59 (2014) APPLICATIONS OF MATHEMATICS No. 5, 489–508

ON THE NUMERICAL PERFORMANCE OF A SHARP A POSTERIORI ERROR ESTIMATOR FOR SOME NONLINEAR ELLIPTIC PROBLEMS

Balázs Kovács, Budapest

(Received March 14, 2013)

Abstract. Karátson and Korotov developed a sharp upper global a posteriori error estimator for a large class of nonlinear problems of elliptic type, see J. Karátson, S. Korotov (2009). The goal of this paper is to check its numerical performance, and to demonstrate the efficiency and accuracy of this estimator on the base of quasilinear elliptic equations of the second order. The focus will be on the technical and numerical aspects and on the components of the error estimation, especially on the adequate solution of the involved auxiliary problem.

Keywords: a posteriori error estimation; quasilinear elliptic problem; numerical experiment

MSC 2010: 65N15, 65N50, 65M60, 65J15

1. Introduction

There is a huge number of numerical methods which are developed to solve mathematical models describing various physical, chemical, biological and other phenomena. A very important issue is to quantify the accuracy of the numerical approximations, i.e., the error estimation of the numerical methods is a crucial point of modeling.

In [12], Karátson and Korotov have developed a sharp upper global a posteriori error estimator for nonlinear elliptic problems, which is perfectly applicable in a finite element framework. Both second and fourth order equations, and also systems of nonlinear elliptic equations are fitting into the theory. These ideas were generalized from a preceding paper covering the same type of error estimation but for linear elliptic problems, see [18]. The estimator is independent of the iterative method which is used to solve the problem, and it is *sharp* in the sense that by the investment of "com-

putation time" we can get as close to the true error as we want. This technique also allows to develop an adaptive method where the errors over some subdomains (e.g. elements of the mesh) are estimated similarly defining elements to refine or coarse. A general reference to error estimation in finite element methods is [3], references on a posteriori estimates for nonlinear problems are [18], [17], duality theory based estimates can be found in [7], [8] deals with functional type error estimation for elliptic problems, on global posteriori error estimation for convection-reaction-diffusion problems the reader is referred to [14], while detailed monographs are [1], [19].

This paper is devoted to the demonstration of the efficiency and accuracy of this estimator with the aid of quasilinear second order elliptic problems. Besides the numerical solution of the nonlinear problem, the estimator involves a smoothing operator, solution of an auxiliary problem on a refined mesh, and calculation of various norms. We are focusing on the technical and numerical aspects of the error estimation: the smoothing itself, the assembly and solution of the refined auxiliary problem (some nested finite element spaces are also involved), fast and accurate norm calculation, and we also briefly discuss the solution of the test problem: the used iterative methods and matrix assemblies. The whole text is written in a finite element framework.

This paper is organized as follows. In Section 2 we introduce the class of elliptic problems which we later use in the numerical tests. We give its weak form, and also briefly discuss its solvability in Sobolev spaces. The corresponding error functional is also introduced, just as some other notation. Section 3 gives a short introduction to the error estimator which is in the center of our investigation. The section ends with a very important remark about the property which gives the estimator its strength, namely how sharpness can be achieved. One of the main parts of this paper is Section 4, which covers the nonlinear solver, the components of the estimator, and also related issues are discussed. Section 5 is devoted to the numerical results, i.e., we consider a given elliptic test problem, we numerically solve the problem, the constants from the estimator are also calculated, and the results of our numerical experiments are displayed in figures and tables.

2. The model problem and notation

In the original paper of Karátson and Korotov [12], the upper error estimator is developed for a very large class of nonlinear problems. Here we focus on the class of second order quasilinear elliptic partial differential equations (PDEs), namely the problems of the form:

$$\begin{cases} -\operatorname{div}(g(|\nabla u|^2)\nabla u) = b & \text{in } \Omega, \\ u|_{\partial\Omega} = 0, \end{cases} \tag{2.1}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain with sufficiently smooth boundary, the r.h.s. b is a real-valued function over Ω. We introduce the notation $f(\xi) := g(|\xi|^2)\xi$ for every $\xi \in \mathbb{R}^d$, i.e., our equation reads $-\operatorname{div} f(\nabla u) = b$.

This problem will be weakly solved by means of a finite element based iterative method over the space $H_0^1(\Omega)$, hence we introduce the weak form of (2.1) including a nonlinear operator F:

$$\langle F(u), v\rangle := \int_\Omega g(|\nabla u|^2)\nabla u \cdot \nabla v = \int_\Omega bv, \quad v \in H_0^1(\Omega), \tag{2.2}$$

where $\langle u, v\rangle := \int_\Omega \nabla u \cdot \nabla v$ denotes the usual inner product (inducing the norm $\|\cdot\|$) on $H_0^1(\Omega)$. An easy computation shows that the Gâteaux derivative of this operator exists for arbitrary $u \in H_0^1(\Omega)$ and it is

$$\langle F'(u)p, v\rangle = \int_\Omega (g(|\nabla u|^2)\nabla p \cdot \nabla v + 2g'(|\nabla u|^2)(\nabla u \cdot \nabla p)(\nabla u \cdot \nabla v)), \quad p, v \in H_0^1(\Omega). \tag{2.3}$$

The weak solution of this problem is denoted by u^*, the existence and uniqueness of u^* is guaranteed by the following theorem under suitable assumptions on the problem.

Let the following conditions hold:

(i) The domain $\Omega \subset \mathbb{R}^d$ is bounded, and it is C^2-diffeomorphic to some convex set.

(ii) The function f is $C^1(\mathbb{R}^d, \mathbb{R}^d)$ and the Jacobian $\partial f(\eta)/\partial\eta$ is symmetric, uniformly bounded and positive, i.e., its eigenvalues can be bounded by some $0 < m \leqslant M$ for arbitrary $\eta \in \mathbb{R}^d$.

(iii) $b \in L^2(\Omega)$.

Then the operator F' is uniformly elliptic, i.e., it can be spectrally bounded by constants $0 < m \leqslant M$ and this implies that the problem has a unique weak solution $u^* \in H_0^1(\Omega)$.

For more details see [5].

The error functional of this problem is non-quadratic, namely, the error corresponding to arbitrary function u from $H_0^1(\Omega)$ is measured by

$$E(u) := \langle F(u) - F(u^*), u - u^*\rangle = \int_\Omega (f(\nabla u) - f(\nabla u^*)) \cdot (\nabla u - \nabla u^*), \tag{2.4}$$

and hence we have

$$m\|u - u^*\|^2 \leqslant E(u) \quad \forall\, u \in H_0^1(\Omega).$$

Since the unknown solution u^* is involved in the computation of this functional, we would like to somehow estimate E. Our goal is to numerically illustrate the sharpness

of a particular estimator of the error functional (which is detailed, besides [12], later in Section 3), and also to show its efficiency, with the aid of problems having the form (2.1) (for the numerical experiments see Section 5).

