Noname manuscript No.
(will be inserted by the editor)

On the interplay between the PDE discretization and numerical solution of the resulting algebraic problems

J. Liesen · Z. Strakoš

the date of receipt and acceptance should be inserted later

Abstract In mathematical modeling, the description of reality via some system of differential or integral equations can be considered a kind of model reduction. This omits, for sake of clarity and solvability of the mathematical model, the less substantial relationships. Discretization of the model means further reduction from an infinite-dimensional to a finite-dimensional function space. This can lead to additional nontrivial issues which are not present in the original mathematical model and which have to be addressed in the solution process. In this paper we address one of these issues that arises in the context of the finite element method (FEM).

The FEM generates an approximate solution of the model in form of a linear combination of functions with strictly *local* supports. The *global* approximation property of the FEM approximate solution is restored by solving a linear algebraic system for the coefficients of this linear combination. The fact that in practice we do not solve this system *exactly* can then have fundamental consequences.

The total approximation error must be evaluated in an appropriate function space. Using a simple model problem we illustrate numerically that in the function space the algebraic error can create significant local components and can dominate locally the total error, even when the globally measured algebraic error (in the energy norm or as the algebraic backward error) is significantly smaller than the globally measured discretization error. Incorporation of the algebraic error into the total error with considering the locality and the interplay between the discretization and the algebraic computation represents a fundamental challenge. This challenge must be addressed in order to put adaptive PDE solvers on a rigorous mathematical ground.

Mathematics Subject Classification (2000) 65F10, 65N15, 65N30, 65N22, 65Y20, 68Q25, 76M12

The work of J. Liesen was supported by the Heisenberg Program of the Deutsche Forschungsgemeinschaft.

The work of Z. Strakoš was supported by the research project MSM0021620839 and partially also by the GACR grant 201/09/0917.

Institute of Mathematics, Technical University of Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, E-mail: liesen@math.tu-berlin.de · Charles University in Prague, Faculty of Mathematics and Physics, Sokolovská 83, 18675 Prague, Czech Republic, E-mail: strakos@karlin.mff.cuni.cz

Keywords numerical PDE, finite element method, discretization error, algebraic error, error estimates, locality of the error, adaptivity

1 Introduction and problem setting

To introduce the setting and notation of this paper we very briefly describe the solution process of a partial differential equation (PDE) boundary value problem, arising from mathematical modeling, by the finite element method (FEM). Further details can be found in any book on the numerical solution of PDEs; see, e.g., [8, 14, 15, 19].

In the first step of the solution process the given PDE or system of PDEs $Lu = f$ (plus appropriate boundary conditions) is transformed into its variational formulation:

$$\text{Find } u \in V \text{ such that } a(u, v) = g(v) \text{ for all } v \in V. \tag{1}$$

Here V is an inifinite-dimensional function space (typically a Sobolev space[1]), a is a bilinear form and g is a linear functional. The Galerkin FEM discretization consists of finding a finite-dimensional subspace $V_h \subset V$ and solving the discretized problem:

$$\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = g(v_h) \text{ for all } v_h \in V_h. \tag{2}$$

If $\phi_1, \ldots, \phi_N$ is a basis of V_h, the discretized variational problem (2) is equivalent to the linear algebraic system

$$Ax = b, \quad A = [a_{ij}] = [a(\phi_j, \phi_i)] \in \mathbb{R}^{N\times N}, \quad b = [g(\phi_1), \ldots, g(\phi_N)]^T \in \mathbb{R}^N, \tag{3}$$

in the sense that the solution vector $x = [\zeta_1, \ldots, \zeta_N]^T$ of (3) contains the coefficients of the solution u_h of (2) with respect to the basis $\phi_1, \ldots, \phi_N$, i.e.,

$$u_h = \sum_{j=1}^{N} \zeta_j \phi_j. \tag{4}$$

In summary, we have to consider:

- the original mathematical model or its variational formulation (1), which typically also requires to understand its origin;
- the discretized problem (2), where the approximate solution is restricted to some finite-dimensional function subspace;
- the algebraic problem (3) that determines the coefficients for the approximate solution with respect to the given basis of the finite-dimensional function subspace.

[1] Here we do not give any specifics of the choice of the appropriate function space and of the concept of solution related to the choice of this space. While in the model problem presented below the situation is simple and the energy norm is appropriate, in many practical cases the choice of an appropriate function space represents a substantial difficulty. As an example, the state-of-the-art theory of nonlinear partial differential equations focuses on solutions that are local in time and exist under the assumptions of sufficient smoothness, this does not give a strong guidance for the (physically) meaningful evaluation of error. In order to obtain such guidance, one must take into account the underlying (physical) principles. Models in continuum thermodynamics as well as thermodynamics of multi-component materials may serve as examples. Here the natural function spaces are determined in relation to the properties of the entropy and the rate of the entropy production; see [16, 17, 24].

