Effective Minimization of Functional Majorant in A Posteriori Error Analysis

Jan Valdman*

June 3, 2008

Abstract

We consider a Poisson problem and its functional a posteriori estimate derived in [20]. The estimate majorizes the L_2 norm of the error of the discrete solution computed by FEM method and contains a free variable from a $H(div)$ space. In order to keep the estimate sharp, the majorant term is minimized with respect to the free variable. A minimization procedure is introduced, containing a solution of linear system of equations as its computationally most expensive part. The linear system is efficiently solved using a conjugate gradient method with a multigrid as a preconditioner. All numerical techniques including the computation of the constant from the Korn's inequality as a part of majorant estimate are demonstrated on one benchmark example.

1 Introduction

A priori rate convergence estimates for finite element approximations of such problems has been investigated in 70s-80s (see e.g., [13]). However, adaptive multi-level algorithms require a posteriori estimates able to (a) provide a reliable and directly computable estimate of the approximation error and (b) efficient error indicator that detects the regions with excessively high errors.

In the recent decades, a posteriori estimates for linear elliptic and parabolic problems were intensively investigated. A reader will find a systematic exposition of the main approaches to a posteriori error estimation of finite element approximations (such as residual or gradient averaging methods) in

*Special Research Program SFB F013 'Numerical and Symbolic Scientific Computing', supported by the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)', at the Johannes Kepler University Linz, Altenbergerstrasse 69, A-4040 Linz; email: jan.valdman@sfb013.uni-linz.ac.at

[1, 6, 7, 5, 4, 11, 27] and in the literature cited therein.

In this paper, a posteriori estimates that majorate the difference between exact solution of a linear elliptic equation and any function in the admissible (energy) class are studied. For the class of uniformly convex variational problems computable error majorants (for any conforming approximation) were derived by the variational techniques in the mid 90th using duality theory of the calculus of variations. Key publications related to this subject are [20, 25, 22]. Another "nonvariational" method was introduced in [23]. In this paper it was stated that for linear elliptic problems both methods lead to the same error majorants. Later it was applied to many problems, including parabolic equations and nonliner problems [8, 9, 14, 15, 24].

As an example of demonstration, let us consider a scalar boundary value (Poisson) problem

$$\triangle u + f = 0 \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega \tag{1}$$

for the searched function u. The right hand side $f \in L_2(\Omega)$ and an open bounded domain $\Omega \in \mathbb{R}^d$, where d denotes a domain dimension, i.e., $d = 1, 2, 3$ are given. Assume that $v \in H_0^1$ is some approximation of u in a Sobolev space $H_0^1(\Omega)$. Then, a functional error estimate

$$\|\nabla(u - v)\|_\Omega \leq \|\nabla v - y\|_\Omega + C_\Omega \|\mathrm{div} y + f\|_\Omega \tag{2}$$

holds for all functions $y \in H(\Omega, \mathrm{div})$, $\|\cdot\|_\Omega$ denotes the $L_2(\Omega)$ norm. Note that no mesh–dependent constants or any assumptions on regularity of an exact solution are contained in this estimate. The only global constant C_Ω included represents a constant from the Fridrichs' inequality

$$\|w\|_\Omega^2 \leq C_\Omega^2 \|\nabla w\|_\Omega^2 \tag{3}$$

which holds for all $w \in H_0^1(\Omega)$. Thus the constant C_Ω only depends on the domain Ω and can be precomputed (it is demonstated in subsection 3.1).

The functional a posteriori estimate (2) holds for any $y \in H(\Omega, \mathrm{div})$ and its quality for a particular choice of y is measured by an index of efficiency

$$I_{\mathrm{eff}} := \frac{\|\nabla v - y\|_\Omega + C_\Omega \|\mathrm{div} y + f\|_\Omega}{\|\nabla(u - v)\|_\Omega}. \tag{4}$$

Under the assumption of an exact solution u and exact integrations of the norm terms it muss hold $I_{\mathrm{eff}} \geq 1$ with an equality for the particular choice $y = \nabla u$, i.e., y is chosen as the flux of an exact solution.

There are several ways [21] how to compute the flux y approximation from the discrete solution v in order to substitute y to (2):

1) Averaging on the mesh of the discrete solution. From a known discrete approximation v of the solution u we choose the testing function $y = \mathcal{G}v$, where $\mathcal{G}$ represents an averaging gradient operator, see e.g., [11] for more details. This is a cheap method providing some preliminary knowledge on the upper bound of the error.

2) Averaging on a refined mesh. This method is similar to 1), only with the difference that the averaging is done for the the solution calculated on once more (or more times) refined mesh. This method can be regarded as a quantitative form of the Runge's rule. This method is more expensive, but provides (generally not always) sharper results.

3) Using partially equilibrated fluxes. By postprocessing of v, a function $y_{\bar{f}}$ is constructed such that $\mathrm{div} y_{\bar{f}} + \bar{f} = 0$ and $\bar{f}$ is sufficiently close to f in $L^2(\Omega)$ norm. Then, the substitution of $y = y_{\bar{f}} + \tau$ into (2) provides an estimate

$$\|\nabla(u-v)\|_\Omega \leq \left\|\nabla v + \mathrm{curl}\tau - y_{\bar{f}}\right\|_\Omega + C_\Omega \left\|f - \bar{f}\right\|_\Omega,$$

where $\tau \in H(\Omega, \mathrm{curl})$ is arbitrary.

