig. 10: Adaptive meshes for the L-domain example.

depicted in Fig. 8 , where N denotes the total number of degrees of freedom. The detailed results of the adaptive algorithms for piecewise constant and linear boundary approximations are listed for $\rho = 1$ in Table 1 and 3 respectively. For the piecewise linear boundary approximation, we find effectivity indices, which are very close to 1. It should be remarked, that no additional quadrature refinements are carried out. The behaviour in the case of the piecewise constant boundary approximation is completely different. Here, we need additional refinements of the quadrature mesh on each refinement level. The reason is the reduced convergence order of the quadrature rule. Furthermore, the numerical error is significantly higher than in the case of the piecewise linear boundary approximation and pollutes the a posteriori error estimate leading to effectivity indices which are oscillating and which are not close to one. The higher-order reconstruction used to evaluate the error identity numerically relies on asymptotic properties of the finite element solution in the nodes, cf. [2, Section 5.2.ii]. However, the relatively large numerical error disturbs these properties. In Table 2, we list the effectivity indices for a fixed mesh under quadrature refinement. We observe that the estimated numerical error should be a factor about 10 smaller than the estimated discretization error to obtain good effectivity indices. However, although the effectivity indices are not good, the adaptive algorithm performs optimally as shown in Fig. 8. All in all, we save a large amount of computational work in the numerical quadrature compared to choices like $\rho = 0.1$ or $\rho = 0.01$. However, there is a loss of accuracy in the error estimate.

For the piecewise constant boundary approximation, we plot the distribution of the quadrature level in Fig. 9. The quadrature level is low at the ends of the fictitious boundary and high in the interior. Also, it can be seen that the quadrature level is higher on elements with a larger diameter. This is due to the fact that a certain quadrature level is required to ensure coercivity and the refinement of the finite-cell mesh induces a refinement of the quadrature.

5.2 Circular domain with reentrant corner

In a second series of experiments, we consider the circular domain with reentrant corner $\Omega := B_1(0) \setminus ([0,1] \times [-1,0])$. The Dirichlet boundary part is $\Gamma_D := ([0,1] \times \{0\}) \cup (\{0\} \times [-1,0])$ and the Neumann boundary part is $\Gamma_N := \partial\Omega \setminus \Gamma_D$. For the discretization via the finite-cell method, we embed Ω into the L-shaped domain $\widehat{\Omega} := (-1,1)^2 \setminus ([0,1] \times [-1,0])$. The initial finite-cell mesh $\mathcal{T}_0$ consists of $3 \cdot 16$ square elements of degree 1. Again, we consider the diffusion–reaction equation $-\Delta u + u^3 = f$, $u|_{\Gamma_D} = 0$, $\partial_n u|_{\Gamma_N} = g_N$. The functions f and g_N are chosen such that $u(r,\varphi) = r^{2/3} \sin\left(\frac{2}{3}\varphi\right)$ is the solution given in polar coordinates. Except for the circular arc on the Neumann boundary, this problem closely resembles the classical L-shaped domain problem. In

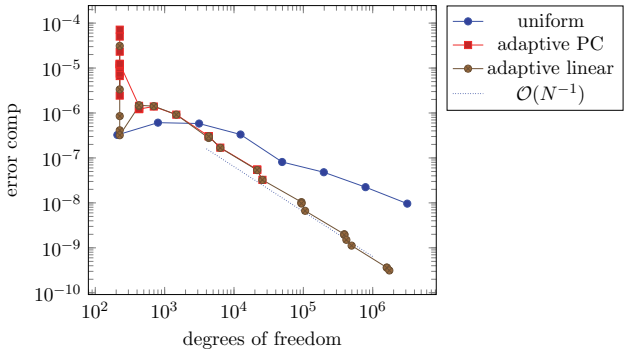


Fig. 11: Convergence results for the L-domain example.

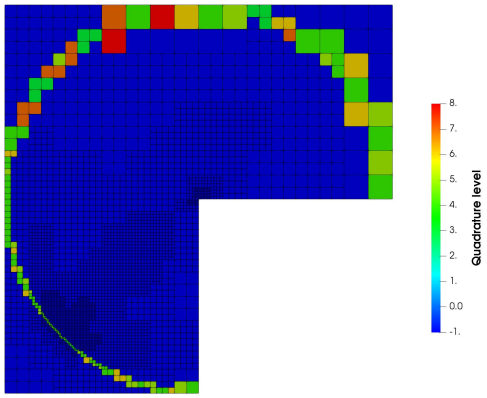


Fig. 12: L-domain: Distribution of the quadrature level in the 13th iteration of adaptive algorithm for the piecewise linear boundary approximation.

