J. Numer. Math., Vol. 17, No. 2, pp. 143–172 (2009)
DOI 10.1515/ JNUM.2009.009

Goal-oriented error control of the iterative solution of finite element equations

D. MEIDNER,* R. RANNACHER,† and J. VIHHAREV†

Received February 19, 2009

Abstract — This paper develops a combined *a posteriori* analysis for the discretization and iteration errors in the computation of finite element approximations to elliptic boundary value problems. The emphasis is on the multigrid method, but for comparison also simple iterative schemes such as the Gauß–Seidel and the conjugate gradient method are considered. The underlying theoretical framework is that of the Dual Weighted Residual (DWR) method for goal-oriented error estimation. On the basis of these *a posteriori* error estimates the algebraic iteration can be adjusted to the discretization within a successive mesh adaptation process. The efficiency of the proposed method is demonstrated for several model situations including the simple Poisson equation, the Stokes equations in fluid mechanics and the KKT system of linear-quadratic elliptic optimal control problems.

Keywords: goal-oriented adaptivity, *a posteriori* error estimation, finite elements, iterative methods, multigrid

1. Introduction

Multigrid methods are extensively used for efficiently solving the discrete equations resulting from the discretization of partial differential equations (see, e.g., [12,18]). In this paper, we consider a general linear elliptic problem discretized by a finite element method as proposed, e.g., in [17]. We develop an adaptive multigrid method for efficient solution of the algebraic equations resulting from the proposed finite element discretization. The idea of the algorithm is as follows: The exact finite element solution approximates the continuous solution only up to discretization accuracy. It seems natural to stop the iteration of the linear solver when the error due to the approximate solution of the discrete equations is comparable to the error due to the finite element discretization itself. To this purpose, we derive an *a posteriori* error estimator which assesses the influences of the discretization and the inexact

*Lehrstuhl für Mathematische Optimierung, Technische Universität München, Fakultät für Mathematik, Boltzmannstraße 3, Garching b. München, Germany

†Institut für Angewandte Mathematik, Universität Heidelberg, Im Neuenheimer Feld 293/294, Heidelberg, Germany

This work has been supported by Deutsche Forschungsgemeinschaft (DFG) through Priority Research Program SPP 1253 ‘Optimization with Partial Differential Equations’.

solution of the arising algebraic equations. This allows us to balance both sources of errors.

The use of adaptive techniques based on *a posteriori* estimation of the discretization error is well accepted in the context of finite element discretization of partial differential equations (see, e.g., [9,20]). Although the convergence properties of multigrid methods are discussed in many publications (cf. [1,13,14]), there are only few results on *a posteriori* error estimation of the iteration error. In the case of solving the Poisson equation, work has been done in [7] and was extended to the Stokes equations in [3]. There, the automatic control of the discretization and multigrid errors has been developed with respect to L^2- and energy norms. The reliability of the proposed adaptive algorithm has been verified on uniformly refined meshes.

However, in many application, the error measured in global norms does not provide useful bounds for the error in terms of a given functional, a so called *quantity of interest*. In this work, we propose the control of both discretization and iteration errors with respect to a general output functional. This approach is based on *a posteriori* error estimation by dual weighted residuals as presented in [9]. We incorporate the adaptive iteration method in the solution process of a given problem. The estimator derived for the discretization error is used on the one hand as stopping criterion for the algebraic iteration and on the other hand provides the necessary information for the construction of locally refined meshes in order to improve the accuracy of the discretization.

A further issue of this paper is the numerical realization of the adaptive method. We explain implementational details and verify the reliability and the efficiency of the proposed algorithm on locally refined meshes.

As starting point, we consider the elliptic problem

$$Au = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \Gamma \tag{1.1}$$

with a linear elliptic operator A and a right-hand side $f \in L^2(\Omega)$ where Ω is assumed to be a bounded domain in $\mathbb{R}^d$, $d \in \{2,3\}$, with polygonal boundary Γ. For simplicity, we impose homogeneous Dirichlet boundary conditions. The case of nonhomogeneous Dirichlet conditions is considered in remarks. However, the techniques provided in this paper can also be applied to problems with other types of boundary conditions.

For the variational formulation of problem (1.1), we introduce the Hilbert space $V := H_0^1(\Omega)$ and the inner product of $L^2(\Omega)$ defined by

$$(v,w) := (v,w)_{L^2(\Omega)} := \int_\Omega vw \, \mathrm{d}x.$$

With the bilinear form $a(\cdot,\cdot)\colon V \times V \to \mathbb{R}$ associated to the linear operator A, the weak formulation of the considered problem (1.1) reads as follows: Find $u \in V$ such that

$$a(u,\varphi) = (f,\varphi) \quad \forall \varphi \in V. \tag{1.2}$$

For the numerical treatment, we discretize this problem leading to a linear system of algebraic equations. Usually, for instance in the articles [8,9], the *a posteriori*

error estimators for the discretization error are derived under the assumption that the exact solution of this linear systems is available. This ensures the crucial property of Galerkin orthogonality. In contrast, here, we assume that the discrete equations are solved only approximately and denote the obtained approximate solution by $\tilde{u}_h$ in contrast to the notation u_h for the 'exact' discrete solution. Our goal is the derivation of an *a posteriori* error estimate with respect to the quantity of interest $J\colon V \to \mathbb{R}$ of the form

$$J(u) - J(\tilde{u}_h) = \eta_h + \eta_m. \tag{1.3}$$

Here, η_h and η_m denote error estimators which can be evaluated from the computed discrete solution $\tilde{u}_h$, where η_h assesses the error due to the finite element discretization and η_m the error due to the inexact solution of the discrete equations.

The outline of the paper is as follows: In Section 2, we describe the finite element discretization of problem (1.2) and develop goal-oriented *a posteriori* error estimates for the discretization as well as the iteration errors. Section 3 discusses the practical evaluation of these error estimates and the implementation of the resulting adaptation strategies. The numerical results presented in Section 4 demonstrate the efficiency and reliability of the proposed method for two prototypical scalar model problems. The last two sections are devoted to the extension of our theory to different types of saddle point problems. Section 5 presents results for the approximation of the Stokes equations in fluid mechanics and Section 6 for the solution of the so-called KKT system (first-order optimality condition) of a linear-quadratic optimization problem.

2. *A posteriori* error analysis for a linear model problem

In this section, we discuss the discretization of problem (1.2) by the Galerkin finite element method. Based on this discretization, we derive the discrete equations which will be solved by iterative schemes, particularly by the multigrid method.

We consider the discretization of problem (1.2) with usual bi-/trilinear H^1-conforming finite elements as explained in the standard literature; see for example [17]. To this end, we consider two- or three dimensional shape-regular meshes $\mathbb{T}_h = \{K\}$ consisting of (convex) quadrilateral or hexahedral cells K, which constitute a nonoverlapping covering of the computational domain Ω. The discretization parameter h is defined as a cellwise constant function by setting $h|_K := h_K$ with the diameter h_K of the cell K. On the mesh $\mathbb{T}_h$, we construct a conforming finite element space $V_h \subset V$ in a standard way:

$$V_h := \left\{ v_h \in V, v_h|_K \in Q^1(K) \text{ for } K \in \mathbb{T}_h \right\}.$$

Here, $Q^1(K)$ consists of shape functions obtained via iso-parametric transformations of bi/trilinear polynomials in $\widehat{Q}^1(\widehat{K})$ defined on the reference cell $\widehat{K} = (0,1)^d$.

With these preliminaries, we formulate the following approximation for (1.2): Find $u_h \in V_h$ such that

$$a(u_h, \varphi_h) = (f, \varphi_h) \quad \forall \varphi_h \in V_h. \tag{2.1}$$

Problem (2.1) is equivalent to the linear system of equations

$$\mathcal{A}\xi = \beta$$

where $u_h = \sum_{i=1}^N \xi_i \varphi_i$ with the usual nodal basis $\{\varphi_i, i = 1,\dots,N\}$ of V_h and the coefficient vector $\xi = (\xi_i)_{i=1}^N$. Further, $\mathcal{A} = (a_{ij})_{i,j=1}^N$ is the $N \times N$ stiffness matrix with entries $a_{ji} = a(\varphi_i, \varphi_j)$ and $\beta = (b_j)_{j=1}^N$, with $b_j = (f, \varphi_j)$, represents the right-hand side.

We introduce the L^2 projection $P_h \colon V \to V_h$ defined by

$$(P_h u, \varphi_h) = (u, \varphi_h) \quad \forall \varphi_h \in V_h \tag{2.2}$$

and the Ritz projection $Q_h \colon V \to V_h$ given by

$$a(Q_h u, \varphi_h) = a(u, \varphi_h) \quad \forall \varphi_h \in V_h. \tag{2.3}$$

Further, we define the discrete operator $A_h \colon V_h \to V_h$ by

$$(A_h v_h, \varphi_h) = a(v_h, \varphi_h) \quad \forall v_h, \varphi_h \in V_h. \tag{2.4}$$

With these notations, we can rewrite equation (2.1) equivalently in the form

$$A_h u_h = P_h f. \tag{2.5}$$

Below, we will consider hierarchies of meshes $\mathbb{T}_j := \mathbb{T}_{h_j}$, $j = 0,\dots,L$, with mesh size parameters h_j. Accordingly, the notation $V_j := V_{h_j}$, $u_j := u_{h_j}$, $P_j := P_{h_j}$, $Q_j := Q_{h_j}$, and $A_j := A_{h_j}$ will be used.

In the next section, we first establish an *a posteriori* error representation with respect to a linear functional for the error arising due to the finite element discretization alone. After that, we derive the error representation which assesses the influence of the discretization error and the error occurring due to the inexact solution of the discrete equations.

2.1. Estimation of the discretization error

We consider the control of the error with respect to some quantity of interest $J(u)$, which is assumed to be given in terms of a linear functional $J : V \to \mathbb{R}$. To this end, we introduce the following continuous dual problem: Find $z \in V$ such that

$$a(\varphi, z) = J(\varphi) \quad \forall \varphi \in V. \tag{2.6}$$

Using Galerkin orthogonality and the definition of the dual problem (2.6), we obtain the error identity for $e_h := u - u_h$

$$\begin{aligned} J(e_h) &= a(e_h, z) = a(e_h, z - \hat{z}_h) \\ &= (f, z - \hat{z}_h) - a(u_h, z - \hat{z}_h) =: \rho(u_h)(z - \hat{z}_h) \end{aligned} \tag{2.7}$$

where $\hat{z}_h$ is an arbitrary element of V_h. As usual the residual term $\rho(u_h)(z-\hat{z}_h)$ can be rewritten in terms of local cell and edge residuals of the computed approximation u_h multiplied by local weights involving the dual solution z (see [9] or [2]). The practical evaluation of this residual term will be described below.

Remark 2.1. In the case of nonhomogeneous Dirichlet boundary conditions, $u|_\Gamma = g$, we have the following error representation:

$$J(e_h) = \rho(u_h)(z-\hat{z}_h) - (g-g_h, \partial_n z)_\Gamma, \quad \hat{z}_h \in V_h.$$

Here, g_h is an approximation of g used as boundary condition for u_h, $u_h|_\Gamma = g_h$.