4) Minimization of the majorant with respect to y on the mesh of the discrete solution v. This is the most expensive method for a detailed knowledge of the error.

A comparison of methods 1), 2) and 4) for a broader class of problems with nonlinear boundary conditions can be found e.g., in [24]. An equilibration techniques based on computation of y as a solution of small local problems are documented e.g., in [10] and [27].

This paper is focused merely on the method 4), i.e., the minimization of the majorant on the same mesh, in particularly from the point of their efficient numerical implementation.

2 Majorant minimization problem

Let us assume to know a discrete solution $v \in H_0^1(\Omega)$ corresponding to a problem (1) with a given f. Then, the choice of the "flux" function $y \in H(\Omega, \mathrm{div})$ determines how small value of the nominator in I_{eff}, is and therefore how sharp the estimate (2) is.

Problem 1 (Minimization problem). *Given a discrete solution $v \in H_0^1$, a right-hand side of the Poisson problem $f \in L_2(\Omega)$, the Friedrich's constant belonging to the domain Ω. Find a function $y \in H(\Omega, \mathrm{div})$ satisfying the condition*

$$\|\nabla v - y\|_\Omega + C_\Omega \|\mathrm{div} y + f\|_\Omega \rightarrow min. \tag{5}$$

In order to avoid complications with the nondiferentibiality of norm terms in (5), we apply the inequality $(a+b)^2 \leq (1+\beta)a^2 + (1+\frac{1}{\beta})b^2$ valid for all $\beta > 0$ to obtain

$$(\|\nabla v - y\|_\Omega + C_\Omega \|\mathrm{div} y + f\|_\Omega)^2 \leq \mathcal{M}(\nabla v, f, \beta, C_\Omega, y), \tag{6}$$

where the upper bound in (6) denotes a functional majorant

$$\mathcal{M}(\nabla v, f, \beta, C_\Omega, y) := (1+\beta)\|\nabla v - y\|_\Omega^2 + (1+\frac{1}{\beta})C_\Omega^2 \|\mathrm{div} y + f\|_\Omega^2 .$$

The majorant arguments f, v, C_Ω are known and only $\beta > 0$ and $y \in H(\Omega, \mathrm{div})$ are free arguments. In particular case, for a fixed choice of β, the majorant represents a quadratic functional in unknown flux y. On the other hand, for a fixed y, the parameter

$$\beta = \frac{C_\Omega \|\mathrm{div} y + f\|_\Omega}{\|\nabla v - y\|_\Omega} \tag{7}$$

minimizes the majorant $\mathcal{M}$ amongst all positive β. It suggests the following solution algorithm to Problem 1.

Algorithm 1 (Majorant minimization algorithm)**.** *Given $\beta > 0$.*

(a) Compute y from the minimization of the quadratic problem

$$\mathcal{M}(\nabla v, f, \beta, C_\Omega, y) \to \mathit{min} \tag{8}$$

(b) Upgrade β from y using the formula (7)*. If the convergence in y is not achieved then goto to step (a).*

For the detailed analysis of step (a) of Algorithm 1, it is convenient to decompose $\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$ in its y-dependent and independent parts

$$\begin{aligned} \mathcal{M}_1 &:= (1+\beta)\|\nabla v\|_\Omega^2 + (1+\frac{1}{\beta})C_\Omega^2 \|f\|_\Omega^2 , \quad (9) \\ \mathcal{M}_2 &:= (1+\beta)\Big(\|y\|_\Omega^2 - 2(\nabla v, y)\Big) + (1+\frac{1}{\beta})C_\Omega^2 \Big(\|\mathrm{div} y\|_\Omega^2 + 2(f, \mathrm{div} y)\Big), \end{aligned}$$

where $(\cdot,\cdot)$ denotes the $L_2(\Omega)$ scalar product. Then, one can replace the minimization problem (8) by

$$\mathcal{M}_2(\nabla v, f, \beta, C_\Omega, y) \to \mathrm{min}. \tag{10}$$

Since $\mathcal{M}_2 = 0$ for the choice $y = 0$, the minimum of (10) must be nonpositive. Besides, it holds $\mathcal{M}_1 \geq 0$.

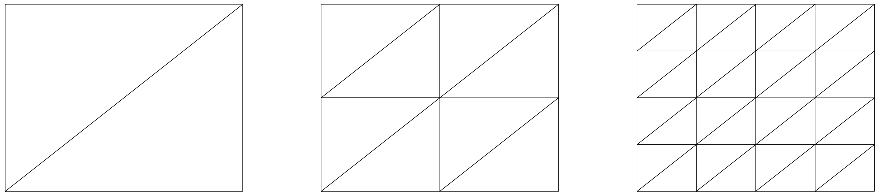

Figure 1: Uniform triangulations at levels 0 (left), 1 (middle), and 2 (right).