If we leave aside, for simplicity, the errors due to modeling and possible uncertainty in the data, we are confronted in this solution process with three different type of errors:

- the *discretization error* $u - u_h$, where u solves (1) and u_h solves (2) (for simplicity we assume that these solutions exist);
- the *algebraic error* $x - x_n$, where x solves (3) and x_n is a computed approximation to x;
- the *total error* $u - u_h^{(n)}$, where u solves (1) and $u_h^{(n)} = \sum_{j=1}^{N} \zeta_j^{(n)} \phi_j$ is determined by the coefficient vector $x_n = [\zeta_1^{(n)}, \ldots, \zeta_N^{(n)}]^T$.

These errors are related by the simple, yet fundamental equation

$$u - u_h^{(n)} \;=\; (u - u_h) \,+\, (u_h - u_h^{(n)}),$$

which means that the total error is the sum of the discretization error and the algebraic error (after being transferred from the coordinate space $\mathbb{R}^N$ to the function space V_h).

A main point of the FEM is that in order to simplify the mathematical issues related to estimation of the discretization error and in order to obtain a sparse matrix A, each basis function ϕ_j is nonzero only on a small subset of the domain Ω. This fact is computationally crucial, because in mathematical modeling of real world phenomena typically the matrices are very large. Thus, the FEM in general gives up the global approximation property of individual basis functions; each FEM basis function approximates the solution *only locally*. The *global* approximation is restored by solving the linear algebraic system (3) and by forming the linear combination (4). (One can also point out the requirement for investigating the approximation error in the regions of interest, with convincing arguments presented, e.g., by Babuška and Stroboulis in [8, p. 417 and Chapter 6] and by Bangerth and Rannacher in [9, Chapter 1].)

If one assumes that the linear algebraic system (3) is solved *exactly*, then the total error reduces to the discretization error. In numerous publications on the numerical analysis of partial differential equations, the exact solution x is indeed assumed to be available. Our major point is that this assumption does not reflect the reality of numerical computations. Moreover, aiming at the smallest possible algebraic error is in conflict with the requirement of *computational efficiency* of numerical PDE solvers. In practice only an approximation x_n to the exact algebraic solution x is available.

The local character of the FEM basis functions on the one hand, and the global character of the linear algebraic problem resulting from the discretization on the other have the following fundamental consequence: The algebraic error $x - x_n$ can have strongly varying individual entries, which potentially lead to a large variation in the sizes of the local components of the *total* error $u - u_h^{(n)}$ on the individual elements, irrespectively of the local value of the solution u or the local value of the discretization error $u - u_h$. These facts will be illustrated numerically in Section 3 below. In practice they always should be taken into consideration when evaluating the total error, unless they can be (rigorously) shown to be insignificant for the given problem.

The goal of the whole computation is to obtain an acceptable approximation to the solution of the original problem. Here the acceptability refers to the mathematical modeling level, which uses the given PDE (or system of PDEs) as a tool, and the error is measured in the proper function space. For an instructive account of the related issues (without considering the algebraic error) we refer to [7, 28]. Here we argue that the algebraic part of the error must also be taken into account, which, in general, brings into the numerical PDE error analysis a fundamental challenge that is very

rarely considered in state-of-the-art investigations. The importance of investigating the algebraic error and its distribution is stated, e.g., in [15, Sections 6.4–6.5 and Chapter 12]. However, the issue is in [15] not pursued or further analyzed. Examples of publications where the algebraic error is included in the analysis will be presented below.

We emphasize that our point goes much beyond the need for investigating numerical stability and conditioning issues sometimes declared in the PDE literature; see, e.g., [36]. Numerical stability and conditioning forms only a small part of it.

2 Standard algebraic tools for error analysis

Clearly, when solving challenging mathematical modelling problems, the question of an acceptable computational error of matrix computations can not be resolved by algebraic methods alone. It rather must take into account that the approximation error is measured within the given function space. We now examine whether standard algebraic approaches can be easily incorporated into the framework described above.

An epochal progress in understanding results of practical algebraic computations is related to the concept of the (algebraic) *backward error*. To briefly describe the principle in the context of iterative methods for linear algebraic systems, let x_n be the approximation to the solution of $Ax = b$ computed at step n of an iterative method. The backward error analysis considers the perturbed linear algebraic system

$$(A + \Delta A)\, x_n = b + \Delta b \tag{5}$$

and answers the question how close the perturbed problem (5), which is solved *exactly* by x_n, is to the original problem $Ax = b$, which is solved *approximately* by x_n. As shown by Rigal and Gaches [32] (also see [22, Theorem 7.1]) the normwise (relative) backward error of x_n, defined by

$$\beta(x_n) \equiv \min\{\, \beta : (A + \Delta A)\, x_n = b + \Delta b\,,\ \|\Delta A\| \le \beta\|A\|\,,\ \|\Delta b\| \le \beta\|b\|\,\}\,, \tag{6}$$

satisfies

$$\beta(x_n) = \frac{\|b - Ax_n\|}{\|b\| + \|A\|\,\|x_n\|} = \frac{\|\Delta A_{\min}\|}{\|A\|} = \frac{\|\Delta b_{\min}\|}{\|b\|}. \tag{7}$$

In other words, $\beta(x_n)$ is equal to the norm of the *smallest* relative perturbations in A and b such that x_n exactly solves the perturbed system. Here $\|\cdot\|$ is any vector and the corresponding induced matrix norm. The componentwise variant can be found in [29]; see also [22, Chapter 7].