2.2. Estimation of the iteration error

The following analysis concerns the approximative solution of the discrete problems by the usual iterative schemes, such as the Gauß–Seidel method, the conjugate gradient (CG) method, or the multigrid method. The goal is the derivation of an *a posteriori* estimate of the resulting 'iteration error' in terms of the difference $J(u_h) - J(\tilde{u}_h)$.

First, we recall the definition of the multigrid algorithm. We suppose that we are given a sequence of refined grids $\mathbb{T}_j = \mathbb{T}_{h_j}$, $j = 0, 1, \ldots, L$, with corresponding finite element spaces $V_j = V_{h_j}$. We assume the meshes $\mathbb{T}_j$ to become finer with increasing j and the spaces V_j to be nested, $V_j \subset V_{j+1}$. We denote by $S_j : V_j \to V_j$ the smoothing operator on the level j. The grid transfer operations are $p_j^{j+1} : V_j \to V_{j+1}$ (prolongation) and $r_j^{j-1} : V_j \to V_{j-1}$ (restriction). We aim at finding an approximation $\tilde{u}_L \in V_L$ on the finest mesh $\mathbb{T}_L$ to the solution $u_L \in V_L$ of the equation

$$A_L u_L = f_L := P_L f \tag{2.8}$$

using a multigrid algorithm based on the hierarchy of meshes $\mathbb{T}_j$, $j = 0, 1, \ldots, L$. Starting with an initial guess $u_L^{(0)}$, the multigrid process produces a sequence of approximations $\tilde{u}_L = u_L^{(t+1)}$ via the procedure $u_L^{(t+1)} = \mathrm{MG}(L, \gamma, u_L^{(t)}, f_L)$ described in Algorithm 2.1.

Algorithm 2.1. Multigrid cycle $\mathrm{MG}(j, \gamma, u_j^{(t)}, f_j)$:

1: **if** $j = 0$ **then**
2: Solve $A_0 u_0^{(t+1)} = f_0$ exactly.
3: **else**
4: Pre-smoothing $\bar{u}_j^{(t)} := S_j^\nu(u_j^{(t)})$.
5: Compute the residual: $d_j^{(t)} := f_j - A_j \bar{u}_j^{(t)}$.
6: Restrict the residual: $\tilde{d}_{j-1}^{(t)} := r_j^{j-1} d_j^{(t)}$.
7: **for** $r = 1$ **to** γ **do**

Starting with $v_{j-1}^{(0)} := 0$ iterate $v_{j-1}^{(r)} := \mathrm{MG}(j-1, \gamma, v_{j-1}^{(r-1)}, \tilde{d}_{j-1}^{(t)})$.
end for
Apply the correction: $\bar{\bar{u}}_j^{(t)} := \bar{u}_j^{(t)} + p_{j-1}^j \tilde{v}_{j-1}^{(\gamma)}$.
Post-smoothing $u_j^{(t+1)} := S_j^{\mu}(\bar{\bar{u}}_j^{(t)})$.
end if

The parameters ν and μ indicate respectively the number of the pre- and post-smoothing steps. The structure of the multigrid algorithm is determined by the parameter γ. The case $\gamma = 1$ corresponds to the so-called V-cycle and $\gamma = 2$ to the W-cycle.

For comparison, we also consider the Gauß–Seidel method for the nodal-value vector $\xi_L \in \mathbb{R}^{N_L}$ corresponding to the finite element solution $u_L \in V_L$,

$$(\mathscr{L}_L + \mathscr{D}_L)\xi_L^{(t)} = \beta_L - \mathscr{R}_L \xi_L^{(t-1)}, \quad t = 1,2,\dots, \quad \xi_L^{(0)} = \tilde{\xi}_{L-1} \tag{2.9}$$

with the usual splitting $\mathscr{A}_L = \mathscr{L}_L + \mathscr{D}_L + \mathscr{R}_L$, or the conjugate gradient (CG) method (without preconditioning),

$$\xi_L^{(t)} \in \mathbb{R}^{N_L}: \quad \|\beta_L - \mathscr{A}_L \xi_L^{(t)}\|_{\mathscr{A}^{-1}} = \min_{y_L \in \mathscr{K}_t} \|\beta_L - \mathscr{A}_L y_L\|_{\mathscr{A}^{-1}} \tag{2.10}$$

with the Krylov spaces $\mathscr{K}_t := \mathrm{span}\{\mathscr{I}, \mathscr{A}, \dots, \mathscr{A}_L^{t-1}\}$, on the different mesh levels. All three iterative methods yield approximative discrete solutions on the fines mesh $\mathbb{T}_L$ which are denoted by $\tilde{u}_L \in V_L$.

To derive an error estimator which includes the error due to the inexact solution of the discrete problems, we replace the 'exact' discrete solution u_L on the current finest mesh $\mathbb{T}_L = \mathbb{T}_{h_L}$ in (2.7) by the computed discrete solution $\tilde{u}_L$. When doing so, an additional term appears in the error estimate which includes the discrete residuals $R_j(\tilde{u}_L) \in V_j$ defined by

$$R_j(\tilde{u}_L) := P_j(f_L - A_L \tilde{u}_L).$$

Proposition 2.1. *Let $u \in V$ be the solution of problem* (1.1) *and $\tilde{u}_L \in V_L$ the approximative finite element solution of the discrete problem* (2.8) *on the finest mesh $\mathbb{T}_L$. Then, we have the following general representation for the error $e := u - \tilde{u}_L$:*

$$J(e) = \rho(\tilde{u}_L)(z - \hat{z}_L) + \rho(\tilde{u}_L)(\hat{z}_L). \tag{2.11}$$

If the multigrid method has been used the following refined representation holds for the iteration residual:

$$\rho(\tilde{u}_L)(\hat{z}_L) = \sum_{j=1}^{L} (R_j(\tilde{u}_L), \hat{z}_j - \hat{z}_{j-1}) + (R_0(\tilde{u}_L), \hat{z}_0). \tag{2.12}$$

Here, $\hat{z}_j \in V_j$, $j = 0,1,\dots,L$, can be chosen arbitrarily and the residual $\rho(\tilde{u}_L)(\cdot) = (f,\cdot) - a(\tilde{u}_L,\cdot)$ is as defined in (2.7).

Proof. We again consider the continuous dual problem (2.6). There holds

$$\begin{aligned} J(e) &= a(e,z) = a(e,z-\hat{z}_L) + a(e,\hat{z}_L) \\ &= (f,z-\hat{z}_L) - a(\tilde{u}_L,z-\hat{z}_L) + (f,\hat{z}_L) - a(\tilde{u}_L,\hat{z}_L) \\ &= \rho(\tilde{u}_L)(z-\hat{z}_L) + \rho(\tilde{u}_L)(\hat{z}_L). \end{aligned} \tag{2.13}$$

This is the asserted error representation (2.11). The first term on the right-hand side corresponds to the discretization error. If the multigrid method has been used, the second term corresponding to the iteration error can be rewritten in the form

$$\rho(\tilde{u}_L)(\hat{z}_L) = \sum_{j=1}^{L} \big\{ (f,\hat{z}_j - \hat{z}_{j-1}) - a(\tilde{u}_L,\hat{z}_j - \hat{z}_{j-1}) \big\} + \big\{ (f,\hat{z}_0) - a(\tilde{u}_L,\hat{z}_0) \big\}. \tag{2.14}$$

Since $V_j \subset V_L$ for $j \leqslant L$, we observe by the definitions of Q_j, P_j, and A_j that for $\varphi_j \in V_j$ there holds

$$(f,\varphi_j) - a(\tilde{u}_L,\varphi_j) = (P_j f,\varphi_j) - (A_j Q_j \tilde{u}_L,\varphi_j)$$

Further, by means of the identity $A_j Q_j = P_j A_L$ for $j \leqslant L$, we have

$$(P_j f,\varphi_j) - (A_j Q_j \tilde{u}_L,\varphi_j) = (P_j(f - A_L\tilde{u}_L),\varphi_j) = (R_j(\tilde{u}_L),\varphi_j).$$

Using these identities for $\varphi_j = \hat{z}_j - \hat{z}_{j-1}$ and $\varphi_0 = \hat{z}_0$ in (2.14) completes the proof of the proposition. □

The error representation (2.11) can be used for approximative solutions $\tilde{u}_L$ obtained by any solution process. Below, we will describe how this can be used for designing automatic stopping criteria for the iteration on the finest mesh level depending on the actual discretization error. We emphasize that the choice of the dual weights in the iteration error representation depends on the weights in the discretization error.

Remark 2.2. The result of Proposition 2.1 does not depend on the special form of the multigrid cycle, i.e., V-, W- or F-cycles are allowed. Moreover, there is no restriction concerning the application of pre/post-smoothing and the choice of the smoother used in the multigrid method.

The error representation (2.12) for the multigrid method exploits the structure of this 'optimal' iteration method. This allows us also to tune the smoothing iteration on the several mesh levels in order to get an easier balancing with the discretization error. The effectivity of this process is demonstrated by the examples presented below. In [7] it has been shown that in the case of the canonically chosen grid transfer operations,

$$p_j^{j+1} = \mathrm{id}, \qquad r_j^{j-1} = P_{j-1}$$

and only pre-smoothing is used, the discrete residuals $R_j(\tilde{u}_L)$ can be identified with the multigrid residuals $R_j(\tilde{v}_j)$ obtained in the course of the correction process from the equation $A_j v_j = P_j d_{j+1}$,

$$\begin{aligned} R_j(\tilde{u}_L) &= P_j(f_L - A_L\tilde{u}_L) \\ &= P_j f_L - P_j A_L S_L^{\nu}(\tilde{u}_L^{(0)}) - P_j A_L p_{L-1}^{L}\tilde{v}_{L-1} \\ &= P_j(d_L - A_{L-1}\tilde{v}_{L-1}) \\ &= P_j d_L - P_j A_{L-1} S_{L-1}^{\nu}(\tilde{v}_{L-1}^{(0)}) - P_j A_{L-1} p_{L-2}^{L-1}\tilde{v}_{L-2} \\ &\vdots \\ &= P_j(d_{j+2} - A_j\tilde{v}_{j+1}) \\ &= P_j d_{j+2} - P_j A_{j+1} S_{j+1}^{\nu}(\tilde{v}_{j+1}^{(0)}) - P_j A_{j+1} p_j^{j+1}\tilde{v}_j \\ &= P_j(d_{j+1} - A_j\tilde{v}_j) = R_j(\tilde{v}_j). \end{aligned} \tag{2.15}$$

This shows that the discrete residuals can be evaluated on the respective grid levels $\mathbb{T}_j$ without explicitly referring to the fine-grid solution $\tilde{u}_L$.

Corollary 2.1. Assume the grid transfer operations in the multigrid algorithm are chosen canonically and the multigrid residual $R_0(v_0)$ on the coarsest level vanishes. Then, under the conditions of Proposition 2.1, the following error representation holds:

$$J(e) = \rho(\tilde{u}_L)(z - \hat{z}_L) + \sum_{j=1}^{L}(R_j(\tilde{v}_j), \hat{z}_j - \hat{z}_{j-1}). \tag{2.16}$$

Proof. The assertion follows by Proposition 2.1 and identity (2.15). □

Remark 2.3. With a similar argumentation as in Proposition 2.1 and Corollary 2.1, one can easily verify the following result in the case of nonhomogeneous Dirichlet data, $u|_\Gamma = g$. There holds

$$J(e) = \rho(\tilde{u}_L)(z - \hat{z}_L) - (g - g_L, \partial_n z)_\Gamma + \sum_{j=1}^{L}(R_j(\tilde{v}_j), \hat{z}_j - \hat{z}_{j-1}) \tag{2.17}$$

where $g_L \in V_L$ are the Dirichlet boundary conditions of u_L, $u_L|_\Gamma = g_L$.