Tab. 4: L-domain: Adaptive iteration, number of degrees of freedom, range of quadrature levels, number of quadrature points and associated computational error, estimated discretization and quadrature error, and effectivity and indicator indices for a piecewise constant boundary approximation.

r	DOF	α	QP	$\lvert\text{comp}\rvert$	$\lvert e_D\rvert$	$\lvert e_Q\rvert$	eff	ind_D
1	225	0	1 194	$7.15 \cdot 10^{-5}$	$2.40 \cdot 10^{-4}$	$1.33 \cdot 10^{-4}$	−1.50	6.74
11	225	6 – 10	161 550	$1.25 \cdot 10^{-5}$	$1.57 \cdot 10^{-5}$	$1.54 \cdot 10^{-7}$	1.25	2.43
12	427	6 – 10	162 242	$1.44 \cdot 10^{-6}$	$2.68 \cdot 10^{-8}$	$2.32 \cdot 10^{-7}$	0.14	24.67
19	427	13 – 17	18 777 410	$1.23 \cdot 10^{-6}$	$1.95 \cdot 10^{-7}$	$1.45 \cdot 10^{-9}$	0.16	32.46
20	697	13 – 17	18 483 434	$1.39 \cdot 10^{-6}$	$1.05 \cdot 10^{-6}$	$1.70 \cdot 10^{-9}$	0.75	25.05
21	1 473	12 – 17	18 019 328	$9.20 \cdot 10^{-7}$	$7.60 \cdot 10^{-7}$	$9.48 \cdot 10^{-10}$	0.83	5.33
22	4 313	11 – 17	18 300 420	$3.06 \cdot 10^{-7}$	$2.21 \cdot 10^{-7}$	$3.69 \cdot 10^{-9}$	0.73	4.38
23	4 313	12 – 18	32 161 284	$3.04 \cdot 10^{-7}$	$2.21 \cdot 10^{-7}$	$1.75 \cdot 10^{-9}$	0.73	4.39
24	6 343	12 – 18	31 603 560	$1.70 \cdot 10^{-7}$	$1.52 \cdot 10^{-7}$	$1.50 \cdot 10^{-9}$	0.91	4.34
25	21 547	11 – 17	29 032 928	$5.52 \cdot 10^{-8}$	$4.41 \cdot 10^{-8}$	$1.33 \cdot 10^{-9}$	0.82	3.34
26	21 547	12 – 18	52 183 520	$5.46 \cdot 10^{-8}$	$4.43 \cdot 10^{-8}$	$6.48 \cdot 10^{-10}$	0.82	3.34
27	21 547	13 – 19	99 172 832	$5.43 \cdot 10^{-8}$	$4.46 \cdot 10^{-8}$	$3.22 \cdot 10^{-10}$	0.83	3.34
28	25 697	13 – 18	94 223 878	$3.26 \cdot 10^{-8}$	$2.91 \cdot 10^{-8}$	$2.99 \cdot 10^{-10}$	0.90	4.02

particular, it features a corner singularity in $(0, 0)$ leading to a reduced regularity. Consequently, we cannot expect that the finite cell method based on uniform refinement yields optimal algebraic convergence rates.

We aim to control the error of $|\nabla u|^2$ at the point $p = (\cos\left(\frac{5}{4}\pi\right), \sin\left(\frac{5}{4}\pi\right)) \in \partial\Omega$, which we approximate by $J(v) := \int_{\widehat{\Omega}} \tilde{I}_S |\nabla v|^2$, where $\tilde{I}_S$ with $S = B_{0.01}(p) \cap \Omega$ is a smoothed indicator function given on S by

$$\tilde{I}_S(x) = \begin{cases} w(|x-p|), & x \in S, \\ 0, & x \in \widehat{\Omega} \backslash S \end{cases}$$

where $w(r) = -10^6 \cdot r^3 + 3 \cdot 10^4 \cdot r^2 - 3 \cdot 10^2 \cdot r + 1.0$. Since the exact solution u is known, the functional J can be evaluated up to machine precision on S. Here, we compute $J(u) \approx 0.0017523852871125355$.

The adaptive meshes generated in the adaptive algorithm with $\rho = 0.01$ are depicted in Fig. 10. We find strong refinements close to the point p and in the reentrant corner as well as between these two regions, which is expected. The convergence is examined in Fig. 11. After some quadrature refinements in the beginning, we find optimal convergence of order $\mathcal{O}(N^{-1})$ for the adaptive approach. Using uniform refinements, the convergence order is reduced as expected. The results for piecewise constant and linear boundary approximations in the quadrature are almost identical. However, the computation with the piecewise constant boundary approximation stops earlier due to the huge number of quadrature points. The two approaches deliver similar results which are given in detail in the Tables 4 and 5. However, we need approximately 690 times more

Tab. 5: L-domain: Adaptive iteration, number of degrees of freedom, range of quadrature levels, number of quadrature points and associated computational error, estimated discretization and quadrature error, and effectivity and indicator indices for a piecewise linear boundary approximation.

