Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARTIN WEISER

On goal-oriented adaptivity for elliptic optimal control problems

ZIB-Report 09-08 (December 2009)

On goal-oriented adaptivity for elliptic optimal control problems*

Martin Weiser

December 8, 2009

Abstract

The paper proposes goal-oriented error estimation and mesh refinement for optimal control problems with elliptic PDE constraints using the value of the reduced cost functional as quantity of interest. Error representation, hierarchical error estimators, and greedy-style error indicators are derived and compared to their counterparts when using the all-at-once cost functional as quantity of interest. Finally, the efficiency of the error estimator and generated meshes are demonstrated on numerical examples.

AMS MSC 2000: 65N50, 65K10, 49M25
Keywords: optimal control, goal oriented error estimation, adaptive mesh refinement

1 Introduction

Solving optimization problems with PDE constraints numerically with finite elements, one faces two fundamental questions: How large is the error of a computed approximate solution, and what is the coarsest mesh on which a required accuracy can be obtained? Both questions are usually addressed simultaneously with the closely related concepts of a posteriori error estimators and error indicators.

The first question can only be answered if a metric is defined that allows to quantify the distance between approximate and exact solution. Classical error estimators for PDE problems use a general norm, often a Sobolev norm, of the difference of exact and approximate solution. More recently, goal-oriented error estimators focus on the difference of observables, so-called quantities of interest. With the error concept defined, a variety of computational techniques can be applied to estimate the error value [1, 2, 14].

*This work has been supported by the DFG Research Center MATHEON "Mathematics for key technologies" in Berlin.

The second question is most often addressed by adaptively refining the finite element mesh in regions that contribute most to the error value and where a local improvement of the discretization accuracy can be expected to decrease the error value most. This leads to a kind of greedy algorithm for solving the nonlinear approximation problem posed by the second question.

In genuine optimal control problems, the quantity of interest is the cost functional J. Quite naturally, the whole point of solving optimization problems is to obtain a minimal value of J, whereas a general measure of distance between exact and approximate solution is usually of minor practical concern. The equality constraint usually takes the form of a state equation that allows to compute an (at least locally) unique state $y = y(u)$ for every value of the control u. This allows to write the optimal control problem either in all-at-once form as $\min_{y,u} J(y,u)$, where both state and control enter as optimization variable into the cost functional J, or equivalently in reduced form $\min_u \tilde{J}(u) = J(y(u),u)$, with the control as the only optimization variable whereas the state is just an implicitly defined intermediate quantity. For stationary problems, the all-at-once formulation is usually preferred for actual computation. Consequently, starting with the seminal paper by Becker, Kapp, and Rannacher [5], the value $|J(y^h,u^h) - J(y,u)|$ has been used throughout the literature to quantify the distance between the exact solution (y,u) and its computed approximation (y^h,u^h). Goal-oriented error estimation techniques have been developed for a large variety of PDE-constrained optimal control problems, see [2,12] for a survey and the more recent papers [6,10,11,16] for control and state constrained problems.

As pointed out by Vexler [15], there are different problems usually written in form of optimal control problems for which the quantity of interest is not the cost functional, but a different function of the control. The most notable example of such problems are parameter identification problems, where the error in the identified parameters is the natural quantity of interest, or the data mismatch in case a regularization parameter needs to be determined from a discrepancy principle [9]. Monitoring convergence of inexact Newton methods for interior point methods in optimal control problems leads to yet another kind of quantity of interest. Goal-oriented error estimation techniques have been developed for those situations as well [13]. Here, however, we will be concerned with genuine optimal control problems, where the cost functional is actually the quantity of interest.

In many practical applications computing an accurate minimal value of the cost functional is of less importance than obtaining a control that, when applied, leads to a cost functional value that is as small as possible. In other words, in these situations the relevant quantity of interest is $J(y(u^h),u^h)$ instead of $J(y^h,u^h)$. After all, y_h is only an approximation of the state $y(u^h)$ that will be obtained by applying u_h, and the error it introduces into the cost functional value $J(y^h,u^h)$ is of no practical consequence. In this paper we will follow this line of thought and develop a goal-oriented error estimator for $\tilde{J}(u)$ in Section 2.1 with associated error indicators in Section 2.3. The computational concept is formulated using hierarchical error estimators for obtaining an approximate error representation in Section 2.2. Algorithmic considerations are discussed in Section 2.4.

The difference between the proposed approach and previous error estimators is largest whenever y^h incurs a discretization error into the cost functional's value that is essentially independent of the control. Vice versa, if such control-independent discretization errors are missing, no significant difference can be expected. Therefore, we report numerical results for illustrative optimal control problems designed to cover both cases in Section 3.

2 Error Estimation

2.1 Abstract error representation

In order not to hide the concept behind technical details, we consider a rather abstract linear optimal control problem

$$\min_{y\in Y, u\in U} J(y,u) = \langle y, \frac{1}{2}H_{yy}y - b_y\rangle + \langle u, \frac{1}{2}H_{uu}u - b_u\rangle \tag{1}$$

subject to the equality constraint

$$Ay + Bu - b_\lambda = 0. \tag{2}$$

Here, Y and U are Banach spaces, $H_{yy} : Y \to Y^*$ and $H_{uu} : U \to U^*$ are bounded, symmetric, and positive semidefinite, and $A : Y \to Y^*$ has a bounded inverse. $B : U \to Y^*$ is merely continuous, and clearly we require $b_y, b_\lambda \in Y^*$ and $b_u \in U^*$. This abstract framework is just meant to simplify notation. In the following sections we will restrict our attention to more concrete problem settings with elliptic PDEs.

Using (2), we can write

$$y(u) = A^{-1}(b_\lambda - Bu)$$

and consider the equivalent unconstrained or reduced problem

$$\min_{u\in U} \tilde{J}(u) = J(y(u), u). \tag{3}$$

We assume that the reduced Hessian

$$\tilde{H} = \tilde{J}'' = B^*A^{-*}H_{yy}A^{-1}B + H_{uu}$$

is positive definite on U, such that (3) has a unique minimizer $\bar{u}$. Equivalent necessary and sufficient conditions are

$$\tilde{J}'(\bar{u}) = 0 \tag{4}$$

in the reduced setting (3) and, introducing the Lagrangian

$$L(y,u,\lambda) = J(y,u) + \langle \lambda, Ay + Bu - b_\lambda \rangle$$

for $\lambda \in Y$,

$$L'(\bar{y}, \bar{u}, \bar{\lambda}) = 0. \tag{5}$$

Given an approximate solution u^h of (3), we are interested in the error in the reduced functional. Introducing the Newton correction $\delta u = \bar{u} - u^h$, the error can be written as

$$\tilde{E}(u^h) = \tilde{J}(u^h) - \tilde{J}(\bar{u}) = \frac{1}{2}\langle \delta u, \tilde{H}\delta u \rangle. \tag{6}$$

Here we used (4) and the fact that $\tilde{J}$ is quadratic. For nonlinear but differentiable problems, remainder terms can be written down, which are, however, often ignored in practical algorithms.