R e m a r k 2.1. We point out here that the test problem could be more general, e.g., other types of boundary conditions (Neumann, mixed, etc.), general nonlinear problems, i.e., $-\operatorname{div}(f(x,\nabla u))$ could be allowed, as long as it fits into the framework of [12], see the assumptions above. Nevertheless, if we considered more general problems it would not give more insight into the behaviour of the estimator, but it would cause many technical problems elsewhere.

3. The a posteriori error estimator

In this section we briefly recall the estimator itself, the theoretical and practical background of it, but we do not go into details, for a full description the reader is referred to [12].

Let us assume that the nonlinear problem $F(u) = b$ is given and has a unique weak solution u^*, with the corresponding error functional $E(u) = \langle F(u) - F(u^*), u - u^*\rangle$, where the operator F satisfies the conditions (i)–(iii) from the previous section and it also satisfies the following:

(iv) $f'\colon \mathbb{R}^d \to \mathbb{R}^{d\times d}$ (i.e. $F'\colon W^{1,\infty}(\Omega) \to B(H_0^1(\Omega))$) is Lipschitz continuous with constant L.

(The original abstract assumptions are the ones listed in ([12], Assumptions 3.2 and 3.3)).

Then the following theorem holds:

Theorem 3.1 ([12])**.** *Let $u_h \in W^{1,\infty}(\Omega)$ be an approximate solution of $F(u) = b$. Then for arbitrary $z \in L^\infty(\Omega)^d$ such that $z \in H(\operatorname{div}) := \{z \in L^2(\Omega)^d\colon \operatorname{div} z \in L^2(\Omega)\}$ and arbitrary $w \in H_0^1(\Omega)$,*

$$
\begin{aligned}
(3.1)\qquad E(u_h) &\leqslant \operatorname{EST}(u_h; z, w)\\
&:= \Big(m^{-1/2} c_\Omega \|\operatorname{div} f(z) + b\|_{L^2(\Omega)} + \frac{L}{2} m^{-3/2} D(u_h; z, w)\\
&\quad + \Big(\langle f(\nabla u_h) - f(z),\ \nabla u_h - z\rangle_{L^2(\Omega)^d}\\
&\quad + \frac{L}{2m} D(u_h; z, w) \|\nabla u_h - z\|_{L^2(\Omega)^d} \Big)^{1/2} \Big)^2,
\end{aligned}
$$

where

$$
(3.2)\quad D(u_h; z, w) := (M\|z - \nabla w\|_{L^2(\Omega)^d} + c_\Omega \|\operatorname{div} f(z) + b\|_{L^2(\Omega)}) \|\nabla u_h - z\|_{L^\infty(\Omega)^d},
$$

where $M \geqslant m > 0$ are the spectral bounds of $F'(u)$, L is its Lipschitz constant, and $c_\Omega > 0$ is the constant appearing in the Poincaré-Friedrichs inequality (it only depends on the domain Ω).

The subscript in u_h only means that this is an approximate solution, which came from an arbitrary solution technique. We point out here that clearly the estimate (3.1) is sharp for $u_h = u^*$, $z^* = \nabla u^*$, and the corresponding w is u^*, due to ([12], Prop. 4.1).

We now turn to the practical aspects of this a posteriori estimator.

Let V_h be a finite element subspace of $H_0^1(\Omega)$ and let $u_h \in V_h$ be the corresponding FEM approximation of u^*, the exact solution of our problem (2.2). In the usual finite element framework, u_h is a continuous piecewise polynomial function, hence $u_h \in W^{1,\infty}(\Omega)$. If we choose z to be any continuous piecewise polynomial function, e.g. a function from V_h, and arbitrary $w \in H_0^1(\Omega)$, then $z \in L^\infty(\Omega)^d \cap H(\mathrm{div})$, and all of the assumptions of the above theorem are fulfilled.

R e m a r k 3.1. The optimal parameters in $\mathrm{EST}(u_h; \cdot, \cdot)$, corresponding to the given numerical solution u_h, will be denoted by a superscript $*$, i.e., z^*, w^*. Using this notation, we would like to compute $\mathrm{EST}(u_h; z^*, w^*)$.

The next paragraph is devoted to the determination of the optimal z^* and w^* in $\mathrm{EST}(u_h; z, w)$: By the above properties, the optimal value of the parameter z^* should be a sufficiently accurate approximation of the gradient of u^*. By the suggestions made in [12] we use an averaging operator G_h, introduced in ([9], pp. 146–150). Namely we replace the unknown function ∇u^* by the averaged gradient of the approximate solution $G_h(\nabla u_h)$, since in the case of linear elements $G_h(\nabla u_h)$ is closer to ∇u^* than ∇u_h, precisely we have

$$\|\nabla u^* - \nabla u_h\| = o(h), \quad \text{while} \quad \|\nabla u^* - G_h(\nabla u_h)\| = o(h^2), \tag{3.3}$$

see [9].

Then we define

$$z^* := G_h(\nabla u_h).$$

Finally, the last missing parameter $w^* \in H_0^1(\Omega)$ is defined as the weak solution of the linear auxiliary equation:

$$\begin{cases} -\Delta w = -\operatorname{div} z^* & \text{in } \Omega, \\ w|_{\partial\Omega} = 0. \end{cases} \tag{3.4}$$

By a solution of this problem we always mean a weak solution.

If pure Dirichlet boundary conditions are posed, then $c_\Omega \leqslant \mathrm{diam}(\Omega)^d$, both for this special case and for a more general case we refer to [20] or [16].

R e m a r k 3.2. The following property of the estimator is crucial. For piecewise linear finite elements the numerical integration of the right-hand side is very easy, therefore the solution of the auxiliary problem is also very easy, to be precise it requires considerably less computation than the solution of the nonlinear problem. Hence, if we solve it on a much finer mesh, we can increase the efficiency of the estimator easily and quickly. This property is one of the strengths of this estimator, it will play a very important role later on.

4. Numerical aspects of the estimator and the tests

This section is devoted to the description of the numerical aspects of both the nonlinear solver and the estimator from Section 3 in detail. We also discuss some particular parts of them, concentrating on the numerical aspects and on the efficiency of the applied techniques.