2.1 Discretization

The finite element method is used to provide discrete approximations of the minimization problem above. The domain Ω is divided by a regular triangulation $\mathcal{T}$ in triangles in the sense of Ciarlet [12], i.e., $\mathcal{T}$ is a finite partition of Ω into closed triangles; two distinct elements T_1 and T_2 are either disjoint, or $T_1 \cap T_2$ is a complete edge or a common node of both T_1 and T_2. The bilinear forms in (10) read after the discretization (using a given $H(\Omega, \mathrm{div})$ finite element basis)

$$\|y\|_\Omega^2 = \hat{y}^T M \hat{y}, \quad \|\mathrm{div} y\|_\Omega^2 = \hat{y}^T DIVDIV \hat{y}, \tag{11}$$

where M and $DIVDIV$ represent the "mass" and "divdiv" matrices, whereas the linear forms discretize as

$$(\nabla v, y) = l_1^T \hat{y}, \ \ (f, \mathrm{div} y) = l_2^T \hat{y},$$

where $\hat{y}$ denotes a vector representing y in the considered $H(\Omega, \mathrm{div})$ finite element basis. With this discretization, the y-dependent majorant part reads

$$\mathcal{M}_2 = \hat{y}^T \left[(1+\beta)M + (1+\frac{1}{\beta})C_\Omega^2 DIVDIV \right] \hat{y} - \\ 2\hat{y}^T \left[(1+\beta)l_1 - (1+\frac{1}{\beta})C_\Omega^2 l_2 \right] \tag{12}$$

and its minimization with respect to $\hat{y}$ leads to the linear system of equations

$$\left[(1+\beta)M + (1+\frac{1}{\beta})C_\Omega^2 DIVDIV \right] \hat{y} = (1+\beta)l_1 - (1+\frac{1}{\beta})C_\Omega^2 l_2. \tag{13}$$

The next example explains a practical application of the majorant estimate (2) and of Algorithm 1.

3 Example

Let us assume the external force $f(x, y) = 2x(1-x) + 2y(1-y)$ in the unit square domain $(x, y) \in \Omega = (0, 1) \times (0, 1)$. For this special setup, the exact solution of (1) and its flux read

$$
\begin{aligned}
u &= x(1-x)y(1-y), \\
\frac{\partial u}{\partial x} &= (1-2x)y(1-y), \\
\frac{\partial u}{\partial y} &= x(1-x)(1-2y)
\end{aligned}
\tag{14}
$$

for all $(x, y) \in \Omega$. In order to approximate (in this case known) exact solution numerically, the finite element method is applied. The unit square geometry is discretized using the sequence of nested uniform meshes as displayed in Figure 1. Then, a discrete solution v is computed by using the nodal linear (Courant) ansatz functions on each triangular mesh, see the left column pictures of Figure 2.

3.1 Friedrich's constant and its computation

level	elements	C_Ω
0	2	0.17677669529664
1	8	0.20912551729722
2	32	0.22083318837774
3	128	0.22400032086691
4	512	0.22480827985755
5	2048	0.22501130936667
6	8192	0.22506213224527
7	32768	0.22507484206717
theory	∞	0.22507907903928

Table 1: Friedrich's constant for the unit square domain computed from the generalized eigenvalue problem (16) on nested uniform meshes.

The computation of (2) requires the knowledge of the constant C_Ω^2 from the Friedrich's inequality (3). The discretization of this inequality (on a given triangulation of Ω) reads

$$\hat{w}^T M_\triangle \hat{w} \le C_\Omega^2 \hat{w}^T K_\triangle \hat{w}, \tag{15}$$

where $K_\triangle$ and $M_\triangle$ represent a stiffness and mass matrices of the Poisson problem (1). Thus C_Ω^2 represent the maximum of all eigenvalue λ of the generalized eigenvalue problem

$$M_\triangle \hat{w} = \lambda K_\triangle \hat{w}. \tag{16}$$

Table 1 reports on approximate values of C_Ω using nested uniform meshes and nodal linear (Courant) ansatz functions and compares them with a