In particular in PDE constrained optimization where A represents an elliptic differential operator, it is often more convenient to solve (5) rather than (4), because $\tilde{H}$ is in general a nonlocal operator and its discretization is usually dense. Therefore we assume an approximate solution $x^h = [y^h, u^h, \lambda^h]^T$ of (5) has been computed, such that the Newton correction $\delta x = \bar{x} - x^h = [\delta y, \delta u, \delta\lambda]^T \in X = Y \times U \times Y$ satisfies the residual equation $H\delta x = r$. Here,

$$H = L'' = \begin{bmatrix} H_{yy} & & A^* \\ & H_{uu} & B^* \\ A & B & \end{bmatrix} \tag{7}$$

denotes the Hessian of the Lagrange function.

An error representation that will turn out to be computationally more useful than (6) is based on the following Lemma.

Lemma 2.1. *For arbitrary $x = [x_y, x_u, x_\lambda]^T \in X$ it holds that*

$$\langle \delta u, \tilde{H} x_u \rangle = \langle w, Hx \rangle, \tag{8}$$

where $w = [w_y, w_u, w_\lambda]^T$ is given by

$$w_y = -A^{-1}B\delta u, \tag{9}$$

$$w_u = \delta u, \tag{10}$$

$$w_\lambda = -A^{-*}H_{yy}w_y. \tag{11}$$

Proof. Solving $Hx = z$ by block elimination via A and A^* we see that $\tilde{H}x_u = z_u - B^*A^{-*}(z_y - H_{yy}A^{-1}z_\lambda)$ and hence

$$\langle \delta u, \tilde{H} x_u \rangle = \frac{1}{2}\left\langle \delta u, z_u - B^*A^{-*}(z_y - H_{yy}A^{-1}z_\lambda) \right\rangle.$$

With w defined as above, this is just (8). $\square$

Applying Lemma 2.1 to (6) and using $r = H\delta x$ we obtain the error representation

$$\tilde{E}(u^h) = \frac{1}{2}\left(\langle w_y, r_y\rangle + \langle w_u, r_u\rangle + \langle w_\lambda, r_\lambda\rangle\right). \tag{12}$$

2.2 Galerkin discretization and hierarchical error estimators

Computing the error $\tilde{E}$ by (12) is as difficult as solving the original problem (1). For the purpose of error estimation, however, the weight functions w_u, w_y, and w_λ can be approximated with moderate cost.

In the following we assume that (5) has been solved by a Galerkin discretization using a finite dimensional ansatz spaces $X^h = Y^h \times U^h \times Y^h \subset X$, such that the discrete solution x^h satisfies

$$\langle L'(x^h), \xi^h\rangle = 0 \quad \text{for all } \xi^h \in X^h. \tag{13}$$

The finite dimensional Galerkin representation $H^{hh} : X^h \to (X^h)^*$ of H in X^h is defined by $\langle H^{hh}\xi^h, v^h\rangle_{(X^h)^*,X^h} = \langle H\xi^h, v^h\rangle_{X^*,X}$ for all $\xi^h, v^h \in X^h$. Similarly, the Galerkin right hand side b^h is the projection of $b = [b_y, b_u, b_\lambda]^T$ onto $(X^h)^*$ and satisfies $\langle b^h, v^h\rangle_{(X^h)^*,X} = \langle b, v^h\rangle_{X^*,X}$ for all $v^h \in X^h$. With this notation, x^h is given as the solution of the linear equation system

$$H^{hh}x^h = b^h,$$

leaving the residual $r = b - Hx^h$. Note that due to Galerkin orthogonality, r is polar to X^h, i.e. $\langle r, \xi^h\rangle = 0$ for all $\xi^h \in X^h$ and thus $r^h = 0$.

Let us assume there are finite dimensional extension spaces $Y^e \subset Y$ and $U^e \subset U$ given, such that $Y^h \cap Y^e = \{0\}$ and $U^h \cap U^e = \{0\}$. The approximation $[\delta x^h, \delta x^e]$ of $\delta x = \bar{x} - x^h$ on the extended ansatz space $X^h \times X^e$ with $X^e = Y^e \times U^e \times Y^e$ then satisfies

$$\begin{bmatrix} H^{hh} & H^{eh} \\ H^{he} & H^{ee} \end{bmatrix} \begin{bmatrix} \delta x^h \\ \delta x^e \end{bmatrix} = \begin{bmatrix} 0 \\ r^e \end{bmatrix}. \tag{14}$$

Solving (14) yields a computable approximation $\delta u^h + \delta u^e$ of δu. Since (14) can be quite expensive to solve, and one is only interested in an approximation for δu anyway, the system is usually simplified. We will address this topic in Section 2.4.

Discretizing (9)–(11) in a similar way, we may define approximate weight functions $w_y^h + w_y^e$ and $w_\lambda^h + w_\lambda^e$ satisfying

$$\begin{bmatrix} A^{hh} & A^{eh} \\ A^{he} & A^{ee} \end{bmatrix} \begin{bmatrix} w_y^h \\ w_y^e \end{bmatrix} = -\begin{bmatrix} B^{hh} & B^{eh} \\ B^{he} & B^{ee} \end{bmatrix} \begin{bmatrix} \delta u^h \\ \delta u^e \end{bmatrix} \tag{15}$$

and

$$\begin{bmatrix} A^{hh} & A^{eh} \\ A^{he} & A^{ee} \end{bmatrix}^* \begin{bmatrix} w_\lambda^h \\ w_\lambda^e \end{bmatrix} = -\begin{bmatrix} H_{yy}^{hh} & H_{yy}^{eh} \\ H_{yy}^{he} & H_{yy}^{ee} \end{bmatrix} \begin{bmatrix} w_y^h \\ w_y^e \end{bmatrix}, \tag{16}$$

respectively. Finally, as $r^h = 0$, a computable estimate $\tilde{E}^e$ of $\tilde{E}$ is given by

$$\tilde{E}(u^h) \approx [\tilde{E}](u^h) = \frac{1}{2}\left(\langle w_y^e, r_y^e\rangle + \langle w_u^e, r_u^e\rangle + \langle w_\lambda^e, r_\lambda^e\rangle\right). \tag{17}$$

2.3 Error indicators

In case $\tilde{E}$ is not sufficiently small, the ansatz space X^h needs to be enlarged. Natural candidates for functions to add are the ones that span the extension space X^e. Let us assume that $\{\xi_1^e, \dots, \xi_n^e\}$ is a basis of X^e and that $w = \sum_{i=1}^n w_i^e \xi_i^e$. A rather straightforward strategy is to extend the ansatz space with those basis functions ξ_i^e carrying the most weight in $\tilde{E}(u^h)$, i.e., for which $|w_i^e \langle \xi_i^e, r\rangle|$ is largest. This approach works quite well. A closer look, however, reveals that extending the ansatz space also affects the part of the Galerkin solution in X^h. In the following we will therefore design a greedy-style error indicator which for each basis function of X^e estimates how much $\tilde{E}$ would be affected if just this one basis function was included into the ansatz spaces.