3. Practical realization

In this section, we describe the numerical evaluation of the *a posteriori* estimator for the discretization error. We particularly consider the discretization with piecewise bilinear finite elements on quadrilateral meshes in two space dimensions.

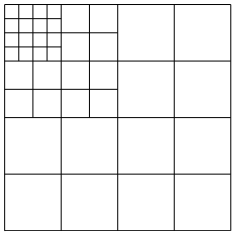

Figure 1. Patched mesh with hanging nodes.

3.1. Evaluation of the estimator for the discretization error

The representation of the discretization error in (2.7) involves interpolation errors, since we set $\hat{z}_h := I_h z$. For their approximation, we discretize the continuous dual problem (2.6) with finite elements accordingly to the primal problem (2.1). Thus, the discretized dual problem reads: Find $z_h \in V_h$ such that

$$a(\varphi_h, z_h) = J(\varphi_h) \quad \forall \varphi_h \in V_h.$$

We solve this problem by a multigrid method on the same grid as that used for the primal problem. To be more precise, we carry out one multigrid iteration for solving the primal problem and then we execute one multigrid iteration for solving the dual problem. With these computed solutions, we evaluate the error estimators. This alternating solving will be continued till the stopping criterion for the multigrid iteration is achieved. For solving the discrete dual problem, there is no need to assemble a new stiffness matrix. The matrix corresponding to the primal problem can simply be transposed.

We approximate the interpolation errors using the computed approximation of the dual problem $\tilde{z}_h$ and patchwise interpolations into higher-order finite element spaces. To this end, we introduce the linear operator Π_h, which maps the computed solution to the approximations of the interpolation errors: $z - I_h z \approx \Pi_h \tilde{z}_h$. In the present case of discretization by bilinear finite elements, Π_h is chosen as

$$\Pi_h := I_{2h}^{(2)} - \mathrm{id}, \qquad I_{2h}^{(2)} : V_h \to V_{2h}^{(2)}.$$

The piecewise biquadratic interpolation $I_{2h}^{(2)}$ can easily be computed if the underlying mesh provides a patch structure. That is, one can always combine four adjacent cells to a macro cell on which the biquadratic interpolation can be defined. An example of such a patched mesh is shown in Fig. 1.

We obtain the following computable estimator for the discretization error:

$$\eta_h := \rho(\tilde{u}_h)(\Pi_h \tilde{z}_h) = (f, \Pi_h \tilde{z}_h) - a(\tilde{u}_h, \Pi_h \tilde{z}_h).$$

This error estimator is used for controlling the discretization error and for steering mesh refinement. For the latter purpose, the error estimator η_h must be localized

to cellwise contributions in order to obtain error indicators to be used within an adaptive algorithm. To this end, we consider the nodal basis $\{\varphi_i, i = 1,2,\dots,N\}$ of V_h. Letting $\tilde{Z}$ denote the coefficient vector of $\tilde{z}_h$, we have the representation

$$\tilde{z}_h = \sum_{i=1}^{N} \varphi_i \tilde{Z}_i.$$

We can rewrite the error estimator as

$$\eta_h = \langle \Psi, \tilde{Z} \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidian inner product on $\mathbb{R}^N$ and Ψ is given by

$$\Psi_i := \rho(\tilde{u}_h)(\Pi_h \varphi_i).$$

However, a direct localization of this term leads to an overestimation of the error due to the oscillatory behavior of the residuals, cf. [16]. To avoid this, we employ the approach described in [11]. We introduce a filtering operator $\pi := \mathrm{id} - I_{2h}^h$, where I_{2h}^h is an interpolation operator in the space of bilinear finite elements defined on patches and denote the coefficient vector of the filtered dual solution $\pi\tilde{z}_h$ by $\tilde{Z}^\pi$:

$$\pi\tilde{z}_h = \sum_{i=1}^{N} \varphi_i \tilde{Z}_i^\pi.$$

Then, the properties of π and $I_{2h}^{(2)}$ and the linearity of the residual ρ with respect to the weight imply

$$\eta_h = \langle \Psi, \tilde{Z}^\pi \rangle.$$

A further localization leads to nodewise error indicators $\eta_{h,i}$ given as

$$\eta_{h,i} = \Psi_i \tilde{Z}_i^\pi, \quad i = 1,2,\dots,N.$$

For the mesh refinement these nodewise contributions are shifted to the corresponding cellwise contributions. Then, these error indicators are used to select the cells which have to be refined within the adaptive algorithm. For possible mesh refinement strategies we refer to [9].

3.2. Evaluation of the estimator for the iteration error

Now, we describe the evaluation of the *a posteriori* error estimator for the iteration error, particularly the multigrid error. Furthermore, we concretize the choice of multigrid components for the technical implementation used in the following numerical examples.

For an arbitrary iteration method, we use the general estimator for the iteration error on the finest mesh $\mathbb{T}_L$ given in Proposition 2.1:

$$\eta_m^{(1)} := (f, \tilde{z}_L) - a(\tilde{u}_L, \tilde{z}_L) \tag{3.1}$$

where $\tilde{z}_L$ is the approximative solution of the dual problem on the finest mesh $\mathbb{T}_L$.

In order to evaluate the representation of the multigrid error given in Proposition 2.1 or its refined version (2.16) in Corollary 2.1, we have to approximate the terms $\hat{z}_j$ and $\hat{z}_{j-1}$ which are defined on the different grid levels $\mathbb{T}_j$ and $\mathbb{T}_{j-1}$. One possibility is to store the calculated dual solutions on each level. Let us assume that we want to calculate $\hat{z}_j - \hat{z}_{j-1}$ for one fixed j. Then, we prolongate the approximated dual solution $\tilde{z}_{j-1}$ from a coarser level on the finer level j. Since we have nested finite element spaces $V_j \subset V_{j+1}$, we use the canonical embedding $p_{j-1}^j = \mathrm{id}$ as the prolongation operator in the multigrid method. By this means, we approximate

$$\hat{z}_j - \hat{z}_{j-1} \approx \tilde{z}_j - p_{j-1}^j \tilde{z}_{j-1},$$

and obtain the following approximative estimator for the multigrid error:

$$\eta_m^{(2)} := \sum_{j=1}^{L} (R_j(\tilde{v}_j), \tilde{z}_j - p_{j-1}^j \tilde{z}_{j-1}) \tag{3.2}$$

where $\tilde{v}_j$ is the approximative solution of the defect equation on mesh $\mathbb{T}_j$.

There is still an alternative way of estimating the multigrid error using the identity (2.16). The computed dual solution $\tilde{z}_L$ on the finest mesh level is restricted to the lower mesh levels. Defining the functions $r_L^j \tilde{z}_L := r_{j+1}^j r_{j+2}^{j+1} \cdots r_L^{L-1} \tilde{z}_L$ for $0 \leqslant j < L$, the dual weights are approximated like $\hat{z}_j - \hat{z}_{j-1} \approx r_L^j \tilde{z}_L - p_{j-1}^j r_L^{j-1} \tilde{z}_L$. On the finest level we compute the difference $\tilde{z}_L - p_{L-1}^L r_L^{L-1} \tilde{z}_L$. Thus, we obtain the following *a posteriori* error estimator for the multigrid error:

$$\eta_m^{(3)} := \sum_{j=1}^{L-1} (R_j(\tilde{v}_j), r_L^j \tilde{z}_L - p_{j-1}^j r_L^{j-1} \tilde{z}_L) + (R_L(\tilde{v}_L), \tilde{z}_L - p_{L-1}^L r_L^{L-1} \tilde{z}_L). \tag{3.3}$$

The approximations $r_L^j \tilde{z}_L$ are defined using the restriction operators r_{j+1}^j, which in the considered situation are chosen as L^2-projections on the grid levels $\mathbb{T}_j$.

In numerical experiments it has turned out that all three iteration error estimators $\eta_m^{(i)}$, $i = 1, 2, 3$, are equally efficient. Therefore, in all the numerical tests involving the multigrid algorithm, we have employed the iteration error estimator $\eta_m^{(2)}$, which also allows us to adapt the smoothing iterations on the different mesh levels.

3.3. Adaptive algorithm

We propose an adaptive algorithm where the discretization and multigrid errors are balanced. That is, we carry out the multigrid iteration until the following relation holds:

$$|\eta_h| \approx |\eta_m|.$$

Moreover, we use the additional information provided by the multigrid error estimator $\eta_m^{(2)}$ and allow the number of smoothing steps to vary over the different mesh

levels in order to reduce the amount of work. In the following, we denote by ν_l and μ_l the number of pre- and post-smoothing steps, respectively, on mesh level l in the multigrid method. On the levels with biggest error contribution, we perform four pre- and post-smoothing steps, while only one pre- and post-smoothing step is used otherwise.

Assuming that we want to compute the value of the quantity of interest up to a given accuracy TOL, we follow the adaptive process given by Algorithm 3.1.

Algorithm 3.1. Adaptive algorithm:

Choose an initial discretization $\mathbb{T}_{h_0}$ and set $l = 0$.
loop
  Set $t = 1$.
  repeat
    if $t = 1$ **then**
      for $j = 0$ to l **do**
        Set $\nu_j = 1, \mu_j = 1$.
      end for
    end if
    Apply one multigrid cycle to the problem $A_l u_l = f_l$.
    Set $t = t + 1$.
    Evaluate the estimators $\eta^{(2)}_{m_l}$ and η_{h_l}.
    According to the error indicators on the different levels,

$$(R_j(\tilde{v}_j), \tilde{z}_j - p^j_{j-1}\tilde{z}_{j-1})$$

    determine the subset of levels $I = \{i_1, \ldots, i_n\}$ with the biggest contribution to the error estimator and increase the number of smoothing steps by
    if $t > 1$ **then**
      for $j = 1$ to n **do**
        Set $\nu_{i_j} = 4, \mu_{i_j} = 4$.
      end for
    end if
  until $|\eta_{m_l}| \leqslant c|\eta_{h_l}|$
  if $|\eta_{h_l} + \eta_{m_l}| \leqslant \mathrm{TOL}$ **then**
    stop
  end if
  Refine the mesh $\mathbb{T}_{h_l} \to \mathbb{T}_{h_{l+1}}$ accordingly to the size of the error indicators $\eta_{h_l,i}$.
  Interpolate the previous solution $\tilde{u}_l$ on the current mesh $\mathbb{T}_{h_{l+1}}$.
  Increment l.
end loop

To start the multigrid algorithm, we use the values from the computation on the previous mesh level as starting data. This allows us to avoid unnecessary iterations

on current mesh.

Furthermore, we use an equilibration factor $c \in (0,1)$ for comparing the two estimators η_m and η_h: $|\eta_m| \leqslant c|\eta_h|$. This ensures also that the local mesh refinement results from the value of the discretization error estimator. For the numerical tests following below, we have chosen the factor c as 0.1. Selecting a smaller value does not improve the accuracy of the computed value very much but increases the number of the multigrid iterations. A greater value, however, can affect the local mesh refinement.