4.1. Solving the nonlinear problem. We numerically solve the problem (2.2) by finite element based iterative methods: a quasi-Newton method with a piecewise constant but stepwise variable preconditioner in one dimension, and a Newton's method in two dimensions. The idea behind using not just one nonlinear solver is to show that the estimator works well with various nonlinear solvers.

First we define a triangulation of our domain Ω, and the corresponding finite element subspace $V_h \subset H_0^1(\Omega)$, consisting of the piecewise linear functions vanishing on the boundary $\partial\Omega$.

The first of our two iterative solvers is a quasi-Newton method:

$$\begin{cases} \text{(a)} & u_0 \equiv 0; \\ \text{(b)} & \text{for a fixed } n \in \mathbb{N}, \text{ if } u_n \in V_h \text{ is known then} \\ & p_n \in V_h \text{ is the solution of the problem:} \\ & \displaystyle\int_\Omega \omega_n(x)\nabla p_n \cdot \nabla v = -\int_\Omega g(\nabla|u_n|^2)\nabla u_n \cdot \nabla v + \int_\Omega bv \quad \forall\, v \in V_h; \\ \text{(c)} & u_{n+1} := u_n + \tau_n\, p_n, \end{cases}$$

where τ_n is a damping parameter and the function ω_n piecewise constantly approximates the nonlinear term in the derivative F'. In our case the domain Ω is usually decomposed into several (usually less than five) parts, where the function ω_n is constant.

The second solver is a damped inexact Newton method:

$$\begin{cases} \text{(a)} & u_0 \equiv 0; \\ \text{(b)} & \text{for a fixed } n \in \mathbb{N}, \quad \text{if } u_n \in V_h \text{ is known then} \\ & p_n \in V_h \text{ is the solution of the problem:} \\ & \int_\Omega (g(|\nabla u_n|^2)\nabla p_n \cdot \nabla v + 2g'(|\nabla u_n|^2)(\nabla u_n \cdot \nabla p_n)(\nabla u_n \cdot \nabla v)) \\ & \quad = -\int_\Omega g(\nabla |u_n|^2)\nabla u_n \cdot \nabla v + \int_\Omega bv \quad \forall\, v \in V_h; \\ \text{(c)} & u_{n+1} := u_n + \tau_n\, p_n, \end{cases}$$

where τ_n is again a damping parameter.

The first method was developed by Karátson and Faragó in [11], and both of these methods have been intensively investigated, see e.g. [5], [15], [13], or generally on nonlinear methods and Newton-type methods see [21] and [4], respectively.

Throughout the paper, all of the finite element related objects, i.e., solving the nonlinear and the auxiliary equations, the computation of norms, are in accordance with the well known element-by-element assembly techniques combined with the use of a reference element.

R e m a r k 4.1. During these algorithms we have to compute the gradient of a piecewise polynomial function from V_h, which can be done by a little modification of the general matrix assembly idea. To be precise, going over the nodes and using sufficient order of finite differences yields an exact derivative of $v \in V_h$.

4.2. Fast and accurate assembly of the stiffness matrix S. The crucial step in the assembly of the linear auxiliary equations is the computation of the integrals (or matrix elements):

$$\int_\Omega g(|\nabla u_n|^2)\nabla p_n \cdot \nabla \phi_k + \int_\Omega 2g'(|\nabla u_n|^2)(\nabla u_n \cdot \nabla p_n)(\nabla u_n \cdot \nabla \phi_k),$$

where $u_n \in V_h$ is obtained as before, and the unknown function is $p_n = \sum_{j=1}^{n(h)} c_j \phi_j$, for every basis function ϕ_k, $k = 1, 2, \ldots, n(h)$ (here $n(h)$ denotes the number of basis functions spanning the FEM subspace V_h, this notation will be used later as well). We have to compute the matrix in each iterative step, hence the assembly needs to be very fast. The elements of the matrix S are the following:

$$S_{kj} = \int_\Omega g(|\nabla u_n|^2)\nabla \phi_j \cdot \nabla \phi_k \tag{4.1}$$

$$+ \int_\Omega 2g'(|\nabla u_n|^2)(\nabla u_n \cdot \nabla \phi_j)(\nabla u_n \cdot \nabla \phi_k) \tag{4.2}$$

for $j, k = 1, 2, \ldots, n(h)$.

We formulate our approach and result in the following proposition.

Proposition 4.1. *For a given $u_h \in V_h$ the matrix S can be assembled as follows:*

$$\begin{aligned} S_{kj} = & \sum_{T\in\mathcal{T}_h} (g(|\nabla u_n|^2))|_T \int_T (\partial_x\phi_j\partial_x\phi_k + \partial_y\phi_j\partial_y\phi_k) \\ & + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_x u_n)^2)|_T \int_T \partial_x\phi_j\partial_x\phi_k \\ & + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_x u_n\partial_y u_n))|_T \int_T (\partial_x\phi_j\partial_y\phi_k + \partial_y\phi_j\partial_x\phi_k) \\ & + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_y u_n)^2)|_T \int_T \partial_y\phi_j\partial_y\phi_k, \end{aligned}$$

where $\mathcal{T}_h$ is a triangulation of the domain. Furthermore, the integrals over the triangles $T \in \mathcal{T}_h$ can be expressed by integrals over the reference triangle T_0 and the basis functions φ over it, e.g.,

$$\begin{aligned} \int_T \partial_x\phi_j\partial_x\phi_k = c_{11}^2 \int_{T_0} \partial_x\varphi_j\partial_x\varphi_k + c_{11}c_{21} \int_{T_0} (\partial_x\varphi_j\partial_y\varphi_k + \partial_y\varphi_j\partial_x\varphi_k) \\ + c_{21}^2 \int_{T_0} \partial_y\varphi_j\partial_y\varphi_k. \end{aligned}$$

The other three integrals have an analogous form. Finally, the integrals over the reference element T_0 can be exactly calculated in advance.

P r o o f. As $u_n \in V_h$ is a continuous piecewise linear function, its gradient is a piecewise constant function. To achieve the goal posed in the title of this subsection, we will highly exploit this fact. We will detail our approach just for the second term (4.2), the same works for the other one too.