For any $\xi \in X^e$, let $x(\xi)$ denote the Galerkin solution of (1) computed in the ansatz space $X^h + \mathbb{R}\xi$, with $u(\xi)$ being its control component.

Theorem 2.2. *Assume that $x(\xi) = x^h + \epsilon\xi + \hat{x}$ for some $\hat{x} \in X^h$. Then the linearized error reduction*

$$\begin{aligned}\tilde{E}(u(\xi)) = \tilde{E}(u^h) &+ \epsilon\left\langle (H^{ee} - H^{he}(H^{hh})^{-1}H^{eh})w^e, \xi\right\rangle \\ &+ \mathcal{O}\left(\epsilon^2 + \epsilon\|w^h + w^e - w\|\right)\end{aligned} \tag{18}$$

holds.

Proof. Since both $x(\xi)$ and x^h are Galerkin solutions in ansatz spaces that include X^h,

$$\langle H\hat{x}, \phi\rangle = \langle H(x(\xi) - \epsilon\xi - x^h), \phi\rangle = -\epsilon\langle H\xi, \phi\rangle$$

holds for all $\phi \in X^h$, such that $\hat{x}$ is given as the solution of $H^{hh}\hat{x} = -\epsilon H^{eh}\xi$. Using Lemma 2.1, we compute the directional derivative

$$\begin{aligned}\langle \tilde{E}'(u^h), \hat{x}_u + \epsilon\xi_u\rangle &= \langle \delta u, \tilde{H}(\hat{x}_u + \epsilon\xi_u)\rangle \\ &= \langle w, H(\hat{x} + \epsilon\xi)\rangle \\ &= \left\langle \begin{bmatrix} w^h \\ w^e \end{bmatrix}, \begin{bmatrix} H^{hh} & H^{eh} \\ H^{he} & H^{ee} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \epsilon\xi \end{bmatrix} \right\rangle + R \\ &= \left\langle \begin{bmatrix} H^{hh} & H^{eh} \\ H^{he} & H^{ee} \end{bmatrix} \begin{bmatrix} w^h \\ w^e \end{bmatrix}, \begin{bmatrix} -\epsilon(H^{hh})^{-1}H^{eh}\xi \\ \epsilon\xi \end{bmatrix} \right\rangle + R \\ &= \epsilon\left\langle (H^{ee} - H^{he}(H^{hh})^{-1}H^{eh})w^e, \xi\right\rangle + R\end{aligned}$$

where the remainder term due to discretization of w is of order $R = \mathcal{O}(\epsilon\|w^h + w^e - w\|)$. □

Suppose that $\delta x^e = \sum_{i=1}^n e^i \xi_i^e$. Let us define the error indicator weight $p = (H^{ee} - H^{he}(H^{hh})^{-1}H^{eh})w^e$. With the influence of including a basis function ξ^i into the ansatz space quantified by (18), we define the error indicators

$$\eta^i = \left| e^i \langle p, \xi_i^e \rangle \right| . \tag{19}$$

Note that if $p \in (X^e)^*$ is computed in the Galerkin setting of X^e, it is represented as a vector in $\mathbb{R}^n$ by the scalar products with the ansatz functions of X^e as $p^i = \langle p, \xi_i^e \rangle$. Thus, (19) reduces to $\eta^i = |e^i p^i|$.

2.4 Practical error estimation for PDE constrained optimization

In this section we focus on second order elliptic PDEs and thus assume that $A : H^1(\Omega) \to H^1(\Omega)^*$ is an elliptic second order differential operator defined over a domain $\Omega \subset \mathbb{R}^d$, and that $Y^h = \{\xi \in C(\Omega) : \xi|_T \in \mathbb{P}_1 \forall T \in \mathcal{T}\}$ is the space of linear finite elements over a conforming simplicial triangulation $\mathcal{T}$ of Ω with vertices $\mathcal{T}^\nu$. Let $Y^e = \{\xi \in C(\Omega) : \xi|_T \in \mathbb{P}_2 \ \forall T \in \mathcal{T} \wedge \xi(t) = 0 \ \forall t \in \mathcal{T}^\nu\}$ be the hierarchical extension of Y^h. The basis functions used for spanning Y^e are chosen with minimal support in order to give a sparse Galerkin matrix. To each edge in the triangulation a piecewise quadratic basis function known as "bubble function" is associated, the support of which is the union of the elements incident to the edge.

In this computational setting, the direct implementation of error estimation and extension of ansatz spaces as worked out above is inefficient and impracticable. Below we discuss practical modifications that lead to more efficient and implementable algorithms.

Approximate computation of weight functions. Since Y^e is significantly larger than Y^h by a factor of around 3 in 2D and 6 in 3D, solving (14) might be quite expensive. As is common for hierarchical error estimators (see [1, 8]), we solve a cheaper approximation by neglecting one off-diagonal block and localizing the defect problem. Consequently, the square subblocks of H^{ee} are reduced to just their diagonal, giving $\hat{H}^{ee}$:

$$\begin{bmatrix} H^{hh} & H^{eh} \\ & \hat{H}^{ee} \end{bmatrix} \begin{bmatrix} \delta x^h \\ \delta x^e \end{bmatrix} = \begin{bmatrix} 0 \\ r^e \end{bmatrix}.$$

Note that in $\hat{H}^{ee}$, all basis functions are decoupled, such that the computations are spatially local. For this reason, long-distance influences, leading to the so-called pollution error, have to be captured by δx^h. Dropping H^{eh} instead of H^{he}, or both off-diagonal blocks, would lead to $\delta x^h = 0$ and consequently neglect the global error transport.

As to the weight functions, in (15) we can again localize A^{ee} to $\hat{A}^{ee}$ and drop either A^{he} or A^{eh}. Since w_y^h does not enter into (17) due to r_y being polar to Y^h, dropping A^{he} and solving for w_y^e first would again neglect the pollution error due to the locality of $\hat{A}^{ee}$. This is in fact extremely critical if the range of B has local support, e.g. in boundary control problems — see Figure 12 in Section 3.2. This is different from energy norm estimates of elliptic PDEs, where the pollution error plays a negligible role [7,8]. Thus we now drop A^{eh} and solve for w_y^h first. The very same consideration holds for approximating w_λ^e via (16).

Having thus computed the weight functions $w^e = (w_y^e, w_u^e, w_\lambda^e)$, evaluating (17) is just a matter of linear algebra.