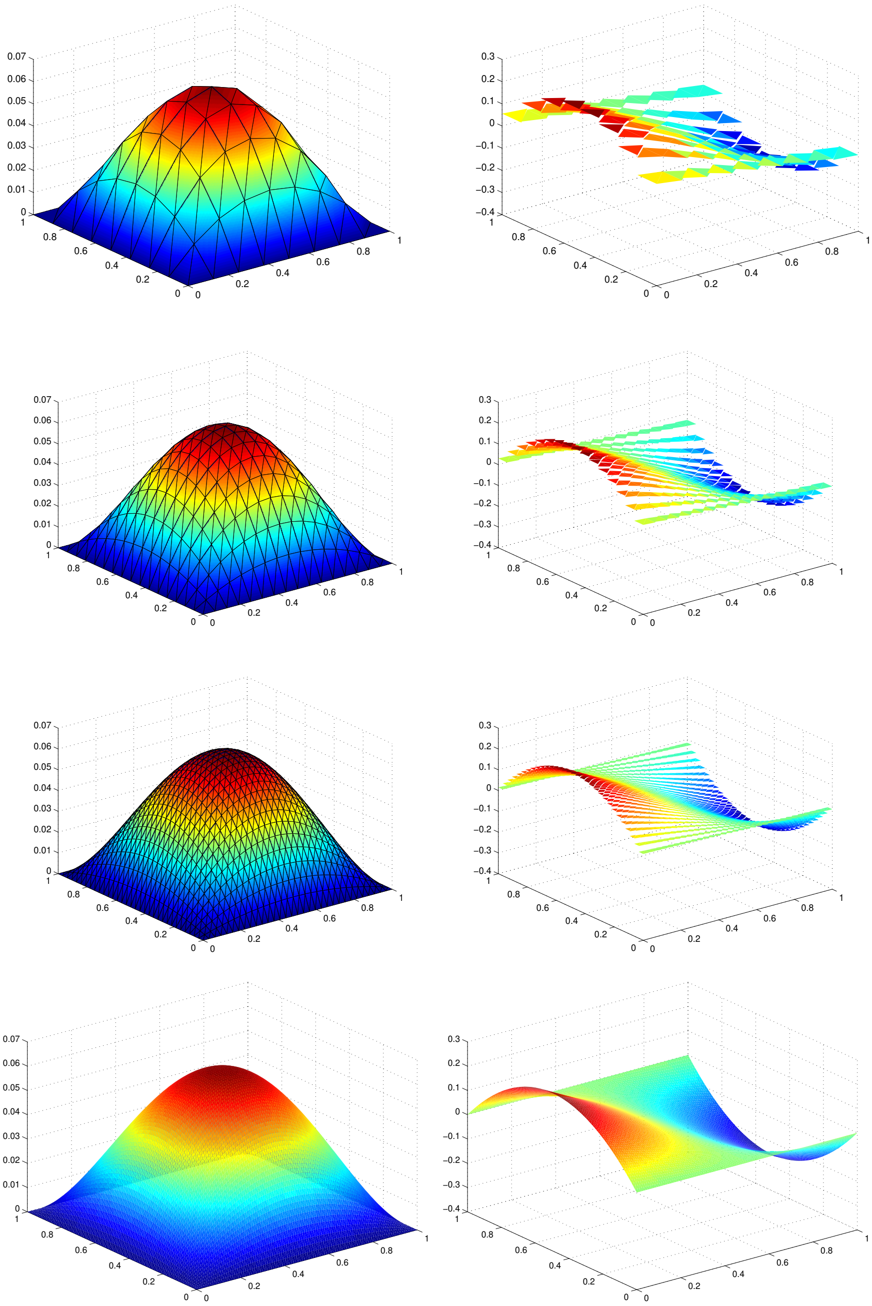

Figure 2: Discrete solutions v and y-component of the flux y computed from using Algorithm 1 on mesh levels 2, 3, 4 (the first three rows from above) and the exact solution u and its flux y-component $\frac{\partial u}{\partial y}$ (bottom row).

level	DOF v	$\|\nabla(u-v)\|_\Omega^2$	DOF y	majorant $\mathcal{M}$	I_{eff}
1	9	8.873457e-03	16	2.135435e-02	1.55
2	25	3.240138e-03	56	9.050160e-03	1.67
3	81	8.950033e-04	208	2.635564e-03	1.72
4	289	2.295153e-04	800	6.858397e-04	1.73
5	1089	5.774695e-05	3136	1.732119e-04	1.73
6	4225	1.445987e-05	12416	4.341359e-05	1.73
7	16641	3.616415e-06	49408	1.086033e-05	1.73
8	66049	9.041944e-07	197120	2.715516e-06	1.73

Table 2: Comparison of the square of exact error $\|\nabla(u-v)\|_\Omega^2$ and majorant $\mathcal{M}$ values and on various uniform mesh levels.

theoretical value

$$C_\Omega = \frac{1}{\sqrt{2}\pi} \approx 0.22507484206717$$

known for the unit square domain.Discrete values of C_Ω converge to the theoretical value quadratically with respect to the mesh size. Discrete values of C_Ω were computed from (16) in MATLAB.

3.2 Majorant computation

Raviart-Thomas elements of the zero degree (known as RT_0 elements [17]) are applied for the computation of the flux $y \in H(\Omega, \text{div})$ in Algorithm 1. Their MATLAB implementation is based on [3] with some modification with respect to the performance and extension towards a multigrid solver.

For simplicity, only one step (a) and one step (b) of Algorithm 1 were applied assuming an initial value $\beta = 1$. Quadrature rules exact for polynomials up to the order two were used for the computation of integrals on triangulations. The right column of Figure 2 displays computed fluxes (only one component due to symmetry reasons) for various uniform mesh levels and the exact flux of the solution provided by (14). It can be observed that both discrete solutions and fluxes (at least visually) converge to exact solution and its flux.

By comparing the values of Table 2 or its visualization (Figure 3), both exact errors and majorant values converge linearly with respect to corresponding degrees of freedom used for their computation. The ratio of their values on the same mesh level is expressed by the index of efficiency which remains bounded and converge to the approximate value $I_{\text{eff}} \approx 1.73$.