At first let us just concentrate on the multiplication:

$$\begin{aligned} & \left(\begin{pmatrix}\partial_x u_n\\ \partial_y u_n\end{pmatrix}\cdot\begin{pmatrix}\partial_x\phi_j\\ \partial_y\phi_j\end{pmatrix}\right)\left(\begin{pmatrix}\partial_x u_n\\ \partial_y u_n\end{pmatrix}\cdot\begin{pmatrix}\partial_x\phi_k\\ \partial_y\phi_k\end{pmatrix}\right) \\ & \quad = (\partial_x u_n\partial_x\phi_j + \partial_y u_n\partial_y\phi_j)(\partial_x u_n\partial_x\phi_k + \partial_y u_n\partial_y\phi_k) \\ & \quad = (\partial_x u_n)^2\partial_x\phi_j\partial_x\phi_k + (\partial_x u_n\partial_y u_n)(\partial_x\phi_j\partial_y\phi_k + \partial_y\phi_j\partial_x\phi_k) \\ & \qquad + (\partial_y u_n)^2\partial_y\phi_j\partial_y\phi_k. \end{aligned}$$

The other term (4.1) has a similar, but a bit simpler structure, therefore the above idea can be easily adapted.

Returning back to the original integral form of the stiffness matrix and using the above formulae, we end up at

$$\begin{aligned}
S_{kj} &= \int_\Omega g(|\nabla u_n|^2)\nabla\phi_j\cdot\nabla\phi_k + \int_\Omega 2g'(|\nabla u_n|^2)(\nabla u_n\cdot\nabla\phi_j)(\nabla u_n\cdot\nabla\phi_k)\\
&= \int_\Omega (g(|\nabla u_n|^2))(\partial_x\phi_j\partial_x\phi_k+\partial_y\phi_j\partial_y\phi_k) + \int_\Omega (2g'(|\nabla u_n|^2)(\partial_x u_n)^2)\partial_x\phi_j\partial_x\phi_k\\
&\quad + \int_\Omega (2g'(|\nabla u_n|^2)(\partial_x u_n\partial_y u_n))(\partial_x\phi_j\partial_y\phi_k+\partial_y\phi_j\partial_x\phi_k)\\
&\quad + \int_\Omega (2g'(|\nabla u_n|^2)(\partial_y u_n)^2)\partial_y\phi_j\partial_y\phi_k,
\end{aligned}$$

where the collected terms (in brackets) are constant over each triangle T of our mesh $\mathcal{T}_h$.

This allows us to compute our matrix S as follows.

$$\begin{aligned}
S_{kj} &= \int_\Omega g(|\nabla u_n|^2)\nabla\phi_j\cdot\nabla\phi_k + \int_\Omega 2g'(|\nabla u_n|^2)(\nabla u_n\cdot\nabla\phi_j)(\nabla u_n\cdot\nabla\phi_k)\\
&= \sum_{T\in\mathcal{T}_h} (g(|\nabla u_n|^2))|_T \int_T (\partial_x\phi_j\partial_x\phi_k+\partial_y\phi_j\partial_y\phi_k)\\
&\quad + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_x u_n)^2)|_T \int_T \partial_x\phi_j\partial_x\phi_k\\
&\quad + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_x u_n\partial_y u_n))|_T \int_T (\partial_x\phi_j\partial_y\phi_k+\partial_y\phi_j\partial_x\phi_k)\\
&\quad + \sum_{T\in\mathcal{T}_h} (2g'(|\nabla u_n|^2)(\partial_y u_n)^2)|_T \int_T \partial_y\phi_j\partial_y\phi_k,
\end{aligned}$$

where we denote by $f|_T$ the value of an arbitrary piecewise constant function f over the element $T\in\mathcal{T}_h$. Here, as usual, the integrals of partial derivatives of the basis functions over the elements T can be exactly calculated with the aid of the reference triangle T_0, and the basis functions φ_k.

Since this is well known, we only show this computation in the case of $\int_T \partial_x\phi_j\partial_x\phi_k$.

$$\begin{aligned}
\int_T \partial_x\phi_j\partial_x\phi_k &= \int_{T_0} \partial_x\Big(\varphi_j\Big(C\begin{pmatrix}x\\y\end{pmatrix}+b\Big)\Big)\partial_x\Big(\varphi_k\Big(C\begin{pmatrix}x\\y\end{pmatrix}+b\Big)\Big)\\
&= \int_{T_0} (c_{11}\partial_x\varphi_j + c_{21}\partial_y\varphi_j)(c_{11}\partial_x\varphi_k + c_{21}\partial_y\varphi_k)\\
&= c_{11}^2\int_{T_0}\partial_x\varphi_j\partial_x\varphi_k + c_{11}c_{21}\int_{T_0}(\partial_x\varphi_j\partial_y\varphi_k+\partial_y\varphi_j\partial_x\varphi_k)\\
&\quad + c_{21}^2\int_{T_0}\partial_y\varphi_j\partial_y\varphi_k,
\end{aligned}$$

where the functions φ_k are the basis functions defined on the reference element, and C, b define the affine linear map between T and T_0. These values can clearly be calculated analytically in advance. □

The assembly of the matrices corresponding to the first method can be done in the same way.

R e m a r k 4.2. The idea of using a reference element is also very useful during the calculation of the right-hand side integrals. We only need to transform the nodes of a chosen quadrature rule to the triangle we are currently working on.

We also note here that in our case the integrals over the reference element were exactly calculated with the aid of Maple.

4.3. The smoothing operator G_h. We would like to define a smoothing operator to achieve better approximation of ∇u^*. Namely, as we mentioned before in (3.3), there exists a smoothing operator which satisfies

$$\|\nabla u^* - G_h(\nabla u_h)\| = o(h^2), \quad u_h \in V_h,\ u^* \in H_0^1(\Omega) \text{ is the solution},$$

instead of the usual first order estimate.

Before defining the operator G_h, we recall that $\mathcal{T}_h := \{T_k\}$ are triangulations of the domain $\Omega \subset \mathbb{R}^d$. Now we introduce the space of piecewise constant functions:

$$V_h^{\text{const}} := \{u\colon\ \Omega \to \mathbb{R};\ u_{|_{T_k}} = \text{const},\ \forall\, T_k \in \mathcal{T}_h\}.$$

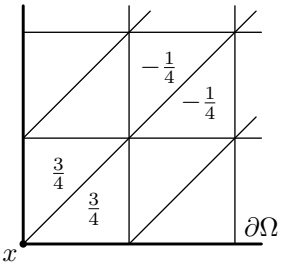
This space is important in the sequel, since if we compute the gradient of a piecewise linear function $u \in V_h$, then it will lie in the space $(V_h^{\text{const}})^d$, i.e., ∇ maps V_h into $(V_h^{\text{const}})^d$.

Later we would like to compare smoothed and original gradients and also apply differential operators to them, see (3.2), therefore the smoothing operator has to map from $(V_h^{\text{const}})^d$ to $(V_h)^d$ to achieve the above goals.