r	DOF	α	QP	$\lvert\text{comp}\rvert$	$\lvert e_D\rvert$	$\lvert e_Q\rvert$	eff	ind_D
1	225	0	1 203	$3.12 \cdot 10^{-5}$	$3.08 \cdot 10^{-5}$	$1.15 \cdot 10^{-5}$	−0.62	1.60
5	225	0 – 4	4 683	$3.21 \cdot 10^{-7}$	$1.61 \cdot 10^{-5}$	$3.53 \cdot 10^{-8}$	50.33	2.41
6	427	0 – 4	5 477	$1.43 \cdot 10^{-6}$	$2.42 \cdot 10^{-8}$	$5.53 \cdot 10^{-8}$	−0.02	41.57
8	427	2 – 6	14 629	$1.48 \cdot 10^{-6}$	$2.19 \cdot 10^{-7}$	$7.16 \cdot 10^{-10}$	0.15	37.26
9	697	2 – 6	15 571	$1.42 \cdot 10^{-6}$	$1.04 \cdot 10^{-6}$	$9.47 \cdot 10^{-9}$	0.72	24.04
10	1 473	2 – 6	18 201	$9.20 \cdot 10^{-7}$	$7.36 \cdot 10^{-7}$	$6.91 \cdot 10^{-10}$	0.80	5.33
11	4 313	1 – 6	29 307	$2.78 \cdot 10^{-7}$	$1.95 \cdot 10^{-7}$	$2.48 \cdot 10^{-8}$	0.61	4.48
13	4 313	3 – 8	56 475	$3.02 \cdot 10^{-7}$	$2.20 \cdot 10^{-7}$	$9.35 \cdot 10^{-10}$	0.72	4.41
14	6 343	2 – 7	63 287	$1.68 \cdot 10^{-7}$	$1.52 \cdot 10^{-7}$	$5.25 \cdot 10^{-10}$	0.90	4.34
15	21 547	1 – 7	121 289	$5.37 \cdot 10^{-8}$	$4.40 \cdot 10^{-8}$	$3.26 \cdot 10^{-10}$	0.81	3.34
16	25 669	1 – 7	136 843	$3.25 \cdot 10^{-8}$	$2.88 \cdot 10^{-8}$	$1.19 \cdot 10^{-10}$	0.89	3.98
17	93 849	1 – 6	410 403	$1.04 \cdot 10^{-8}$	$7.87 \cdot 10^{-9}$	$6.97 \cdot 10^{-10}$	0.82	4.02
19	93 849	3 – 8	496 371	$9.70 \cdot 10^{-9}$	$8.46 \cdot 10^{-9}$	$5.27 \cdot 10^{-11}$	0.88	4.02
20	105 963	2 – 8	544 173	$6.67 \cdot 10^{-9}$	$6.08 \cdot 10^{-9}$	$5.11 \cdot 10^{-11}$	0.92	4.38
21	389 419	1 – 7	1 667 969	$2.01 \cdot 10^{-9}$	$1.68 \cdot 10^{-9}$	$5.93 \cdot 10^{-11}$	0.86	4.84
23	389 419	3 – 9	2 014 449	$1.95 \cdot 10^{-9}$	$1.64 \cdot 10^{-9}$	$5.03 \cdot 10^{-12}$	0.84	4.84
24	416 891	2 – 8	2 115 655	$1.51 \cdot 10^{-9}$	$1.50 \cdot 10^{-9}$	$5.08 \cdot 10^{-12}$	1.00	4.63
25	495 939	1 – 8	2 431 635	$1.12 \cdot 10^{-9}$	$1.14 \cdot 10^{-9}$	$5.19 \cdot 10^{-12}$	1.02	5.41
26	1 596 013	1 – 7	6 814 267	$3.64 \cdot 10^{-10}$	$4.56 \cdot 10^{-10}$	$5.14 \cdot 10^{-12}$	1.26	5.26
27	1 596 013	2 – 8	7 204 059	$3.62 \cdot 10^{-10}$	$3.81 \cdot 10^{-10}$	$2.82 \cdot 10^{-12}$	1.06	5.26
28	1 736 841	1 – 8	7 762 007	$3.14 \cdot 10^{-10}$	$3.39 \cdot 10^{-10}$	$2.79 \cdot 10^{-12}$	1.09	6.60