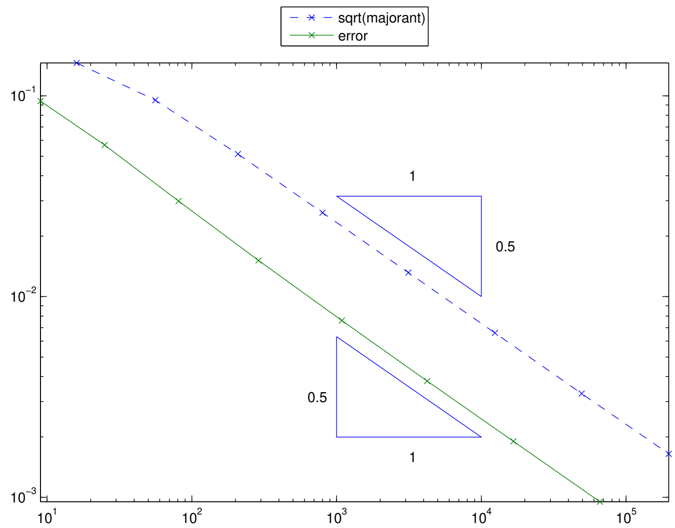

Figure 3: Values of exact error $\|\nabla(u-v)\|_\Omega$ and the square root of the majorant $\sqrt{\mathcal{M}}$ displayed versus degrees of freedom (displayed as DOF v and DOF y in Table 2) of their calculation on various uniform mesh levels.

4 Improving computational costs

It is obvious from the example in Section 2.1, that the highest computation costs are caused due to the solution of the linear system of equation (13) in step (a) of Algorithm 1. Note that the system matrix

$$\left[(1+\beta)M + (1+\frac{1}{\beta})C_\Omega^2 DIV DIV\right]$$

is symmetric and positive definite. The positive definiteness follows automatically from the estimate (6), where the quadratic functional on the right side must be nonnegative in order to bound the nonnegative error from above (reliability of the estimate).

Let consider an iterative method for solving the linear system of equation (13). The advantage of iterative over direct methods is clear, since each iteration flux vector $\hat{y}_k$ for $k = 0, 1, \ldots$ can be insterted into the majorant (12). It is therefore not necessary to solve the linear system exactly. Let us consider the preconditioned conjugate gradient (PCG) method for the iterative solution of the linear system of equation (13). The PCG method can be schematically described (for a reformulated system $Ax = b$) as

Algorithm 2 (PCG for Ax=b)**.** *Let an initial iteration x_0 is given. Compute the initial residual $r_0 = b - Ax_0$, the initial energy $E_0 = \frac{1}{2}(Ax_0, x_0) -$*

(b, x_0) and $s_0 = C^{-1} r_0$, $p_0 = s_0$, where C is a given preconditioning matrix. For the iterations $j = 0, 1, \ldots$ do the loop

(1) $\gamma_j = \frac{(r_j, s_j)}{(p_j, Ap_j)}$

(2) $E_{j+1} = e_j - \frac{\gamma_j (r_j, s_j)}{2}$

(3) $x_{j+1} = x_j + \gamma_j p_j$

(4) $r_{j+1} = r_j - \gamma_j A p_j$

(5) $s_{j+1} = C^{-1} r_{j+1}$

(6) $\delta_{j+1} = \frac{(r_{j+1}, s_{j+1})}{(r_j, s_j)}$

(7) $p_{j+1} = s_{j+1} + \delta_{j+1} p_j$

After the end of loop output the solution x_{j+1} and the energy E_{j+1}.

This algorithm recalls Algorithm 2 from [26] with an extension for the computation of the energy (in the step (2)) defined as

$$E_j := \frac{1}{2}(Ax_j, x_j) - (b, x_j). \tag{17}$$

The knowledge of the energy is required for the computation of the flux-dependent functional majorant part $\mathcal{M}_2$, since it holds (cf. (12))

$$\mathcal{M}_2(\nabla v, f, \beta, C_\Omega, y_j) = 2E_j, \tag{18}$$

where A and b are the matrix and the right-hand side of the linear system (13). In order to save extra matrix-vector multiplication in (17), which is already provided in step (1) of Algorithm 2, the formula in step (2) provides a cheap way of computing E_j. It can be directly derived from the combination of two formulae. The first one is the known relation between the energy error of the CG-iterations and their energy

$$\begin{aligned} E_j - E =& \frac{1}{2}(Ax_j, x_j) - \frac{1}{2}(Ax, x) - (b, x_j) + (b, x) \\ =& \frac{1}{2}\left((Ax_j, x_j) + (Ax, x) - 2(Ax, x_j)\right) \\ =& \frac{1}{2}(A(x - x_j), x - x_j) := \frac{1}{2}||x_j - x||_A^2, \end{aligned} \tag{19}$$

where x denotes an exact solution of the linear system $Ax = b$. The second one is a special version of the formula (3.6) for $d = 1$ from [26]

$$||x - x_{j+1}||_A^2 = ||x - x_j||_A^2 - \gamma_j (r_j, s_j). \tag{20}$$

Since $\gamma_j(r_j, s_j) \geq 0$ in (20), the PCG reduces (or at least does not increase) the energy in each iteration. Therefore, Algorithm 1 is modified to obtain

