Research Article

Bernhard Endtmayer and Thomas Wick*

A Partition-of-Unity Dual-Weighted Residual Approach for Multi-Objective Goal Functional Error Estimation Applied to Elliptic Problems

DOI: 10.1515/cmam-2017-0001
Received October 5, 2016; revised February 3, 2017; accepted February 24, 2017

Abstract: In this work, we design a posteriori error estimation and mesh adaptivity for multiple goal functionals. Our method is based on a dual-weighted residual approach in which localization is achieved in a variational form using a partition-of-unity. The key advantage is that the method is simple to implement and backward integration by parts is not required. For treating multiple goal functionals we employ the adjoint to the adjoint problem (i.e., a discrete error problem) and suggest an alternative way for its computation. Our algorithmic developments are substantiated for elliptic problems in terms of four different numerical tests that cover various types of challenges, such as singularities, different boundary conditions, and diverse goal functionals. Moreover, several computations with higher-order finite elements are performed.

Keywords: Finite Element Method, Mesh Adaptivity, Dual-Weighted Residual, Partition-of-Unity, Multi-Objective Goal Functionals, Adjoint to the Adjoint Problem

MSC 2010: 65N30, 65M60, 49M15, 35Q74

1 Introduction

In many physical applications the target is to compute a quantity of interest up to a certain accuracy rather than the entire solution. Moreover, fluid flow (Navier–Stokes) and aerodynamics flow simulations as well as multiphysics problems such as fluid-structure interaction, fracture problems, poroelastic problems, Maxwell equations, and magnetohydrodynamics are of importance. Here, several physical phenomena interact and consequently the accurate evaluation of more than one goal functional might be of interest. However, before we can address such nonlinear coupled PDEs, we need a reliable framework that is tested and validated for a single PDE, different boundary conditions, and different types of goal functionals.

In this study, accurate functional evaluations are based on adaptive mesh refinement. The method of choice is based on dual-weighted residual (DWR) a posteriori error estimation [9, 10], which is based on the ideas presented in [18] and motivated by duality principles well known in optimization [3].

In addition to the primal problem (the given PDE), a corresponding adjoint problem needs to be solved, which provides (local) sensitivity measures with respect to an error goal functional. In the early stages, further extensions of the DWR method have been accomplished in [1, 2, 8, 12, 21, 27–29]. Most of these studies have in common that the strong formulation [10] for error localization is needed. A weak form with patched meshes has been proposed in [12]. One advantage of a weak localization of the DWR technique lies in its application to multiphysics problems because the classical localization works with strong (second-order) operators that are costly to evaluate and additionally (often several) face integration terms need to be evaluated.

Bernhard Endtmayer: Institute of Computational Mathematics, JKU Linz, Altenberger Str. 69, 4040 Linz, Austria, e-mail: bernhard.endtmayer@dk-compmath.jku.at
***Corresponding author: Thomas Wick:** Centre de Mathématiques Appliquées, École Polytechnique, Université Paris-Saclay, 91128 Palaiseau, France, e-mail: thomas.wick@polytechnique.edu

Recently, in [31], another weakly-based localization technique has been suggested. It is straightforward to employ and easy to implement. Here (similar to [12]), partial integration back to the strong operator is not necessary. Therefore, no face terms need to be evaluated. Rather, solution information about neighboring cells (which is very important in particular for low-order finite elements [15]) is gathered by employing a partition-of-unity (PU) leading to a nodal-based error indicator representation. To realize the PU, a lowest-order finite element is sufficient. We notice that a PU for strongly localized DWR error estimation has been previously suggested in [26].

On the other hand, Hartmann and Houston [24] and Hartmann [23] considered multiple target functionals. However, literature on this topic is rare and there exist a few other studies [19, 25, 33] from which [33] is quite recent. One crucial difficulty is the computational cost, not only for single goal functional evaluations due to the fact that a (linear) adjoint problem needs to be solved. But this adjoint solution must deliver 'more' information than the primal problem and for this reason it is usually more expensive than a linear primal problem. Of course, for nonlinear problems, solving the adjoint problem does only correspond to one additional Newton solve. Thus, the cost of the adjoint problem becomes much less significant.

For multiple goal functionals, say N, a naive approach would mean to solve N adjoint problems, which makes the method not attractive at all. Therefore, Hartmann and Houston [23, 24] considered an adjoint to the adjoint problem (which is equivalent to saying a discrete error problem), which only requires two additional solutions and therefore significantly reduces the computational cost.

Based on this approach we suggest two modifications in this work. First, we apply the PU-DWR method to multiple target functionals. Second, we propose an idea how to decouple the two additional problems (associated with the adjoint to the adjoint problem) such that they can be performed in principle in parallel. These algorithmic developments are complemented with a series of numerical examples using the finite element method in which different challenges are addressed. In particular, several higher-order finite element computations are provided, which have not yet been shown for such problems in existing literature.

This paper is organized as follows: In Section 2, the model problem is presented as well as the basics of DWR mesh adaptivity. Then, in Section 3, the approach for treating multiple goal functionals is presented. Next in Section 4, various numerical examples are presented that cover different aspects of smooth solutions, singularities, L-shaped and slit domains. For residual-based error estimates and adaptivity in form of graded-mesh refinement for L-shaped domains (and domains with other corners), we refer to [5]. We also consider different boundary conditions of homogeneous and non-homogeneous Dirichlet type and homogeneous Neumann conditions. Moreover, different types of goal functionals are taken into account such as point values, line integration, and domain integrals. Finally, in Section 5, we recapitulate our findings and provide a few ideas for future work.

2 The DWR Method for Goal Functional Evaluations

In this section, we first provide the problem statement and spatial discretization. Then, we briefly review the DWR method for single goal functionals and recapitulate the classical strategy and the partition-of-unity DWR approach for goal functional evaluations.

2.1 The Model Problem

By $\Omega \subset R^d$ with $d = 2$, we denote a domain with polygonal or polyhedral boundary. The boundary $\partial\Omega$ is split into two non-overlapping parts $\Gamma_D \cap \Gamma_N = \emptyset$ and $\partial\Omega := \Gamma_D \cup \Gamma_N$. On Ω, we denote by $(\cdot,\cdot)$ the L^2-inner product and by $\|\cdot\|$ the corresponding L^2-norm. By $\langle\cdot,\cdot\rangle$ we denote as usual the pairing between H^{-1} and H_0^1 functions [20]. By $H^{r+1}(\Omega)$ we denote the space of Lebesgue functions with square integrable weak derivatives up to degree $r+1$. In particular, by

$$V := \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D\}$$

we denote the space of $H^1(\Omega)$ functions with trace zero on the Dirichlet boundary Γ_D.

We consider a weak form of the diffusion problem and assume that $\tilde{f} \in L^2(\Omega)$, the domain is sufficiently regular so that the trace theorem (see, e.g., [37]) holds true, i.e., $g_N \in H^{-\frac{1}{2}}(\Gamma_N)$, and finally $g_D \in H^{\frac{1}{2}}(\Omega)$. The weak form of the problem reads: Find $u \in \{g_D + V\}$ such that

$$(\alpha \nabla u, \nabla \varphi) = \langle f, \varphi \rangle \quad \text{for all } \varphi \in V, \tag{2.1}$$

where

$$\langle f, \varphi \rangle := \int_\Omega \tilde{f}(x)\varphi(x)\, dx + \int_{\Gamma_N} g_N(x)\varphi(x)\, ds_x,$$

and the diffusion coefficient $\alpha := \alpha(x) \in L^\infty(\Omega)$. In this setting $\int_{\Gamma_N} g_N(x)\varphi(x)\, ds_x$ has to be understood as duality product as for instance in [22]. If $g_N \in L^2(\Gamma_N)$ then it coincides with the integral.

Remark 2.1. The classical (strong) formulation of (2.1) reads as follows:

$$-\nabla \cdot (\alpha \nabla u) = \tilde{f} \quad \text{in } \Omega, \tag{2.2a}$$

$$u = g_D \quad \text{on } \Gamma_D, \tag{2.2b}$$

$$\alpha \nabla u \cdot n = g_N \quad \text{on } \Gamma_N, \tag{2.2c}$$

where n is the normal vector.

The non-homogeneous Dirichlet boundary condition (2.2b) is imposed on Γ_D in the trace sense of a H^1 function. The unknown solution $u \in \{g_D + V\}$ is approximated in a finite-dimensional function space V_h, which is discussed in Section 2.2.

2.2 Spatial Discretization

All formulations in this work are spatially discretized with a Galerkin finite element scheme, introducing H^1 conforming discrete spaces $V_h \subset V$ consisting of functions Q_r^c of degree r. Here Q_r^c denotes the finite element space

$$Q_r^c := \{v_h \in [C(\Omega)]^d \mid v_h|_K \in Q_r(K) \text{ for all } K \in \mathbb{T}_h,\ v_h|_{\Gamma_D} = 0\},$$

where the mesh is denoted by $\mathbb{T}_h$ consisting of quadrilateral elements K and the corresponding mesh size parameter is labeled by h. Moreover,

$$Q_r(K) := \Big\{ v \in C^\infty(K) \mid v(x) = \sum_{\beta \in B_r} c_\beta x^\beta \Big\},$$

with $c_\beta \in \mathbb{R}$ and $x^\beta := \prod_{i=1}^d x_i^{\beta_i}$ and the set

$$B_r := \{\beta \in (\mathbb{N}_0)^d \mid \beta_i \le r \text{ for all } i \in \{1, \dots, d\}\}.$$

Specifically, we use isoparametric tensor-product finite elements [16]. To this end, the definition of the discrete space reads

$$V_h := Q_r^c.$$

Since we only work with continuous elements in this work, we set $Q_r := Q_r^c$. Next, we notice that mesh adaptivity yields locally refined cells, which leads to hanging nodes [14] in the mesh. For convenience of the reader, we often denote explicitly the degree r for the spaces $V_h^{(r)}$ in order to avoid misunderstanding.

With these preparations, we state the discretized version of our problem: Find $u_h \in \{g_D + V_h\}$ such that

$$a(u_h, \varphi_h) = \langle f, \varphi_h \rangle \quad \text{for all } \varphi_h \in V_h,$$

where $a(u_h, \varphi_h) := (\alpha \nabla u_h, \nabla \varphi_h)$ and the duality product is defined as before.

2.3 A Brief Review for Single Goal Functionals

In the following we describe the DWR method for linear goal functionals and linear primal problems. The aim is to compute a certain quantity of interest $J(u)$ with a desired accuracy at low computational cost. Possible examples are mean values, line integration or point values:

$$J(u) = \int_\Omega u \, dx, \quad J(u) = \int_\Gamma \partial_n u \, ds, \quad J(u) = u(x_0, y_0).$$

Remark 2.2. The second and third goal functionals are a priori not well defined. In case of the second functional we know the $\nabla u \in [L^2(\Omega)]^d$. Adopting the trace theorem as in [17], we can deduce that the trace in normal direction belongs to $H^{-\frac{1}{2}}(\partial\Omega)$. This however leads to the problem that the second functional is not always well defined. Concerning the third functional, it is well known (see, e.g., [13]) that for H^1 functions in dimension $d > 1$ the solution u is not any more continuous and the last evaluation is not well defined. If the domain and boundaries are sufficiently regular in 2D, the resulting solution is however H^2 regular and thanks to embedding theorems also continuous.

The above goal functionals are however computed with a numerical method leading to a discrete version $J(u_h)$. Thus the key goal is to control the error $J(u) - J(u_h)$ in terms of local residuals, which are computable on each mesh cell K. To address this goal, we assign an associated adjoint problem as it is standard in optimization (for example via defining the Lagrangian); see, e.g., [3]. Such an adjoint solution yields local sensitivity information with respect to the given error functional. For a posteriori error estimation with dual-weighted residuals these concepts have been explained in detail in [10]. To this end, we seek the adjoint variable $z \in V$:

$$a(\psi, z) = J(\psi) \quad \text{for all } \psi \in V. \tag{2.3}$$

Specifically, the adjoint bilinear form is given by

$$a(\psi, z) = (\alpha \nabla \psi, \nabla z).$$

The boundary conditions on Γ_D are built into V and are of homogeneous Dirichlet type. For symmetric problems (as we deal with in this work), the adjoint bilinear form $a(\cdot, \cdot)$ is the same as the original one, but differs for non-symmetric problems like transport for example.