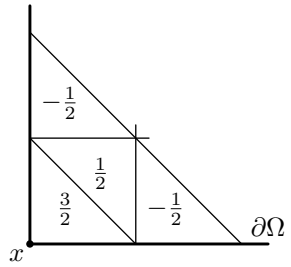
In [9] an appropriate smoothing operator was introduced, defined for $v \in (V_h^{\text{const}})^d$ as follows (and here we immediately drop the components subscript). In the nodes of the mesh we have

$$(G_h(\nabla v))(x) := \sum_{j=1}^{m_t} w_j^x \nabla v_{|_{T_j^x}},$$

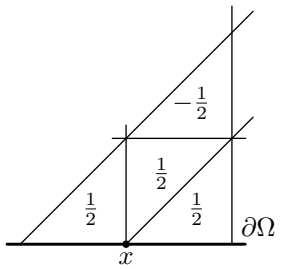
where $t := t(x) \in \{1, 2, \dots, 6\}$ is the number of triangles which contain the node $x \in \Omega$, and m_t is the number of triangles T_j^x sketched in Figure 1 (we only displayed the cases which we will need later), just as the real valued weights w_j^x. Otherwise the function is a piecewise linear interpolation over the nodes.



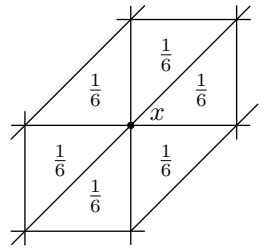
(A) Vertex node type 1



(B) Vertex node type 2



(C) Boundary node



(D) Inner node

Figure 1. Elements and coefficients of the smoothing operator [9].

This smoothing operator is sharp for arbitrary quadratic polynomial v. This and the proof of estimate (3.3) can also be found in [9].

R e m a r k 4.3. We mention here that in this case the operator G_h can be represented by a matrix, but we do not need it exactly, just its action over a vector. This can be done by following the ideas of the assembly, but for the nodes. (The same idea was mentioned in Remark 4.1, but for the gradient.)

4.4. The auxiliary problem and its solution, mesh refinement and prolongation. We would like to weakly solve our linear auxiliary problem

$$\begin{cases} -\Delta w = -\operatorname{div} z^*, \\ w|_{\partial\Omega} = 0, \end{cases}$$

on a finer mesh exploiting the property of $\mathrm{EST}(\cdot;\cdot,\cdot)$ from Remark 3.2, namely

$$\int_\Omega \nabla w \cdot \nabla v = \int_\Omega z^* \cdot \nabla v, \quad v \in V_{h_f} \subset H_0^1(\Omega), \tag{4.3}$$

where, instead of our original triangulation $\mathcal{T}_h := \{T_k;\ k = 1, 2, \ldots, n(h)\}$ of $\Omega \subset \mathbb{R}^N$, we define a finer mesh $\mathcal{T}_{h_f} := \{T_k;\ k = 1, 2, \ldots, n(h_f)\}$, where the ratio $\varrho = h/h_f$

is given, and every triangle of $\mathcal{T}_h$ is divided into ϱ^2 similar triangles, these latter ones form $\mathcal{T}_{h_f}$. The unique weak solution of (4.3), denoted by w^*, is the optimal parameter w corresponding to u_h.

Again, because of the involved norms and the easier computation of the right-hand side of (4.3), it is convenient to use prolongation operators from multigrid theory, especially the theory of nested finite element subspaces plays an important role here.

We briefly recall these ideas here. On the coarser mesh we still use the space V_h, and on the finer one we use the subspace V_{h_f}, namely we have

$$V_h := \operatorname{span}\{v_1, v_2, \dots, v_{n(h)}\}, \quad V_{h_f} := \operatorname{span}\{v_1^f, v_2^f, \dots, v_{n(h_f)}^f\}$$

of the usual linear Lagrangian basis functions. Hence, the important relation

$$V_h \subset V_{h_f}$$

holds. This yields the following form of the basis functions $v_k \in V_h$:

$$v_k(x) = \sum_{j=1}^{n(h_f)} \beta_{jk} v_j^f(x), \quad k = 1, 2, \dots, n(h),$$

which in matrix form is $\underline{v}(x) = \beta\ \underline{v^f}(x)$.

Now let $u := \sum_{j=1}^{n(h)} y_j v_j \in V_h$, then by the above equation, we have

$$u(x) = \underline{y}^T \underline{v}(x) = \underline{y}^T (\beta \underline{v^f}(x)) = (\underline{y^f})^T \underline{v^f}, \quad \text{if } \underline{y^f} = \beta^T \underline{y}.$$

Hence, by β^T we defined the prolongation operator $P\colon\ V_h \to V_{h_f}$.

With the aid of this operator the assembly of matrices and vectors to solve (4.3) is faster, since the standard techniques (from Proposition 4.1 and the following remark) are applicable.

R e m a r k 4.4. (i) The advantage of this approach is not just its convenience, but mainly it is the fact that if we avoid this technique, then during the computation of the values

$$\int_\Omega z^* \cdot \nabla v_k^f, \quad k = 1, 2, \dots, n(h),$$

we have to calculate the values of $z^* \in V_h$ in the nodes of the quadrature rule on the finer triangles. The creation of a finer mesh (i.e., nodes and elements) is also inevitable, but if we compute them the calculation of the prolongation operator is nearly at no cost.

(ii) Its advantage is more significant if we use higher order finite element basis functions, or we apply any but the standard refinement, e.g., in some adaptive *hp*-FEM solvers.

4.5. The norms in the estimator $\mathrm{EST}(\cdot;\cdot,\cdot)$**.** The three involved norms of the estimator from (3.1) and (3.2), namely $\|\cdot\|_{L^2(\Omega)}$, $\|\cdot\|_{L^2(\Omega)^d}$ and $\|\cdot\|_{L^\infty(\Omega)^d}$, and the scalar product in $L^2(\Omega)^d$ can be computed by an assembly-based idea.

Suppose that we have a quadrature rule defined over the reference triangle, then it is easy to compute the values of the basis functions in the given quadrature nodes. By going over the triangles of our mesh one can easily compute the values of the functions in the norm (or in the inner product).

Since one has to compute many integrals numerically, it is crucial to use a quadrature rule that requires as few function evaluations as posible for both the basis functions and others (g, b or f). This can be achieved by rules where a node and the corresponding value can be used multiple times, e.g., points over the element's boundary. In our case we used a simple three-point rule using the vertices of the triangles with weights $1/3$.