Mesh refinement. Enlarging the ansatz space in case the estimated error is too large is often realized as h-adaptivity. Instead of including basis functions from X^e into the ansatz space, mesh edges are marked for refinement if their associated quadratic basis would have been included into the ansatz space.

3 Numerical Examples

Before selecting example problems, let us briefly compare (12) with representations of the error quantity $E(y^h, u^h) = J(y^h, u^h) - J(\bar{y}, \bar{u})$ proposed in [5]. A qualitative comparison will allow us to design test problems highlighting the differences.

Assuming that (9) is solved by a Galerkin discretization, implying $\langle \lambda^h, Ay^h + Bu^h - b_\lambda \rangle = 0$, we have

$$
\begin{aligned}
E^h(y^h, u^h) = J(y^h, u^h) - J(\bar{y}, \bar{u}) &= L(y^h, u^h, \lambda^h) - L(\bar{y}, \bar{u}, \bar{\lambda}) \\
&= \frac{1}{2} \left\langle (\delta y, \delta u, \delta\lambda), L'' \, [\delta y, \delta u, \delta\lambda]^T \right\rangle \\
&= \frac{1}{2} \left(\langle \delta y, r_y \rangle + \langle \delta u, r_u \rangle + \langle \delta\lambda, r_\lambda \rangle \right).
\end{aligned}
$$

First of all, since L'' is indefinite, E can be both positive or negative, and in particular it can be zero even far away from the solution. Even if one is only interested in the optimal value $\tilde{J}(\bar{u})$, computational estimates of $[E]$ will be less reliable on coarser meshes and far away from the solution, such that the reliability of a vanishing error estimate is questionable. For this reason, breaking $[E]$ up into local contributions and summing up their absolute values is taken as a remedy, but this may sacrifice the efficiency of the error estimator. And of course, by construction $[E]$ gives little information about the objective value $\tilde{J}(u^h)$ that will actually be achieved if the computed control is implemented in practice.

In contrast, $\tilde{E}$ is always nonnegative, and zero only at the exact solution. Thus, a negative computational estimate $[\tilde{E}]$ is a strong indicator that the computation of the error representations and weight functions is not accurate enough.

The weight functions in E corresponding to w_y and w_λ are

$$\delta y = A^{-1}(r_\lambda - B\delta u) = w_y + A^{-1} r_\lambda$$

and

$$\delta\lambda = A^{-*}(r_y - H_{yy}\delta y) = w_\lambda + A^{-*} r_y - A^{-*} H_{yy} A^{-1} r_\lambda.$$

Therefore the differences between $\tilde{E}$ and E, and between the generated meshes, will be most pronounced when w_y is small but r_λ is large, or $\|w_\lambda\| \ll \|r_y\|$, respectively. Such situations occur, e.g., when the state equation error is large where it affects the cost functional but, due to spatial distance or averaging effects, has little influence on u. In particular, boundary control with state observation far away from the control boundary and finite dimensional controls come to mind.

Below we present one boundary control example and one with finite dimensional control. We will use linear finite elements on unstructured triangular meshes and examine both, the differences between $\tilde{E}$ and E^h, and the impact of approximating δu via (18). Computations are performed using the KASKADE 7 code based on DUNE [3, 4]. Since DUNE does not support marking of edges for refinement, the edge-related error indicators η^i are distributed equally to the incident elements. A fixed fraction (25%) of the elements with largest error indicator contributions is then marked for refinement.

3.1 Finite dimensional control

Example 1. We construct a test problem with finite dimensional scalar control where a large state equation error is created by a reentrant corner of the domain. We choose $\Omega = \{x \in [0,6] \times [0,2] : x_2 \le 6 - x_1 \vee x_2 \ge x_1 - 4\}$. The problem is

$$\min \frac{1}{2}\|y-1\|^2_{L^2(\Omega)} \quad \text{s.t.} \quad \begin{aligned} -\Delta y &= u \quad \text{in } \Omega \\ y &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

with an optimal control value of $\bar{u} = 3.26358$ computed on a very fine discretization. Due to symmetry, actual computations have been restricted to the lower half of Ω. The computational domain with the coarse grid is shown in Figure 1 together with the optimal state y^*. The errors $\tilde{E}(u^h)$ and $E(x^h)$ for adaptive refinement according to $[E]$ and $[\tilde{E}]$ are shown in Figure 2, refined grids in Figure 3. Both refinement criteria produce very similar grids with the same efficiency. In contrast, the errors $\tilde{E}(u^h)$ and $E(x^h)$, respectively, are strikingly different, showing an order of approximation difference. Both estimators provide good estimates of the true errors.

Example 2. As an extreme example we consider a problem where the control acts only on a subset $\Omega_c = [0,1] \times [0,2]$ of the domain Ω, and where the influence

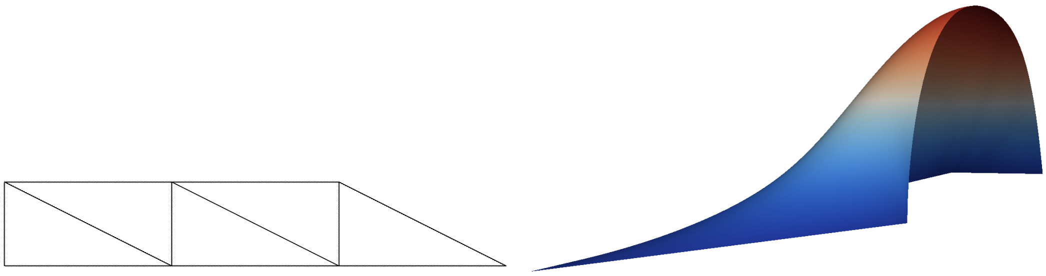

Figure 1: Setting of Example 1. Left: computational domain with coarse grid. Right: optimal state.

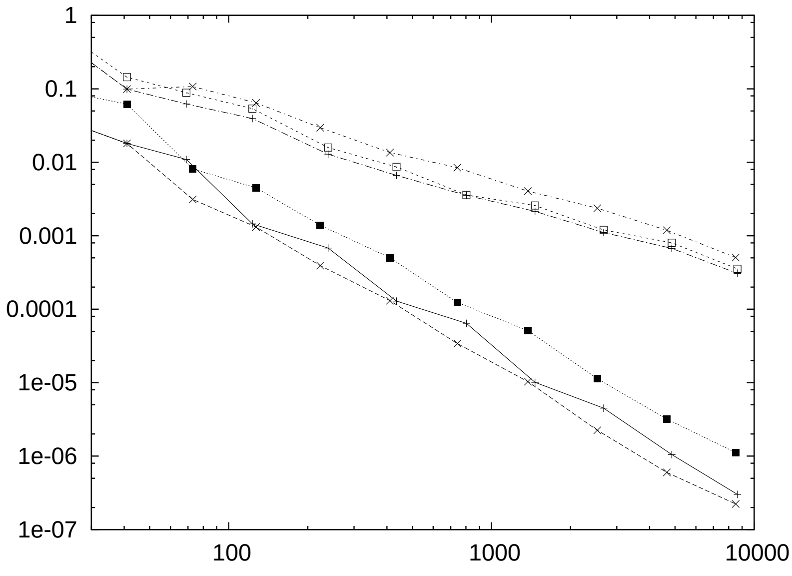

Figure 2: Error versus total number of degrees of freedom. Refinement according to $[E]$ is shown as $+$, according to $[\tilde{E}]$ as $\times$. Top lines: errors $E(x^h)$ for both refinement strategies and estimated error $[E]$ ($\square$) for ($+$). Bottom lines: errors $\tilde{E}(u^h)$ for both refinement strategies and estimator $[\tilde{E}]$ ($\blacksquare$) for ($\times$).