Although the concept of backward error arose from investigations of numerical instabilities (see, e.g., [30], [22, Chapter 7] that describe the role of Goldstine, von Neumann, Turing and the epochal contribution of Wilkinson), it can be used irrespectively of the source of the error (truncation and/or roundoff). The algebraic backward error ingeniously separates the properties of the method (and even of the particular individual computation) from the *conditioning* of the problem. Their combination allows to estimate the size of the algebraic error $x - x_n$ measured in an appropriate *norm*; see the essays [38,6], [10, Section 3.2] and the monograph [22]. Arioli, Noulard and Russo [5] used the function backward errors and extended the concept to function spaces; see also [1,3] and [26, Section 4.3].

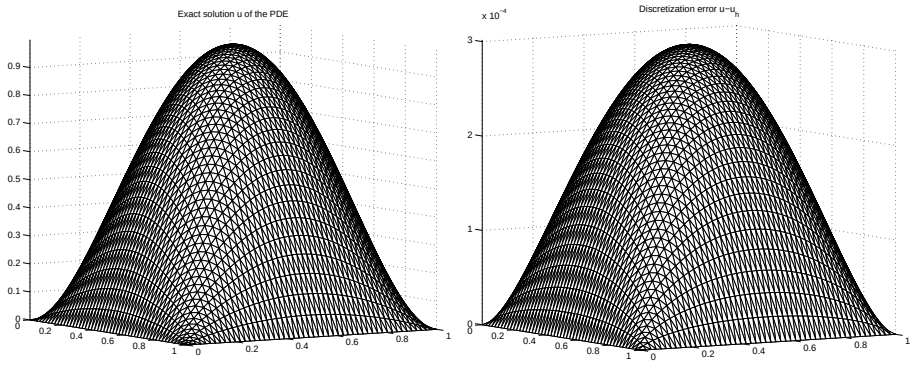

Fig. 1 Left: MATLAB plot of the exact solution u of the Poisson model problem (8)–(9). Right: MATLAB plot of the discretization error $u - u_h$ (the vertical axis is scaled by 10^{-3}). It should be emphasized that plots show the piecewise linear approximations of the actual functions, which is, as explained in Remark 1, for the discretization error misleading.

At first sight the incorporation of the algebraic backward error concept into the estimates of the total error measured in the function space seems to be just a technical exercise. Due to the error of the model, the discretization error and the uncertainties in the data, the system $Ax = b$ represents a whole class of admissible systems. Each system in this class corresponds (possibly in a stochastic sense) to the original real-world problem. One can therefore argue that as long as the algebraic backward error $\beta(x_n)$ in (6)–(7) is *small enough,* the computed algebraic solution x_n is with respect to the subject of the mathematical modeling as good as the solution x of $Ax = b$. The meaning of *small enough* is sometimes intuitively interpreted as, say, *an order of magnitude below the size of the discretization error* (all measured in the norms which physically correspond to each other). It is worth to point out that the balance between the discretization and the algebraic errors is typically evaluated globally (in norms).

The practical situation is, however, much more subtle. In particular, in order to perform the computations efficiently, we need tight *a posteriori* estimates of the local distribution of the total error which incorporate the algebraic error; a more detailed argumentation and example can be found in [23]. Whether and to which extent the algebraic backward error can serve this purpose is yet to be found. The experimental results presented in the following section indicate the nontrivial problems which need to be resolved.

3 Experimental results

We consider the following two-dimensional Poisson model problem,

$$-\Delta u = f \quad \text{in } \Omega = (0,1) \times (0,1), \quad f = 32(\eta_1 - \eta_1^2 + \eta_2 - \eta_2^2), \tag{8}$$

$$u = 0 \quad \text{on } \partial\Omega. \tag{9}$$

This problem has the nicely smooth exact solution

$$u(\eta_1, \eta_2) = 16\eta_1\eta_2(1 - \eta_1)(1 - \eta_2). \tag{10}$$

The variational formulation of (8)–(9) is given by (2) with

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega, \quad g(v) = \int_\Omega f v \, d\Omega.$$

We discretize the variational problem using the (conforming) Galerkin finite element method (FEM) with linear basis functions on a regular triangular grid with the mesh size $h = 1/(m+1)$, where m is the number of inner nodes in each direction. The basis function ϕ_j, $j = 1, 2, \ldots, m^2$, corresponding to the jth inner node has its support composed of six triangle elements with the node j as the central point.