5. Numerical performance of the estimator

5.1. Test problem. We consider (2.1), where $\Omega = (0,1)^2$, and our model problem has the source of nonlinearity g given by

$$g(\eta) := \begin{cases} \dfrac{1.02}{1+\sqrt{1-\eta/3}}, & \text{if } 0 \leqslant \eta \leqslant \eta_0 := 2.6; \\ g(\eta_0) \approx 0.7951, & \text{if } \eta \geqslant \eta_0. \end{cases} \tag{5.1}$$

The reason to cut the function g is to ensure that the nonlinear operator above is defined on $H^2(\Omega) \cap H_0^1(\Omega)$. Finally, the right-hand side b is chosen so that the exact classical solution u^* is known precisely. For more details of this problem the interested reader is referred to [6]. This kind of quasilinear problems is also considered in [2].

We also need the constants appearing in the estimator, namely M, m, L and c_Ω from Theorem 3.1.

Theorem 5.1. *Let Ω be the domain given above. If in addition the function g is given as in* (5.1)*, then the following statements are true for the operator F:*

(i) *There exist constants $M \geqslant m > 0$, independent of u and p, such that the derivative of F satisfies*

$$m\|p\|^2 \leqslant \langle F'(u)p, p\rangle \leqslant M\|p\|^2,$$

where

$$m = g(0) = 0.51 \quad \text{and} \quad M = g(\eta_0) + 2g'(\eta_0)\eta_0 \approx 2.046213.$$

(ii) *The derivative* $F'\colon W^{1,\infty}(\Omega) \to B(H^1_0(\Omega))$ *is also Lipschitz continuous with constant*

$$L \approx 11.935094.$$

(iii) *The Poincaré-Friedrichs constant is* $c_\Omega = 2/\pi$.

P r o o f. (i) Since the function g and its derivative g' are monotonically increasing functions on $[0, \eta_0]$, constants elsewhere, and both nonnegative, the lower bound

$$\begin{aligned}\langle F'(u)p, p\rangle &\geqslant \int_\Omega (g(|\nabla u|^2)|\nabla p|^2 + 2g'(|\nabla u|^2)(\nabla u \cdot \nabla p)^2)\\ &\geqslant g(0)\|p\|^2_{H^1_0(\Omega)}, \quad u, p \in H^1_0(\Omega)\end{aligned}$$

is straightforward, i.e., $m = g(0) = 0.51$. The upper bound comes from the estimate

$$\begin{aligned}\langle F'(u)p, p\rangle &\leqslant \int_\Omega (g(|\nabla u|^2) + 2g'(|\nabla u|^2)|\nabla u|^2)|\nabla p|^2 \leqslant\\ &\leqslant (g(\eta_0) + \widetilde{M})\|p\|^2_{H^1_0(\Omega)}, \quad u, p \in H^1_0(\Omega),\end{aligned}$$

where $\widetilde{M} := \max\limits_{\eta\in[0,\eta_0]} |2g'(\eta)\eta|$, since g' vanishes outside the interval $[0, \eta_0]$. The value of $\widetilde{M}$ is $2g'(\eta_0)\eta_0$, and therefore $M = g(\eta_0) + \widetilde{M} \approx 2.046213$.

(ii) According to ([10], Section 3) the derivative of $g(r^2)r$ is Lipschitz continuous if the following conditions hold:

$$0 < m \leqslant g(r) \leqslant M, \quad 0 < m \leqslant (g(r^2)r)' \leqslant M \quad \forall\, r \geqslant 0, \tag{5.2}$$

$$|(g(r^2)r)''| \leqslant L_1 \quad \forall\, r \geqslant 0, \tag{5.3}$$

and the Lipschitz constant is $L := \max\{L_1, 3L_2\}$, where $L_2 := \sup\limits_{r\geqslant 0}(g(r^2))'$. This shows that under the above conditions the weak operator F' is also Lipschitz continuous with the same constant L.

The estimates in (5.2) easily follow from (i) above, with the same bounds. To compute L_1 and L_2 we again use the facts that g, g' and g'' are monotonically increasing non negative functions on $[0, \eta_0]$, hence we have

$$|(a(r^2)r)''| = 6a'(r^2)r + 4a''(r^2)r^3 \leqslant 6a'(\eta_0)\eta_0^{1/2} + 4a''(\eta_0)\eta_0^{3/2} =: L_1.$$

and

$$(a(r^2))' = 2a'(r^2)r \leqslant 2a'(\eta_0)\eta_0^{1/2} =: L_2.$$

The computed constants are

$$L_1 \approx 11.935094 \quad \text{and} \quad L_2 \approx 0.805631.$$

Hence, $L = L_1 \approx 11.935094$.

(iii) This follows from the sharp constant of Steklov's inequality over the interval $[0, 1]$. □

5.2. Numerical results. Our experiments were carried out in the following way:

- ▷ The FEM discretization was done by using linear Courant elements over a (not necessarily) uniform mesh (of squares divided into two equal triangles, as in Figure 1).
- ▷ We carried out element-by-element assembly, with the aid of a reference element. The numerical integrations were done with a sufficient order, just as the numerical differentiation.
- ▷ The Newton-type methods were damped. We used different piecewise constant coefficient variable preconditioners: the domain was decomposed into at most $d = 4$ pieces.
- ▷ The stopping criterion for the nonlinear solvers was

$$\frac{\|F_h(u_n) - b_h\|_{H_0^1(\Omega)}}{\|F_h(u_0) - b_h\|_{H_0^1(\Omega)}} < 10^{-10}.$$

- ▷ The code was written in MATLAB and the auxiliary linear algebraic systems were solved using the built-in solver `\mldivide`.

The following tables and their corresponding plots show the error functional compared to the estimator, also the effectivity index is displayed, while the log-log plots only show the values of $E(.)$ and $\mathrm{EST}(\cdot;\cdot,\cdot)$ (the true error E is marked by $*$).

The CPU times were reasonable but they are not displayed here, since our numerical tests were carried out on a standard desktop computer.

5.2.1. One-dimensional case. Here we set the right-hand side function such that our exact solution u^* is the bubble function $x(1 - x)$.

In Table 1 we can see the basic results, where the auxiliary problem (the one which yields w^*) is solved on the original mesh, i.e., $\varrho = 1$.

According to Remark 3.2, in Tables 2 and 3 we refine our mesh by a ratio ϱ to solve the auxiliary problem. The improving effect of this step can be nicely seen in the tables and even better in the plots.