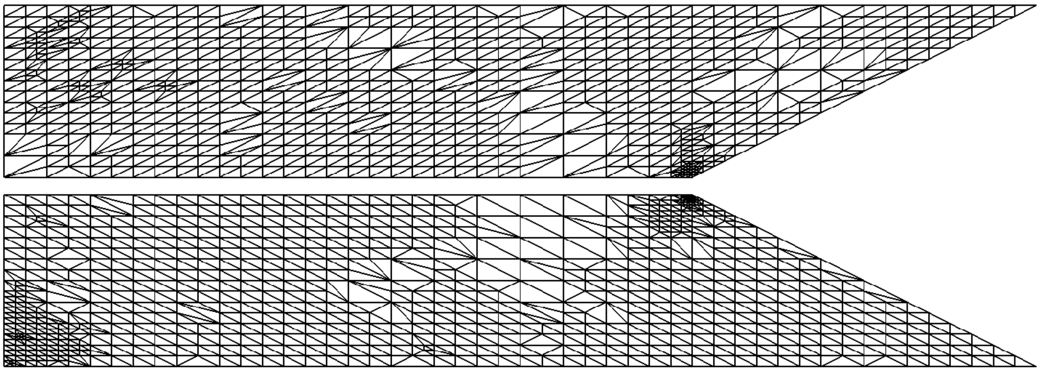

Figure 3: Meshes obtained by refining eight times according to $[\tilde{E}(u^h)]$ (top, 1377 dofs) and $[E(x^h)]$ (bottom, 1539 dofs).

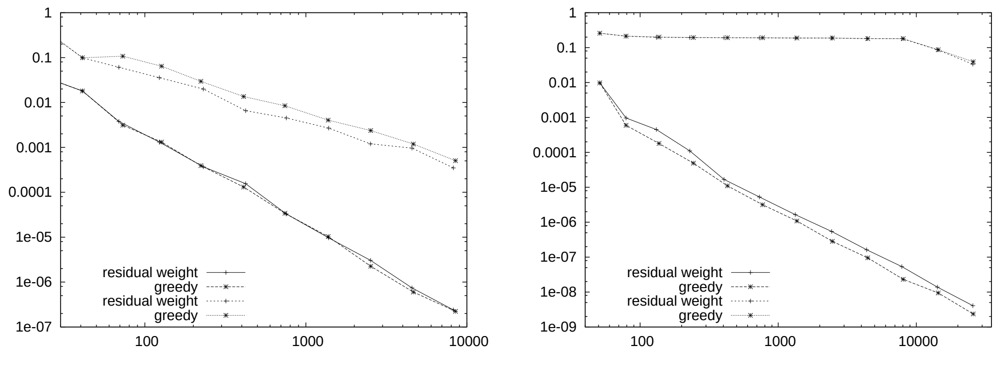

Figure 4: Effect of error indicator formulation on the errors $E(x^h)$ (top lines) and $\tilde{E}(u^h)$ (bottom lines) for Example 1 (left) and Example 2 (right). In both cases, refinement is according to $[\tilde{E}]$.

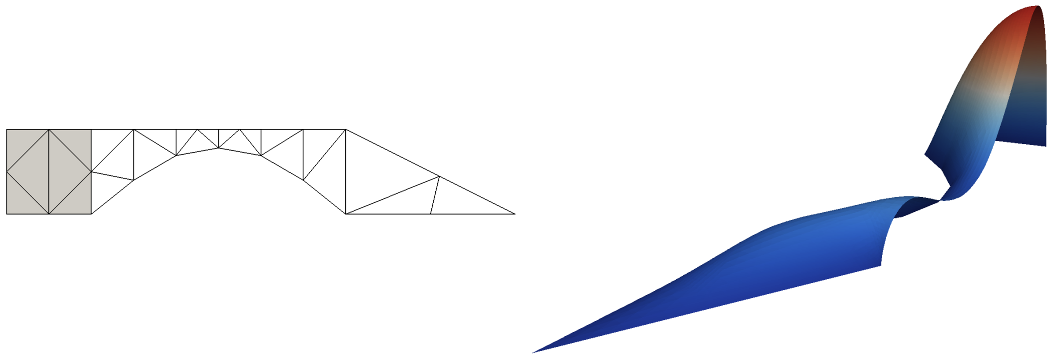

Figure 5: Setting of Example 2. Left: computational domain with coarse grid and shaded control region. Right: optimal state.

of the state error caused by the intruding corner on the control is attenuated by a bottleneck in the domain. An additional source term is applied on the whole domain such that the intruding corner leads to a relevant gradient singularity.

$$\min_{y\in H^1(\Omega),u\in\mathbb{R}} \frac{1}{2}\|y-1\|^2_{L^2(\Omega)} \quad \text{s.t.} \quad \begin{aligned} -\Delta y &= 3+\chi_{\Omega_c}u && \text{in } \Omega \\ y &= 0 && \text{on } \partial\Omega \end{aligned}$$

Again, computation is restricted to the lower half of the domain as shown in Figure 5 together with the optimal state. Here the optimal control value is $\bar{u} = 4.74727$. The errors and their estimates are shown in Figure 6, refined meshes in Figure 7. As expected, the smaller influence of the corner singularity on the control leads to significantly less refinement in that area when using $[\tilde{E}]$ as refinement criterion. The degrees of freedom saved on the right of the domain are spent on the left in order to compute a more accurate control. This is also reflected in the error, where a small but persistent accuracy gain of factor 2.5 for the same number of degrees of freedom can be observed when refining according to $[\tilde{E}]$. As a downside, the actually computed value of $J(y^h, u^h)$ is much less accurate for refinement driven by $[\tilde{E}]$.