Existence and uniqueness of this adjoint solution follows by standard arguments. The regularity of $z \in V$ depends on the regularity of the functional J. For $J \in H^{-1}(\Omega)$ it holds $z \in H^1(\Omega)$. Given a more regular functional like the L^2-error $J(\varphi) = \|e_h\|^{-1}(e_h, \varphi)$ (where $e_h := u - u_h$) with $J \in L^2(\Omega)^*$ (denoting the dual space), it holds $z \in H^2(\Omega)$ on suitable domains (convex polygonal or smooth boundary with C^2-parameterization).

Inserting as special test function $\psi := u - u_h$ into (2.3) yields

$$a(u - u_h, z) = J(u - u_h),$$

and therefore we have now a representation for the error in the goal functional.

Next, we use Galerkin orthogonality $a(u - u_h, \psi_h) = 0$ for all $\psi_h \in V_h$, and we obtain

$$a(u - u_h, z) = a(u - u_h, z) - \underbrace{a(u - u_h, \psi_h)}_{=0} = a(u - u_h, z - \psi_h) = J(u - u_h). \tag{2.4}$$

The previous step allows us to choose ψ_h in such a way that $z - \psi_h$ can be bounded using interpolation estimates. Indeed, since ψ_h is an arbitrary discrete test function, we can for example use a projection $\psi_h := i_h z$ in (2.4):

$$a(u - u_h, z - i_h z) = J(u - u_h). \tag{2.5}$$

Since z is an unknown itself, we cannot yet simply evaluate the error estimator because z is only known analytically in very special cases.

2.4 Approximation of the Adjoint Solution

In order to obtain a computable error representation, z is approximated through a discrete function $z_h \in V_h$, that is obtained from solving

$$a(\psi_h, z_h) = J(\psi_h) \quad \text{for all } \psi_h \in V_h.$$

Then,

$$a(u - u_h, z_h - i_h z_h) \approx J(u - u_h). \tag{2.6}$$

The difficulty is that if we compute the adjoint problem with the same polynomial degree as the primal problem, then $z_h - i_h z_h \equiv 0$, and thus the whole error identity defined in (2.6) would vanish, i.e., $J(u - u_h) \equiv 0$. To overcome this point, either a global-higher order approximation (using a higher order finite element), a solution on a finer mesh, or local higher-order approximations using a patch-wise higher-order interpolation can be adopted [8, 10]. Clearly, the last possibility is the cheapest. In this work, however, for simplicity, we simply use a global-higher order finite element of degree $r + 1$ (in case that the primal problem has been computed with degree r).

We finally end up with the (primal) error estimator:

$$a(u - u_h, z_h^{(r+1)} - i_h z_h^{(r+1)}) \approx J(u - u_h).$$

Thus, the error in the functional $J(u - u_h)$ can be expressed in terms of a residual, that is weighted by (the local) adjoint sensitivity information $z^{(r+1)} - i_h z^{(r+1)}$.

As quality measure we use the effectivity index I_{eff}:

$$I_{\text{eff}} := I_{\text{eff}}(u_h, z_h) = \frac{\eta}{J(u - u_h)} = \frac{a(u - u_h, z_h^{(r+1)} - i_h z_h^{(r+1)})}{J(u - u_h)}. \tag{2.7}$$

Problems with good I_{eff} have the property $I_{\text{eff}} \to 1$ for $h \to 0$. For the localization of the error on each cell or each degree of freedom, we forward the reader to the next two sections.

2.5 The Classical Way of Error Localization

We briefly recapitulate the classical way and then explain a variational technique that uses a partition-of-unity (PU). Both techniques have in common that they start from

$$J(u - u_h) = a(u - u_h, z - i_h z).$$

In the classical way, the error identity (2.5) is treated with integration by parts on every mesh element $K \in \mathcal{T}_h$, which yields

$$J(u - u_h) = \sum_{K \in \mathcal{T}_h} \langle f + \nabla \cdot (\alpha \nabla u_h), z - i_h z \rangle_K + \int_{\partial K} \alpha \partial_n u_h \cdot (z - i_h z)\, ds. \tag{2.8}$$

Remark 2.3. In praxis, this primal error estimator needs to be evaluated in the dual space. Here, we proceed as follows:

- Prolongate the primal solution u_h into the dual space.
- Next, we compute the interpolation $i_h z_h^{(r+1)} \in Q_r$ with respect to the primal space.
- Then, we compute $z_h^{(r+1)} - i_h z_h^{(r+1)}$ (here, $i_h z_h^{(r+1)}$ is prolongated to Q_{r+1} in order to compute the difference).
- Evaluate the duality product $\langle \cdot, \cdot \rangle$ and face terms.

From (2.8), we can set-up the error estimator. Following the usual procedure for residual based error estimators [34], we combine each two boundary integrals over element edges to a normal jump and proceed with Cauchy–Schwarz to get

$$|J(u) - J(u_h)| \le \eta := \sum_{K \in \mathcal{T}_h} \rho_K \omega_K,$$

with

$$\rho_K := \|\tilde{f} + \nabla \cdot (\alpha \nabla u_h)\|_K + \frac{1}{2} h_K^{-\frac{1}{2}} \|[\alpha \partial_n u_h]\|_{\partial K}, \quad \omega_K := \|z - i_h z\|_K + h_k^{\frac{1}{2}} \|z - i_h z\|_{\partial K},$$

where by $[\alpha \partial_n u_h]$ we denote the jump of the u_h derivative in normal direction. The residual part ρ_K only contains the discrete solution u_h and the problem data. On the outer Dirichlet boundary Γ_D, we set $[\alpha \partial_n u_h] = 0$ and on the Neumann part we evaluate $\alpha \partial_n u_h = g_N$. Of course, we implicitly assume here that $g_N \in L^2(\Gamma_N)$ such that these terms are well-defined.

2.6 A Variational Error Estimator with PU Localization

In this section, we recapitulate an alternative way and use a localization approach based on the variational formulation [31]. This idea combines the simplicity of the approach proposed in [12] (as it is given in terms of variational residuals) in terms of a very simple structure, which makes it particularly interesting for coupled PDE systems (see further comments below). Localization is based on introducing a partition of unity (PU) $\{\psi_1, \ldots, \psi_M\} =: V_{\text{PU}}$ with $\dim(V_{\text{PU}}) = M$ and the property $\sum \psi_i \equiv 1$. We then insert the PU into the global error identity (2.5):

Proposition 2.4. *For the finite element approximation of* (2.1)*, we have the a posteriori error estimate*

$$|J(u) - J(u_h)| \leq \eta := \sum_{i=1}^{M} |\eta_i|, \tag{2.9}$$

where

$$\eta_i = \{-a(u - u_h, (z - i_h z)\psi_i)\},$$

and more specifically for our diffusion problem:

$$\eta_i = \{\langle f, (z - i_h z)\psi_i\rangle - (\alpha \nabla u_h, \nabla(z - i_h z)\psi_i)\}. \tag{2.10}$$

To set-up the PU, one can simply work with lowest-order finite elements, i.e., a bilinear function on quadrilaterals in two dimensions. Thus, as finite element space we can choose $V_{\text{PU}} := V_h^{(1)}$.

The previous error indicators η_i are node-wise contributions of the error. Mesh adaptivity can be carried out in two ways:

- In a node-wise fashion: if a node i is picked for refinement, all elements touching this node will be refined.
- Alternatively, one could also first assemble element-wise for each $K \in \mathcal{T}_h$ indicators by summing up all indicators belonging to nodes of this element and then carry out adaptivity in the usual element-wise way.

On adaptive meshes with hanging nodes, the evaluation of the PU indicator is straightforward: First, the PU is assembled in (2.10) employing the basis functions $\psi_i \in V_{\text{PU}}$ for $i = 1, \ldots, M$. In a second step, the contributions belonging to hanging nodes are condensed in the usual way by distribution to the neighboring indicators. This localization technique can be readily applied in two and three dimensions [31] and has also been extended to general polygonal meshes [35]. As it has been already demonstrated for the similar approach (for a single goal functional) proposed in [12], for instance for variational inequalities in solid mechanics [32] or fluid-structure interaction [30], a major advantage of a weak localization is the easy application to nonlinear coupled PDE systems, where the evaluation of strong residuals can be cumbersome. We refer the reader again to fluid-structure interaction [30, 38] as example. On the other hand, a first application of PU-DWR to nonlinear-coupled PDE problems has been recently undertaken in [36]. Finally, we want to comment that it is well known that for dual-weighted residual goal-oriented adaptivity we have no theoretical justification for convergence and optimality of the adaptive algorithm, but only excellent practical observations.

3 PU-DWR for Multiple Goal Functionals

In the previous section we recapitulated the DWR method for computing a single goal functional. Now we assume that we are given N linear functionals with $N \in \mathbb{N}$. Let $\mathbb{J}$ be defined as $\mathbb{J} := \{J_0, \dots, J_{N-1}\}$. We can use (2.9) for each $J_i \in \mathbb{J}$ where $i \in \{0, \dots, N-1\}$ to compute the node-wise contributions of the error. But to do so we have to solve N adjoint problems. Therefore we seek a method to avoid these computations. We follow the idea from [23, 24] of combining the functionals in $\mathbb{J}$. We create a linear combination of the goal functionals to one functional $\tilde{J}_c$ resulting in

$$\tilde{J}_c(\psi) := \sum_{i=0}^{N-1} w_i J_i(\psi) \quad \text{for all } \psi \in V,$$

where $w_i \in \mathbb{R}$. We call $\tilde{J}_c$ the combined functional. Now we have to find out how to choose the weights w_i. One crucial aspect is the sign of w_i because it may lead to error canceling. Furthermore, we are interested in having similar relative errors in our functional evaluations. One idea (as in [23, 24]) is to choose w_i as

$$w_i := \frac{\operatorname{sign}(J_i(u) - J_i(u_h))\omega_i}{|J(u_h)|}, \tag{3.1}$$

where ω_i describe some self-chosen, but positive weights. This choice leads to no error canceling and also the relative errors are similar (if the weights ω_i are almost equal). But unfortunately we do not know $J_i(u)$. Hence, we have to find a way to get $\operatorname{sign}(J_i(u) - J_i(u_h))$. To do so, we consider the adjoint to the adjoint problem (which is akin to saying a discrete error problem) [23, 24]: Find the error function e such that

$$a(e, \psi) = \langle R_{u_h}, \psi \rangle \quad \text{for all } \psi \in V, \tag{3.2}$$

where $\langle R_{u_h}, \psi \rangle := \langle f, \psi \rangle - a(u_h, \psi)$.

By solving this problem, we obtain e where $e = u - u_h$. Therefore we can compute $J_i(u) - J_i(u_h)$. The adjoint to the adjoint problem provides information with respect to the error in the goal functionals $J_i(u)$, but it does not yield local error information, which are required for mesh refinement. Thus the solution is to solve both the adjoint to the adjoint problem and another adjoint problem leading to two additional problems. In summary, using this approach for multiple goal functionals, three problems need to be solved: primal, adjoint, adjoint to the adjoint.

For treating the adjoint to the adjoint problem we again have to solve a PDE discretized by finite elements. For this problem we have to use a discrete subspace $V_h^{(r+1)} \subset V$ which fulfills $V_h \subsetneq V_h^{(r+1)}$ because otherwise $\langle R_{u_h}, \psi_h \rangle = 0$ for all $\psi_h \in V_h$, which must be avoided. Moreover, we have to solve the primal problem to compute u_h and then compute e as solution of the adjoint to the adjoint problem, so we have to solve two systems sequentially.