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.055037791312483	371.914078575348410	6757.4310
2	0.013856122697177	19.345738838764426	1396.1870
3	0.003470203009080	1.329380121949488	383.0843
4	0.000867929813010	0.098033794168086	112.9513
5	0.000217006037276	0.007817070596375	36.0224
6	0.000054252981611	0.000699770435089	12.8983
7	0.000013563337394	0.000073833802861	5.4436
8	0.000003390840098	0.000009569410155	2.8221
9	0.000000847710384	0.000001531570408	1.8067
10	0.000000211927618	0.000000291727910	1.3765

Table 1. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^0$ (1-dimensional case)

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.055037791312483	114.652692531208420	2083.1630
2	0.013856122697177	5.684624286593208	410.2608
3	0.003470203009080	0.405274348836577	116.7869
4	0.000867929813010	0.031888528183674	36.7409
5	0.000217006037276	0.002830004525359	13.0411
6	0.000054252981611	0.000297136298263	5.4769
7	0.000013563337394	0.000038405295416	2.8316
8	0.000003390840098	0.000006137233652	1.8099
9	0.000000847710384	0.000001168004210	1.3778
10	0.000000211927618	0.000000250505246	1.1820

Table 2. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^1$ (1-dimensional case)

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.055037791312483	8.678976984913346	157.6912
2	0.013856122697177	0.277163428649300	20.0030
3	0.003470203009080	0.023957199535661	6.9037
4	0.000867929813010	0.002739424227838	3.1563
5	0.000217006037276	0.000411223495871	1.8950
6	0.000054252981611	0.000076207701695	1.4047
7	0.000013563337394	0.000016168017295	1.1920
8	0.000003390840098	0.000003707872800	1.0935
9	0.000000847710384	0.000000886810389	1.0461
10	0.000000211927618	0.000000216782162	1.0229

Table 3. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^4$ (1-dimensional case)

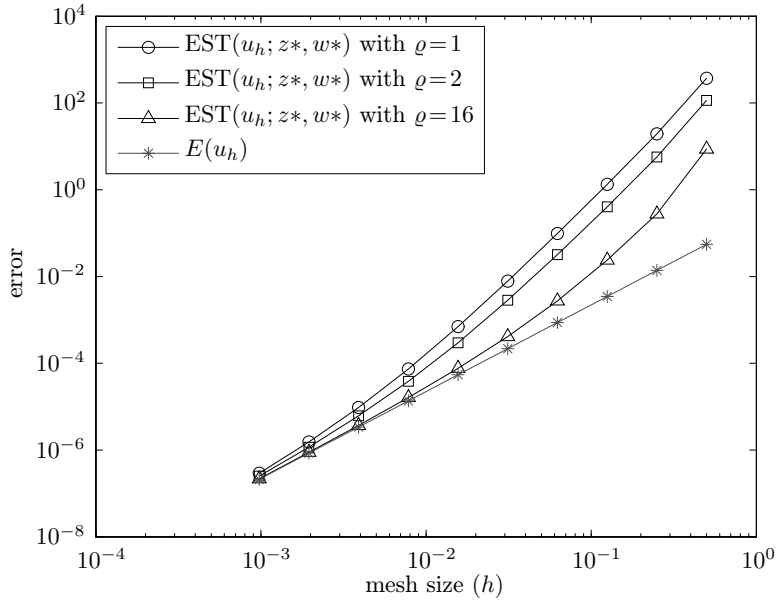


Figure 2. Log-log plot of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for different values of the mesh refinement parameter ϱ (one-dimensional case)

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.006666023505318	0.451035483031462	67.6619
2	0.002102844038482	0.487467362953306	231.8134
3	0.000553382662118	0.071086894070267	128.4588
4	0.000140081092830	0.007876048605962	56.2249
5	0.000035128827031	0.000943996768005	26.8724
6	0.000008788992755	0.000135996305551	15.4735
7	0.000002197672335	0.000023652672057	10.7626

Table 4. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^0$ (2-dimensional case)

5.2.2. Two-dimensional case. Switching to the two-dimensional case, our solution is now $u^*(x, y) = 16x(1-x)y(1-y)$. This time the efficiency of the estimator can be better seen on the effectivity index in the tables. The idea of refining the mesh of the auxiliary problem, see Remark 3.2, is more important now.

The best effectivity index in the two-dimensional case is not as good as in the one-dimensional case, which is due to the lack of computational capacity, but we expect the same good results for finer meshes ($k = 8, 9, 10, \ldots$). The coarse mesh ($k \leqslant 4$) accuracy could be improved by applying a rule with more nodes, or some composite quadrature rule over the reference triangle.

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.006666023505318	0.375125295146493	56.2742
2	0.002102844038482	0.351096532462108	166.9627
3	0.000553382662118	0.048700998105800	88.0060
4	0.000140081092830	0.005456781742943	38.9544
5	0.000035128827031	0.000700434137849	19.9390
6	0.000008788992755	0.000110502057647	12.5728
7	0.000002197672335	0.000020819161520	9.4733

Table 5. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^1$ (2-dimensional case)

$h = 1/2^k$, $k =$	$E(u_h)$	$\mathrm{EST}(u_h, y^*, w^*)$	effectivity index
1	0.006666023505318	0.337032061419479	50.5597
2	0.002102844038482	0.292017331281910	138.8678
3	0.000553382662118	0.039548543368707	71.4669
4	0.000140081092830	0.004430256762882	31.6264
5	0.000035128827031	0.000591643020265	16.8421
6	0.000008788992755	0.000098619829525	11.2208
7	0.000002197672335	0.000015934911938	7.2508

Table 6. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for the refined mesh by a ratio $\varrho = 2^2$ (2-dimensional case)

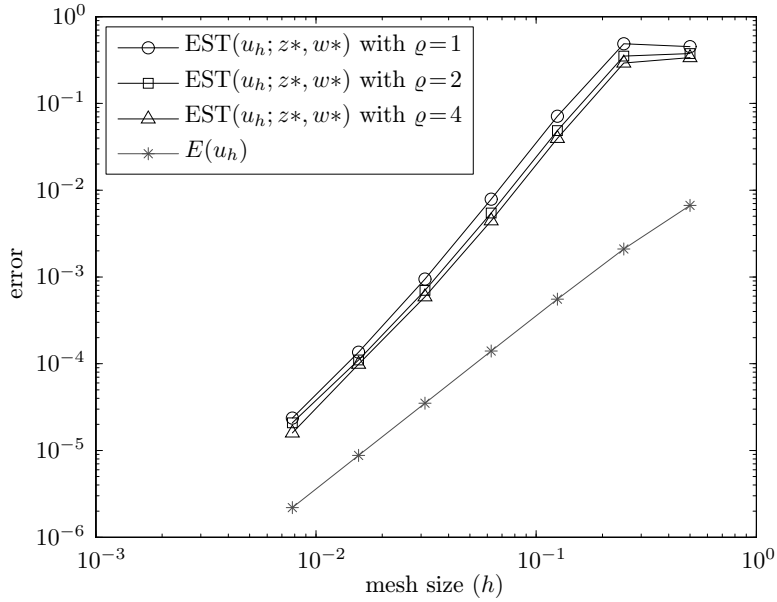


Figure 3. Comparison of $E(u_h)$ and $\mathrm{EST}(u_h; z^*, w^*)$ for different values of the mesh refinement parameter ϱ (2 dimensional case)

The first and second values in the tables are very close, since for $k = 1$ the mesh has only boundary nodes, hence the FEM approximation is "very good" in this irrelevant case due to the Dirichlet boundary conditions.