Algorithm 3 (Majorant minimization CG based algorithm)**.** *Let* $v \in H_0^1$ *be a given discrete solution. Let* $\beta > 0$ *is a given value.*

(a) Compute $\hat{y}$ *as a solution of a linear system* (13) *using CG method with a given stopping criterion.*

(b) Upgrade β *from* y *using the formula* (7)*. If the convergence in* y *is not achieved then goto to step (a).*

The left column of Figure 4 reports on the behavior of one step (a) of Algorithm 3 applied to levels 6 and 7 solutions from Example 2.1. The conjugate gradient methods is applied for the solution of the linear system (13), in which $\beta = 1$ is set. The initial choice $\hat{y}_0 = 0$ leads to an estimate

$$\|\nabla v - y\|_\Omega^2 \leq \mathcal{M}_1(\nabla v, f, \beta, C_\Omega), \tag{21}$$

where $\mathcal{M}_1$ is defined in (9). This estimated is improved with each PCG iteration y_j, since the flux-dependent part $\mathcal{M}_2(\nabla v, f, \beta, C_\Omega, y_j)$ of the majorant $\mathcal{M}$ decreases (or stays constant it the worst theoretical case).

4.1 Numerical performance of PCG

The following test are measuring the performance of PCG solver from the step (a) of Algorithm 3 in MATLAB. For practical computation, the PCG terminated in the j-th iteration, if the stopping criterion (with $\varepsilon = 10^{-6}$)

$$\sqrt{\frac{(r_j, s_j)}{(r_0, s_0)}} \leq \varepsilon \tag{22}$$

was fulfilled.

Remark 1. *If the preconditioner* C *approximates* A *very well, i.e.,* $C \approx A$*, it obviously holds*

$$(r_j, s_j) = (A(x - x_j), C^{-1}A(x - x_j)) \approx ||x - x_j||_A^2$$

and (22) *is equivalent to a stopping criterion based on the relative the relative A-norm of the error*

$$\frac{||x - x_j||_A}{||x - x_0||_A} \leq \varepsilon. \tag{23}$$

In the extremal case of no preconditioning (then we speak of CG method without preconditioning), i.e., C *is an identity matrix,* (22) *is equivalent to a classical stopping criterion (default in MATLAB)*

$$\frac{||b - Ax_j||_2}{||b||_2} \leq \varepsilon. \tag{24}$$

Then, it obviously holds $(r_j, s_j) = (A(x - x_j), A(x - x_j)) = ||x - x_j||^2_{A^2}$ *and* (22) *is equivalent to another stopping criterion*

$$\frac{||x - x_j||_{A^2}}{||x - x_0||_{A^2}} \leq \varepsilon. \tag{25}$$

As an operation of the preconditioner C, we apply a simple V-cycle of a geometrical multigrid method [16] based on provided hierarchy of nested triangulations $\mathcal{T}_0, \mathcal{T}_1, \ldots$. As the linear system (13) arises from a $H(div)$ problem, a special smother as a part of the multigrid method is required. Our choice is the additive version of the smoother of Arnold, Falk and Winther [2] using one presmoothing and one postsmoothing steps.

Table 3 compares numbers of iterations of non-preconditioned (CG) and multigrid-preconditioned (MPCG) method for various levels of triangulation. Single CG or MPCG iterations and the corresponding majorant values are displayed on Figure 4 for mesh levels 6 and 7. The number of iterations reflects typical properties of conjugate gradients and system matrices arising in elliptic partial boundary value problems. For shape regular triangulations, the condition number of A is known to be proportional to h^{-2}, where h is the mesh-size parameter, i.e., $cond(A) \approx h^{-2}$. Furthermore, the number of CG- iterations (with respect to the same stopping tolerance ε) satisfies $ITER \approx \sqrt{cond(A)}$. Together, it holds

$$ITER \approx h^{-1}. \tag{26}$$

The mesh parameter h is halfened after each uniform refinement and therefore the number of CG-iterations is according to (26) expected to be doubled in the non-preconditioned case.

For the multigrid-preconditioned CG method (MPCG) we observe that the number of iterations remains bounded, in our case

$$ITER \leq 11. \tag{27}$$

This property is so called mesh- independence and it demonstrates the optimality of the chosen multigrid preconditioner. A detailed observation of the right column pictures of Figure 4 indicates that 4 iterations, i.e., $ITER = 4$ provide already a very sharp knowledge of the majorant values without need for additional iterations. Thus a better understanding and application of different termination criteria of the conjugate gradient might lead to even higher effectivity in the computation of the majorant values in the future.