As described in Algorithm 3.1, we have to evaluate the error estimators η_{h_l} and η_{m_l} in every multigrid step. In order to reduce the computational work, we propose the following strategy for the adaptive algorithm: After solving the discrete equations via multigrid on the mesh $\mathbb{T}_{h_l}$, we save the value η_{h_l} by setting $\eta_{\text{old}} := \eta_{h_l}$. On the next finer mesh $\mathbb{T}_{h_{l+1}}$, we do not evaluate the discretization error estimator $\eta_{h_{l+1}}$ until

$$|\eta_{m_{l+1}}| \leqslant c|\eta_{\text{old}}|. \tag{3.4}$$

Then, we save the new value $\eta_{h_{l+1}}$ in η_{old}. In the next multigrid iteration condition (3.4) will be verified again. Thus, we reduce the number of evaluations of the discretization error estimator on each mesh. In the numerical tests presented below the error estimator η_h has been evaluated at most twice on every mesh.

4. Numerical examples: scalar model problems

In this section, we demonstrate the efficiency and reliability of the proposed adaptive algorithm. We compare the adaptive multigrid method described above with a multigrid method employing a residual based stopping criterion such as commonly used for iterative methods. To this end, we replace the adaptiv stopping criterion in the multigrid solver by requiring that the initial multigrid residual is to be reduced by a factor of 10^{-11}. The discretization error estimator will still be used for the construction of locally refined meshes and the error control. We denote this algorithm by MG I and the adaptive multigrid algorithm by MG II. Further, we show the results obtained by using the Gauß–Seidel and the CG method for computing the discrete solutions on the different mesh levels.

In the tables below, the following notation is used for the discretization error $e_h := u - u_h$:

$$\begin{aligned} J(e_h) &\quad \text{'exact' functional discretization error,} \\ \eta_h &\quad \text{estimator of discretization error,} \\ I_{\text{eff}}^h = \left|\frac{\eta_h}{J(e_h)}\right| &\quad \text{effectivity index of discretization error estimator,} \end{aligned}$$

for the iteration error $e_m := u_h - \tilde{u}_h$:

$$\begin{aligned} J(e_m) &\quad \text{'exact' functional iteration error,} \\ \eta_m^{(1)} &\quad \text{estimator of general iteration error,} \\ \eta_m^{(2)} &\quad \text{first special estimator of multigrid iteration error,} \\ \eta_m^{(3)} &\quad \text{second special estimator of multigrid iteration error,} \\ I_{\text{eff}}^{m,j} := \left|\frac{\eta_m^{(j)}}{J(e_m)}\right| &\quad \text{effectivity index of iteration error estimators,} \end{aligned}$$

and for the total error $e := u - \tilde{u}_h$:

$$I_{\text{eff}}^{\text{tot}} := \left|\frac{\eta_h + \eta_m}{J(e)}\right| \quad \text{effectivity index of total error estimator.}$$

In order to access the 'exact' iteration error $J(e_m)$, we solve the discrete equations on each mesh additionally by the algorithm MG I using the ILU-iteration as smoother and requiring that the multigrid residual to be reduced down to round-off error level 10^{-15}. The 'exact' value of $J(e_h)$ is obtained by using an approximation to u on a very fine mesh.

4.1. Example 1

The first numerical test examines the sharpness and practical relevance of the *a posteriori* error estimators derived above on locally refined meshes. We consider the singularly perturbed elliptic problem

$$-\varepsilon\Delta u + u = 1 \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega \tag{4.1}$$

on the two-dimensional unit square $\Omega := (0,1)^2$ with $\varepsilon = 10^{-4}$. As quantity of interest, we take the mean value

$$J(u) := |\Omega|^{-1} \int_\Omega u(x)\,\mathrm{d}x.$$

In this case, boundary layers occur and the meshes resulting from the adaptive algorithm are strongly refined near the boundary (see Fig. 2). Due to the particular choice of the right hand side $f \equiv 1$ and the functional $J(\cdot)$ the dual solution z coincides with the primal solution u. We calculate the corresponding discrete solutions $\tilde{u}_L$ and $\tilde{z}_L$ using the V-cycle, in the first set of tests, with four Jacobi-steps for pre- and post-smoothing, i.e., in the multigrid Algorithm 3.1, we set $\gamma = 1$ and $\nu = \mu = 4$. These tests are performed on the isotropic meshes shown in Fig. 2. The anisotropic meshes are used for the test of the multigrid algorithm with adaptively chosen smoothing separately on the different mesh levels. On such meshes the

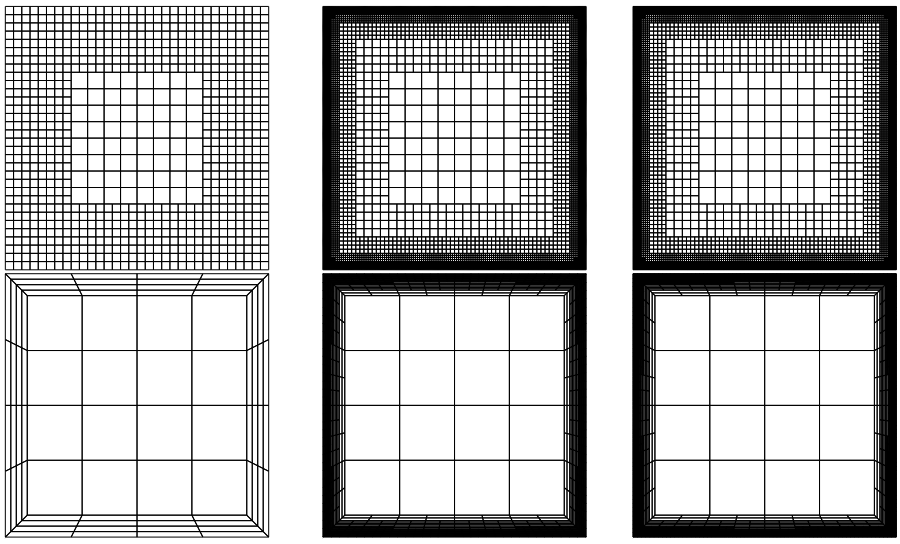

Figure 2. Example 1: Isotropic and anisotropic coarse initial meshes and locally refined meshes.

proper choice of smoothing is particularly critical for the efficiency of the multigrid method.

Table 1 demonstrates the effectivity of the different error estimators $I_{\text{eff}}^{m,j}$ defined above for the multigrid iteration on a coarse and on a very fine mesh. All three estimators turn out to be surprisingly efficient even on coarser meshes. This observation has also been made for the other test problems described below and will therefore not be repeated there.

Next, Table 2 shows the development of the discretization and multigrid errors and the *a posteriori* error estimators from a solution process on a very fine mesh with 341 721 knots. This table also demonstrates the sufficient quality of the error estimators. We note that we would have stopped the multigrid iteration in the adaptive algorithm already after one respectively two steps. Carrying out the multigrid iteration beyond this point does not significantly improve the obtained solution. This shows that the stopping criterion in the multigrid method is efficient. Tables 3 and 4 show the corresponding results for the Gauß–Seidel and the CG iteration. Also in these cases the stopping criterion turns out to be efficient. These two primitive iterative methods have been tested only on rather coarse meshes because of there extremely slow convergence on finer meshes.

Next, we demonstrate the interaction of the two error estimators for the discretization and iteration errors by showing the full history of the adaptive algorithm (Algorithm 4.1) for MG I and MG II in Tables 5 and 6, respectively. We see that only a few multigrid iterations are sufficient to reduce the iteration error below the discretization error level.

Finally, we test the effectivity of the multigrid error estimator $\eta_m^{(2)}$ with respect

Table 1.
Example 1: Effectivity of the different error estimators for the multigrid iteration on a coarse mesh with 913 knots and on a very fine mesh with 341 721 knots.

	mesh with 913 cells			mesh with 341 721 cells		
It	$I_{\text{eff}}^{m,1}$	$I_{\text{eff}}^{m,2}$	$I_{\text{eff}}^{m,3}$	$I_{\text{eff}}^{m,1}$	$I_{\text{eff}}^{m,2}$	$I_{\text{eff}}^{m,3}$
1	1.00	1.00	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00	1.00	1.00

Table 2.
Example 1: Effectivity of error estimators on a mesh with 341 721 knots.

It	$J(e_h)$	η_h	I_{eff}^{h}	$J(e_m)$	$\eta_m^{(2)}$	$I_{\text{eff}}^{m,2}$
1	5.89e-06	5.11e-06	0.87	4.65e-06	4.65e-06	1.00
2	5.89e-06	5.12e-06	0.87	4.41e-07	4.41e-07	1.00
3	5.89e-06	5.12e-06	0.87	7.06e-08	7.06e-08	1.00

Table 3.
Example 1: Gauß–Seidel iteration on a locally refined mesh with 5489 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^{h}	$J(e_m)$	$\eta_m^{(1)}$	$I_{\text{eff}}^{m,1}$	$\|u^{(t)}-u\|_\infty$
1	9.90e-04	8.42e-04	0.85	1.40e-03	1.26e-03	0.90	2.50e-01
5	9.90e-04	8.54e-04	0.86	7.53e-05	7.52e-05	1.00	4.29e-02
10	9.90e-04	8.53e-04	0.86	5.08e-06	5.08e-06	1.00	9.35e-03
15	9.90e-04	8.53e-04	0.86	3.42e-07	3.42e-07	1.00	2.49e-03
20	9.90e-04	8.53e-04	0.86	2.28e-08	2.28e-08	1.00	6.54e-04

Table 4.
CG iteration on a locally refined mesh with 5489 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^{h}	$J(e_m)$	$\eta_m^{(1)}$	$I_{\text{eff}}^{m,1}$	$\|b_L-A_L\xi^{(t)}\|_{A^{-1}}$
1	9.90e-04	8.40e-04	0.85	1.19e-03	1.02e-03	0.86	1.32e-02
5	9.90e-04	8.56e-04	0.86	2.17e-04	2.13e-04	0.98	1.85e-03
10	9.90e-04	8.54e-04	0.86	2.54e-05	2.54e-05	1.00	2.62e-04
15	9.90e-04	8.53e-04	0.86	1.19e-05	1.19e-05	1.00	5.42e-05
20	9.90e-04	8.53e-04	0.86	4.21e-07	4.21e-07	1.00	6.11e-06

Table 5.
Example 1: Iteration with MG I (iteration towards round-off error level).

N	#It	$J(e)$	$\eta_h+\eta_m^{(2)}$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
25	6	2.28e-01	4.03e-02	4.03e-02	4.31e-14	0.18
81	7	1.05e-01	1.82e-02	1.82e-02	6.21e-16	0.17
289	6	4.13e-02	9.46e-03	9.46e-03	3.58e-15	0.23
913	9	1.36e-02	5.07e-03	5.07e-03	9.61e-14	0.37
2 369	9	3.82e-03	2.42e-03	2.42e-03	7.46e-13	0.63
5 489	10	9.90e-04	8.53e-04	8.53e-04	3.38e-14	0.86
11 985	8	2.51e-04	2.41e-04	2.41e-04	6.52e-13	0.96
31 129	9	6.51e-05	6.46e-05	6.46e-05	3.92e-13	0.99
100 697	9	1.73e-05	1.80e-05	1.80e-05	1.80e-12	1.04
341 721	10	5.89e-06	5.12e-06	5.12e-06	8.03e-14	0.87

Table 6.
Example 1: Iteration with MG II (adaptive stopping criterion).