3.2 Boundary control

This example is modeled after [5]. We choose $\Omega = [1/4, 3/4] \times [0, 3/4] \cup [0, 1] \times [1/2, 3/4]$ and define the control boundary $\partial\Omega_c = \{x \in \partial\Omega : x_2 = 0\}$. The observation boundary is either $\partial\Omega_o = \partial\Omega_c$ (a) or $\partial\Omega_o = \{x \in \partial\Omega : x_2 = 3/4\}$ (b). On this geometry we consider the tracking type problem

$$\min \frac{1}{2}\|y-y_d\|^2_{L^2(\Omega_o)} + \frac{\alpha}{2}\|u\|^2_{L^2(\Omega_c)}$$

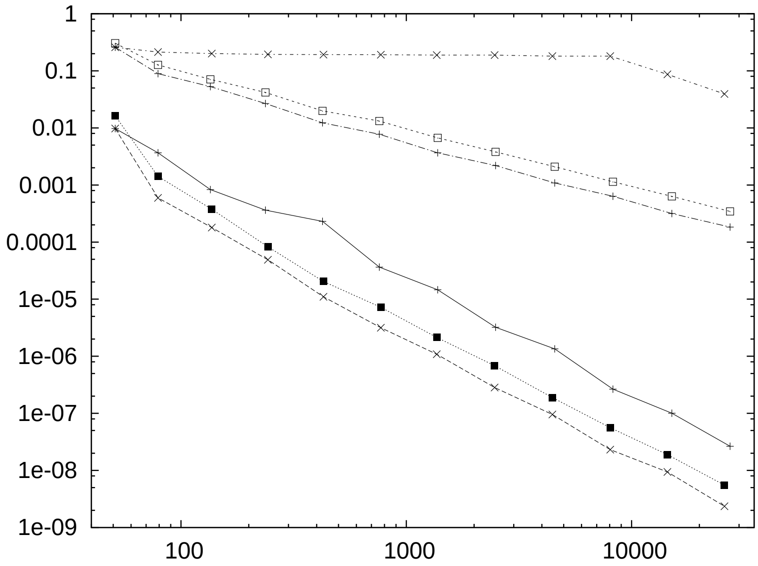

Figure 6: Error versus total number of degrees of freedom. Refinement according to $[E]$ is shown as $+$, according to $[\tilde{E}]$ as $\times$. Top lines: errors $E(x^h)$ for both refinement strategies and estimated error $[E]$ ($\square$) for ($+$). Bottom lines: errors $E(u^h)$ for both refinement strategies and estimator $[\tilde{E}]$ ($\blacksquare$) for ($\times$).

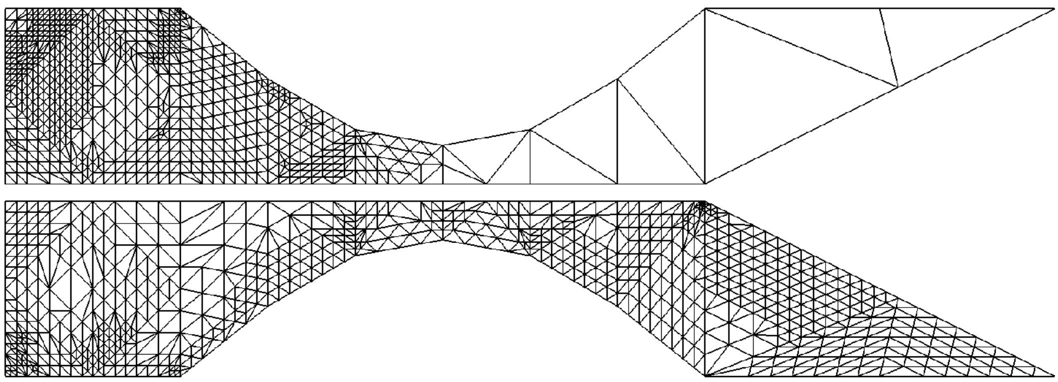

Figure 7: Meshes obtained by refining six times according to $[\tilde{E}]$ (top, 1365 dofs) and $[E]$ (bottom, 1417 dofs) with approximately the same number of vertices.

subject to

$$\begin{aligned} -\kappa\Delta y + y &= 0 && \text{in } \Omega \\ \kappa\partial_n y &= u && \text{on } \partial\Omega_c \\ \partial_n y &= 0 && \text{on } \partial\Omega\backslash\partial\Omega_c \end{aligned}$$

Following [5], we choose $y_d = 1$ and $\alpha = 1$, but $\kappa = 0.1$.

The unique stationary point of the Lagrangian

$$L(y,u,\lambda) = \frac{1}{2}\|y - y_d\|^2_{L^2(\partial\Omega_o)} + \frac{\alpha}{2}\|u\|^2_{L^2(\partial\Omega_c)} + \int_\Omega (\nabla y \cdot \nabla\lambda + y\lambda)dx - \int_{\partial\Omega_c} u\lambda\, dx,$$

is the solution. As before, the space Y^h of piecewise linear finite elements on a triangular mesh is chosen as ansatz space for y and λ. U^h is just the trace space of Y^h on $\partial\Omega_c$. Together with the stationarity condition

$$0 = \partial_u L(y,u,\lambda) = \alpha u - \lambda \tag{20}$$

this ensures $r_u = 0$.

Equivalently, we can use (20) to eliminate u pointwisely, which is just block elimination of u in (7) via the Nemyckii operator H_{uu}, and obtain a control reduced Lagrangian

$$\hat{L}(y,\lambda) = \frac{1}{2}\|y - y_d\|^2_{L^2(\partial\Omega_o)} - \frac{1}{2\alpha}\|\lambda\|^2_{L^2(\partial\Omega_c)} + \int_\Omega (\nabla y \cdot \nabla\lambda + y\lambda)dx.$$

In this case, the Hessian of the Lagrangian turns out to be

$$L'' = \begin{bmatrix} H_{yy} & A^* \\ A & Q \end{bmatrix}, \quad Q = -BH_{uu}^{-1}B^*,$$

and accordingly the weight functions are $w_y = -A^{-1}Q\delta\lambda$ and $w_\lambda = -A^{-*}H_{yy}w_y$.

4 Conclusion

Goal-oriented error estimation and adaptive refinement for optimal control problems can equally well be performed with respect to the error quantities $E(x^h)$ and $\tilde{E}(u^h)$, respectively. The error quantities E and $\tilde{E}$ can differ by orders of magnitude, such that using the one that is of actual interest can be crucial, in particular if used in termination criteria. Depending on the problem, the adaptively refined meshes can be quite similar or very different, in the latter case leading to a modest efficiency gain.

Acknowledgement. Support by DFG Research Center MATHEON "Mathematics for key technologies", project A1, is gratefully acknowledged.