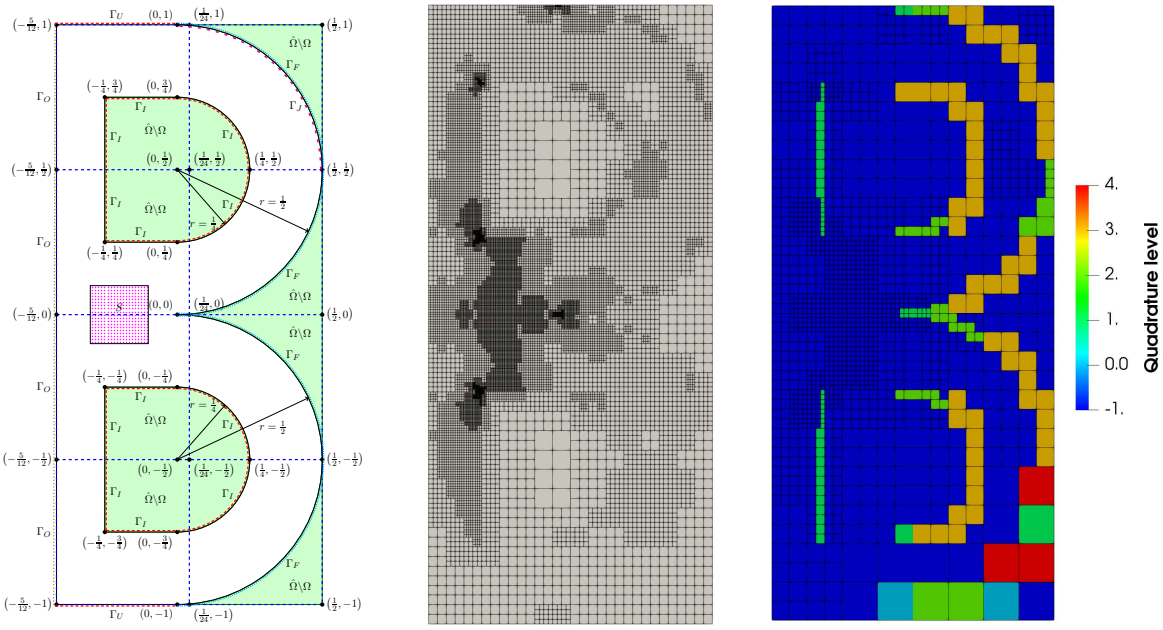
quadrature points when we use the piecewise constant boundary approximation to achieve the same accuracy as with the linear approach. This is also substantiated by the significantly higher quadrature levels for the piecewise constant boundary approximation. In Fig. 12, the distribution of the quadrature level is shown for the piecewise linear boundary approximation. We find a high level near to S if the cells are comparatively coarse. Furthermore, in the upper left corner, we find high quadrature levels. The effectivity indices lie in an acceptable range between 0.8 and 1.3 for higher N even though the problem is of low regularity, cf. Tables 4 and 5.

5.3 B-domain

In this section, we consider a domain, which we call B-domain. It is illustrated together with the initial mesh in Fig. 13a, where the used notation is also defined. The domain has two holes and a strong singularity in the point $(0, 0)$, which is located on the fictitious boundary and not in a node of the mesh. The underlying problem formulation is model plasticity with linear isotropic hardening, we refer to [38] for a complete description. It is given by the PDE

$$-\operatorname{div}(C(\nabla u)) = f, \qquad C(\nabla u) = \begin{cases} \nabla u, & |\nabla u| < \sigma_0 \\ \zeta \nabla u + \dfrac{1-\zeta}{|\nabla u|}\nabla u, & \text{else.} \end{cases} \tag{5.1}$$

Here, $\sigma_0 > 0$ denotes the yield stress and $0 \leqslant \zeta \ll 1$ the hardening parameter. We choose $\sigma_0 = 1$ and $\zeta = 0.01$. We assume homogeneous Neumann boundary conditions on the boundary of the holes and on two small parts of the outer boundary, $\Gamma_N = \Gamma_I \cup \Gamma_U$, and homogeneous Dirichlet boundary conditions on the outer boundary, $\Gamma_D = \Gamma_O \cup \Gamma_F$. Since the homogeneous Dirichlet boundary conditions on Γ_F cannot be realized in the usual finite element way in the finite cell method, we approximate them by a penalty approach with a large penalty parameter $\gamma \gg 0$. The initial γ is set to 10^2 as well as $\varepsilon = 10^{-10}$. The right hand side f is



(a) Sketch of the B-domain and the initial mesh.

(b) Mesh in the 14th adaptive iteration with about 20 000 DOF.

(c) Distribution of the quadrature level in the 8th iteration

Fig. 13: Sketch of the B-domain example, adaptive mesh, and distribution of the quadrature level in the 8th iteration of adaptive algorithm.

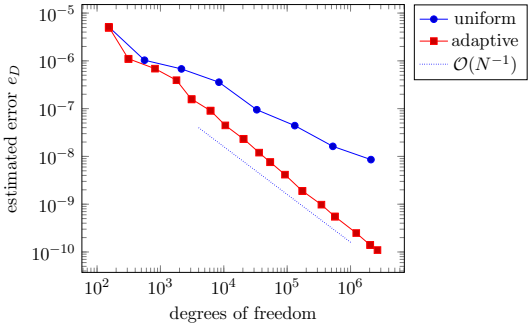


Fig. 14: Convergence results for the B-domain example.