It is well known that nodes can be ordered to obtain the discrete Laplacian matrix A of the form

$$A = [a(\phi_j, \phi_i)] = \text{tridiag}(-I, T, -I) \in \mathbb{R}^{m^2 \times m^2}, \quad T = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m \times m};$$

see, e.g., [15, Section 15.1]. The matrix A is symmetric and positive definite, with its extreme eigenvalues given by

$$\lambda_{\min}(A) = 8 \sin^2\left(\frac{h\pi}{2}\right), \quad \lambda_{\max}(A) = 8 \sin^2\left(\frac{mh\pi}{2}\right);$$

see, e.g., [19, Chapter 4]. We assemble the right hand side b using a two-dimensional Gaussian quadrature formula that is exact for polynomials of degree at most three.

In our numerical experiment we use $m = 50$, and thus A is of size 2500×2500. Similar numerical results can be obtained for any other choice of m. All computations have been performed using MATLAB. The extreme eigenvalues of A and the resulting condition number (with respect to the matrix 2-norm) are

$$\lambda_{\min}(A) = 7.5867 \times 10^{-3}, \quad \lambda_{\max}(A) = 7.9924, \quad \kappa(A) = 1.0535 \times 10^{3}.$$

We have computed the (approximate) solution of $Ax = b$ using the MATLAB backslash operator. Neglecting the algebraic error in this computation, the (closely approximated) squared energy norm of the discretization error is

$$\|\nabla(u - u_h)\|^2 \;=\; a(u - u_h, u - u_h)^2 \;= 5.8299 \times 10^{-3}. \tag{11}$$

The shape of the discretization error on the MATLAB plots seems very similar to the shape of the solution, see Fig. 1. As explained in the following remark, the discretization error is, however, much less smooth than shown on the right part of Fig. 1.

Remark 1 All figures shown in this paper have been generated by the MATLAB `trisurf` command, which generates a triangular surface plot. The inputs of `trisurf` are the coordinates of the nodes in the given triangular mesh and the respective values of the plotted function at these nodes. In the plot the function values in the triangle interiors are interpolated linearly from the values at the nodes, and hence *the figures do not show the actual function values inside the triangles*. For the solution u the difference is not significant. In case of the discretization error $u - u_h$ (see the right part of Fig. 1), the plot is, however, misleading . The discretization error is not as smooth as suggested by the plot, but contains "bubbles" inside the triangles, which can be (depending on the size of the error) significant. The same holds for the total errors shown in Figs. 2–6.

Table 1 Errors and CG iterations in our numerical experiment.

γ	Total error $\|\nabla(u-u_h^{(n)})\|^2$	Algebraic error $\|x-x_n\|_A^2$	Componentwise backward error	No. of CG iterations
50.0	1.0195×10^{-2}	4.3656×10^{-3}	2.6831×10^{-2}	27
1.00	5.8444×10^{-3}	1.4503×10^{-5}	4.2274×10^{-4}	35
0.50	5.8304×10^{-3}	5.6043×10^{-7}	5.4886×10^{-5}	42
0.10	5.8299×10^{-3}	1.6639×10^{-8}	4.0418×10^{-6}	50
0.02	5.8299×10^{-3}	5.5286×10^{-10}	2.1091×10^{-6}	56

Now we apply the conjugate gradient method (CG) of Hestenes and Stiefel [21] to the linear algebraic system $Ax = b$. We use $x_0 = 0$ and stop the iteration when the normwise backward error drops below the level $\gamma\, h^\alpha$, i.e., when

$$\frac{\|b - Ax_n\|}{\|b\| + \|A\|\,\|x_n\|} < \gamma\, h^\alpha, \tag{12}$$

where $\gamma > 0, \alpha > 0$ are positive parameters and $\|\cdot\|$ denotes the 2-norm. If the size of the backward error is small enough, then the algebraic approximate solution x_n *exactly* solves an algebraic problem that is very close to $Ax = b$. Then one might expect that the algebraic error does not have a noticeable impact on the total error (here we use the normwise backward error; the componentwise variant, which is also reported in the table below, would not lead to any significant change).

In order to examine this reasoning and, in particular, in order to examine quantitatively an intuitive understanding of the term *small enough* in relation to the size of the discretization error, we have used

$$\alpha = 3$$

which may seem sufficient, with the choice $\gamma = 1$, to keep the algebraic error insignificant in comparison to the discretization error. The other values of γ used in the experiment are given in Table 1. With $m = 50$, the values $\alpha = 3$ with $\gamma = 50$ closely resemble the situation $\alpha = 2$ with $\gamma = 1$, which corresponds to the size of the inaccuracies in determining of A and b being proportional to $h^2 = (51)^{-2}$. With decreasing γ, the algebraic error measured in the algebraic energy norm quickly drops very significantly below the discretization error (11).