N	#It	$J(e)$	$\eta_h+\eta_m^{(2)}$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
25	1	2.28e-01	4.49e-02	4.13e-02	2.02e-03	0.20
81	1	1.05e-01	1.86e-02	1.82e-02	3.11e-04	0.18
289	1	4.13e-02	9.52e-03	9.47e-03	4.87e-05	0.23
913	1	1.36e-02	5.07e-03	5.07e-03	4.34e-06	0.37
2 369	1	3.82e-03	2.42e-03	2.42e-03	1.47e-08	0.63
5 489	1	9.90e-04	8.57e-04	8.53e-04	3.39e-06	0.87
11 985	1	2.51e-04	2.47e-04	2.41e-04	6.12e-06	0.98
31 129	1	6.51e-05	7.10e-05	6.44e-05	6.50e-06	1.09
100 697	2	1.73e-05	1.96e-05	1.79e-06	1.66e-06	1.13
341 721	2	5.89e-06	5.76e-06	5.12e-06	4.41e-07	0.98

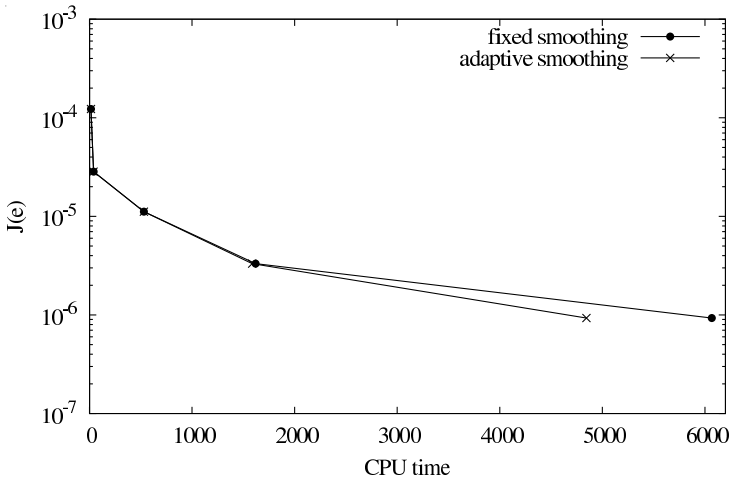

Figure 3. Example 1: Gain in efficiency of the multigrid algorithm by the adaptive choice of smoothing steps on the different mesh levels.

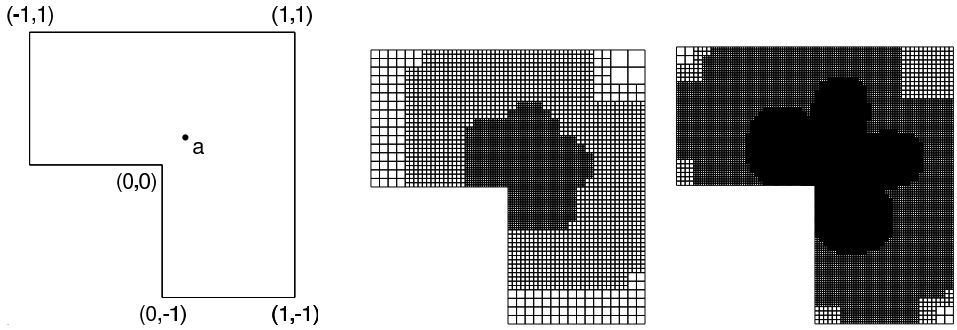

Figure 4. Example 2: Configuration and locally refined meshes.

to the fine tuning of the smoothing process on the different mesh levels according to Algorithm 3.1. To this end, we allow the reduction to only one pre- and post-smoothing step and the use of a damped Jacobi iteration on the basis of the size of the components $(R_j(\tilde{v}_j), \tilde{z}_j - p^j_{j-1}\tilde{z}_{j-1})$ of the error estimator related to the different mesh levels. Figure 3 shows that a significant gain in computing efficiency can be achieved by this fine tuning of the smoothing process. The same observation has been made for the next test example.

4.2. Example 2

The second test demonstrates the efficiency of the adaptive multigrid algorithm. We compare the CPU time needed by the methods MG I and MG II to achieve a prescribed error TOL. We consider the Poisson problem

$$-\Delta u = 1 \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega \tag{4.2}$$

on an L-shaped domain $\Omega \subset \mathbb{R}^2$ (see Fig. 4). As target functional we choose the pointvalue $J(u) := u(a)$ with $a = (0.2, 0.2)$. Since this functional is not in $H^{-1}(\Omega)$ it has to be regularized,

$$J(u) := |B_\varepsilon(a)|^{-1} \int_{B_\varepsilon(a)} u(x)\, \mathrm{d}x = u(a) + O(\varepsilon^2)$$

where $B_\varepsilon(a) := \{z \in \Omega, |z-a| \leqslant \varepsilon\}$ is a ball around the point a with radius $\varepsilon =$ TOL. In this case, the resulting meshes are locally refined as depicted in Fig. 4.

We solve the discrete equations by the multigrid algorithm using a V-cycle and four ILU-steps for pre- and post-smoothing, i.e., as in Example 1, we set $\gamma = 1$ and $\nu = \mu = 4$ in the multigrid algorithm. In order to test the performance of the adaptive multigrid algorithm, we consider the CPU time needed for achieving the error tolerance TOL $= 5 \times 10^{-7}$. Figure 5 depicts the CPU times for both algorithms MG I and MG II. These results show that the adaptive algorithm MG II is twice as fast as the algorithm MG I.

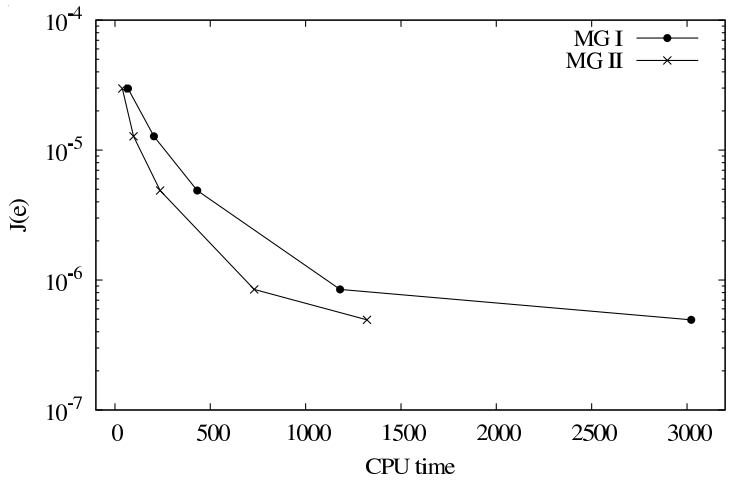

Figure 5. Example 2: Comparison of the CPU time.

Table 7.
Example 2: Iteration with MG I (iteration towards round-off error level).

N	#It	$J(e)$	$\eta_h+\eta_m^{(2)}$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
65	3	8.85e-03	1.14e-02	1.14e-02	5.04e-15	1.29
225	5	4.06e-03	1.57e-03	1.15e-03	6.01e-14	0.39
721	6	1.16e-03	9.57e-04	9.57e-04	3.95e-14	0.83
1 625	7	4.35e-04	2.26e-04	2.26e-04	4.70e-14	0.52
4 573	8	1.43e-04	9.95e-05	9.95e-05	7.71e-13	0.70
11 565	8	5.50e-05	2.98e-05	2.98e-05	1.67e-12	0.54
31 077	10	1.85e-05	1.28e-05	1.28e-05	6.33e-13	0.70
67 669	9	5.94e-06	4.89e-06	4.89e-06	2.67e-12	0.82
174 585	10	8.47e-07	2.00e-06	2.00e-06	1.79e-12	2.35
427 185	10	4.94e-07	7.63e-07	7.63e-07	1.37e-12	1.54

Table 8.
Example 2: Iteration with *MG II* (adaptive stopping criterion).

N	#It	$J(e)$	$\eta_h+\eta_m^{(2)}$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
65	1	8.85e-03	1.14e-02	1.14e-02	9.36e-06	1.29
225	1	4.06e-03	1.67e-03	1.58e-03	9.42e-05	0.41
721	2	1.16e-03	9.58e-04	9.57e-04	1.35e-06	0.83
1 625	1	4.35e-04	2.44e-04	2.26e-04	1.89e-05	0.56
4 573	2	1.43e-04	1.01e-04	9.95e-05	1.28e-06	0.70
11 565	2	5.50e-05	3.04e-05	2.98e-05	6.43e-07	0.55
31 077	2	1.85e-05	1.40e-05	1.28e-05	1.23e-06	0.76
67 669	2	5.94e-06	5.36e-06	4.89e-06	4.71e-07	0.90
174 585	3	8.47e-07	2.05e-06	2.00e-06	5.04e-08	2.41
427 185	3	4.94e-07	8.04e-07	7.63e-07	4.07e-08	1.62

Table 9.
Example 2: Gauß–Seidel iteration on locally refined mesh with 721 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(1)}$	$I_{\text{eff}}^{m,1}$	$\|u_L^{(t)} - u_L\|_\infty$
1	1.16e-03	9.71e-04	0.84	2.72e-03	2.47e-03	0.91	6.91e-02
10	1.16e-03	9.42e-04	0.81	1.68e-03	1.65e-03	0.98	4.21e-02
20	1.16e-03	9.48e-04	0.82	1.21e-03	1.20e-03	0.99	3.66e-02
40	1.16e-03	9.53e-04	0.82	6.86e-04	6.81e-04	0.99	2.78e-02
60	1.16e-03	9.55e-04	0.82	3.90e-04	3.88e-04	1.00	2.10e-02
80	1.16e-03	9.56e-04	0.82	2.21e-04	2.21e-04	1.00	1.59e-02
100	1.16e-03	9.56e-04	0.82	1.25e-04	1.25e-04	1.00	1.19e-02

Table 10.
Example 2: CG iteration on a locally refined mesh with 721 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(1)}$	$I_{\text{eff}}^{m,1}$	$\|b_L - A_L\xi^{(t)})\|_{A^{-1}}$
1	1.16e-03	8.06e-04	0.69	1.54e-03	7.33e-04	0.47	6.50e-02
5	1.16e-03	9.50e-04	0.81	1.85e-03	1.80e-03	0.97	7.57e-03
10	1.16e-03	9.54e-04	0.82	4.60e-04	4.50e-04	0.97	6.34e-03
15	1.16e-03	9.50e-04	0.81	3.10e-05	2.99e-05	0.96	1.17e-03
20	1.16e-03	9.55e-04	0.82	2.17e-05	2.17e-05	0.99	3.08e-04
25	1.16e-03	9.57e-04	0.82	4.12e-06	4.12e-06	0.99	1.01e-04
30	1.16e-03	9.57e-04	0.82	1.09e-06	1.09e-06	1.00	1.32e-05
35	1.16e-03	9.57e-04	0.82	2.72e-07	2.72e-07	0.99	2.02e-06
40	1.16e-03	9.57e-04	0.82	8.22e-09	8.22e-09	1.00	2.31e-07
45	1.16e-03	9.57e-04	0.82	2.05e-09	2.05e-09	1.00	2.46e-08
50	1.16e-03	9.57e-04	0.82	1.93e-10	1.93e-10	1.00	1.94e-09
55	1.16e-03	9.57e-04	0.82	3.40e-12	3.40e-12	1.00	6.85e-11

In Tables 7 and 8, we show the convergence history of the two algorithms MG I and MG II. The results in Table 8 show that in most of the adapted meshes, we only need two or three multigrid iterations to reduce the iteration error well below the discretization error.