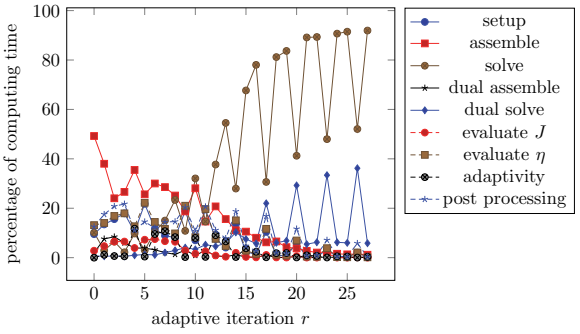


Fig. 15: Percentage of the computing time in each adaptive iteration.

given by a constant function with $f \equiv 2$. The weak form of equation (5.1) reads

$$a(u)(v) = (C(\nabla u), \nabla v) + \gamma \int_{\Gamma_F} u^3 v = F(v) = (f, v) \qquad \forall v \in V.$$

It should be remarked that a need not be three times directional differentiable. Nevertheless, our approach works as shown in the following results. We are interested in the quantity of interest J given by

$$J(v) = \int_S \omega_D v^2 + \int_{\Gamma_J} \omega_F v, \qquad S = (-0.3, -0.1) \times (-0.1, 0.1)$$

where $\omega_D(x, y) = 10^8 \cdot (x + 0.3)^2 (x + 0.1)^2 (y + 0.1)^2 (y - 0.1)^2$ and $\omega_F(\varphi, r) = \varphi^2(\varphi - 0.5\pi)^2$.

We have seen in the last example that the results for a piecewise linear and a piecewise constant boundary approximation are almost the same except for the number of quadrature points needed. To shorten the presentation we focus here on the linear boundary approximation with $\rho = 0.01$ and $k = 2$. In Fig. 13b, the

Tab. 6: B-domain: Adaptive iteration, number of degrees of freedom, number of quadrature points, range of quadrature levels, estimated discretization and quadrature error, as well as effectivity and indicator indices for a piecewise linear boundary approximation.

r	DOF	α	QP	|comp|	$\lvert e_D \rvert$	$\lvert e_Q \rvert$	eff	ind_D
1	153	0	1 028	$1.99 \cdot 10^{-6}$	$4.91 \cdot 10^{-6}$	$3.96 \cdot 10^{-7}$	2.26	1.14
2	153	0 – 1	1 364	$2.19 \cdot 10^{-6}$	$4.98 \cdot 10^{-6}$	$2.42 \cdot 10^{-7}$	2.16	1.15
3	153	0 – 2	2 028	$2.39 \cdot 10^{-6}$	$5.10 \cdot 10^{-6}$	$6.11 \cdot 10^{-8}$	2.11	1.15
4	153	0 – 3	3 324	$2.43 \cdot 10^{-6}$	$5.13 \cdot 10^{-6}$	$1.58 \cdot 10^{-8}$	2.10	1.15
5	313	0 – 3	3 952	$2.10 \cdot 10^{-6}$	$1.09 \cdot 10^{-6}$	$1.82 \cdot 10^{-8}$	0.51	2.09
6	313	0 – 4	6 504	$2.12 \cdot 10^{-6}$	$1.11 \cdot 10^{-6}$	$4.62 \cdot 10^{-9}$	0.52	2.09
7	821	0 – 4	8 750	$2.02 \cdot 10^{-6}$	$6.87 \cdot 10^{-7}$	$4.42 \cdot 10^{-9}$	0.34	1.49
8	1 791	0 – 4	12 686	$1.05 \cdot 10^{-6}$	$3.93 \cdot 10^{-7}$	$1.87 \cdot 10^{-9}$	0.37	1.43
9	3 105	1 – 3	18 142	$4.73 \cdot 10^{-7}$	$1.55 \cdot 10^{-7}$	$3.97 \cdot 10^{-9}$	0.32	1.49
10	3 105	1 – 4	23 950	$4.76 \cdot 10^{-7}$	$1.58 \cdot 10^{-7}$	$1.00 \cdot 10^{-9}$	0.33	1.49
11	6 175	1 – 4	36 200	$2.45 \cdot 10^{-7}$	$9.03 \cdot 10^{-8}$	$9.68 \cdot 10^{-10}$	0.36	1.43
12	6 175	1 – 5	47 624	$2.46 \cdot 10^{-7}$	$9.09 \cdot 10^{-8}$	$2.82 \cdot 10^{-10}$	0.37	1.43
13	10 569	1 – 5	64 954	$1.01 \cdot 10^{-7}$	$4.46 \cdot 10^{-8}$	$3.03 \cdot 10^{-10}$	0.44	1.53
14	20 473	1 – 4	104 258	$4.35 \cdot 10^{-8}$	$2.31 \cdot 10^{-8}$	$3.33 \cdot 10^{-10}$	0.52	1.45
15	20 473	1 – 5	127 698	$4.38 \cdot 10^{-8}$	$2.34 \cdot 10^{-8}$	$8.36 \cdot 10^{-11}$	0.53	1.45
16	36 087	1 – 5	189 172	$1.90 \cdot 10^{-8}$	$1.20 \cdot 10^{-8}$	$9.03 \cdot 10^{-11}$	0.63	1.54
17	54 089	1 – 5	260 416	$1.00 \cdot 10^{-8}$	$7.55 \cdot 10^{-9}$	$8.78 \cdot 10^{-11}$	0.75	1.51
18	54 089	2 – 6	313 200	$1.01 \cdot 10^{-8}$	$7.59 \cdot 10^{-9}$	$2.20 \cdot 10^{-11}$	0.75	1.51
19	92 367	1 – 5	462 994	$5.05 \cdot 10^{-9}$	$4.15 \cdot 10^{-9}$	$2.12 \cdot 10^{-11}$	0.82	1.51
20	173 843	1 – 5	783 662	$2.37 \cdot 10^{-9}$	$1.89 \cdot 10^{-9}$	$2.23 \cdot 10^{-11}$	0.78	1.71
21	173 843	2 – 6	883 054	$2.39 \cdot 10^{-9}$	$1.90 \cdot 10^{-9}$	$5.65 \cdot 10^{-12}$	0.79	1.70
22	347 209	1 – 5	1 568 260	$1.15 \cdot 10^{-9}$	$9.77 \cdot 10^{-10}$	$5.77 \cdot 10^{-12}$	0.84	1.59
23	565 929	1 – 5	2 432 740	$6.34 \cdot 10^{-10}$	$5.49 \cdot 10^{-10}$	$5.62 \cdot 10^{-12}$	0.86	1.71
24	565 929	2 – 6	2 615 156	$6.39 \cdot 10^{-10}$	$5.52 \cdot 10^{-10}$	$1.54 \cdot 10^{-12}$	0.86	1.71
25	1 230 765	1 – 6	5 249 306	$2.96 \cdot 10^{-10}$	$2.49 \cdot 10^{-10}$	$1.51 \cdot 10^{-12}$	0.84	1.72
26	2 046 323	1 – 5	8 504 968	$1.76 \cdot 10^{-10}$	$1.39 \cdot 10^{-10}$	$1.52 \cdot 10^{-12}$	0.78	1.81
27	2 046 323	2 – 6	8 891 944	$1.77 \cdot 10^{-10}$	$1.40 \cdot 10^{-10}$	$4.10 \cdot 10^{-13}$	0.79	1.81
28	2 661 737	1 – 6	11 318 682	$1.37 \cdot 10^{-10}$	$1.10 \cdot 10^{-10}$	$4.15 \cdot 10^{-13}$	0.80	1.72