The componentwise backward error given in Table 1 is computed by the formula

$$\max_i \frac{(|b - Ax_n|)_i}{(|A|\,|x_n| + |b|)_i},$$

where $(y)_i$ denotes the ith entry of the vector y and $|\cdot|$ means that we take the corresponding matrix or vector with the absolute values of its entries; see [22, Theorem 7.3].

The discretization error (11) and the values in the second and third column of Table 1 satisfy (up to a small inaccuracy proportional to machine precision) the Galerkin orthogonality relation

$$\|\nabla(u - u_h^{(n)})\|^2 = \|\nabla(u - u_h)\|^2 + \|\nabla(u_h - u_h^{(n)})\|^2 = \|\nabla(u - u_h)\|^2 + \|x - x_n\|_A^2;$$

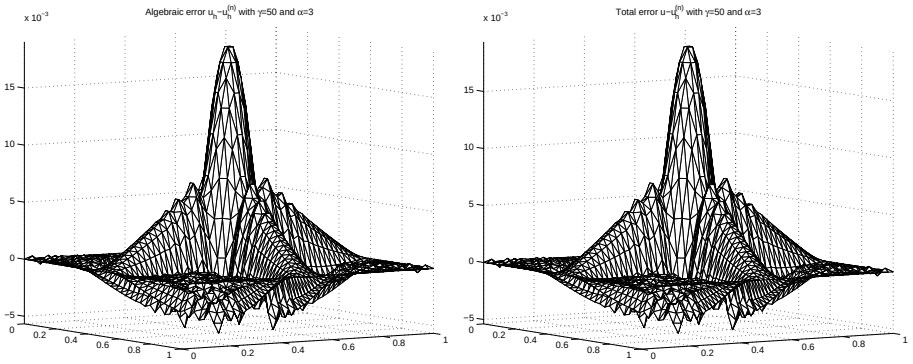

Fig. 2 $\gamma = 50.0$: algebraic error $u_h - u_h^{(n)}$ (left) and total error $u - u_h^{(n)}$ (right); the vertical axes are scaled by 10^{-3}. While the algebraic error is piecewise linear, the total error is not (the MATLAB plot does not show the small bubbles over individual elements; see Remark 1).

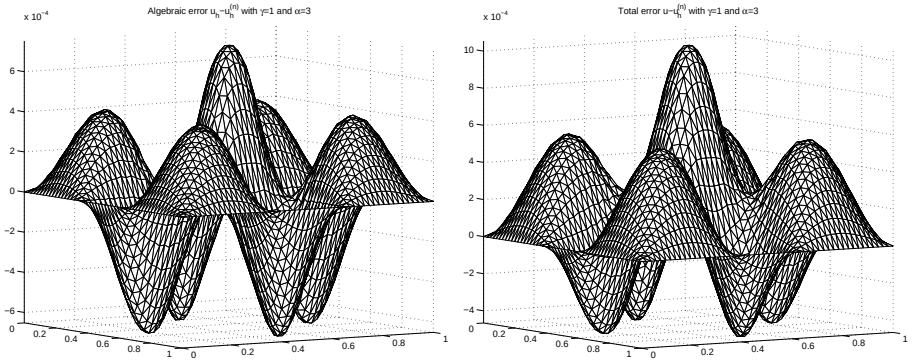

Fig. 3 $\gamma = 1.0$: algebraic error $u_h - u_h^{(n)}$ (left) and total error $u - u_h^{(n)}$ (right).The vertical axis are scaled by 10^{-4}; see also Remark 1.

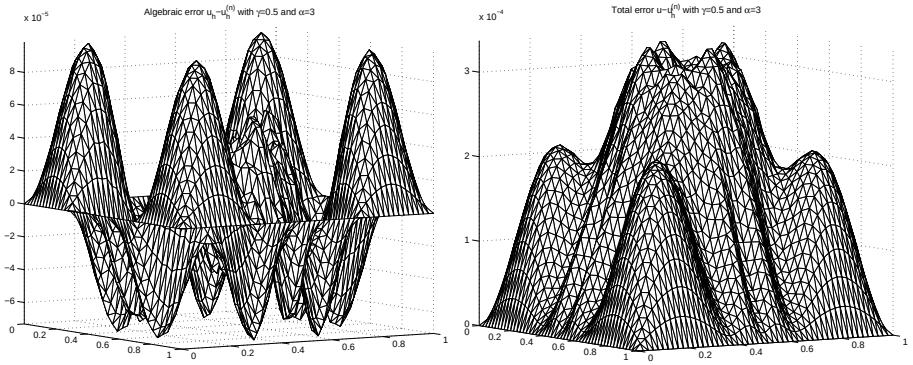

Fig. 4 $\gamma = 0.5$: algebraic error $u_h - u_h^{(n)}$ (left) and total error $u - u_h^{(n)}$ (right). The vertical axes are scaled by 10^{-5} (left) and by 10^{-4} (right); see also Remark 1.