Next, we consider the computation of the approximate solution u_h on a fixed locally refined, but still rather coarse, mesh by the Gauß–Seidel and the conjugate gradient (CG) method.

5. Application to saddle point problems: the Stokes equations

Next, we turn to elliptic problems of saddle point form. As a prototypical example, we consider the Stokes equations of fluid mechanics describing the behavior of a

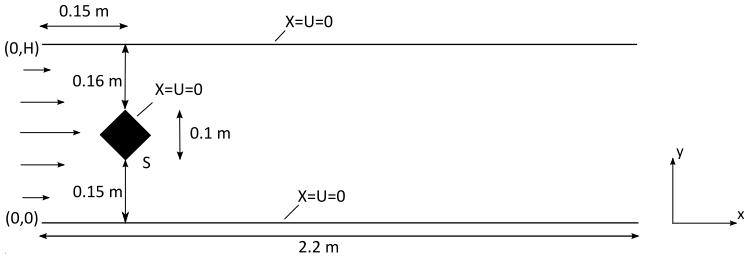

Figure 6. Configuration of the flow example.

creeping incompressible fluid under the action of an external volume force f,

$$\begin{aligned} -\nu\Delta v + \nabla p = f, \quad \nabla\cdot v = 0 \quad \text{in } \Omega \\ v = 0 \ \text{ on } \Gamma_{\text{rigid}}, \quad v = v^{\text{in}} \ \text{ on } \Gamma_{\text{in}}, \quad \nu\partial_n v - pn = 0 \ \text{ on } \Gamma_{\text{out}} \end{aligned} \tag{5.1}$$

on a two-dimensional domain Ω as depicted in Fig. 6 with boundary $\partial\Omega = \Gamma_{\text{rigid}} \cup \Gamma_{\text{in}} \cup \Gamma_{\text{out}}$. The unknowns are the velocity vector v and the scalar pressure p of the fluid. The viscosity parameter $\nu \in \mathbb{R}_+$ is given and the density is normalized to $\rho \equiv 1$. A parabolic velocity profile is prescribed at the inlet Γ_{in}.

The quantity of interest is the drag coefficient of the square obstacle in the flow,

$$J(u) = \frac{2}{U^2 D}\int_S n^T(2\nu\tau(v) - pI)e_1 \,\mathrm{ds}$$

where $u = \{v, p\}$, $\tau(v) := \nabla v + \nabla v^T$ the strain tensor, n the outer normal unit vector along S, D the diameter of the obstacle, $\bar{U}$ the maximum inflow velocity, and e_1 the unit vector in the main flow direction.

The discretization of the Stokes problem (5.1) is based on its standard variational formulation: Find $\{v, p\} \in (\hat{v} + H) \times L$ satisfying

$$\begin{aligned} \nu(\nabla v, \nabla\varphi) - (p, \nabla\cdot\varphi) = (f, \varphi) \quad \forall\varphi \in H \\ (\chi, \nabla\cdot v) = 0 \quad \forall\chi \in L \end{aligned} \tag{5.2}$$

where $H := H_0^1(\Omega)^2$, $L := L^2(\Omega)$, and $\hat{v}$ is a suitable (solenoidal) extension of the boundary data to Ω. The discretization of the saddle point problem (5.2) uses equal-order (bilinear) Q_1 elements for velocity and pressure with additional pressure stabilization for circumventing the usual 'inf-sup' stability condition. The resulting discrete equations read

$$\begin{aligned} \nu(\nabla v_h, \nabla\varphi_h) - (p_h, \nabla\cdot\varphi_h) = (f, \varphi_h) \quad \forall\varphi_h \in H_h \\ (\chi_h, \nabla\cdot v_h) + s_h(\chi_h, p_h) = 0 \quad \forall\chi_h \in L_h \end{aligned} \tag{5.3}$$

where $V_h \subset V$ and $L_h \subset L$ are the finite element subspaces and $s_h(\chi_h, p_h)$ is a stabilizing form. The details of this Galerkin discretization are described, e.g., in [19] or [5]. The discrete saddle point problem (5.3) is solved by a multigrid method

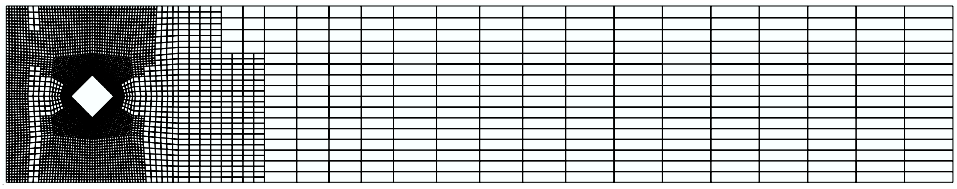

Figure 7. Refined mesh with 4898 knots in the flow example.

using the standard mesh transfer operations and a 'block ILU' iteration as smoother (with four pre- and post-smoothing steps), where at every mesh point the velocity and pressure unknowns are grouped together. By arguments completely analogous to the ones used above for the Poisson model situation the following combined *a posteriori* error estimator for the discretization and iteration error on the finest mesh $\mathbb{T}_L$ can be derived

$$\begin{aligned} J(v,p) - J(\tilde{v}_L,\tilde{p}_L) \;&\approx\; \eta_h + \eta_m^{(2)} \\ &:= \rho(\tilde{v}_L,\tilde{p}_L)(\Pi\tilde{z}_L) + \sum_{j=1}^{L} (R_j(\tilde{x}_j), \tilde{z}_j - p_{j-1}^{j}\tilde{z}_{j-1}) \end{aligned} \tag{5.4}$$

where $R_j(\tilde{x}_j) = P_j((f - A_h S_{j+1}^{\nu}(\tilde{u}_{j+1})) - A_h\tilde{x}_j)$ for $1 \leqslant j < L$ is the residual of the approximated defect correction, $R_L(\tilde{x}_L) := P_L(f - A_h\tilde{u}_L)$ and $\tilde{z}_j$ represents the approximated dual solution on level j. For the practical evaluation of the error estimator (5.4), we use the same strategy as described above in Section 3. Figure 7 shows an adapted mesh for computing the drag coefficient.

Tables 11, 12, and 13 show the results obtained by the MG I iteration (with algebraic stopping criterion) on a coarse and on a very fine mesh and on a sequence of adaptively refined meshes. The corresponding results obtained by the MG II iteration (with adaptive stopping criterion) are shown in Tables 14, 15, and 16. We clearly see the efficiency and reliability of the adaptive stopping criterion. Finally the superiority of the adaptive stopping criterion over the 'algebraic' stopping criterion with respect to CPU time is demonstrated in Fig. 8.

6. Application to PDE-constrained optimization problems

The final application is in linear-quadratic optimal control. Through the first-order necessary optimality condition (the KKT system), we are lead to the discretization of linear elliptic saddle point problems. We consider the prototypical model problem

$$\begin{aligned} &J(u,q) := \tfrac{1}{2}\|u-\bar{u}\|^2 + \tfrac{1}{2}\alpha\|q\|^2 \;\to\; \min \\ &-\Delta u = q + f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{6.1}$$

Table 11.
Iteration with MG I (algebraic stopping criterion) on a locally refined mesh with 708 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(2)}$	$I_{\text{eff}}^{m,2}$
1	5.69e-05	7.69e-05	1.35	3.49e-04	3.54e-04	1.01
2	5.69e-05	9.17e-05	1.61	5.62e-06	5.56e-06	0.99
3	5.69e-05	9.19e-05	1.61	8.34e-07	8.34e-07	1.00
4	5.69e-05	9.19e-05	1.61	6.04e-08	6.04e-08	1.00
5	5.69e-05	9.19e-05	1.61	8.49e-10	8.49e-10	1.00
6	5.69e-05	9.19e-05	1.61	2.16e-10	2.16e-10	1.00
7	5.69e-05	9.19e-05	1.61	4.78e-12	4.78e-12	1.00
8	5.69e-05	9.19e-05	1.61	6.76e-13	6.76e-13	1.00
9	5.69e-05	9.19e-05	1.61	3.83e-14	3.83e-14	1.00
10	5.69e-05	9.19e-05	1.61	8.33e-16	8.33e-16	1.00
11	5.69e-05	9.19e-05	1.61	1.77e-16	1.76e-16	1.01
12	5.69e-05	9.19e-05	1.61	2.03e-18	1.37e-18	1.48

Table 12.
Iteration with MG I (algebraic stopping criterion) on a locally refined mesh with 306 308 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(2)}$	$I_{\text{eff}}^{m,2}$
1	1.85e-07	2.97e-07	1.60	1.18e-06	1.18e-06	1.00
2	1.85e-07	2.97e-07	1.60	1.22e-08	1.22e-08	1.00
3	1.85e-07	2.97e-07	1.60	1.17e-09	1.17e-09	1.00
4	1.85e-07	2.98e-07	1.60	4.04e-11	4.04e-11	1.00
5	1.85e-07	2.98e-07	1.60	5.98e-12	5.98e-12	1.00
6	1.85e-07	2.98e-07	1.60	7.64e-13	7.64e-13	1.00
7	1.85e-07	2.98e-07	1.60	7.99e-14	7.99e-14	1.00
8	1.85e-07	2.98e-07	1.60	6.77e-15	6.75e-15	1.00
9	1.85e-07	2.98e-07	1.60	6.92e-16	6.79e-16	1.02
10	1.85e-07	2.98e-07	1.60	5.80e-17	4.43e-17	1.31
11	1.85e-07	2.98e-07	1.60	2.02e-17	9.35e-19	21.65
12	1.85e-07	2.98e-07	1.60	8.74e-19	1.13e-17	0.08

Table 13.
Iteration with MG I (algebraic stopping criterion).

N	#It	$J(e)$	$\eta_h+\eta_m^{(2)}$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
708	12	5.69e-05	9.19e-05	9.19e-05	2.03e-18	1.61
1 754	9	3.12e-05	2.81e-05	2.81e-05	1.05e-16	0.90
4 898	9	1.83e-05	1.21e-05	1.21e-05	2.20e-15	0.66
11 156	9	1.05e-05	7.01e-06	7.01e-06	9.49e-15	0.67
22 526	10	5.34e-06	3.77e-06	3.77e-06	8.36e-17	0.71
44 874	10	2.75e-06	2.12e-06	2.12e-06	3.39e-16	0.77
82 162	10	1.26e-06	1.09e-06	1.09e-06	4.29e-17	0.86
159 268	11	5.76e-07	6.11e-07	6.11e-07	1.26e-17	1.06
306 308	12	1.85e-07	2.98e-07	2.98e-07	8.74e-19	1.60

Table 14.
Iteration with MG II (adaptive stopping criterion) on locally refined mesh with 708 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(2)}$	$I_{\text{eff}}^{m,2}$
1	5.69e-05	7.69e-05	1.35	3.49e-04	3.54e-04	1.01
2	5.69e-05	9.17e-05	1.61	5.62e-06	5.56e-06	0.99

Table 15.
Iteration with MG II (adaptive stopping criterion) on locally refined mesh with 306 308 knots (starting value taken from the preceding mesh).