6. Conclusions

Our experiments show that this estimator cooperates well with different nonlinear iterative solvers, and that it is indeed efficient and highly applicable for a posteriori error estimation.

The sharpness of the estimation is in a close connection with the accuracy of the numerical solution of the auxiliary problem (which yields the parameter w^*), to be precise its accuracy highly depends on the applied mesh refinement.

Our experiments suggest that the sharpness of the estimator in higher dimensions can be increased in two ways. One can raise the computational capacity or parallelize the method, or apply a better smoothing operator, e.g., one that is sharp for higher order polynomials, therefore decreasing the error coming from the numerical solution of the auxiliary problem (4.3).

Another strength of the estimator, which was numerically observed, is that both in EST and D (i.e., (3.1) and (3.2)) the dominant terms are multiplied by a factor which can be easily decreased by refining the mesh of the auxiliary problem.

Altogether we can see that this a posteriori error estimator is both efficient and sharp, it cooperates well with different iterative methods, and that it requires just a few parameters to compute (the constants m, M, L and c_Ω). We were also able to demonstrate the importance of the auxiliary problem.

References

[1] *M. Ainsworth, J. T. Oden*: A Posteriori Error Estimation in Finite Element Analysis. Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs, and Tracts, Chichester, Wiley, 2000. zbl MR

[2] *O. Axelsson, J. Maubach*: On the updating and assembly of the Hessian matrix in finite element methods. Comput. Methods Appl. Mech. Eng. *71* (1988), 41–67. zbl MR

[3] *R. Becker, R. Rannacher*: A feed-back approach to error control in finite element methods: Basic analysis and examples. East-West J. Numer. Math. *4* (1996), 237–264. zbl MR

[4] *C. Brezinski*: A classification of quasi-Newton methods. International Conference on Numerical Algorithms, Vol. I (Marrakesh, 2001). Numer. Algorithms *33* (2003), 123–135. zbl MR

[5] *I. Faragó, J. Karátson*: Numerical Solution of Nonlinear Elliptic Problems via Preconditioning Operators. Theory and Applications. Advances in Computation: Theory and Practice 11, Nova Science Publishers, Huntington, 2002. zbl MR

[6] *I. Faragó, J. Karátson*: The gradient-finite element method for elliptic problems. Numerical Methods and Computational Mechanics (Miskolc, 1998). Comput. Math. Appl. *42* (2001), 1043–1053. zbl MR

[7] *W. Han*: A Posteriori Error Analysis via Duality Theory. With Applications in Modeling and Numerical Approximations. Advances in Mechanics and Mathematics 8, Springer, New York, 2005. zbl MR
[8] *A. Hannukainen, S. Korotov*: Techniques for a posteriori error estimation in terms of linear functionals for elliptic type boundary value problems. Far East J. Appl. Math. *21* (2005), 289–304. zbl MR
[9] *I. Hlaváček, M. Křížek*: On a superconvergent finite element scheme for elliptic systems, I. Dirichlet boundary condition. Apl. Mat. *32* (1987), 131–154. zbl MR
[10] *J. Karátson*: On the Lipschitz continuity of derivatives for some scalar nonlinearities. J. Math. Anal. Appl. *346* (2008), 170–176. zbl MR
[11] *J. Karátson, I. Faragó*: Variable preconditioning via quasi-Newton methods for nonlinear problems in Hilbert space. SIAM J. Numer. Anal. (electronic) *41* (2003), 1242–1262. zbl MR
[12] *J. Karátson, S. Korotov*: Sharp upper global a posteriori error estimates for nonlinear elliptic variational problems. Appl. Math., Praha *54* (2009), 297–336. zbl MR
[13] *J. Karátson, B. Kovács*: Variable preconditioning in complex Hilbert space and its application to the nonlinear Schrödinger equation. Comput. Math. Appl. *65* (2013), 449–459. MR
[14] *S. Korotov*: Global a posteriori error estimates for convection-reaction-diffusion problems. Appl. Math. Modelling *32* (2008), 1579–1586. zbl MR
[15] *B. Kovács*: A comparison of some efficient numerical methods for a nonlinear elliptic problem. Cent. Eur. J. Math. *10* (2012), 217–230. zbl MR
[16] *S. G. Mikhlin*: Constants in Some Inequalities of Analysis. Transl. from the German. A Wiley-Interscience Publication, John Wiley & Sons, Chichester, 1986. zbl MR
[17] *P. Neittaanmäki, S. Repin*: Reliable Methods for Computer Simulation. Error Control and a Posteriori Estimates. Studies in Mathematics and its Applications 33, Elsevier, Amsterdam, 2004. zbl MR
[18] *S. I. Repin*: A posteriori error estimation for nonlinear variational problems by duality theory. J. Math. Sci., New York *99* (2000), 927–935; Transl. from the Russian. Zap. Nauchn. Semin. POMI *243* (1997), 201–214. zbl MR
[19] *R. Verfürth*: A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques. Wiley-Teubner Series Advances in Numerical Mathematics, John Wiley & Sons, Stuttgart, Chichester, 1996. zbl
[20] *V. S. Vladimirov*: Equations of Mathematical Physics. Transl. from the Russian. Mir, Moskva, 1984. MR
[21] *E. Zeidler*: Nonlinear Functional Analysis and its Applications. III: Variational Methods and Optimization. Springer, New York, 1985. zbl MR

Author's address: *Balázs Kovács*, Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University, Pázmány P. sétány 1/C, 1117 Budapest, Hungary, e-mail: `koboaet@cs.elte.hu`; MTA-ELTE Numerical Analysis and Large Networks Research Group, www.cs.etle.hu/numnet.