The dependence of the adjoint and adjoint to the adjoint problem slightly limits the possibility to further reduce the computational cost. Therefore, we suggest the following alternative in case we approximate the solution in our discrete subspace $V_h^{(r+1)}$, $r \geq 1$:

Proposition 3.1. *Let $a : V \times V \to \mathbb{R}$ be a bilinear form, $f \in V^*$, where V^* is the dual space of V, fulfilling the assumptions of Lax–Milgram (e.g., [20, 22]). Let $u_h, u_h^{(2)}, e_h^{(2)}$ be the solutions of the following problems, respectively: Find $u_h \in V_h$ and $e_h^{(2)}, u_h^{(2)} \in V_h^{(2)}$ such that*

$$a(u_h, \psi_h) = \langle f, \psi_h \rangle \qquad \text{for all } \psi_h \in V_h, \tag{3.3}$$

$$a(u_h^{(2)}, \psi_h^{(2)}) = \langle f, \psi_h^{(2)} \rangle \qquad \text{for all } \psi_h^{(2)} \in V_h^{(2)}, \tag{3.4}$$

$$a(e_h^{(2)}, \psi_h^{(2)}) = \langle R_{u_h}, \psi_h^{(2)} \rangle \quad \text{for all } \psi_h^{(2)} \in V_h^{(2)}, \tag{3.5}$$

where $V_h \subset V$, $V_h^{(2)} \subset V$ and $\langle R_{u_h}, \psi_h^{(2)} \rangle$ are defined as in (3.2). Then there exists a projection $P_{[V_h^{(2)}]} : V \to V_h^{(2)}$ such that

$$e_h^{(2)} = u_h^{(2)} - P_{[V_h^{(2)}]} u_h. \tag{3.6}$$

Specifically, if $V_h \subseteq V_h^{(2)}$, it holds

$$e_h^{(2)} = u_h^{(2)} - u_h. \tag{3.7}$$

Proof. Let u_h be the solution of (3.3). Then there exists a unique $f_{u_h} \in V^*$ such that

$$a(u_h, \psi) = \langle f_{u_h}, \psi \rangle \quad \text{for all } \psi \in V. \tag{3.8}$$

If we want to approximate the solution of (3.8) on the finite element space $V_h^{(2)}$, we obtain the approximation u_{u_h} of u_h which is given by the unique solution of the problem: Find $u_{u_h} \in V_h^{(2)}$ such that

$$a(u_{u_h}, \psi_h^{(2)}) = \langle f_{u_h}, \psi_h^{(2)} \rangle \quad \text{for all } \psi_h^{(2)} \in V_h^{(2)}.$$

It can be shown that the mapping $u_h \mapsto u_{u_h}$ is a projection, which is denoted by $P_{[V_h^{(2)}]}$. For this projection, it holds

$$a(u_h - \underbrace{P_{[V_h^{(2)}]} u_h}_{u_{u_h}}, \psi_h^{(2)}) = \langle f_{u_h}, \psi_h^{(2)} \rangle - \langle f_{u_h}, \psi_h^{(2)} \rangle = 0 \quad \text{for all } \psi_h^{(2)} \in V_h^{(2)}.$$

A simple calculation shows

$$\begin{aligned} a(e_h^{(2)}, \psi_h^{(2)}) &= \langle R_{u_h}, \psi_h^{(2)} \rangle \\ &= \langle f, \psi_h^{(2)} \rangle - a(u_h, \psi_h^{(2)}) \\ &= a(u_h^{(2)}, \psi_h^{(2)}) - a(P_{[V_h^{(2)}]} u_h, \psi_h^{(2)}) \\ &= a(u_h^{(2)} - P_{[V_h^{(2)}]} u_h, \psi_h^{(2)}) \quad \text{for all } \psi_h^{(2)} \in V_h^{(2)}. \end{aligned}$$

The Lax–Milgram lemma yields a unique solution in the space $V_h^{(2)}$ and we can conclude that

$$e_h^{(2)} = u_h^{(2)} - P_{[V_h^{(2)}]} u_h,$$

hence

$$a(e_h^{(2)}, \psi_h^{(2)}) = \langle R_{u_h}, \psi_h^{(2)} \rangle = a(u_h^{(2)} - P_{[V_h^{(2)}]} u_h, \psi_h^{(2)}) \quad \text{for all } \psi_h^{(2)} \in V_h^{(2)}.$$

And this shows the first statement (3.6). If $V_h \subseteq V_h^{(2)}$ holds, then $P_{[V_h^{(2)}]} u_h = u_h$ because $u_h \in V_h^{(2)}$ and henceforth (3.7) has been shown. □

Remark 3.2. The assumptions of the Lax–Milgram lemma can be relaxed by any condition which guarantees only that $a(u, \psi) = \langle f, \psi \rangle$ for all $\psi \in V$, (3.3), (3.4) and (3.5) have unique solutions for all $f \in V^*$.

Remark 3.3. Furthermore, we notice that our previous theory does hold not only for $V_h^{(1)} \subset V_h^{(r+1)}$ but for general spaces which are not necessarily subspaces of $V_h^{(r+1)}$.

Corollary 3.4. *From Proposition 3.1 we obtain that if we work on the spaces V_h and $V_h^{(r+1)}$, the error can be simply computed as*

$$e_h^{(r+1)} = u_h^{(r+1)} - u_h.$$

In particular, the two subproblems for obtaining $u_h^{(r+1)}$ and u_h can be computed in parallel without communication using the spaces V_h and $V_h^{(r+1)}$.

Furthermore to avoid problems with prolongation operators in programming, we can compute immediately $J_i(u_h)$ and just communicate this value. With the help of Proposition 3.1, the combined functional $\tilde{J_c}$ is approximated by J_c with

$$J_c(\psi) := \sum_{i=0}^{N-1} J_i(\psi) \frac{\operatorname{sign}(J_i(u_h^{(r+1)}) - J_i(u_h))\omega_i}{|J_i(u_h)|} \quad \text{for all } \psi \in V, \tag{3.9}$$

for some self-chosen but positive weights ω_i. Now we can use the PU approach for the functional J_c in a similar way as discussed in Section 2.6, resulting in the following proposition.

Proposition 3.5. *For the finite element approximation of* (2.1)*, and considering N goal functionals $J_i(\cdot)$ we have the a posteriori error estimate*

$$|J_c(u) - J_c(u_h)| \leq \eta := \sum_{i=1}^{M} |\eta_i|,$$

where

$$\eta_i = \{\langle f, (z - i_h z)\psi_i\rangle - (\alpha \nabla u_h, \nabla(z - i_h z)\psi_i)\}. \tag{3.10}$$

Specifically, the adjoint problem is given by: Find $z \in V$ such that

$$a(\varphi, z) = J_c(\varphi) \quad \textit{for all } \varphi \in V,$$

for which J_c has been constructed by (3.9).

Remark 3.6. Since in praxis we are in general not able to compute the exact adjoint solution z (see also Section 2.4), we approximate z by z_h solving the following problem: Find $z_h \in V_h^{(r+1)}$ such that

$$a(\varphi_h, z_h) = J_c(\varphi_h) \quad \text{for all } \varphi_h \in V_h^{(r+1)},$$

for which J_c has been constructed by (3.9).

Remark 3.7. An advantage of this approach is that we have to solve (as in [23, 24]) only two linear systems instead of N, and furthermore we do not lose the possibility of parallelization.

Remark 3.8. If we use the same finite element space for the second primal problem and the adjoint problem, then we just have to assemble one matrix instead of the two system matrices A_{primal}, A_{adjoint} since it holds $A_{\text{adjoint}} = A_{\text{primal}}^T$.

3.1 The Adaptive Algorithm

Let error tolerances TOL_i be given for each $J_i \in \mathbb{J}$ and $i \in \{0, \dots, N-1\}$ where N is the number of functionals of interest. Mesh adaptation is realized by extracting local error indicators η_i from the a posteriori error estimate in Proposition 3.5 in which the adjoint solution has been approximated as explained in Remark 3.6. To this end, we can adapt the mesh using the following strategy:

(1) *Solve two primal problems:* Compute the primal solutions u_h and $u_h^{(r+1)}$ for two finite element spaces, respectively. This can be done completely in parallel.
(2) *Construct combined functional:* Construct J_c as in (3.9).
(3) *Solve the adjoint problem:* Compute the adjoint solution $z_h^{(r+1)}$ by solving the adjoint problem $a(\varphi_h, z_h) = J_c(\varphi_h)$ on a larger FE-space than for u_h.
(4) *Estimate:*
   - Determine the indicator η_i at each node i by (3.10).
   - Compute the sum of all indicators $\eta := \sum_i \eta_i$.
   - Check if the stopping criterion $\eta < \text{TOL}_c$ is satisfied, where $\text{TOL}_c := \inf_{i\in\{0,\dots,N-1\}}\{\omega_i \text{TOL}_i/|J_i(u_h)|\}$. If this criterion is satisfied, stop the computation since $J_c(u_h)$ has been computed with a desired accuracy. Otherwise, proceed to the following step.
(5) *Mark* all cells K_i that touch nodes that have values η_i above the average $\frac{\alpha\eta}{N}$ (where N denotes the total number of cells of the mesh $\mathbb{T}_h$ and $\alpha \approx 1$).
(6) *Refine* the mesh.
(7) *Go back to 1.*

Remark 3.9. The reason for the special choice of TOL_c is to ensure that $|J_i(u) - J_i(u_h)| < \text{TOL}_i$ holds for all functionals J_i, where we tacitly assume that the exact adjoint solution would be known.

Remark 3.10. Despite that we formulated our algorithm for r and $r+1$, we notice that we could also have worked with the same polynomial degree but locally using finer meshes to obtain the second space. This is very similar to the various options that we have to approximate the adjoint problem itself as described in Section 2.4.

4 Numerical Tests

In this section, we consider several numerical tests to substantiate our algorithmic developments. More specifically, the algorithm was tested for the Poisson equation. Furthermore, we used always $\omega_i = 1$ for the weights in J_c (as in (3.9)) and for simplicity the diffusion coefficient has been chosen to be $\alpha = 1$. In more detail, we analyze our algorithm with the help of the following examples:

- *Example 1a, b:* Smooth and discontinuous right-hand sides on an L-shaped domain.
- *Example 2a, b:* Eigenvalue of the Laplacian as right-hand side on the unit square. In Example 2a, we particularly focus on whether global refinement can deliver better results than local refinement for one functional. In Example 2b, we use again the L-shaped domain and we are especially interested if the sign, i.e., $\text{sign}(J_i(u_h^{(r+1)}) - J_i(u_h))$, needs to be computed in every step.
- *Example 3:* Singular right-hand side f for which $f \notin L^2(\Omega)$ but $f \in L^{2-\varepsilon}(\Omega)$ for all $\varepsilon > 0$ on an L-shaped domain. Here, we also perform comparisons with higher-order finite elements.
- *Example 4a, b:* Zero right-hand side on a slit domain with non-homogeneous Dirichlet and homogeneous Neumann conditions. In Example 4b, we focus again on higher-order computations.

For all examples in this section we always consider the error in J as the relative error $|J(u_h) - J(u)|/|J(u_h)|$. Furthermore we consider one refinement step for global refinement as refining every single element and one refinement step for DWR as one step of the algorithm in Section 3.1. Our programming code is based on the open-source finite element package deal.II (see [6, 7]).

4.1 Example 1

4.1.1 Example 1a

Configuration. In the first example we considered the Poisson equation on an L-shaped domain

$$\Omega = (-1, 1) \times (-1, 1) \setminus (-1, 0) \times (-1, 0)$$

with smooth right-hand side:

$$\begin{aligned} -\Delta u(x, y) &= f(x, y) \quad \text{for all } (x, y) \in \Omega, \\ u(x, y) &= 0 \qquad\quad \text{for all } (x, y) \in \partial\Omega, \end{aligned}$$

where

$$f(x, y) = x\big(8 - 2x^2 - 6y^2 + e^{3y}(1 - 3y(4 + y) + x^2(-7 + 3y(4 + 3y)))\big).$$

Here the exact solution u is given by

$$u(x, y) = x(y^2 - 1)(x^2 - 1)(e^{3y} - 1).$$

Goal Functionals of Interest. We consider the following three goal functionals:

$$J_0(u) := u(0.5, 0.5), \quad J_1(u) := \int_{\Omega_1} u(x, y)\, d(x, y), \quad J_2(u) := \int_{\Gamma_1} \nabla u(x, y).n\, d(x, y),$$

where $\Omega_1 = (-0.5, 0) \times (0.5, 1)$ and $\Gamma_1 = \{1\} \times (0, 1)$.