mesh in the 14^{th} iteration of the adaptive algorithm is depicted. We find strong refinements around the singularity in $(0, 0)$ as well as around S and Γ_J and around the edges of the holes, which are expected. Since the exact solution of this example is unknown, we cannot compute $J(u)$ exactly. Hence, we approximate $J(u)$ using extrapolation techniques based on the results of the adaptive algorithm and obtain

$$J(u) \approx 4.3067544444371864 \cdot 10^{-4}.$$

This numerically computed reference value is used in the calculation of the error and the effectivity index. We compare the convergence properties of the uniform and the adaptive approach in Fig. 14. The adaptive approach outperforms the uniform approach by far. The results are detailed in Table 6. For finer meshes the effectivity indices are in the range of 0.8. This is a very good value for such a complex example especially in the presence of the nonsmooth differential operator. The indicator indices are smaller than 1.85 in most iterations, which is also a good result. We find that the overall quadrature refinement levels are lower than for the L-domain example. The distribution of the quadrature level in the 8^{th} iteration of the adaptive algorithm for the piecewise linear boundary approximation is depicted in Fig. 13c. The quadrature levels are higher on coarser mesh cells and smaller on fine mesh cells similar to the other examples. It should be pointed out that the quadrature is exact on the straight lines of the holes such that no quadrature refinements at all are carried out. The level is one because the quadrature level is adjusted to the minimal quadrature level during the refinement process of the mesh cells.

5.4 Implementation details and analysis of computing time

Finally, we give some remarks on the implementation and take a look at the computing time of the adaptive algorithm. We have implemented the numerical quadrature in parallel on shared memory computers, so that all integration operations as the assembling of vectors and matrices or the evaluation of J and the error estimator are performed in parallel with a good strong scaling. However, all other parts of the code especially the linear solvers are not parallelized. The nonlinear systems are solved with a damped Newton iteration. In this example, we need up to 40 Newton iterations with strong damping in the first iterations, when the mesh is refined. We refer to [38] for more information concerning the solution algorithm used for this class of problems. If a quadrature refinement is performed, the quadratic convergence of the Newton scheme is realized and only two or three Newton iterations are needed. The linear system in each Newton step is solved with a conjugate gradients (CG) method preconditioned by a symmetric SOR scheme.