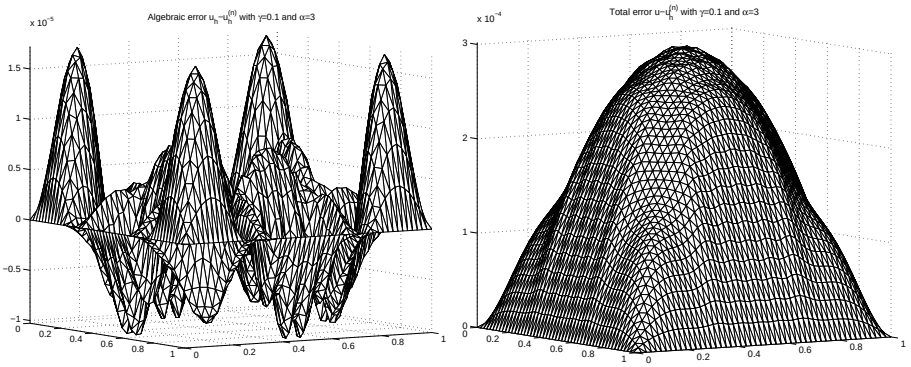

Fig. 5 $\gamma = 0.1$: algebraic error $u_h - u_h^{(n)}$ (left) and total error $u - u_h^{(n)}$ (right). The vertical axes are scaled by 10^{-5} (left) and by 10^{-4} (right); see also Remark 1.

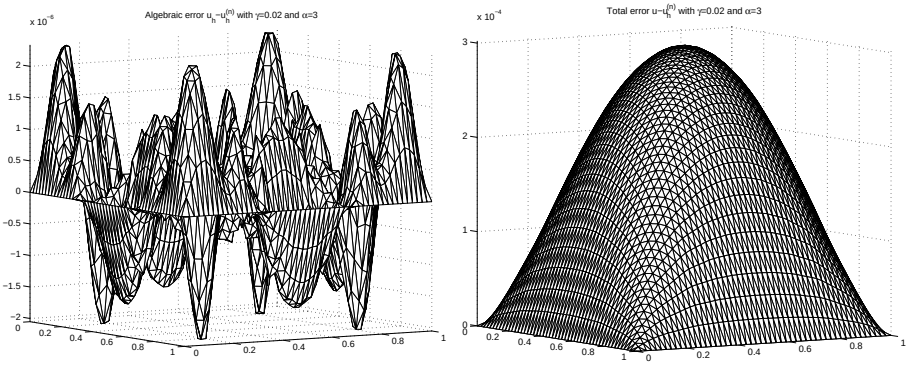

Fig. 6 $\gamma = 0.02$: algebraic error $u_h - u_h^{(n)}$ (left) and total error $u - u_h^{(n)}$ (right). The vertical axes are scaled by 10^{-6} (left) and by 10^{-4} (right); see also Remark 1.

see [14, Theorem 1.3, p. 38]. Therefore, except for $\gamma = 50$, the total error *measured in the energy norm* is dominated by the discretization error, with the globally measured contribution of the algebraic error being orders of magnitude smaller.

When one considers the local distribution of error, the whole picture dramatically changes. Figs. 2–6 show the algebraic and total errors for our choice of parameters. For $\gamma = 50$ the global discretization and algebraic errors measured in the energy norm are of the same order. Both u_h and $u_h^{(n)}$ are piecewise linear and their gradients as well as the gradient of the algebraic error in the function space $\nabla(u_h - u_h^{(n)})$ are piecewise constant. In contrast to that, the gradient of the solution ∇u and therefore also the gradient of the discretization error $\nabla(u - u_h)$ are not piecewise constant. Since we use, for simplicity, zero Dirichlet boundary conditions, we can even write

$$\|\nabla(u - u_h)\|^2 = \|\nabla u\|^2 - \|\nabla u_h\|^2 ;$$

see, e.g., [14, Section 1.5, relation (1.61) and Problem 1.11]. This suggests that the local distribution of the discretization and the algebraic errors can be very different, which is indeed demonstrated by our experiment. Despite the comparable size of the

values

$$\|\nabla(u - u_h)\|^2$$

and

$$\|\nabla(u_h - u_h^{(n)})\|^2 = \|x - x_h\|_A^2$$

for $\alpha = 3$ and $\gamma = 50$, the shape of the total error is fully determined by its algebraic part. With decreasing γ the algebraic error gets smaller and it eventually becomes insignificant. Still, it seems counterintuitive that this happens only after $\|x - x_h\|_A^2$ drops seven orders of magnitude below the squared energy norm of the discretization error $\|\nabla(u - u_h)\|^2$.