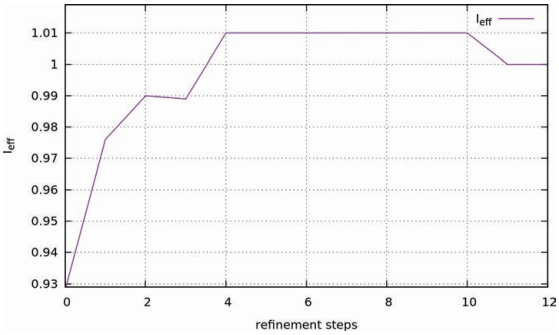

Figure 1. Example 1a: I_{eff} for J_c vs. refinement steps.

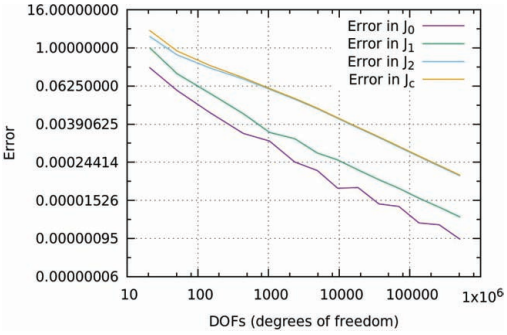

Figure 2. Example 1a: Comparison of relative errors.

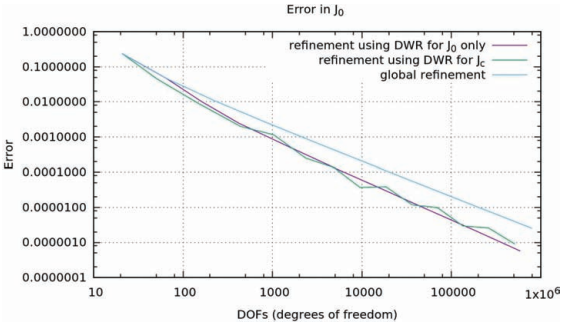

Figure 3. Example 1a: Comparison of relative errors for different refinements for J_0.

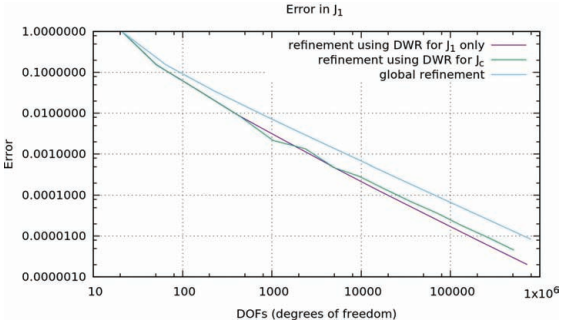

Figure 4. Example 1a: Comparison of relative errors for different refinements for J_1.

Details on Discretization. To solve the first primal problem (to obtain u_h as discussed in Section 3) we used the Galerkin finite element scheme with the discrete space $V_h \subset V$. Here $r = 1$, which results in the space of bilinear functions Q_1 as defined in Section 2.2. For the second primal problem (to obtain $u_h^{(2)}$ as discussed in Section 3) and the adjoint problem we used the discrete space $V_h^{(2)} \subset V$ which is the space of Q_2 functions on the same quadrilateral elements as in V_h. In the following sections, we always use this discretization if not otherwise denoted.

Discussion of Our Findings. First of all, we take a look at the I_{eff} (as in (2.7)) for J_c which measures how good we estimate the true error of J_c with our error estimator. In Figure 1 we observe that for this problem we obtain $I_{\text{eff}} \approx 1$ which shows that we almost approximate the real error with our error estimator.

In the following, let us have a look at the errors in the goal functionals J_i. In Figure 2 we observe that the error in J_c nearly approximates the functional with the largest error. Furthermore we recognize that the error in the other functionals behaves more inconsistent with more than 1000 DOFs. We compare the error using local mesh refinement with the error using global refinement for the single goal functionals. The relative error for the refinement for J_c shown in Figure 3 decreases almost as well as if we just use DWR for J_0. The relative error is less than 10^{-5} with approximately 7×10^4 DOFs (degrees of freedom) for DWR for J_c, 5×10^4 DOFs for DWR for J_0 and 25×10^4 DOFs for global refinement. Therefore both refinements deliver better results than global refinement. The same behavior also appears for the error in J_1 shown in Figure 4. But the most interesting part is the error in J_2 shown in Figure 5.

From the observation in Figure 2, we deduce that J_c shows similar behavior as the functional with the largest error. At the beginning of the algorithm we observe that we have worse convergence for refinement with respect to J_c in comparison to J_2. But later, we achieve a very similar rate because the weight w_2 is the dominating one. For the error in J_2 we obtain a big advantage. To get an error below 10^{-2} we just need approximately 5000 DOFs instead of 5×10^4 and to get an error below 10^{-3} we need for global refinement more than 10^6 DOFs and we just need about 5×10^4, so we obtain an error of 10^{-3} instead of 10^{-2} for the same

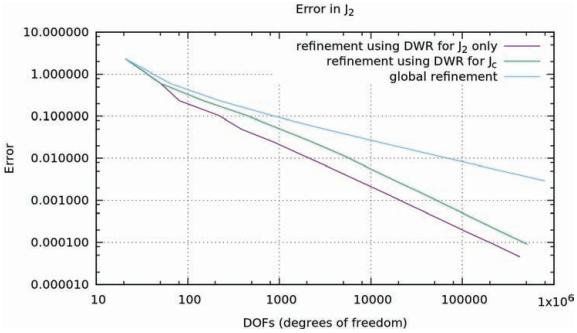

Figure 5. Example 1a: Comparison of relative errors for different refinements for J_2.

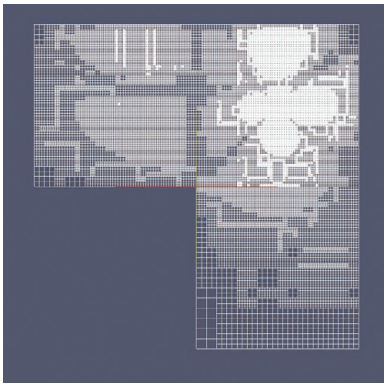

Figure 6. Example 1a: Mesh for DWR for J_0 after 7 refinement steps.

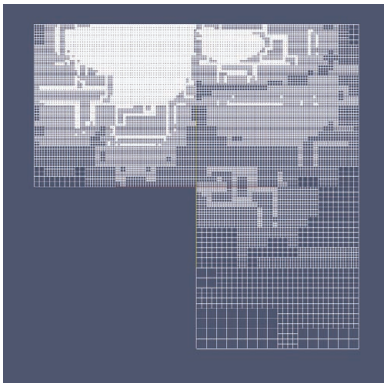

Figure 7. Example 1a: Mesh for DWR for J_1 after 7 refinement steps.

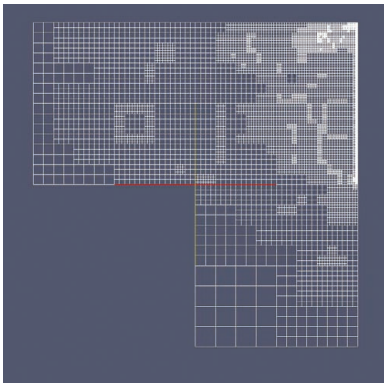

Figure 8. Example 1a: Mesh for DWR for J_2 after 9 refinement steps.

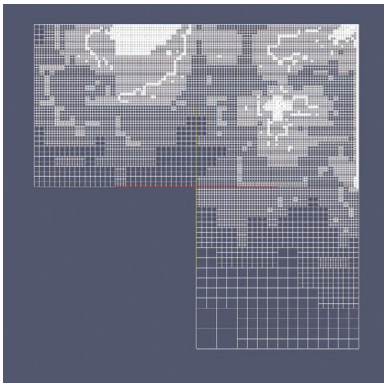

Figure 9. Example 1a: Mesh for DWR for J_c after 9 refinement steps.

number of DOFs. However, the results we obtained are not as good as when using DWR just for J_2 through the effect at the beginning, but we also get the reduction in the other functionals of interest.

Finally, we take a look at the refined meshes in Figures 6–9. The corresponding initial mesh is displayed in Figure 14. We observe that the mesh created by refinement with respect to J_c looks like a combination of the meshes where refinement for one functional is used. We monitor that J_2 is again the dominating part in J_c.

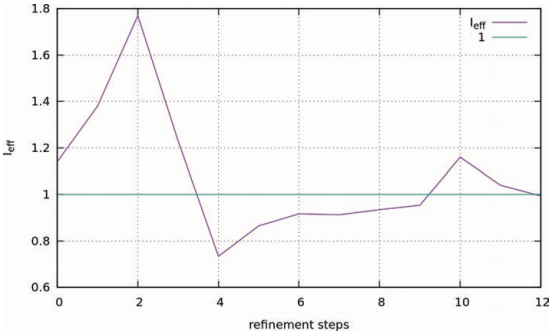

Figure 10. Example 1b: I_{eff} for J_c vs. refinement steps.

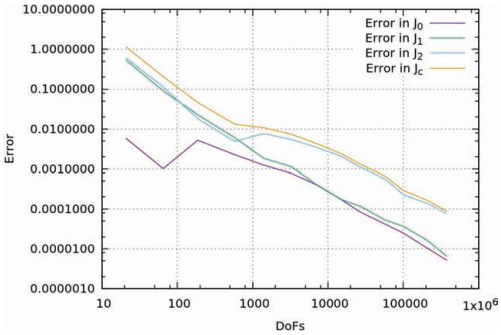

Figure 11. Example 1b: Comparison of relative errors.

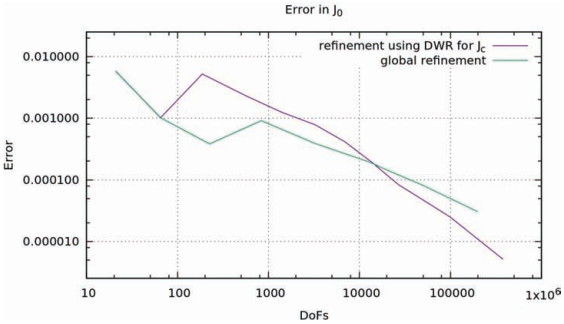

Figure 12. Example 1b: Comparison of relative errors for different refinements for J_0.

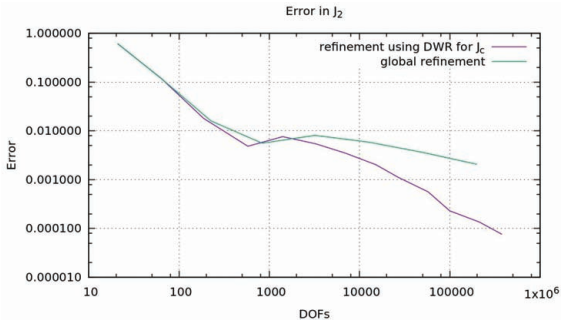

Figure 13. Example 1b: Comparison of relative errors for different refinements for J_2.

4.1.2 Example 1b

In this part, we work with the same domain and same discretization as before, but consider a discontinuous right-hand side

$$f(x,y) = \begin{cases} \sqrt{|x|} + \sqrt{|y|} & \text{if } x^2 + y^2 < 1, \\ -x^2 + y & \text{else.} \end{cases}$$

Here no exact solution u is known.

Goal Functionals of Interest. We consider the following three goal functionals:

$$J_0(u) := u(0.5, 0.5), \quad J_1(u) := \int_{\Omega_2} u(x,y)\, d(x,y), \quad J_2(u) := \int_{\Gamma_2} \nabla u(x,y).n\, d(x,y),$$

where $\Omega_2 = (-1, -0.5) \times (0, 0.5)$ and $\Gamma_2 = \{1\} \times (0, 1)$. Since the exact solution is not known, we approximate the exact functional values on a very fine mesh:

$$J_0(u) \approx 0.15389345606, \quad J_1(u) \approx -0.012801283700, \quad J_2(u) \approx -0.36864857768.$$

Discussion of Our Findings. The effectivity index I_{eff} in this case is shown in Figure 10. It is not as close to 1 as for Example 1a. This may happen through the loss of regularity.