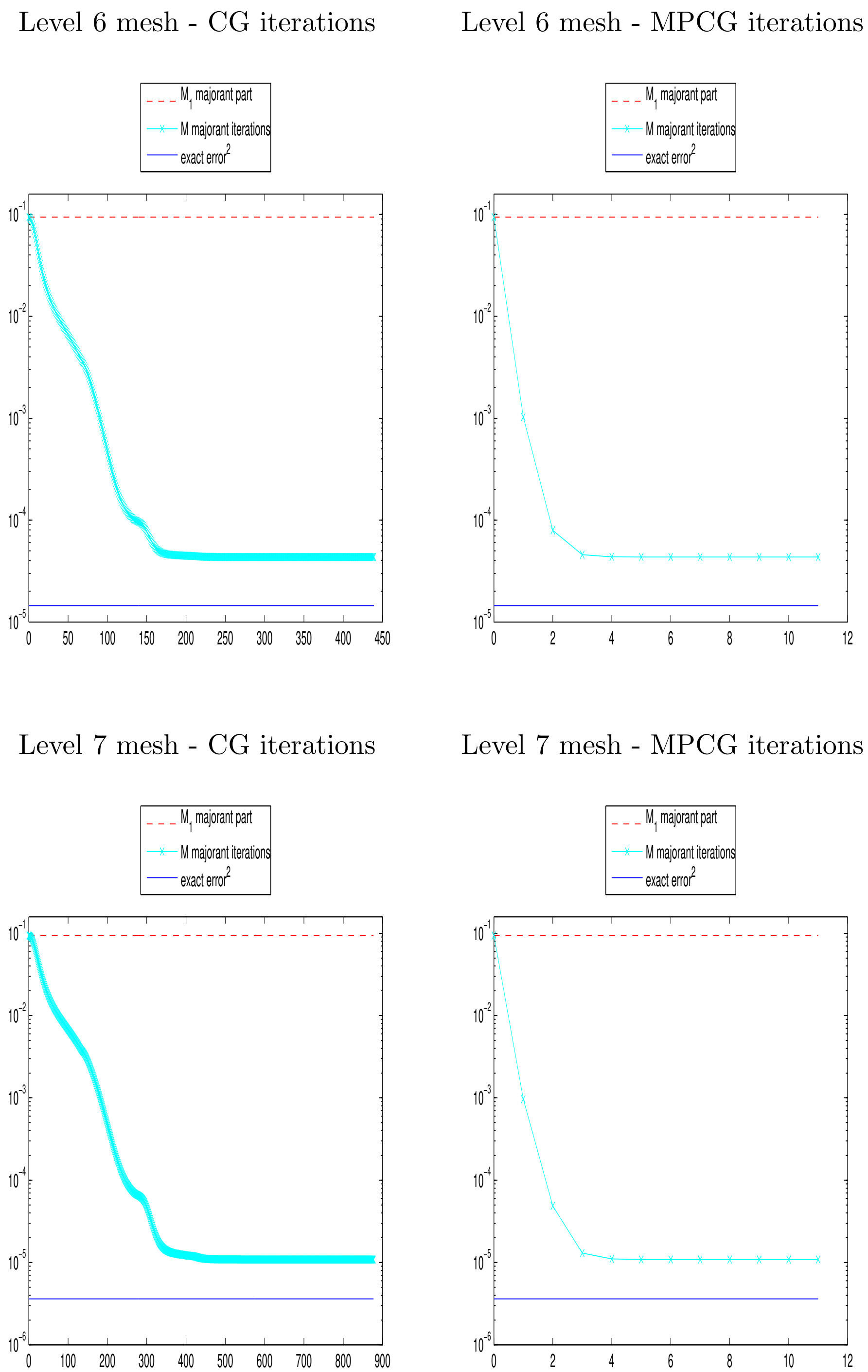

Figure 4: Majorant values displayed versus iterations of CG and MPCG method for level 6 and 7 meshes. For better illustration, the majorant part $\mathcal{M}_1$ (the upper bound) from (9) and the square of the exact error, i.e., $\|\nabla(u-v)\|_\Omega^2$ (the lower bound) are displayed.

level	problem size	no preconditioner	multigrid preconditioner
0	5	2	2
1	16	5	5
2	56	15	8
3	208	45	10
4	800	108	11
5	3136	219	11
6	12416	349	11
7	49408	877	11
8	197120	1773	11
9	787456	3506	11

Table 3: Number of iterations of the CG method using no preconditioner or the multigrid (V cycles) preconditioner with the additive smoother of Arnold, Falk and Winther for 1 smoothing step, $\varepsilon = 10^{-6}$.

5 Conclusions

The minimization of the majorant term in the functional aposteriori estimate can be done by solving a sequence of systems of linear equations for an unknown approximation of the flux of an exact solution. The solution of the first linear system was obtained by the conjugate gradient method in combination with a (Hdiv) multigrid preconditioner. For higher efficiency, the conjugate gradient method is terminated after few iteration providing already very good flux approximation in a considered benchmark example. However, an optimal strategy for the termination process of the preconditioned conjugate gradiennt method in connection to the majorant computation remains an interesting open question.

Acknowledgments

The author acknowledges support from the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' for support under grant SFB F013/F1306 in Linz, Austria. The comments of S. Repin (St. Petersburg) improved an overview on aposteriori estimates, the "Linz multigrid comunity" (U. Langer, R. Simon, S. Zaglmayr) provided practical hints concerning computational issues and multigrid application and A. Janka (Fribourg) assisted at the vectorization of the smoother of Arnold, Falk and Winther in Matlab.