It	$J(e_h)$	η_h	I_{eff}^h	$J(e_m)$	$\eta_m^{(2)}$	$I_{\text{eff}}^{m,2}$
1	1.86e-07	2.98e-07	1.60	1.24e-06	1.24e-06	1.00
2	1.86e-07	2.98e-07	1.60	1.31e-08	1.31e-08	1.00

Table 16.
Iteration with MG II (adaptive stopping criterion).

N	#It	$J(e)$	$\eta_h+\eta_m$	η_h	$\eta_m^{(2)}$	$I_{\text{eff}}^{\text{tot}}$
708	2	5.69e-05	9.74e-05	9.17e-05	5.62e-06	1.71
1 754	2	3.12e-05	2.82e-05	2.81e-05	6.81e-08	0.90
4 898	2	1.83e-05	1.21e-05	1.21e-05	1.60e-08	0.66
11 156	2	1.05e-05	7.05e-06	7.01e-06	3.42e-08	0.67
22 526	2	5.34e-06	3.82e-06	3.77e-06	5.48e-08	0.72
44 874	2	2.75e-06	2.16e-06	2.12e-06	4.04e-08	0.78
82 162	2	1.27e-06	1.11e-06	1.09e-06	2.63e-08	0.88
159 268	2	5.76e-07	6.41e-07	6.10e-07	3.07e-08	1.11
306 308	2	1.86e-07	3.10e-07	2.97e-07	1.31e-08	1.67

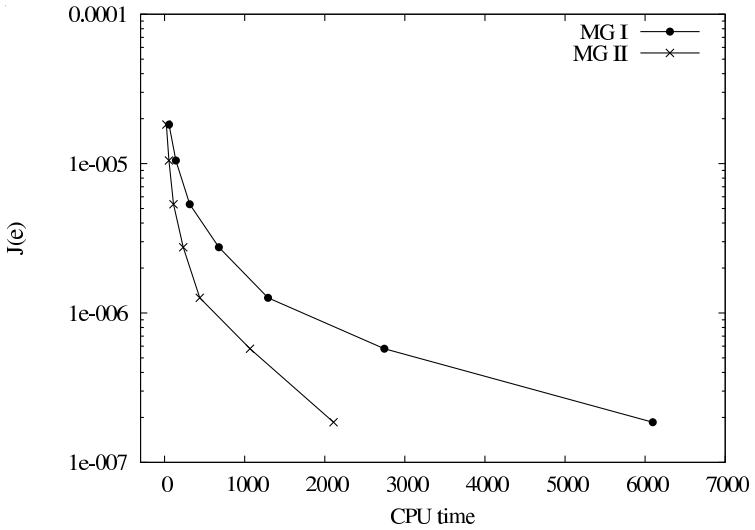

Figure 8. Comparison of the CPU time.

on the two-dimensional unit square $\Omega := (0,1)^2$ with prescribed volume force f and target distribution $\overline{u}$. The regularization parameter is chosen in the range $\alpha = 10^{-3} - 10^{-6}$. The variational formulation of the state equation is as follows: For $q \in Q := L^2(\Omega)$ find $u \in V := H_0^1(\Omega)$ satisfying

$$(\nabla u, \nabla \varphi) - (q, \varphi) = (f, \varphi) \quad \forall \varphi \in V. \tag{6.2}$$

For solving this optimization problem, we use the Euler–Lagrange approach. Introducing the Lagrangian

$$\mathscr{L}(u,q,\lambda) := J(u,q) + (f+q,\lambda) - (\nabla u, \nabla \lambda)$$

with the adjoint variable (Lagrangian multiplier) $\lambda \in V$. Then, the Lagrange principle states that for any optimal solution $\{u,q\} \in V \times Q$ there exists an adjoint solution $\lambda \in V$ such that the triplet $\{u,q,\lambda\}$ is a stationary point of the Lagrangian, i.e., it satisfies the following saddle point system:

$$\begin{aligned} (\nabla \varphi, \nabla \lambda) - (u, \varphi) &= -(\overline{u}, \varphi) \quad \forall \varphi \in V \\ (\chi, \lambda) + \alpha(\chi, q) &= 0 \quad \forall \chi \in Q \\ (\nabla u, \nabla \psi) - (q, \psi) &= (f, \psi) \quad \forall \psi \in V. \end{aligned} \tag{6.3}$$

In strong form this reads like

$$\begin{aligned} -\Delta \lambda - u &= -u^d \quad \text{in } \Omega, \quad \lambda|_{\partial\Omega} = 0 \\ \lambda + \alpha q &= 0 \quad \text{in } \Omega \\ -\Delta u - q &= f \quad \text{in } \Omega, \quad u|_{\partial\Omega} = 0. \end{aligned} \tag{6.4}$$

Using conforming bilinear Q_1 elements for discretizing all three variables $\{u,q,\lambda\}$ in associated finite element subspaces $V_h \subset V$ and $Q_h \subset Q$ leads to the discrete saddle point problems

$$\begin{aligned}(\nabla\varphi_h,\nabla\lambda_h)-(u_h,\varphi_h) &= -(\bar{u},\varphi_h) \quad \forall\varphi_h\in V_h\\ (\chi_h,\lambda_h)+\alpha(\chi_h,q_h) &= 0 \quad \forall\chi_h\in Q_h\\ (\nabla u_h,\nabla\psi_h)-(q_h,\psi_h) &= (f,\psi_h) \quad \forall\psi_h\in V_h.\end{aligned} \tag{6.5}$$

These linear saddle point problems are solved by the multigrid method using a block ILU iteration as smoother, which couples the nodal values of the three unknowns $\{u_h,q_h,\lambda_h\}$ at every mesh point. For measuring the error in this approximation, we choose the cost functional J itself, which seems to be the most natural option.

Proposition 6.1. *Let $\{u,q,\lambda\}\in V\times Q\times V$ be the solution of the KKT system* (6.3) *and $\{\tilde{u}_h,\tilde{q}_h,\tilde{\lambda}_h\}\in V_h\times Q_h\times V_h$ the approximative finite element solution of the discrete KKT system* (6.5) *on the finest mesh $\mathbb{T}_h$. Then, we have the following error representation, in which the first three terms correspond to the discretization error and the last four to the iteration error:*

$$\begin{aligned}J(u,q)-J(\tilde{u}_h,\tilde{q}_h) = {}& \tfrac{1}{2}\tilde{\rho}^*(u-\varphi_h)+\tfrac{1}{2}\tilde{\rho}^q(q-\chi_h)+\tfrac{1}{2}\tilde{\rho}(\lambda-\psi_h)\\ &+\tfrac{1}{2}\tilde{\rho}^*(\varphi_h-\tilde{u}_h)+\tfrac{1}{2}\tilde{\rho}^q(\chi_h-\tilde{q}_h)+\tfrac{1}{2}\tilde{\rho}(\psi_h-\tilde{\lambda}_h)\\ &+\tilde{\rho}(\tilde{\lambda}_h)\end{aligned} \tag{6.6}$$

for arbitrary elements $\varphi_h,\psi_h\in V_h$ and $\chi_h\in Q_h$, with the error residuals

$$\begin{aligned}\tilde{\rho}^*(\cdot) &:= (\tilde{u}_h-\bar{u},\cdot)-(\nabla\cdot,\nabla\tilde{\lambda}_h)\\ \tilde{\rho}^q(\cdot) &:= \alpha(\cdot,\tilde{q}_h)+(\cdot,\tilde{\lambda}_h)\\ \tilde{\rho}(\cdot) &:= (f+\tilde{q}_h,\cdot)-(\nabla\tilde{u}_h,\nabla\cdot).\end{aligned}$$

Proof. We introduce the tensor product spaces $X := V\times Q\times V$ and $X_h := V_h\times Q_h\times V_h$ with elements $x := \{u,q,\lambda\}$ and $x_h := \{u_h,q_h,\lambda_h\}$, respectively. We denote by $\tilde{x}_h := \{\tilde{u}_h,\tilde{q}_h,\tilde{\lambda}_h\}$ the approximative solution of the discretized KKT system obtained by any iterative method. Further, on X, we define the functional $L(x) := \mathscr{L}(u,q,\lambda)$. Hence solving the KKT systems (6.3) and (6.5) is equivalent to determining stationary points $x\in X$ and $x_h\in X_h$ of L:

$$L'(x)(y)=0 \quad \forall y\in X, \qquad L'(x_h)(y_h)=0 \quad \forall y_h\in X_h. \tag{6.7}$$

We will use elementary calculus for the error $e := x-\tilde{x}_h$,

$$L(x)-L(\tilde{x}_h)=\int_0^1 L'(\tilde{x}_h+se)(e)\,\mathrm{d}s$$

and the general error representation for the trapezoidal rule.

$$\int_0^1 f(s)\,\mathrm{d}s = \tfrac{1}{2}\big(f(0)+f(1)\big) + \tfrac{1}{2}\int_0^1 f''(s)s(s-1)\,\mathrm{d}s.$$

Observing that $L'(x)(y) = 0, y \in X$, and $L'''(\cdot) \equiv 0$ for the quadratic functional $L(\cdot)$, we conclude

$$L(x) - L(\tilde{x}_h) = \tfrac{1}{2}L'(\tilde{x}_h)(x-\tilde{x}_h).$$

Hence, recalling the particular structure of the functional $L(\cdot)$ and observing that u satisfies the state equations,

$$\begin{aligned} J(u,q) - J(\tilde{u}_h,\tilde{q}_h) &= L(x) - (f+q,\lambda) + (\nabla u,\nabla\lambda) \\ &\quad - L(\tilde{x}_h) + (f+\tilde{q}_h,\tilde{\lambda}_h) - (\nabla\tilde{u}_h,\nabla\tilde{\lambda}_h) \\ &= L(x) - L(\tilde{x}_h) + (f+\tilde{q}_h,\tilde{\lambda}_h) - (\nabla\tilde{u}_h,\nabla\tilde{\lambda}_h) \\ &= \tfrac{1}{2}L'(\tilde{x}_h)(x-\tilde{x}_h) + \tilde{\rho}(\tilde{\lambda}_h). \end{aligned}$$

Since, for arbitrary $y_h \in X_h$,

$$L'(\tilde{x}_h)(x-\tilde{x}_h) = L'(\tilde{x}_h)(x-y_h) + L'(\tilde{x}_h)(y_h-\tilde{x}_h)$$

and

$$L'(\tilde{x}_h)(\cdot) = \tilde{\rho}^*(\cdot) + \tilde{\rho}^q(\cdot) + \tilde{\rho}(\cdot)$$

the proof is complete. □

Remark 6.1. The choice of the cost functional J for error control may not be considered as appropriate in the present case of a tracking problem where the particular least-squares form of the functional is somewhat arbitrary. Instead one may want to measure the solution accuracy rather in terms of some more relevant quantity depending on control and state, such as for example the. norm $\|q-\tilde{q}_h\|_Q$ of the error in the control. This can be accomplished by utilizing an additional dual problem such as described in [10] and [6].