Hence we do not always get the optimal refinement. As before, the interesting part is the error in the functionals which is analyzed in Figure 11. We detect that the decrease of the error in J_c is again similar to the decrease of the functional with the largest error. Hence J_c changes the behavior after the error of J_2 starts to dominate. As it can been seen in Figure 12, the refinement with respect to J_c even delivers worse results for J_0 than for global refinement at the beginning, but we get a better decrease in the error afterwards. This raises the question whether it can happen that the refinement with respect to J_c does deliver a worse error than global refinement with the same number of DOFs for a single functional. This question will be answered

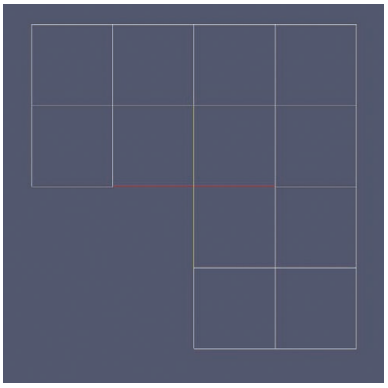

Figure 14. Initial mesh used in Example 1, 2a, and 3.

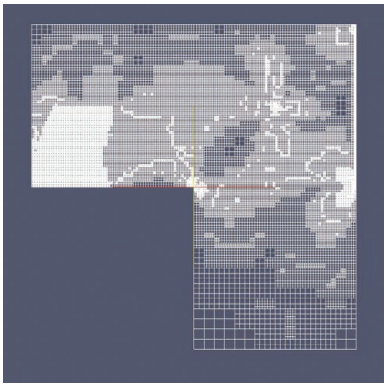

Figure 15. Example 1b: Mesh for DWR for J_c after 9 refinement steps.

in Example 2a. In J_2 we again monitor a similar behavior as in Example 1a (see Figure 13). To achieve an error less than 10^{-3} we have to use approximately 10^6 DOFs and for the refinement with respect to J_c we just need approximately 3×10^4 DOFs. However we see that the error in J_2 even increases for one refinement step and this occurs in the global and the refinement for J_c.

If we take a look at the mesh in Figure 15, we can also see that the DWR method captures low regularity areas like the vertex $(0, 0)$ in the corner of the L-shaped domain.

4.2 Example 2

In this example we consider an eigenfunction of the Laplacian as right-hand side on the unit square $\Omega = (0, 1) \times (0, 1)$ and an L-shaped domain $\Omega = (-1, 1) \times (-1, 1) \setminus (-1, 0) \times (-1, 0)$, respectively, with homogeneous Dirichlet boundary conditions. The discretization is the same as in Section 4.1.

4.2.1 Example 2a

Configuration. Find u such that

$$\begin{aligned} -\Delta u(x, y) &= f(x, y) \quad \text{for all } (x, y) \in \Omega, \\ u(x, y) &= 0 \qquad\quad \text{for all } (x, y) \in \partial\Omega, \end{aligned}$$

where

$$f(x, y) = 10 \sin(\pi x) \sin(3\pi y).$$

The exact solution u is given by

$$u(x, y) = \frac{\sin(\pi x) \sin(3\pi y)}{\pi^2}.$$

The initial mesh is displayed in Figure 17.

Goal Functionals of Interest. We consider the following three goal functionals:

$$J_0(u) := u(0.5, 0.5), \quad J_1(u) := u(0.75, 0.25), \quad J_2(u) := \int_{\Gamma_3} \nabla u(x, y).n \, d(x, y),$$

where $\Gamma_3 = \{1\} \times (0, 1)$.

Discussion of Our Findings. The effectivity index I_{eff} has values in $(1.06, 1.45)$, which is better than in Example 1b but worse than in Example 1a.

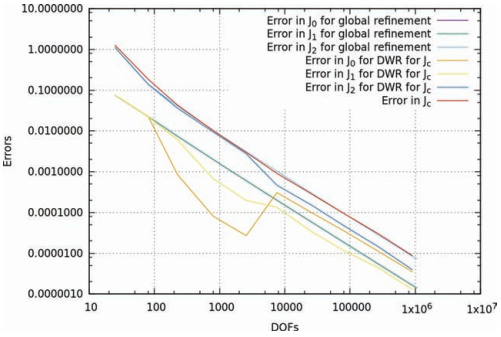

Figure 16. Comparison of relative errors for Example 2a.
We note that the error in J_0 and J_1 for global refinement coincides.

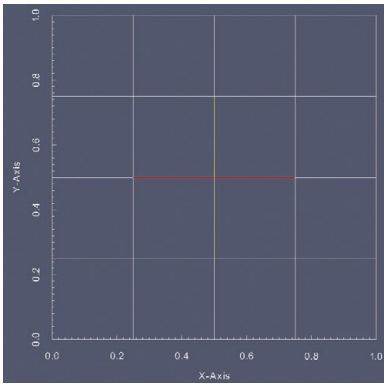

Figure 17. Initial mesh used for Example 2a.

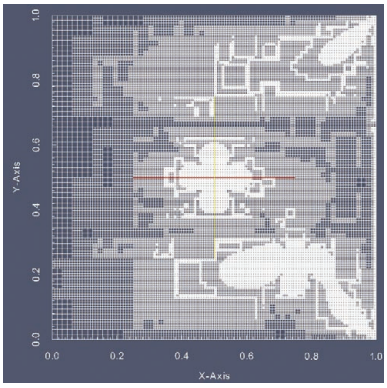

Figure 18. Mesh after 7 refinement steps for Example 2a.

With regard to the error, we make the following observations. In Figure 16 it is observed that we hardly achieve an advantage at the beginning for J_2, but at one specific refinement step we get a much better decrease in the error than in the other steps. But in this step we also get a worse error reduction in the other functionals and in J_0 even an increase. We also observe that we do not necessarily obtain a better result for one functional (here J_0) if we use DWR with respect to J_c than for global refinement. An interesting fact is that the errors of J_0 and J_1 exactly coincide for global refinement but for the refinement using J_c they do not. One reason could be that the errors accumulate and therefore some of the refinement areas for the single functionals are geometrically connected (as visualized in Figure 18).

4.2.2 Example 2b

In the second part of Example 2, we compute now on an L-shaped domain with other functionals of interest and we investigate the sign of the combined functional J_c.

Goal Functionals of Interest. We consider the following three goal functionals:

$$J_0(u) := u(0.5, 0.5), \quad J_1(u) := \int_{\Omega_4} u(x,y)\, d(x,y), \quad J_2(u) := \int_{\Gamma_4} \nabla u(x,y).n\, d(x,y),$$

where $\Omega_4 = (0.75, 1) \times (0, 0.25)$ and $\Gamma_4 = \{1\} \times (0, 1)$.

Discussion of Our Findings. The relevant effectivity index I_{eff} is always in $(1.03, 1.08)$ except on the coarsest mesh in step 0 where $I_{\text{eff}} = 0.75$. In the examples before, we were mostly concerned about the error. But if we construct J_c as in (3.9), we have (as explained in Section 3.1) to solve a bigger linear system just to get a sign for each weight. The question is whether this is really necessary. Therefore, we investigate in the following

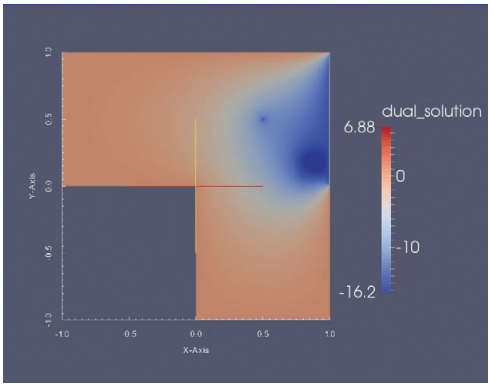

Figure 19. Example 2b: Adjoint solution after 6 refinement steps.

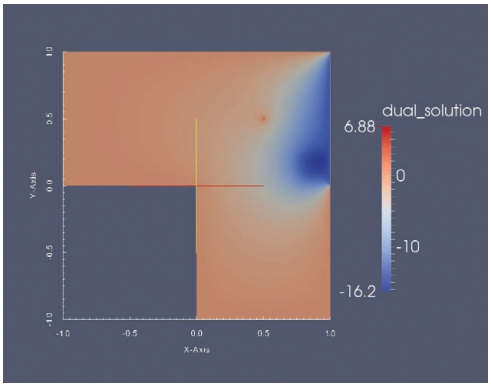

Figure 20. Example 2b: Adjoint solution after 7 refinement steps.

the adjoint solution in more detail. Comparing Figures 19 and 20, we monitor that there is indeed a change of the sign of w_0 (as defined in (3.1)) during the computation. Hence the computation of the sign cannot be avoided in the algorithm presented in Section 3.1.

4.3 Example 3

In this numerical test we consider a non-homogeneous Dirichlet boundary condition on the L-shaped domain. The right-hand side is non-homogeneous and has a pole at (0, 0). Moreover, we perform computations with higher-order finite elements.

Configuration. Find u such that

$$\begin{aligned} -\Delta u(x,y) &= -\frac{1}{\|(x,y)\|_{l_2}} \quad \text{for all } (x,y) \in \Omega, \\ u(x,y) &= \|(x,y)\|_{l_2} \quad \text{for all } (x,y) \in \partial\Omega, \end{aligned}$$

where

$$\Omega = (-1,1)\times(-1,1) \setminus (-1,0)\times(-1,0).$$

The exact solution u is given by

$$u(x,y) = \|(x,y)\|_{l_2}.$$

The initial mesh is displayed in Figure 25.

Goal Functionals of Interest. We consider the following three goal functionals:

$$J_0(u) := u(0.5,-0.5), \quad J_1(u) := u(0.5,0.5), \quad J_2(u) := \int_{\Gamma_5} \nabla u(x,y).n \, d(x,y),$$

where $\Gamma_5 = (0,1)\times\{-1\}$. The exact solution u yields

$$J_0(u) = \sqrt{0.5}, \quad J_1(u) = \sqrt{0.5}, \quad J_2(u) = \log(1+\sqrt{2}).$$

Discretization. Here we use higher-order finite elements for discretization. We denote the configuration, where we used Q_r for the first primal problem and Q_{r+1} for the second primal problem and the adjoint problem, by Q_r/Q_{r+1} finite elements. We use

- Q_1/Q_2 finite elements,
- Q_4/Q_5 finite elements,
- Q_8/Q_9 finite elements.

Remark 4.1. In our numerical test we do not see any effect when using Q_r/Q_s finite elements, where $s > r+1$ in comparison to Q_r/Q_{r+1}. Since the latter is less computational work, we always use this FE combination.

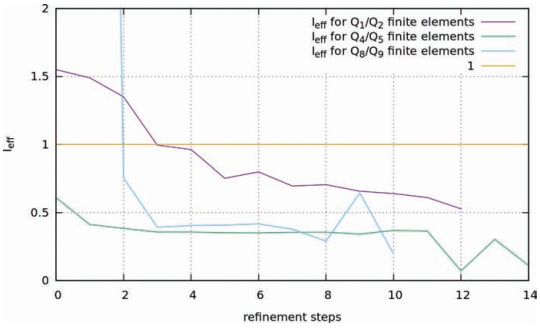

Figure 21. I_{eff} for Example 3.

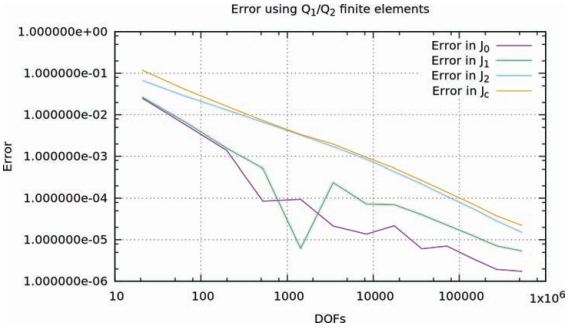

Figure 22. Example 3: Error using Q_1/Q_2 finite elements.