In Fig. 15, we compare the computing time of the different parts of the adaptive algorithm. We summarize all operations concerning memory allocation, quadrature and parallelization preparation and so on under the designation 'setup'. By 'assemble' all assembling operations during the solution of the primal problem are collected, while 'solve' is the cumulated computing time for solving all linear systems in this part. The counterparts concerning the dual problem are named by 'dual assemble' and 'dual solve'. The evaluation of the quantity of interest is given by 'evaluate J'. The process of the calculation of the error estimator η is summarized in 'evaluate η', where we note that several operations especially concerning the localization are not parallelised. The adaptive refinement routines for the mesh and the quadrature are considered under the designation 'adaptivity'. The graphical output of the mesh and the output of the data for the tables and graphs are collected in 'post processing'. The computations are carried out on a Sun Fire compute server with 8 AMD Quad-Core Opteron 8356 CPUs (2.3 GHz) and 64GB RAM, where we use 16 cores for the calculations concerning this example. Figure 15 shows that the computing time for a large number of unknowns is dominated by the solution process of the linear systems. In the last step nearly 92% of the computing time is needed for the solution of the linear systems. Approximately 1.2% is used for the assembling and 5.9% for the solution of the dual problem. All other parts are less than 1%. Note that fewer Newton iterations are required during the quadrature refinement steps. Thus the computational amount of solving the dual problem increases and the computational amount of solving the primal problem decreases. It is an interesting question for further research how the distribution of the computing time changes if the solver is also parallelized and if more sophisticated preconditioners, for instance, based on algebraic multigrid adapted to the special challenges of the finite cell method are used. However, such research is out of the scope of the article at hand.

6 Conclusion

In this article, we present a dual weighted residual (DWR) error estimator for the finite cell method (FCM). The DWR method allows for goal-oriented error control and incorporates the information of a user-defined quantity of interest into the solution of a dual problem that has to be solved alongside the primal problem.

In the FCM, the computational domain is replaced by a simpler enclosing domain on which the finite element space is constructed. The original, possibly complicated domain is approximated by a quadrature mesh. Thereby, a quadrature error is introduced. The presented method allows for splitting the DWR error contribution into an error term related to the discretization, an error term related to the quadrature, and involves several additional terms which cannot be computed numerically. These additional terms may be neglected if the quadrature is sufficiently accurate. From the numerical results one may conclude that the quadrature error term has to be several magnitudes smaller than the discretization error term to obtain accurate error estimates. However, according to the numerical results, this accuracy of the quadrature is not required to obtain optimal convergence rates. We present a refinement strategy that adapts the FCM mesh or the quadrature mesh to keep the quadrature error below the discretization error up to a user-defined multiplicative constant.

Its efficiency is underlined by several examples involving complex geometries and nonsmooth differential operators.

Funding: The first and the last author gratefully acknowledge support by the Deutsche Forschungsgemeinschaft in the Priority Programme 1748 'Reliable simulation techniques in solid mechanics. Development of non-standard discretization methods, mechanical and mathematical analysis' under the project 'High-order immersed-boundary methods in solid mechanics for structures generated by additive processes', Grant SCHR 1244/4-1.

References

[1] A. Abedian, J. Parvizian, A. Düster, and E. Rank, Finite cell method compared to *h*-version finite element method for elasto-plastic problems, *Appl. Math. Mech.* **35** (2014), No. 10, 1239–1248.
[2] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser, 2013.
[3] R. Becker, V. Heuveline, and R. Rannacher, An optimal control approach to adaptivity in computational fluid mechanics, *Int. J. Numer. Methods Fluids* **40** (2002), No. 1-2, 105–120.
[4] R. Becker, C. Johnson, and R. Rannacher, Adaptive error control for multigrid finite element, *Computing* **55** (1995), No. 4, 271–288.
[5] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numerica* **10** (2001), 1–102.
[6] H. Blum, A. Schröder, and F.-T. Suttmeier, *A Posteriori Estimates for FE-Solutions of Variational Inequalities*, pp. 669–680, Springer Milan, Milano, 2003.
[7] H. Blum and F.-T. Suttmeier, An adaptive finite element discretisation for a simplified Signorini problem, *Calcolo* **37** (2000), No. 2, 65–77.
[8] M. Braack and A. Ern, *Adaptive Computation of Reactive Flows with Local Mesh Refinement and Model Adaptation*, Numerical Mathematics and Advanced Applications, Springer, 2004, pp. 159–168.
[9] M. Braack and A. Ern, Coupling multimodelling with local mesh refinement for the numerical computation of laminar flames, *Combustion Theory Model.* **8** (2004), No. 4, 771–788.
[10] D. Braess, *Finite Elemente: Theorie, Schnelle Löser Und Anwendungen in Der Elastizitätstheorie*, Springer-Verlag, 2013.
[11] A. Byfut and A. Schröder, A fictitious domain method for the simulation of thermoelastic deformations in NC-milling processes, *Int. J. Numer. Methods Engrg.* **113** (2017), No. 2, 208–229.
[12] M. Dauge, A. Düster, and E. Rank, Theoretical and numerical investigation of the finite cell method, *J. Sci. Comp.* **65** (2015), No. 3, 1039–1064.
[13] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*, Springer, 2000.
[14] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, *SIAM J. Numer. Anal.* **33** (1996), No. 3, 1106–1124.
[15] A. Düster, J. Parvizian, Z. Yang, and E. Rank, The finite cell method for three-dimensional problems of solid mechanics, *Comput. Methods Appl. Mech. Engrg.* **197** (2008), No. 45, 3768–3782.
[16] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, Introduction to adaptive methods for differential equations, *Acta Numerica* **4** (1995), 105–158.
[17] R. Glowinski and Y. Kuznetsov, Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems, *Comput. Methods Appl. Mech. Engrg.* **196** (2007), No. 8, 1498–1506.
[18] R. Glowinski, T.-W. Pan, and J. Periaux, A fictitious domain method for Dirichlet problem and applications, *Comput. Methods Appl. Mech. Engrg.* **111** (1994), No. 3-4, 283–303.
[19] T. Grätsch and K.-J. Bathe, Goal-oriented error estimation in the analysis of fluid flows with structural interactions, *Comput. Methods Appl. Mech. Engrg.* **195** (2006), No. 41, 5673–5684.
[20] S. Hubrich, P. Di Stolfo, L. Kudela, S. Kollmannsberger, E. Rank, A. Schröder, and A. Düster, Numerical integration of discontinuous functions: moment fitting and smart octree, *Comp. Mech.* **60** (2017), No. 5, 863–881.
[21] M. Joulaian, S. Duczek, U. Gabbert, and A. Düster, Finite and spectral cell method for wave propagation in heterogeneous materials, *Comput. Mech.* **54** (2014), No. 3, 661–675.
[22] M. Joulaian, S. Hubrich, and A. Düster, Numerical integration of discontinuities on arbitrary domains based on moment fitting, *Comp. Mech.* **57** (2016), No. 6, 979–999.
[23] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank, Efficient and accurate numerical quadrature for immersed boundary methods, *Adv. Modeling Simul. Engrg. Sci.* **2** (2015), No. 1, 10.
[24] L. Kudela, N. Zander, S. Kollmannsberger, and E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3D, *Comput. Methods Appl. Mech. Engrg.* **306** (2016), 406–426.