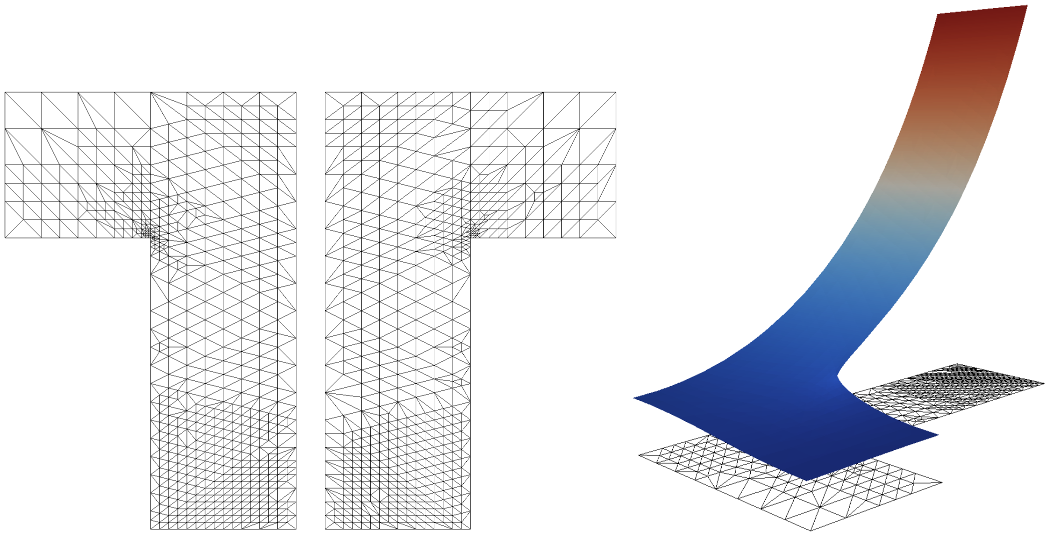

Figure 8: Adaptive grids obtained for situation (a) by refining according to $[E]$ (left) and according to $[\tilde{E}]$ (center). Optimal state y (right).

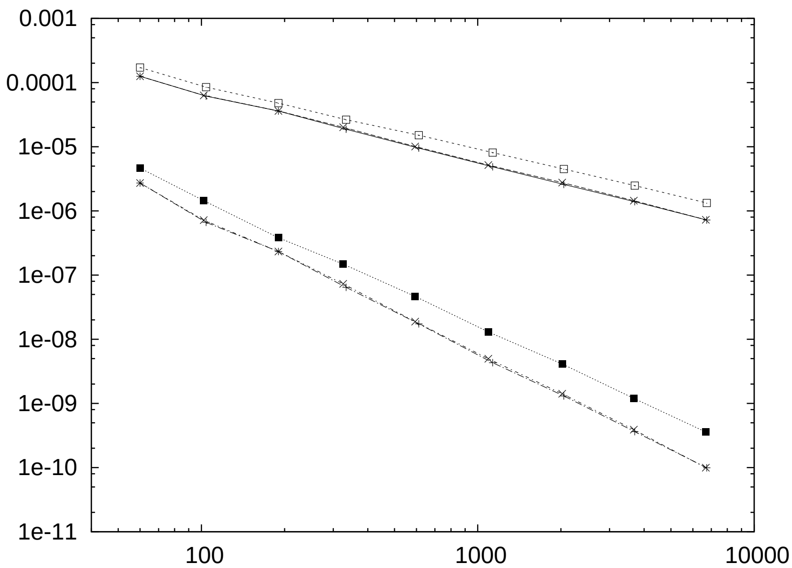

Figure 9: Error versus total number of degrees of freedom. Refinement according to $[E]$ is shown as $+$, according to $[\tilde{E}]$ as $\times$. Top lines: errors $E(x^h)$ for both refinement strategies and estimated error $[E]$ ($\square$) for ($+$). Bottom lines: errors $\tilde{E}(u^h)$ for both refinement strategies and estimator $[\tilde{E}]$ ($\blacksquare$) for ($\times$).

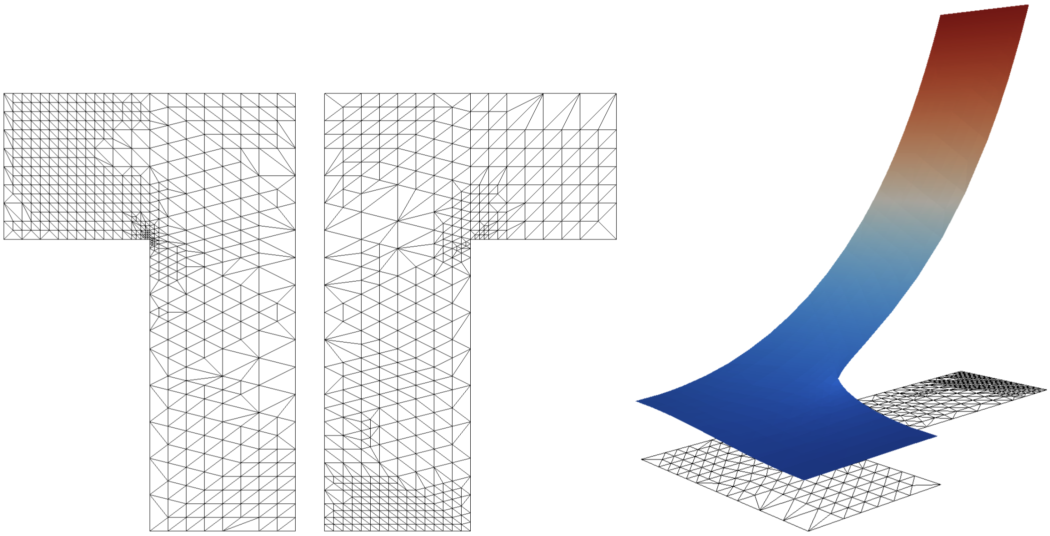

Figure 10: Adaptive grids obtained for situation (b) after five steps of refinement according to $[E]$ (left, 1074 dof) and according to $[\tilde{E}]$ (center, 870 dof). Optimal state $y/2$ (right, scaled for graphical presentation).

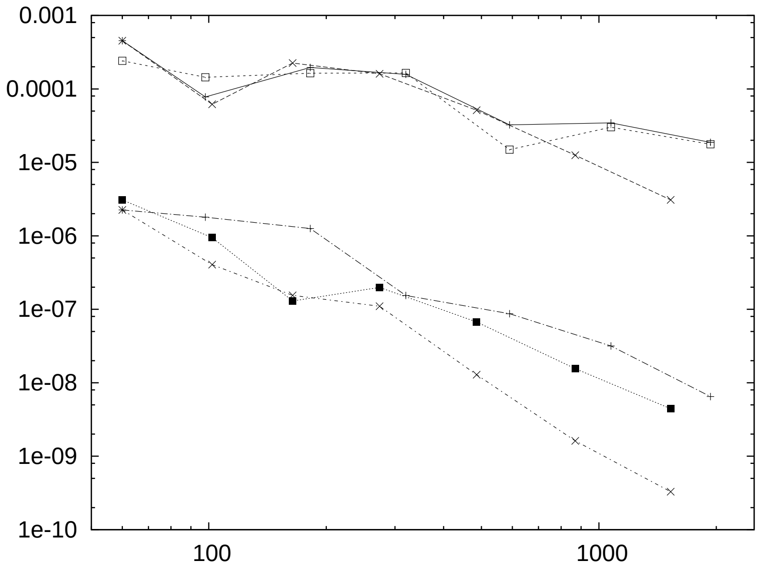

Figure 11: Error versus total number of degrees of freedom. Refinement according to $[E]$ is shown as $+$, according to $[\tilde{E}]$ as $\times$. Top lines: errors $E(x^h)$ for both refinement strategies and estimated error $[E]$ ($\square$) for ($+$). Bottom lines: errors $\tilde{E}(u^h)$ for both refinement strategies and estimator $[\tilde{E}]$ ($\blacksquare$) for ($\times$).