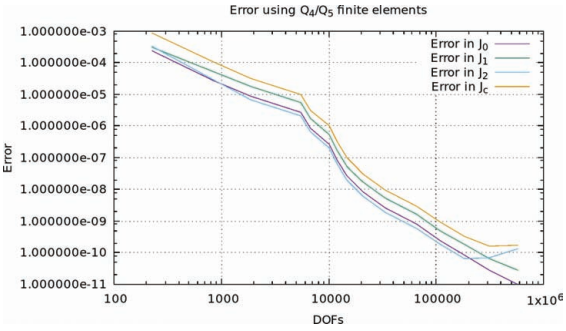

Figure 23. Example 3: Error using Q_4/Q_5 finite elements.

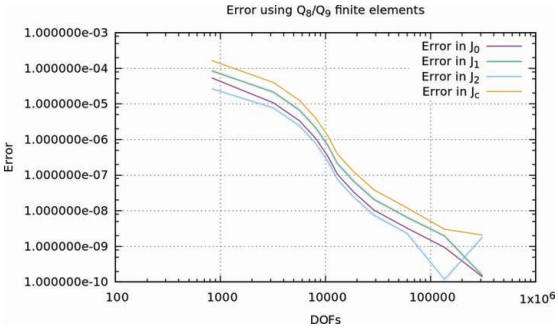

Figure 24. Example 3: Error using Q_8/Q_9 finite elements.

Discussion of Our Findings. In this example we do have a singularity in the right-hand side and we are interested in the behavior of our algorithm employing different finite elements. For the I_{eff} we obtain the following results:

- $I_{\text{eff}} \in (0.5, 1.6)$ for Q_1/Q_2 finite elements,
- $I_{\text{eff}} \in (0.07, 0.65)$ for Q_4/Q_5 finite elements,
- $I_{\text{eff}} \in (0.15, 13.5)$ for Q_8/Q_9 finite elements.

Observing our findings, we obtain an error estimator that works quite well if we use Q_1/Q_2 finite elements, but it also seems that for a higher polynomial degree the error estimator gets worse. However if we take a look at Figure 21 we can see that the high and low values for higher polynomial degrees appear just at the beginning and the end of our algorithm. This may happen through the worse approximation due the coarse grid at the beginning and numerical errors at the end of our computation, because our error estimator just estimates the discretization error and not the numerical error. Nevertheless, in the steps between, our error estimator is still better for Q_1/Q_2 finite elements than for higher order.

As in the previous examples, we are also interested how the error is affected in the functionals of interest. For Q_1/Q_2 finite elements we monitor in Figure 22 that we have a high inconsistency in the decrease of error and we even get some increases in both point evaluations. If we use higher polynomial degrees, we nearly always have a similar reduction in all errors (except at the end where the error for J_2 increases because of numerical errors). Comparing the errors themselves, we do get much better results if we use Q_4/Q_5 finite elements or Q_8/Q_9 finite elements. On the other hand, once we work with higher order, we do not conclude an advantage of using very high polynomial degrees. Consequently, there is a big advantage of going from Q_1/Q_2 to Q_4/Q_5. But even higher polynomial degrees do not pay off.

We note that the error increase towards the maximal refinement steps in Figures 23 and 24 results from numerical inaccuracies towards low tolerances, e.g., from solving the linear system or more likely that the flux evaluation is not accurate enough (since we deal with a derivative). If we take a look at our refined meshes shown in Figures 26–28, we see that we always obtain refinements at the position of the singularity of the right-hand side. This effect becomes stronger if we use polynomials with higher degree. Here, very localized mesh refinement is observed.

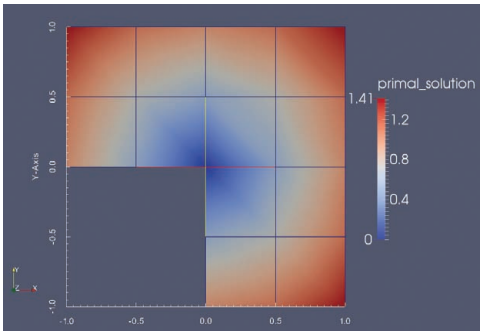

Figure 25. Example 3: Initial mesh (solution for Q_1/Q_2 finite elements).

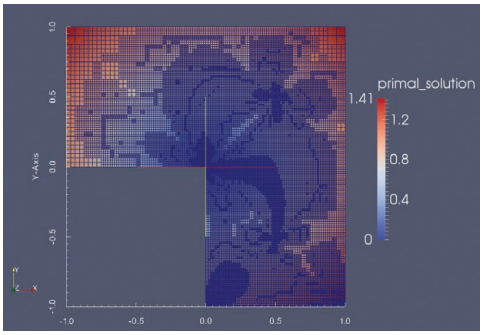

Figure 26. Mesh for Q_1/Q_2 finite elements after 9 refinement steps (71565 DOFs).

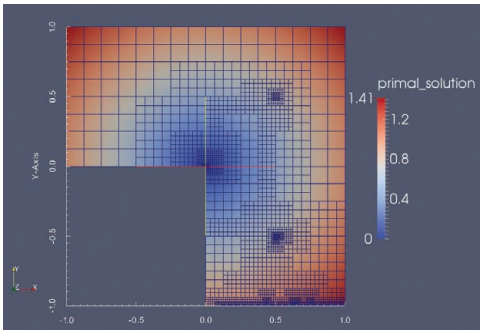

Figure 27. Example 3: Mesh for Q_4/Q_5 finite elements after 10 refinement steps (65463 DOFs).

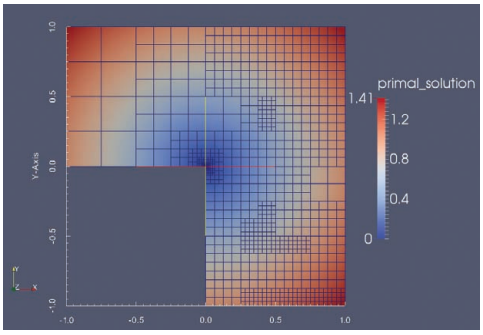

Figure 28. Example 3: Mesh for Q_8/Q_9 finite elements after 8 refinement steps (59639 DOFs).

4.4 Example 4: A Slit Domain

4.4.1 Example 4a

In this final example we consider a slit domain with displacement discontinuity (which can be interpreted as a crack) as pictured in Figure 29. On the slit, homogeneous Neumann boundary conditions are prescribed. In Example 4a, non-homogeneous Dirichlet and homogeneous Neumann conditions are prescribed on the outer boundary. In Example 4b, non-homogeneous Dirichlet on the entire outer boundary are used for which, on the other hand, a manufactured solution can be constructed.

Configuration. We consider the Laplace equation on a slit domain: Find u such that

$$\begin{aligned} -\Delta u(x,y) &= 0 && \text{for all } (x,y) \in \Omega, \\ u(x,y) &= g(x,y) && \text{for all } (x,y) \in \Gamma_D, \\ \nabla u(x,y) \cdot n(x,y) &= 0 && \text{for all } (x,y) \in \Gamma_N, \end{aligned}$$

where

$$\Omega = (-1,1) \times (-1,1) \setminus \{(x,0) \mid -1 \le x \le 0\}.$$

The boundary parts are given as

$$\Gamma_D = \{(-1,y) \mid -1 \le y \le 1\}, \quad \Gamma_N = \partial\Omega \setminus \Gamma_N.$$

Here, $g(x,y)$ is defined as

$$g(x,y) := \operatorname{sign}(y) \frac{\lambda_{G_c}}{\sqrt{2}} \sqrt{\sqrt{x^2+y^2} - x},$$

where $\lambda_{G_c} = 1$. This coefficient is a material parameter and related to the fracture toughness.

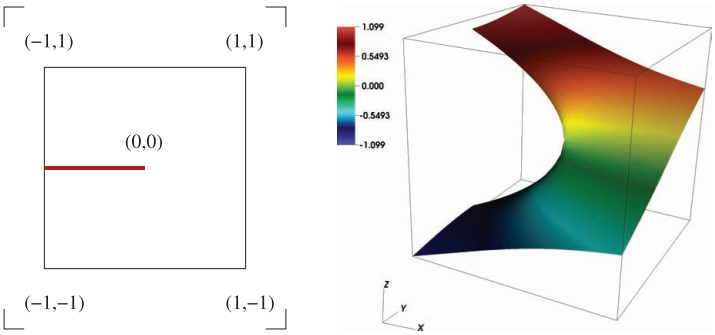

Figure 29. Example 4a: Discontinuity location (in red; left figure) and related numerical solution in a 3D plot (right) to Example 4b, which additionally highlights the non-homogeneous Dirichlet boundary conditions and the jump of displacements across the crack.

These conditions introduce a discontinuity on the boundary at $(-1, 0)$ and consequently a crack with displacement discontinuity as displayed in Figure 29.

Goal Functionals of Interest. We deal with the following four goal functionals (but we notice that not always all goal functionals are simultaneously considered):

$$J_0(u) := u(0.75, 0.75), \qquad J_1(u) := u(-0.5, -0.25),$$
$$J_2(u) := \int_{\Gamma_6} \nabla u(x,y).n \, d(x,y), \quad J_3(u) := \int_{\Omega_6} u(x,y) \, d(x,y),$$

where $\Gamma_6 = \{-1\} \times (-1, -0.25)$ and $\Omega_6 = (0, 1) \times (-1, 0)$.

However no exact solution is known, and therefore we approximate the exact functionals by values that have been obtained from a very fine mesh:

- $J_0(u) \approx +0.18949212064$,
- $J_1(u) \approx -0.66061009755$,
- $J_2(u) \approx -0.54411579542$,
- $J_3(u) \approx -0.18268521784$.

Discussion of Our Findings. First, we consider detailed studies for a single goal functional. The reason is that this test might serve as kind of benchmark for testing algorithms which compute fractures. Even for single goal functionals the existing literature is rare. A fracture in 2D is nothing else than a discontinuity along a line in the primal (displacement) solution. Thus the slit domain is a limiting case for a fracture setting. After having revisited a single goal functional, we add as in the previous examples more quantities of interest. We use four different configurations for our goal functionals:

- Configuration 1: evaluating $J_0(u)$,
- Configuration 2: evaluating $J_0(u)$ and $J_1(u)$,
- Configuration 3: evaluating $J_0(u)$, $J_1(u)$ and $J_2(u)$,
- Configuration 4: evaluating $J_0(u)$, $J_1(u)$, $J_2(u)$ and $J_3(u)$.

In Figure 30 we see that in all configurations, the error in J_c is underestimated, but we cannot expect to have $I_{\text{eff}} = 1$ for this problem due to the loss of regularity. Furthermore the I_{eff} seems not to differ that much for the different configurations. In Figures 31–34 we observe that for all configurations we get a much better reduction for our refinement if we compare to global refinement. An interesting aspect is that in this example all errors are of similar order and we also have a similar reduction in the error.

Another interesting observation can be made in Figure 35 where we even obtain better results if we refine for J_c instead for J_0. Usually we would expect a worse result if we try to refine for more than one functional at the same time. But unfortunately we do not have such findings in general as it can be monitored in Figure 5 for J_2. Nevertheless, we observe that the results are also good for less regular problems even if the error

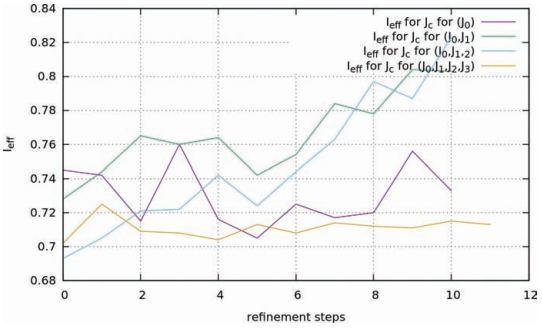

Figure 30. Example 4a: I_{eff} for Configurations 1–4.

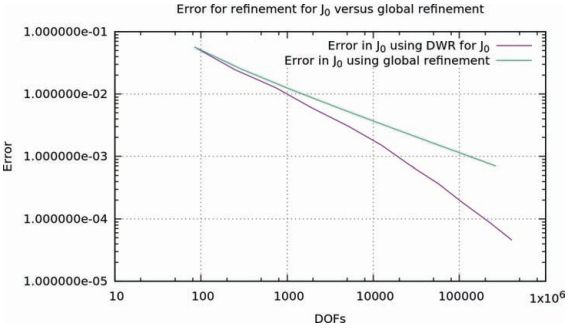

Figure 31. Example 4a: Error for Configuration 1.