For the practical evaluation of the error estimator in Proposition 6.1, we again use the strategy described in Section 3. The discretization error is estimated by the estimator

$$\tilde{\eta}_h := \tfrac{1}{2}\tilde{\rho}^*(\Pi_h\tilde{u}_h) + \tfrac{1}{2}\tilde{\rho}^q(\Pi_h\tilde{q}_h) + \tfrac{1}{2}\tilde{\rho}(\Pi_h\tilde{\lambda}_h)$$

where $\Pi_h := I_{2h}^{(2)} - \mathrm{id}$, with the patchwise biquadratic interpolation $I_{2h}^{(2)} : V_h \to V_{2h}^{(2)}$.

Due to the choice of the dual weights in the discretization error estimator the iteration error estimator reduces to

$$\tilde{\eta}_m := \tilde{\rho}_m(\tilde{\lambda}_h).$$

Table 17.
MG II with block ILU smoothing, $\alpha = 10^{-3}$.

N	E	#It	E_h	$\tilde{\eta}_h$	I^h_{eff}	E_m	$\tilde{\eta}_m$	I^m_{eff}
25	9.35e-04	2	9.35e-04	1.83e-03	1.96	1.14e-07	1.97e-07	1.73
81	1.64e-04	2	1.78e-04	2.19e-04	1.22	1.42e-05	1.68e-05	1.18
289	3.75e-05	2	4.16e-05	4.39e-05	1.05	4.13e-06	4.33e-06	1.04
1 089	1.05e-05	2	1.02e-05	1.03e-05	1.01	3.48e-07	3.52e-07	1.01
3 985	2.67e-06	2	2.54e-06	2.55e-06	1.00	1.28e-07	1.28e-07	1.00
13 321	6.65e-07	2	6.48e-07	6.49e-07	1.00	1.63e-08	1.63e-08	1.00
47 201	1.76e-07	2	1.70e-07	1.69e-07	0.99	6.76e-09	6.77e-09	1.00
163 361	4.89e-08	2	4.69e-08	4.68e-08	0.99	1.97e-09	1.97e-09	1.00
627 697	1.23e-08	2	1.21e-08	1.21e-08	0.99	2.13e-10	2.13e-10	1.00

6.1. Numerical example

We consider the optimization problem with $\bar{u} = (2\pi^2)^{-1}(2\pi^2 - 1)\sin(\pi x)\sin(\pi y)$ and the exact solution

$$u = \frac{-1}{2\pi^2}\sin(\pi x)\sin(\pi y), \quad q = \frac{1}{2\alpha\pi^2}\sin(\pi x)\sin(\pi y), \quad \lambda = \frac{-1}{2\pi^2}\sin(\pi x)\sin(\pi y).$$

The forcing term f is accordingly adjusted. For simplicity the discrete state and control spaces are chosen the same, $V_h = Q_h$, using isoparametric bilinear shape functions. In order to access the 'exact' multigrid error, we solve the discrete equations on each mesh additionally by the multigrid method with block ILU smoothing until the initial residual is reduced by a factor 10^{-15}. For this test, we use the multigrid algorithm MG II with the stopping criterion that the iteration error estimator η_m is ten times smaller than the discretization error estimator η_h. Using a similar notation as in Proposition 6.1, $x = \{u, q, \lambda\}, x_h = \{u_h, q_h, \lambda_h\}, \tilde{x}_h = \{\tilde{u}_h, \tilde{q}_h, \tilde{\lambda}_h\}$, we denote the errors by

$$E := J(x) - J(\tilde{x}_h), \quad E_h := J(x) - J(x_h), \quad E_m := J(x_h) - J(\tilde{x}_h)$$

and the effectivity indices by

$$I^h_{\text{eff}} := \left|\frac{\tilde{\eta}_h}{E_h}\right|, \qquad I^m_{\text{eff}} := \left|\frac{\tilde{\eta}_m}{E_m}\right|.$$

First, we solve the discretized KKT system (6.5) by the multigrid method using the V-cycle and again four steps of block-ILU pre- and post-smoothing on each level. The corresponding results for different values of the regularization parameter α are presented in Tables 17–19. The stopping criterion used in the computation turns out to be efficient and reliable.

Finally, in Table 20 we present the results for the same problem with $\alpha = 10^{-3}$ solved by the multigrid method using only one undamped block Jacobi step as smoothing. Obviously for the present problem the simplest block Jacobi smoothing works almost as well as the much more expensive block ILU smoother. Further, the stopping criterion used in the computation is efficient and reliable.

Table 18.
MG II with block ILU smoothing, $\alpha = 10^{-4}$.

N	E	#It	E_h	$\tilde{\eta}_h$	I^h_{eff}	E_m	$\tilde{\eta}_m$	I^m_{eff}
25	1.46e-03	1	1.40e-03	8.22e-03	5.85	5.92e-05	1.18e-04	1.99
81	1.97e-04	2	1.89e-04	4.69e-04	2.47	7.91e-06	9.85e-06	1.24
289	4.25e-05	3	4.20e-05	5.73e-05	1.36	4.49e-07	4.77e-07	1.06
1 089	1.08e-05	3	1.02e-05	1.11e-05	1.09	6.29e-07	6.39e-07	1.01
3 985	2.70e-06	3	2.53e-06	2.60e-06	1.03	1.74e-07	1.75e-07	1.00
13 321	6.63e-07	3	6.19e-07	6.56e-07	1.06	4.40e-08	4.41e-08	1.00
47 201	1.78e-07	3	1.65e-07	1.70e-07	1.02	1.33e-08	1.33e-08	1.00
163 409	5.03e-08	3	4.69e-08	4.69e-08	1.00	3.48e-09	3.48e-09	1.00

Table 19.
MG II with block ILU smoothing, $\alpha = 10^{-5}$.

N	E	#It	E_h	$\tilde{\eta}_h$	I^h_{eff}	E_m	$\tilde{\eta}_m$	I^m_{eff}
25	5.12e-03	1	5.12e-03	6.89e-02	13.45	6.94e-07	1.40e-06	2.03
81	2.35e-04	2	2.45e-04	2.84e-03	11.57	1.00e-05	1.25e-05	1.25
289	6.01e-05	2	4.29e-05	1.81e-04	4.22	1.71e-05	1.80e-05	1.05
1 089	9.48e-06	3	1.02e-05	1.87e-05	1.82	7.87e-07	7.99e-07	1.01
3 985	2.62e-06	3	2.34e-06	3.12e-06	1.33	2.75e-07	2.77e-07	1.00
13 561	5.19e-07	4	5.22e-07	6.95e-07	1.33	2.51e-09	2.52e-09	1.00
47 913	1.52e-07	3	1.59e-07	1.73e-07	1.08	7.41e-09	7.43e-09	1.00
164 217	4.90e-08	3	4.59e-08	4.71e-08	1.02	3.08e-09	3.08e-09	1.00

Table 20.
MG II with block Jacobi smoothing, $\alpha = 10^{-3}$.

N	E	#It	E_h	$\tilde{\eta}_h$	I^h_{eff}	E_m	$\tilde{\eta}_m$	I^m_{eff}
25	9.44e-04	4	1.83e-03	9.35e-04	1.96	1.55e-05	8.99e-06	1.73
81	1.84e-04	5	2.20e-04	1.78e-04	1.23	7.59e-06	6.44e-06	1.18
289	4.36e-05	5	4.40e-05	4.16e-05	1.05	2.04e-06	1.96e-06	1.04
1 089	1.10e-05	4	1.03e-05	1.02e-05	1.01	8.53e-07	8.44e-07	1.01
3 985	2.69e-06	4	2.55e-06	2.56e-06	0.99	1.31e-07	1.30e-07	1.00
13 321	6.94e-07	4	6.47e-07	6.69e-07	0.96	2.51e-08	2.51e-08	1.00
47 201	1.95e-07	4	1.69e-07	1.90e-07	0.88	4.39e-09	4.40e-09	1.00
171 969	7.24e-08	3	4.42e-08	6.93e-08	0.63	3.07e-09	3.10e-09	0.99

References

1. R. E. Bank and T. Dupont, An optimal order process for solving finite element equations. *Math. Comp.* (1981) **36**, 35 – 51.
2. W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations.* Birkhäuser, Basel-Boston-Berlin, 2003.
3. R. Becker, An adaptive finite element method for the Stokes equations including control of the iteration error. In: *Numerical Mathematics and Advanced Applications* (Eds. H. G. Bock et al.), Proc. of ENUMATH 1997, World Scientific, London, 1998, pp. 609 – 620.
4. R. Becker and M. Braack, Multigrid techniques for finite elements on locally refined meshes. *Numer. Linear Algebra Appl.* (2000) **7**, 363 – 379.
5. R. Becker, M. Braack, R. Rannacher, and Th. Richter, Mesh and model adaptivity for flow problems. In: *Reactive Flow, Diffusion, and Transport* (Eds. W. Jäger, R. Rannacher, and J. Warnatz), Springer, Berlin-Heidelberg, 2007, pp. 47 – 75.
6. R. Becker, M. Braack, D. Meidner, R. Rannacher, and B. Vexler, Adaptive finite element methods for PDE-constrained optimal control problems. In: *Reactive Flow, Diffusion, and Transport* (Eds. W. Jäger, R. Rannacher, and J. Warnatz), Springer, Berlin-Heidelberg, 2007, pp. 177 – 205.
7. R. Becker, C. Johnson, and R. Rannacher, Adaptive error control for multigrid finite element methods. *Computing* (1995) **55**, 271 – 288.
8. R. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.* (1996) **4**, No. , 237 – 264.
9. R. Becker and R. Rannacher, An optimal control approach to *a posteriori* error estimation. In: *Acta Numerica* (Ed. A. Iserles), Cambridge University Press, Cambridge, 2001, pp. 1 – 102.
10. R. Becker and B. Vexler, *A posteriori* error estimation for finite element discretization of parameter identification problems. *Numer. Math.* (2004) **96**, 435 – 459.
11. M. Braack and A. Ern, *A posteriori* control of modeling errors and discretization errors. *Multiscale Model. Simul.* (2003) **1**, 221 – 238.
12. J. H. Bramble, *Multigrid Methods*. Pitman Research Notes in Mathematics, Longman, Harlow-New York, 1993, Vol. 294.
13. J. H. Bramble, J. E. Pasciak, J. Wang and J. Xu, Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.* (1993) **60**, 447 – 471.
14. J. H. Bramble and J. E. Pasciak, New estimates for multilevel algorithms including the V-cycle. *Math. Comp.* (1991) **57**, 23 – 45.
15. A. Brandt, Multi-level adaptive solutions to boundary-value problems. *Math. Comp.* (1977) **31**, 333 – 390.
16. C. Carstensen and R. Verfürth, Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM J. Numer. Anal.* (1999) **36**, 1571 – 1587.
17. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems.* Classics Appl. Math., SIAM, Philadelphia, 2002, Vol. 40.
18. W. Hackbusch, *Multigrid Method and Applications*. Springer, Berlin-Heidelberg-New York, 1985.
19. R. Rannacher, Incompressible viscous flow. In: *Encyclopedia of Computational Mechanics* (Eds. E. Stein, R. de Borst, and T. J. R. Hughes), John Wiley, Chichester, 2004, Vol. 3.
20. R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley/Teubner, New York-Stuttgart, 1996.
21. J. Vihharev, *Adaptive Mehrgittermethoden für Finite Elemente-Diskretisierungen.* Diploma thesis, Institute of Applied Mathematics. University of Heidelberg, Heidelberg, 2008.