It seems also surprising that the algebraic error exhibits such a strongly oscillating pattern. This can be explained in the following way. It is well known that the CG method tends to approximate well the largest and smallest eigenvalues of the system matrix. Assuming exact arithmetic, a close approximation of an eigenvalue means that the corresponding spectral component of the error is diminished, and the method continues in the subsequent iterations as if the given component was not present (leading to "superlinear convergence" of CG); see, e.g., [35] and [27, Theorem 3.3]. In finite precision arithmetic this issue is, in general, more complicated, because due to rounding errors (large) outlying eigenvalues are approximated by computed multiple copies and the convergence of the CG method is delayed; for a survey see [27, Sections 4 and 5]. For the discretized Laplace operator and the relatively small number of iterations this finite precision arithmetic phenomenon is, however, not significant, and the largest and the smallest eigenvalues are approximated at a similar rate.

The approximation of the largest and the smallest eigenvalues means that in the observed range of iterations in our experiment the smooth and the high frequency parts of the error are gradually suppressed by the CG method, while the middle frequency components prevail. Because of the very smooth solution (10), the effect of eliminating the smooth components is dominating, and the algebraic error exhibits an increasingly oscillating pattern as the iteration step n grows.

One may suggest to apply a postprocessing smoothing by performing a few additional steps of the Jacobi, Gauss-Seidel or SOR iterations. In our experiment, such postprocessing smoothing is not efficient. While it smooths out some high frequencies (which are here not significant), it does not change the moderate frequencies which determine the oscillating pattern of the algebraic error.

4 Conclusions

Let us first stress that we do not advocate to solve the Poisson problem on regular domains by the CG method; here CG has no chance to compete with some other specialized fast Poisson solvers. We are also well aware that a proper preconditioner can suppress the reported oscillations of the algebraic error, and that a multigrid solver can in our example naturally balance the local error on individual elements. These facts do not diminish the message which is on purpose kept as simple as possible. Our goal is to show on a simple model problem some important phenomena which should be taken into account when solving large scale mathematical modeling problems in general, where the easy remedies mentioned above might not be applicable. Summarizing, our main message is twofold:

1. The problem of verification in scientific and engineering computing is even more complicated than outlined so nicely in, e.g., [7,18,33]. From the numerical PDE side, a considerable effort should be devoted to overcoming the unrealistic assumption that the matrix computations can be, or even *should be*, performed exactly. If we admit that the linear algebraic systems arising from discretization are not solved exactly (and apparently we have no other option), then the approaches to estimation of the local total error in any method where the discretization basis functions are having local supports (such as in FEM) must be carefully reconsidered. There is a price to pay for using locally supported basis functions. If we do not compute exactly, and we indeed do not, the price can be high. It seems that this has not been fully realized before. Using global estimates and arguing that it is sufficient to make the global algebraic error *sufficiently small*, may not, in general, deliver. Whenever possible, one should aim at the local distribution of the *total error*; an example is for a simple model problem given in [23]. Adaptivity should be based on the local distribution of the total error, not on the estimates for the discretization error with plugging in the computed approximations.
2. From the matrix computation point of view, measuring the error purely on the algebraic level using the backward error analysis and the perturbation theory seems not sufficient. The user needs information about the local behavior of the error in the function space. Application of the state-of-the-art algebraic backward error analysis and perturbation theory, however, do not make it easy to obtain this information.

From both the numerical PDE and the numerical linear algebra sides it should be admitted that matrix computations can not be considered a separate (black box) part of the numerical PDE solution process. Apart from relatively simple cases, black box approaches may not work. Even worse, they are philosophically wrong. Even if direct algebraic solvers are applicable, the resulting algebraic error might not be small and it should be considered (or the opposite should be rigorously justified). The stopping criteria in iterative algebraic solvers should be linked, in an optimal case, with fully computable and locally efficient (on individual elements) *a posteriori* error bounds that allow to keep an appropriate balance between the discretization and the algebraic parts of the error; see, e.g., the discussion in the book by Bangerth and Rannacher [9], in the recent papers [31,25,4,2,12,13,11,23,34,20], in the habilitation thesis [37], in the Ph.D. thesis [26], and the references given there. Although this goal seems highly ambitious and is certainly very difficult to achieve, the near future will certainly bring new exciting results in that direction.

Acknowledgments. We thank André Gaul and Petr Tichý for their help with the numerical experiments, and Howard Elman, Tomáš Vejchodský and Martin Vohralík for pointing out several inaccuracies in the original manuscript.

References

1. M. Arioli, *A stopping criterion for the conjugate gradient algorithms in a finite element method framework*, Numer. Math., 97 (2004), pp. 1–24.
2. M. Arioli, E. H. Georgoulis, and D. Loghin, *Convergence of inexact adaptive finite element solvers for elliptic problems*, Technical Report RAL-TR-2009-021, SFTC RAL (2009).