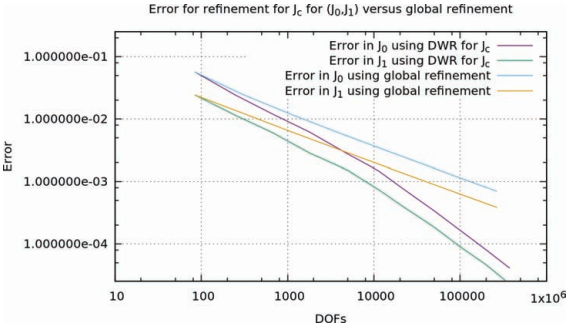

Figure 32. Example 4a: Error for Configuration 2.

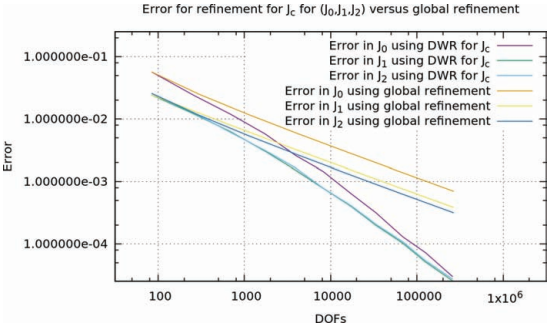

Figure 33. Example 4a: Error for Configuration 3.

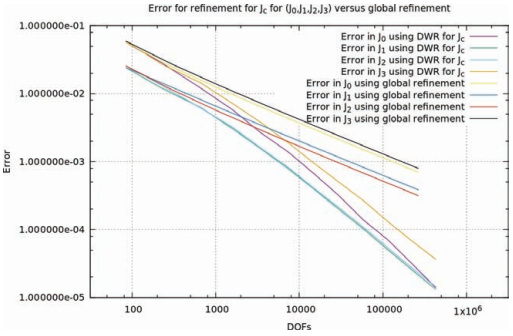

Figure 34. Example 4a: Error for Configuration 4.

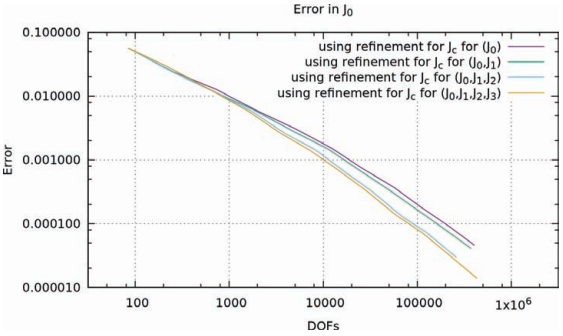

Figure 35. Example 4a: Error for J_0 for Configurations 1–4.

estimator is not that accurate. We also find that our refinement scheme takes care of low regularity domains for all our configurations as displayed in Figures 36–39. That is why the mesh is also refined at the fracture tip in the middle of the domain. We finally notice that the initial mesh for these computations is shown in Figure 49.

4.4.2 Example 4b

In this example, one goal is to perform studies for different polynomial degrees. As in Example 4a, we consider the Laplace equation on a slit domain with displacement discontinuity (i.e., the crack) as pictured in Figure 29. The domain and location of the crack are chosen in such a way that we can work with the manufactured solution constructed in [4, 11]:

$$u(x,y) = \lambda_{G_c} r^{1/2} \sin(\phi/2), \quad \lambda_{G_c} = 1, \quad 0 \le \phi < 2\pi,$$

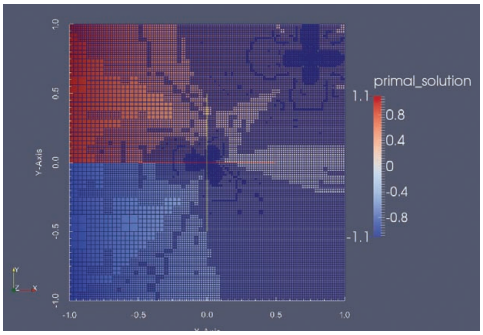

Figure 36. Example 4a: Mesh for Configuration 1 after 7 refinement steps (56809 DOFs).

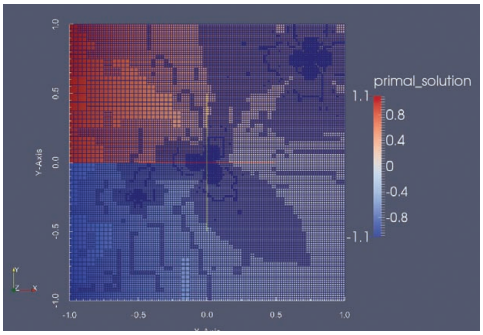

Figure 37. Example 4a: Mesh for Configuration 2 after 7 refinement steps (49883 DOFs).

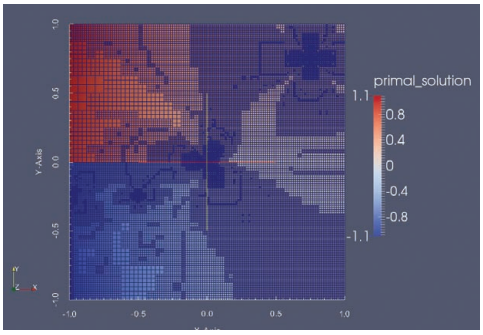

Figure 38. Example 4a: Mesh for Configuration 3 after 8 refinement steps (68809 DOFs).

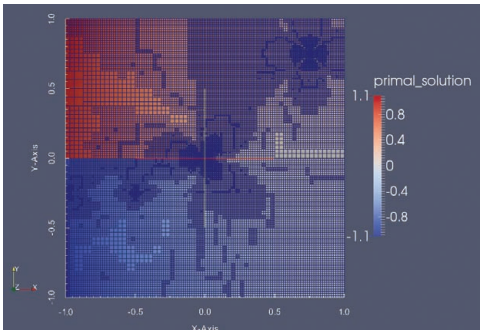

Figure 39. Example 4a: Mesh for Configuration 4 after 8 refinement steps (58425 DOFs).

in polar coordinates. In Cartesian coordinates the solution reads

$$u(x,y) = \operatorname{sign}(y)\frac{\lambda_{G_c}}{\sqrt{2}}\sqrt{\sqrt{x^2+y^2}-x}.$$

We prescribe

$$g(x,y) := \operatorname{sign}(y)\frac{\lambda_{G_c}}{\sqrt{2}}\sqrt{\sqrt{x^2+y^2}-x},$$

on the entire outer boundary.

Goal Functionals of Interest. We consider the following four goal functionals:

$$J_0(u) := u(0.75, 0.75), \qquad J_1(u) := u(-0.5, -0.25),$$
$$J_2(u) := \int_{\Gamma_7} \nabla u(x,y).n \, d(x,y), \quad J_3(u) := \int_{\Omega_7} u(x,y) \, d(x,y),$$

where $\Gamma_7 = \{-1\} \times (-1, -0.25)$ and $\Omega_7 = (0,1) \times (-1,0)$.

Discretization. Here we use different FE for discretization (as in Example 3). We define Q_r/Q_{r+1}, which means that we use Q_r for the first primal problem and Q_{r+1} for the second primal problem and the adjoint problem. We work with the combinations Q_1/Q_2, Q_2/Q_3, Q_3/Q_4, Q_4/Q_5.

Discussion of Our Findings. First, we are interested in how the error decreases with respect to the number of refinement steps. Here, we observe that for all tested polynomial degrees, we achieve a similar decrease as $O(h)$ for global refinement. In more detail, the error of each functional for both refinement methods and all tested polynomial degrees is approximately halved at each refinement step as we detect in Figures 40–43. Therefore, we almost get the same behavior as global refinement in the errors if we just compare it to the

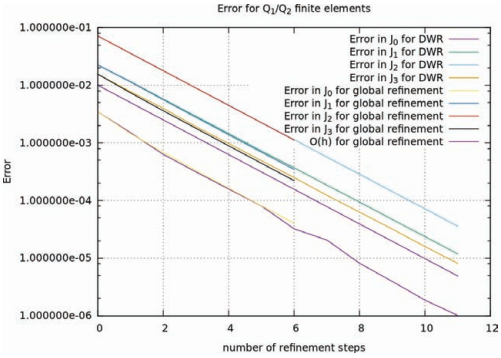

Figure 40. Example 4b: Error versus refinement steps for Q_1/Q_2 finite elements.

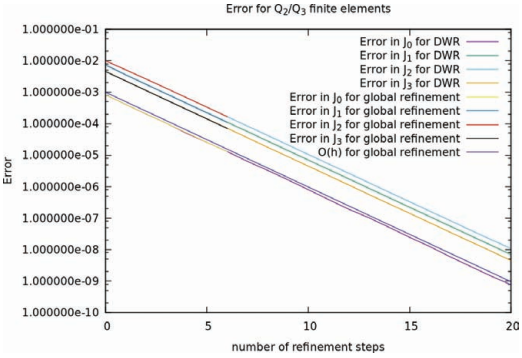

Figure 41. Example 4b: Error versus refinement steps for Q_2/Q_3 finite elements.

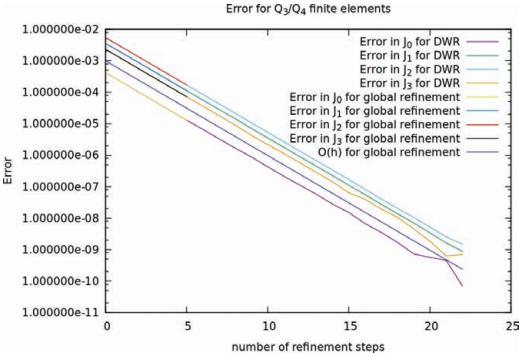

Figure 42. Example 4b: Error versus refinement steps for Q_3/Q_4 finite elements.

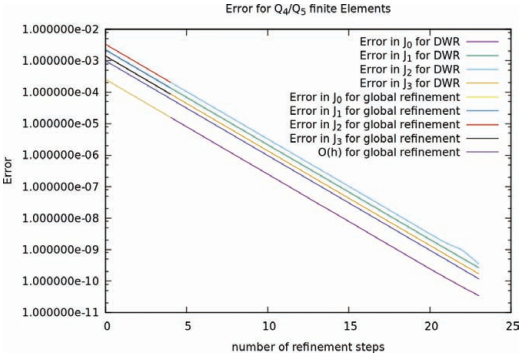

Figure 43. Example 4b: Error versus refinement steps for Q_4/Q_5 finite elements.

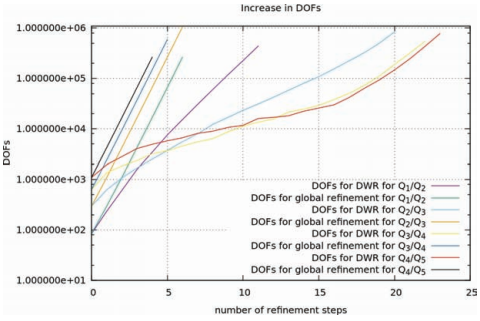

Figure 44. Example 4b: Increase in DOFs for different finite elements and refinement schemes.

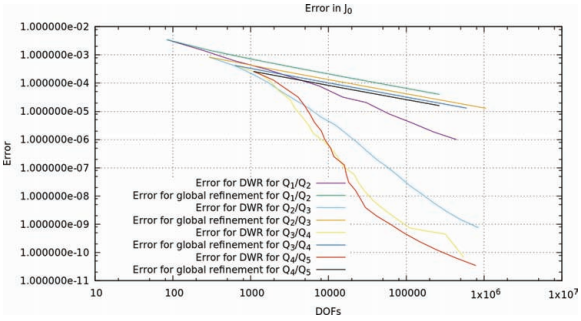

Figure 45. Example 4b: Error in J_0 for different finite elements and refinement schemes.

refinement steps. However, as we monitor in Figure 44, we save many DOFs for our refinement scheme in comparison to global refinement (behaves like $O(h^{-2})$) for almost the same accuracy. If we now compare the different polynomial degrees, we observe that for a higher polynomial degree, we start with a higher number of DOFs, but we do get less increase than for lower polynomial degrees such that we get less DOFs for a higher polynomial degree after a certain number of refinement steps. With regard to the maximal number of DOFs, we remark that the plot curves for global refinement stop because we did not solve linear systems with more than 1 100 000 unknowns. The behavior of the last refinement steps shown in Figures 42 and 43 results from numerical errors.