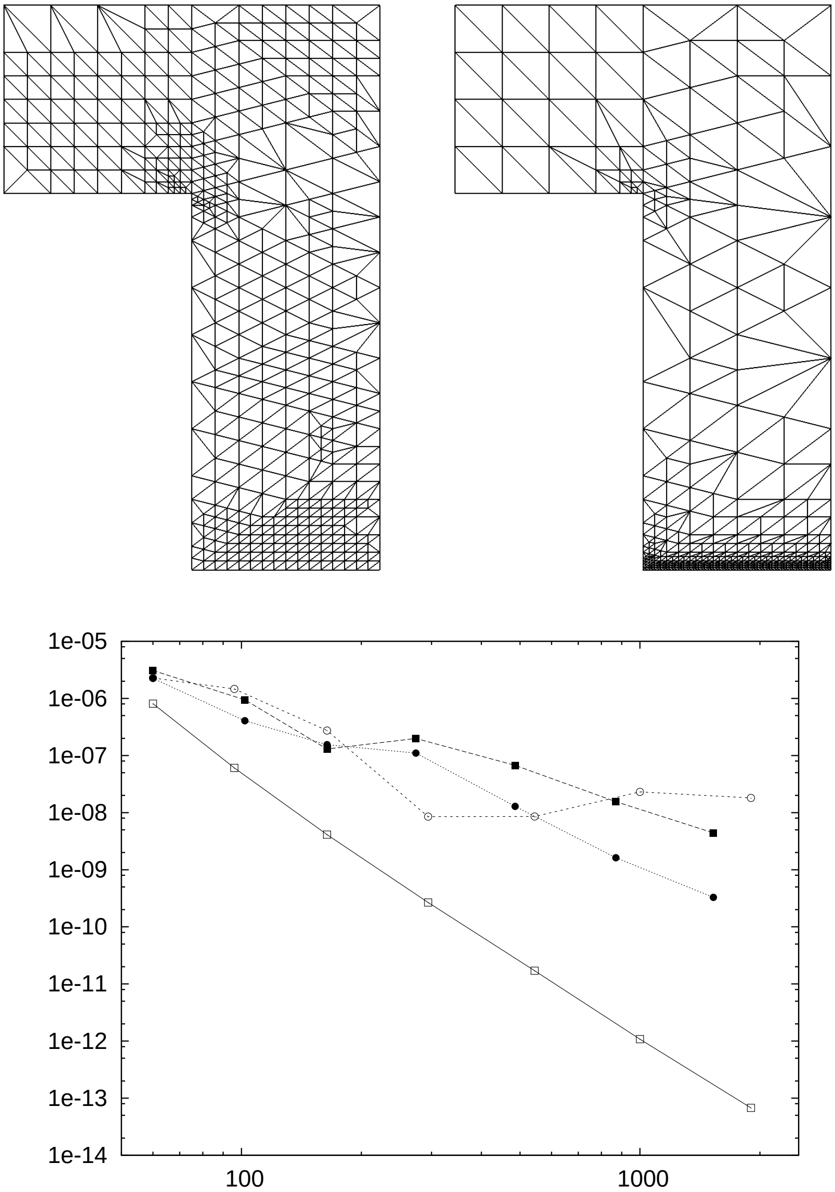

Figure 12: Effect of neglecting the pollution error by considering considering local defect problems only. Top left: pollution error taken into account (same as Fig. 10 center, 870 dof). Tor right: pollution error not taken into account (1000 dof). Bottom: error $\tilde{E}$ ($\bullet$) and its estimate $[\tilde{E}]$ ($\blacksquare$) taking the pollution error into account, error $\tilde{E}$ ($\circ$) and its estimate $[\tilde{E}]$ ($\square$) neglecting the pollution error.

References

[1] M. Ainsworth and J.T. Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics. Wiley, 2000.

[2] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics. Birkhäuser, 2003.

[3] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, R. Kornhuber, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. Part II: Implementation and tests in DUNE. *Computing*, 82(2–3):121–138, 2008.

[4] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. *Computing*, 82(2–3):103–119, 2008.

[5] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: basic concepts. *SIAM J. Control Optim.*, 39:113–132, 2000.

[6] O. Benedix and B. Vexler. A posteriori error estimation and adaptivity for elliptic optimal control problems with state constraints. *Comput. Optim. Appl.*, to appear.

[7] F.A. Bornemann, B. Erdmann, and R. Kornhuber. A posteriori error estimates for elliptic problems. *SIAM J. Numer. Anal.*, 33:1188–1204, 1996.

[8] P. Deuflhard, P. Leinen, and H. Yserentant. Concepts of an adaptive hierarchical finite element code. *IMPACT Comput. Sci. Engrg.*, 1:3–35, 1989.

[9] A. Griesbaum, B. Kaltenbacher, and B. Vexler. Efficient computation of the Tikhonov regularization parameter by goal oriented adaptive discretization. *Inverse Problems*, 24(2), 2008.

[10] A. Günther and M. Hinze. A posteriori error control of a state constrained elliptic control problem. *J. Num. Math.*, 16(4), 2008.

[11] M. Hintermüller and R. Hoppe. Goal-oriented adaptivity in control constrained optimal control of partial differential equations. Preprint 013/2007, University Augsburg, 2007.

[12] R. Rannacher. On the adaptive discretization of PDE-based optimization problems. In M. Heinkenschloss and et al., editors, *PDE Constrained Optimization*. Springer, 2006.

[13] A. Schiela and A. Günther. Interior point methods in function space for state constraints — inexact Newton and adaptivity. Report 2009-01, Zuse Institute Berlin, 2009.

[14] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques.* Advances in Numerical Mathematics. Wiley, 1996.

[15] B. Vexler. *Adaptive Finite Element Methods for Parameter Identication Problems.* PhD thesis, U Heidelberg, 2004.

[16] B. Vexler and W. Wollner. Adaptive finite elements for elliptic optimization problems with control constraints. *SIAM J. Control Optim.*, 47(1):509–534, 2008.