References

[1] M. Ainsworth and J. T. Oden, *A posteriori error estimation in finite element analysis*, Wiley and Sons, New York, 2000.

[2] D. N. Arnold, R. S. Falk, R. Winther, *Preconditioning in $H(div)$ and applications,* Mathematics of computation, 66 (1997), 219, 957-984.

[3] Carstensen, C. and Bahriawati, C. *Three matlab implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control* CMAM, 5 (2005), 4, 333-361.

[4] I. Babuška and T. Strouboulis, *The finite element method and its reliability*, Oxford University Press, New York, 2001.

[5] W. Bangerth and R. Rannacher, *Adaptive finite element methods for differential equations*, Birkhäuser, Berlin, 2003.

[6] I. Babuška and W. C. Rheinboldt, *A-posteriori error estimates for the finite element method*, Internat. J. Numer. Meth. Engrg. 12(1978), pp. 1597–1615.

[7] I. Babuška and W. C. Rheinboldt, *Error estimates for adaptive finite element computations*, SIAM J.Numer. Anal. 15(1978), pp. 736–754.

[8] M. Bildhauer, M. Fuchs and S. Repin, *A posteriori error estimates for stationary slow flows of power–law fluids,* J. Non-Newtonian Fluid Mech. 142 (2007), 112–122.

[9] M. Bildhauer, M. Fuchs and S. Repin, *A functional type a posteriori error analysis for Ramberg–Osgood Model,* Z. Angew. Math. Mech. (ZAMM), 87 (2007), 11–12, 860–876.

[10] D. Braess and J. Schöberl, *Equilibrated Residual Error Estimator for Maxwell's Equations,* Mathematics of Computation, 77 (262), 651-672, 2008.

[11] C. Carstensen and S. Bartels, *Each averaging technique yields reliable a posteriori error control in FEM on unstructured grids. I: Low order conforming, nonconforming, and mixed FEM,* Math. Comp., 71(2002), 239, pp. 945–969.

[12] Ciarlet, P. G., *The finite element method for elliptic problems,* North-Holland, Amsterdam, 1978.

[13] R. S. Falk, *Error estimates for the approximation of a class of variational inequalities*, *Math. Comput.*, 28(1974), 963–971.

[14] M. Fuchs and S. Repin, *Estimates for the deviation from the exact solutions of variational problems modeling certain classes of generalized Newtonian fluids,* Math. Meth. Appl. Sci., 29 (2006), 2225–2244.

[15] M. Frolov, P. Neittaanmäki, and S. Repin, *Guaranteed functional error estimates for the Reissner-Mindlin plate problem,* J. Math. Sci (New York), 132 (2006), 4, 553-561.

[16] Hackbusch, W. *Multi-Grid Methods and Applications*, Springer-Verlag Berlin, Heidelberg, 1985.

[17] P.A. Raviart, J.M. Thomas, *A mixed finite element method for 2nd order elliptic problems,* Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975), pp. 292–315. Lecture Notes in Math., Vol. 606, Springer, Berlin, 1977.

[18] S. Repin, *A posteriori error estimation for variational problems with uniformly convex functionals*, Mathematics of Computations, 69 (2000), 230, 481–500.

[19] S. Repin, *A posteriori error estimation for nonlinear variational problems by duality theory*, Zapiski Nauchn. Semin. (POMI), 243(1997), 201–214.

[20] S. Repin, *A posteriori estimates for approximate solutions of variational problems with strongly convex functionals,* Problems of Mathematical Analysis, 17 (1997), 199–226 (in Russian). English translation in Journal of Mathematical Sciences, 97(1999), 4, 4311–4328.

[21] S. Repin, *A Posteriori Estimates for Partial Differential Equations.* Lectures on Advanced Computational Methods in Mechanics (Ed. by J. Kraus and U. Langer), Radon Series on Computational and Applied Mathematics, Vol. 1, de Gruyter Verlag, Berlin 2007.

[22] S. Repin, *A posteriori error estimates for variational problems with uniformly convex functionals,* Math. Comput., 69(230), 2000, pp. 481-500.

[23] S. Repin, *Two-sided estimates of deviation from exact solutions of uniformly elliptic equations,* Proc. St. Petersburg Math. Society, IX(2001), pp. 143–171, translation in Amer. Math. Soc. Transl. Ser. 2, 209, Amer. Math. Soc., Providence, RI, 2003.

[24] S. Repin and J. Valdman, *Functional a posteriori error estimates for problems with nonlinear boundary conditions,* Journal of Numerical Mathematics 16 (1), 51-81 (2008).

[25] S. I. Repin and L. S. Xanthis, *A posteriori error estimation for nonlinear variational problems,* Comptes Rendus de l'Adadémie des Sciences Mathématique, 324 (1997), 1167-1174.

[26] Z. Strakoš and P. Tichý, *Error estimation in preconditioned conjugate gradients,* BIT Numerical Mathematics, 45, 2005, pp. 789-817.

[27] T. Vejchodský, *Guaranteed and locally computable a posteriori error estimate,* IMA J. Numer. Anal. 26 (2006), no. 3, 525-540.