But as we visualize in Figure 44 the advantage of Q_4/Q_5 finite elements compared to Q_3/Q_4 finite elements is not as big as if we compare Q_3/Q_4 and Q_2/Q_3 finite elements. Since all errors for all polynomial degrees show a similar behavior, we take a look at the error in J_0 in this example. If we plot the error in J_0

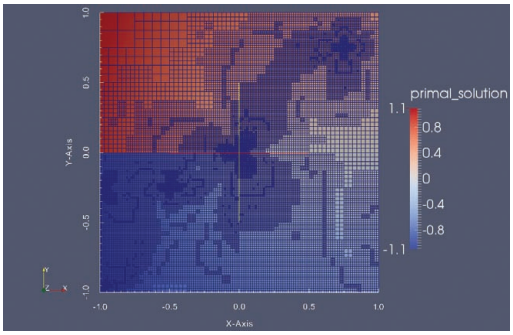

Figure 46. Example 4b: Mesh and solution for Q_1/Q_2 finite elements after 8 refinement steps (60119 DOFs).

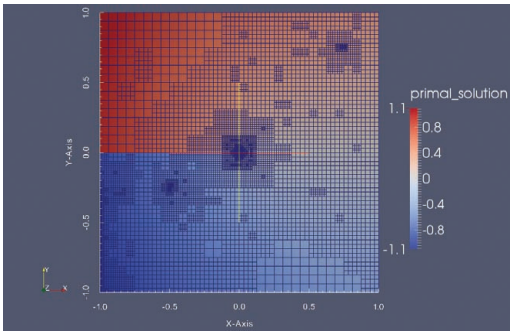

Figure 47. Example 4b: Mesh and solution for Q_2/Q_3 finite elements after 18 refinement steps (58289 DOFs).

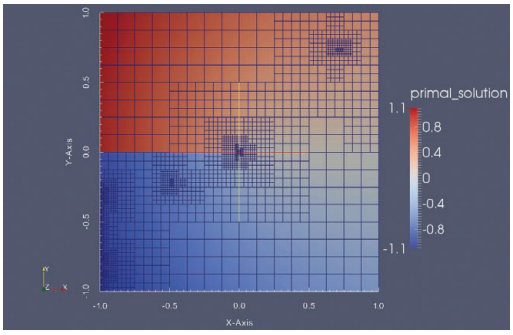

Figure 48. Example 4b: Mesh and solution for Q_4/Q_5 finite elements after 18 refinement steps (63403 DOFs).

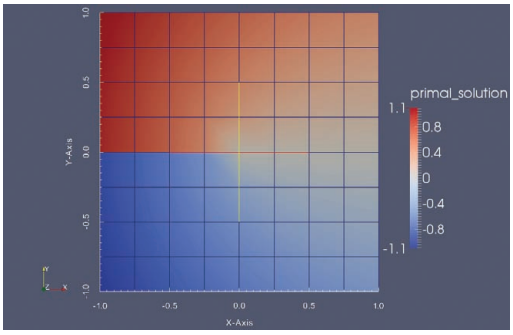

Figure 49. Example 4b: Initial mesh and solution for Example 4a (Q_4/Q_5 solution for Example 4b).

against the DOFs as shown in Figure 45, we see the advantage of the algorithm in Section 3.1. Here we again see the advantage of using higher polynomial degrees. To reach an error less than 10^{-6} we need for Q_1/Q_2 finite elements of about 5×10^5 DOFs and for Q_4/Q_5 finite elements of about 9000 DOFs. But also the findings for Q_1/Q_2 finite elements are satisfying in comparison to global refinement for all tested polynomial degrees.

By taking a look at the meshes constructed by our algorithm (Figures 46–48), one can observe that for higher polynomial degrees more refinements in low regularity regions than in specific functional areas take place. This is in agreement with our findings in Example 3.

5 Conclusions

In this work, we further developed dual-weighted residual error estimation for multiple goal functionals with application to elliptic problems. We addressed in Section 3.5 a variational localization of the error estimator using a partition-of-unity. Next, we proposed an alternative way to solve the adjoint to the adjoint problem. In Section 4, we provided extensive numerical computations for various domains, different boundary conditions, and different types of goal functionals as well as higher-order finite element calculations. From our observations we can deduce that the functional J_c (introduced in (3.9)) delivers a similar behavior than the functional, contained in $\mathbb{J}$, with the highest relative error if the self-chosen weights ω_i from (3.9) are equal. This leads to a better decrease in the error of this functional but not necessarily for the other functionals (as shown in Example 2a). However, we do obtain a better decrease in the maximal relative error than global refinement. If the errors in the different functionals are of similar order, we observe a decrease in each functional as shown in Figures 23 and 24 and Figures 32–34. Example 2b demonstrates that the computation of the sign of the procedure described in Section 3.1 can unfortunately not be avoided. We found out that there can be a big advantage using a higher polynomial degree, even we may get a worse error estimator (as

shown in Example 3). Furthermore we also obtained very good findings for a problem on a low regularity domain (as shown in Example 4a). Also for singular right-hand sides, we obtained satisfying results and we found that for higher polynomial degrees we do more refinement steps at the low regularity regions (see again Example 3). We briefly comment that we also observed the general benefit of using an adaptive algorithm: using global refinement limits significantly to reach low tolerances because the linear systems are simply too big and memory-consuming. For instance, in Example 4b we could only reach a tolerance of 10^{-5} using global mesh refinement. Finally, we notice that the last test, namely Example 4, might serve as basis for developing mesh adaptivity for sophisticated computational methods for treating fracture (or damage) settings (in which the crack is not nicely aligned with the mesh) such as extended/generalized finite elements or phase-field methods. Thus the provided methodology has an immediate potential to be extended for current practical applications.

Acknowledgment: We want to thank Professor Ulrich Langer for supporting this work at the Institute of Computational Mathematics at JKU Linz.

References

[1] M. Ainsworth and J. T. Oden, A posteriori error estimation in finite element analysis, *Comput. Methods Appl. Mech. Engrg.* **142** (1997), no. 1–2, 1–88.
[2] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure Appl. Math. (New York), John Wiley & Sons, New York, 2000.
[3] G. Allaire, *Analyse numerique et optimisation*, Les Éditions de l'École Polytechnique, Palaiseau, 2005.
[4] J. Andersson and H. Mikayelyan, The asymptotics of the curvature of the free discontinuity set near the cracktip for the minimizers of the Mumford–Shah functional in the plain. A revision, preprint (2015), https://arxiv.org/abs/1204.5328v2.
[5] T. Apel, A.-M. Saendig and J. R. Whiteman, Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains, *Math. Methods Appl. Sci.* **19** (1996), no. 1, 63–85.
[6] W. Bangerth, R. Hartmann and G. Kanschat, deal.II – A general purpose object oriented finite element library, *ACM Trans. Math. Softw.* **33** (2007), no. 4, Article ID 24.
[7] W. Bangerth, T. Heister and G. Kanschat, deal. II Differential equations analysis library, Technical Reference (2013).
[8] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Lectures in Math. ETH Zürich, Birkhäuser, Basel, 2003.
[9] R. Becker and R. Rannacher, Weighted a posteriori error control in FE methods, in: *ENUMATH'97* (Heidelberg 1997), World Scientific, Singapore (1998), 621–637.
[10] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numer.* **10** (2001), 1–102.
[11] A. Bonnet and G. David, *Cracktip is a Global Mumford–Shah Minimizer*, Astérisque 274, Société Mathématique de France, Paris, 2001.
[12] M. Braack and A. Ern, A posteriori control of modeling errors and discretization errors, *Multiscale Model. Simul.* **1** (2003), no. 2, 221–238.
[13] D. Braess, *Finite Elemente. Theory, Fast Solvers and Applications in Elasticity Theory*, Springer, Berlin, 2007.
[14] G. F. Carey and J. T. Oden, *Finite Elements. Volume III: Computational Aspects*, Texas Finite Elem., Prentice-Hall, Englewood Cliffs, 1984.
[15] C. Carstensen and R. Verfürth, Edge residuals dominate a posteriori error estimates for low order finite element methods, *SIAM J. Numer. Anal.* **36** (1999), no. 5, 1571–1587.
[16] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1987.
[17] M. Dobrowolski, *Angewandte Funktionalanalysis: Funktionalanalysis, Sobolev-Räume und elliptische Differentialgleichungen*, Springer, Berlin, 2006.
[18] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, Introduction to adaptive methods for differential equations, in: *Acta Numerica 1995*, Cambridge University Press, Cambridge (1995), 105–158.
[19] D. Estep, M. Holst and M. Larson, Generalized green's functions and the effective domain of influence, *SIAM J. Sci. Comput.* **26** (2005), no. 4, 1314–1339.
[20] L. C. Evans, *Partial Differential Equations*, American Mathematical Society, Providence, 2010.
[21] M. Giles and E. Süli, Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality, *Acta Numer.* **11** (2002), 145–236.
[22] C. Großmann, H.-G. Roos and M. Stynes, *Numerical Treatment of Partial Differential Equations*, Springer, Berlin, 2007.

[23] R. Hartmann, Multitarget error estimation and adaptivity in aerodynamic flow simulations, *SIAM J. Sci. Comput.* **31** (2008), no. 1, 708–731.
[24] R. Hartmann and P. Houston, Goal-oriented a posteriori error estimation for multiple target functionals, in: *Hyperbolic Problems: Theory, Numerics, Applications*, Springer, Berlin (2003), 579–588.
[25] P. Houston, B. Senior and E. Sueli, hp-discontinuous galerkin finite element methods for hyperbolic problems: Error analysis and adaptivity, *Internat. J. Numer. Methods Fluids* **40** (2002), no. 1–2, 153–169.
[26] D. Kuzmin and S. Korotov, Goal-oriented a posteriori error estimates for transport problems, *Math. Comput. Simulation* **80** (2010), no. 8, 1674–1683.
[27] J. Oden and S. Prudhomme, On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors, *Comput. Methods Appl. Mech. Engrg.* **176** (1999), 313–331.
[28] J. Peraire and A. Patera, Bounds for linear-functional outputs of coercive partial differential Equations: Local indicators and adaptive refinement, in: *Advances in Adaptive Computational Methods in Mechanics*, Elsevier, Amsterdam (1998), 199–215.
[29] R. Rannacher and F.-T. Suttmeier, A feed-back approach to error control in finite element methods: Application to linear elasticity, *Comput. Mech.* **19** (1997), no. 5, 434–446.
[30] T. Richter, Goal-oriented error estimation for fluid-structure interaction problems, *Comput. Methods Appl. Mech. Engrg.* **223–224** (2012), 38–42.
[31] T. Richter and T. Wick, Variational localizations of the dual weighted residual estimator, *J. Comput. Appl. Math.* **279** (2015), no. 0, 192–208.
[32] A. Schroeder and A. Rademacher, Goal-oriented error control in adaptive mixed FEM for signorini's problem, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), no. 1–4, 345–355.
[33] E. van Brummelen, S. Zhuk and G. van Zwieten, Worst-case multi-objective error estimation and adaptivity, *Comput. Methods Appl. Mech. Engrg.* **313** (2017), 723–743.
[34] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley, New York, 1996.
[35] S. Weisser and T. Wick, The dual-weighted residual estimator realized on polygonal meshes, Preprint Number 384, Saarland University, Department of Mathematics, 2016, https://www.math.uni-sb.de/service/preprints/.
[36] T. Wick, Goal functional evaluations for phase-field fracture using PU-based DWR mesh adaptivity, *Comput. Mech.* **57** (2016), no. 6, 1017–1035.
[37] J. Wloka, *Partial Differential Equations*, Cambridge University Press, Cambridge, 1987.
[38] K. Zee, E. Brummelen, I. Akkerman and R. Borst, Goal-oriented error estimation and adaptivity for fluid-structure interaction using exact linearized adjoints, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), 2738–2757.