[25] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **36** (1971), No. 1, 9–15.
[26] M. Paraschivoiu, J. Peraire, and A. Patera, A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations, *Comput. Methods Appl. Mech. Engrg.* **150** (1997), No. 1-4, 289–312.
[27] J. Parvizian, A. Düster, and E. Rank, Finite cell method, *Comput. Mech.* **41** (2007), No. 1, 121–133.
[28] J. Parvizian, A. Düster, and E. Rank, Topology optimization using the finite cell method, *Optim. Engrg.* **13** (2012), No. 1, 57–78.
[29] S. Prudhomme and J. T. Oden, On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors, *Comput. Methods Appl. Mech. Engrg.* **176** (1999), No. 1-4, 313–331.
[30] A. Rademacher, *Adaptive Finite Element Methods for Nonlinear Hyperbolic Problems of Second Order*, Ph.D. thesis, TU Dortmund, 2010.
[31] A. Rademacher and A. Schröder, Dual weighted residual error control for frictional contact problems., *Comput. Meth. Appl. Math.* **15** (2015), No. 3, 391–413.
[32] E. Rank, S. Kollmannsberger, C. Sorger, and A. Düster, Shell finite cell method: a high order fictitious domain approach for thin-walled structures, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), No. 45, 3200–3209.
[33] R. Rannacher and J. Vihharev, Adaptive finite element analysis of nonlinear problems: balancing of discretization and iteration errors, *J. Numer. Math.* **21** (2013), No. 1, 23–62.
[34] T. Richter, Goal-oriented error estimation for fluid–structure interaction problems, *Comput. Methods Appl. Mech. Engrg.* **223** (2012), 28–42.
[35] T. Richter and T. Wick, Variational localizations of the dual weighted residual estimator, *J. Comp. Appl. Math.* **279** (2015), 192–208.
[36] V. Saul'ev, On the solution of some boundary value problems on high performance computers by fictitious domain method, *Siberian Math. J.* **4** (1963), No. 4, 912–925.
[37] A. Schröder and A. Rademacher, Goal-oriented error control in adaptive mixed FEM for Signorini's problem, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), No. 1, 345–355.
[38] F.T. Suttmeier, On plasticity with hardening: an adaptive finite element discretisation, *Int. Math. Forum* **5** (2010), No. 52, 2591–2601.
[39] K. van der Zee, E. van Brummelen, I. Akkerman, and R. de Borst, Goal-oriented error estimation and adaptivity for fluid–structure interaction using exact linearized adjoints, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), No. 37, 2738–2757.
[40] T. Wick, Goal-oriented mesh adaptivity for fluid-structure interaction with application to heart-valve settings, *Arch. Mech. Engrg.* **59** (2012), No. 1, 73–99.
[41] Z. Yang, S. Kollmannsberger, A. Düster, M. Ruess, E. Garcia, R. Burgkart, and E. Rank, Non-standard bone simulation: interactive numerical analysis by computational steering, *Comput. Visualization Sci.* **14** (2011), No. 5, 207–216.
[42] Z. Yang, M. Ruess, S. Kollmannsberger, A. Düster, and E. Rank, An efficient integration technique for the voxel-based finite cell method, *Int. J. Numer. Methods Engrg.* **91** (2012), No. 5, 457–471.
[43] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, J. Parvizian, M. Ruess, D. Schillinger, and E. Rank, FCMLab: A finite cell research toolbox for MATLAB, *Adv. Engrg. Software* **74** (2014), 49–63.