3. M. Arioli, D. Loghin, and A. J. Wathen, *Stopping criteria for iterations in finite element methods*, Numer. Math., 99 (2005), pp. 381–410.
4. M. Arioli and D. Loghin, *Stopping criteria for mixed finite element problems*, Electronic Trans. Numer. Anal., 29 (2008), pp. 178–192.
5. M. Arioli, E. Noulard, and A. Russo, *Stopping criteria for iterative methods: applications to PDE's*, Calcolo, 38 (2001), pp. 97–112.
6. I. Babuška, *Numerical stability in problems of linear algebra*, SIAM J. Numer. Anal., 9 (1972), pp. 53–77.
7. I. Babuška and J. T. Oden, *Verification and validation in computational engineering and science*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 4057–4066.
8. I. Babuška and T. Strouboulis, *The Finite Element Method and Its Reliability*, Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, New York, 2001.
9. W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser Verlag, Basel, 2001.
10. B. J. C. Baxter and A. Iserles, *On the foundations of computational mathematics*, in Handbook of Numerical Analysis, Vol. XI, North-Holland, Amsterdam, 2003, pp. 3–34.
11. R. Becker and S. Mao, *Convergence and quasi-optimal complexity of a simple adaptive finite element method*, M2AN Math. Model. Numer. Anal., 43 (2009), pp. 1203–1219.
12. C. Burstedde and A. Kunoth, *Fast iterative solution of elliptic control problems in wavelet discretization*, J. Comput. Appl. Math., 196 (2006), pp. 299–319.
13. C. Burstedde and A. Kunoth, *A wavelet-based ested iteration-inexact conjugate gradient algorithm for adaptively solving elliptic PDEs*, Numer. Algorithms, 48 (2008), pp. 161–188.
14. H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.
15. K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equations*, Cambridge University Press, Cambridge, 1996.
16. E. Feireisl, *Dynamics of Viscous Compressible Fluids*, vol. 26 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2004.
17. E. Feireisl and A. Novotný, *Singular Limits in Thermodynamics of Viscous Fluids*, Advances in Mathematical Fluid Mechanics, Birkhäuser Verlag, Basel, 2009.
18. W. N. Gansterer, Y. Bai, R. M. Day, and R. C. Ward, *A framework for approximating eigenpairs in electronic structure computations*, Comp. Sci. Eng., 6 (2004), pp. 50–59.
19. W. Hackbusch, *Elliptic Differential Equations – Theory and Numerical Treatment*, Springer-Verlag, Berlin, 1992.
20. H. Harbrecht and R. Schneider, *On error estimation in finite element methods without having Galerkin orthogonality*, technical report, Universität Bonn, 2009.
21. M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
22. N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.
23. P. Jiránek, Z. Strakoš, and M. Vohralík, *A posteriori error estimates including algebraic error and stopping criteria for iterative solvers*, SIAM J. Sci. Comput., 32 (2010), pp. 1567–1590.
24. J. Málek and K. R. Rajagopal, *On the modeling of inhomogeneous incompressible fluid-like bodies*, Mechanics of Materials, 38 (2006), pp. 233–242.
25. D. Meidner, R. Rannacher, and J. Vihharev, *Goal-oriented error control of the iterative solution of finite element equations*, J. Numer. Math., 17 (2009), pp. 143–172.
26. A. Międlar, *Inexact Adaptive Finite Element Methods for Elliptic PDE Eigenvalue Problems*, PhD thesis, Institut für Mathematik, TU Berlin, 2010.
27. G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
28. J. T. Oden, I. Babuška, F. Nobile, Y. Feng, and R. Tempone, *Theory and methodology for estimation and control of errors due to modeling, approximation and uncertainty*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 195–204.
29. W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409.
30. B. N. Parlett, *The contribution of J. H. Wilkinson to numerical analysis*, in A History of Scientific Computing (Princeton, NJ, 1987), ACM Press Hist. Ser., ACM, New York, 1990, pp. 17–30.

31. R. Rannacher, A. Westenberger, and W. Wollner, *Adaptive finite element solution of eigenvalue problems: Ballancing of discretization and iteration error*, J. Numer. Math., 18 (2010), pp. 303–327.
32. J.-L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. Assoc. Comput. Mach., 14 (1967), pp. 543–548.
33. P. J. Roache, *Building PDE codes to be verifiable and validatable*, Comp. Sci. Eng., 6 (2004), pp. 30–38.
34. D. Silvester and V. Simoncini, *An optimal iterative solver for symmetric indefinite systems stemming from mixed approximation*, TOMS, to appear (2011).
35. A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients*, Numerische Mathematik, 48 (1986), pp. 543–560.
36. E. Stein, ed., *Error-Controlled Adaptive Finite Elements in Solid Mechanics*, J. Wiley, Chichester, 2003.
37. M. Vohralík, *A Posteriori Error Estimates, Stopping Criteria, and Inexpensive Implementations for Error Control and Efficiency in Numerical Simulations*, Habilitation thesis, Université Pierre et Marie Curie, 2010.
38. J. H. Wilkinson, *Modern error analysis*, SIAM Rev. 13 (1971), pp. 548–568.