An adaptive WEM algorithm for solving elliptic boundary value problems in fairly general domains

SIAM J. Sci. Comput.
Vol. 28, No. 6, pp. 2114–2138

AN ADAPTIVE WEM ALGORITHM FOR SOLVING ELLIPTIC BOUNDARY VALUE PROBLEMS IN FAIRLY GENERAL DOMAINS[*]

S. BERRONE[†] AND T. KOZUBEK[‡]

Abstract. In this paper, we introduce a simple adaptive wavelet element algorithm similar to the Cohen–Dahmen–DeVore algorithm [A. Cohen, W. Dahmen, and R. DeVore, *Math. Comp.*, 70 (2001), pp. 27–75]. The main difference is that we do not assume knowledge of the many constants appearing therein. The algorithm is easy to implement and applicable to a large class of problems in fairly general domains. The efficiency is illustrated by several two-dimensional numerical examples and compared with an adaptive finite element method.

Key words. adaptive wavelet and finite element methods, elliptic operator equations, rates of convergence

AMS subject classifications. 65T60, 42C40, 65N30, 65N50, 65N55

DOI. 10.1137/04062014X

1. Introduction.

Wavelets have become an efficient tool for solving many problems arising in theory and in practice. Let us mention, for example, signal analysis, compression of information, geometry representation, and numerical analysis. In this paper we deal with wavelets used for the *adaptive* solution of elliptic boundary value problems in $\mathbb{R}^d$, $d = 1, 2, 3$. Adaptivity is a key ingredient in modern numerical computation which helps to improve significantly the performance of the method. When dealing with the finite element method (FEM), adaptivity occurs in the context of grid refinement, in contrast to the function selection used in wavelet discretization methods such as the wavelet element method (WEM). A detailed discussion of the adaptive (nonlinear) refinement advantages with respect to the uniform (linear) refinement can be found in [12, 23].

We introduce an adaptive WEM algorithm applicable to a large class of problems defined in fairly general domains. The key ideas of our algorithm follow the fundamental Cohen–Dahmen–DeVore approach to wavelet adaptivity for PDEs [2, 12, 13], but we do not assume knowledge of the many constants appearing therein. Industrial applications of the Cohen–Dahmen–DeVore algorithm in all its aspects still seem to be quite hard. Here we propose an adaptive algorithm based on the same ideas, but whose practical application is easier. For the proposed algorithm we do not provide any proof of either convergence or optimality, as in [12]. Our goal is rather (i) *to easily employ the standard solvers, such as* CG, BiCG, GMRES*, in order to improve the actual approximation of the coefficient vector;* (ii) *to easily control the solution process by a small set of parameters;* (iii) *to extend the applicability of the adaptive algorithm to more realistic problems in fairly general geometrical domains* Ω *in* $\mathbb{R}^d$; (iv) *to intro-*

[*] Received by the editors October 24, 2005; accepted for publication (in revised form) May 31, 2006; published electronically December 5, 2006. This research was supported by the European project Breaking Complexity, HPRN-CT-2002-00286, by grants IAA1075402 and 1ET400300415 of the Grant Agency of the Czech Academy of Sciences, and by the Italian funds Miur PRIN 2004 “Adattività e avanzamento in tempo nei modelli numerici alle derivate parziali” and Indam-GNCS-2005 “Metodi numerici per lo studio di problemi evolutivi multiscala.”

http://www.siam.org/journals/sisc/28-6/62014.html

[†] Dipartimento di Matematica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy (sberrone@calvino.polito.it).

[‡] Department of Applied Mathematics, VSB-Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic (tomas.kozubek@vsb.cz).

duce an approximate computation of the integrals for computing the nonzero entries of the stiffness matrix $\mathbb{A}$ and the right-hand side vector $\mathbf{f}$ *and to possibly minimize their number.*

To construct multiresolution systems on fairly general domains in $\mathbb{R}^d$, we use a WEM construction described in [10, 11, 4], but the proposed algorithm itself does not depend on the type of the used wavelet basis [20, 26] nor on the WEM construction [22, 14, 10]. The key point of the WEM is the splitting of the domain into subdomains in such a way that each of them can be mapped to a reference hypercube. Bases on the reference domain in $\mathbb{R}^d$ are constructed by the tensor product of scaling and wavelet functions defined on the unit interval. In our implementation wavelet bases on the interval are formed by biorthogonal B-spline wavelets with optimal localization introduced in [26]. To obtain a globally continuous biorthogonal wavelet basis, one has to enforce matching conditions across interelement boundaries. The construction enables one to fulfill additional important features such as symmetry, vanishing moments, and a minimal support.

Problem setting and general requirements on the wavelet bases are briefly mentioned in section 2. For space reasons we assume some knowledge of these aspects, referring for more details to [16, 17, 18] and the references therein. In section 3, we introduce the adaptive WEM algorithm and describe all ingredients in details. In section 4, we illustrate the applicability of this adaptive WEM method to quite general situations; some numerical examples involving singularities and two-dimensional complex geometries requiring mapping and matching are considered. Conclusions are summarized in section 5. Finally, in the appendix, we compare some results obtained by the proposed algorithm with results obtained using a well-assessed adaptive finite element code [30, 1]. This comparison does not attempt to claim that one method outperforms the other, but aims at comparing some aspects of the solution process of the two adaptive methods.

2. General settings.

In this section, we introduce a class of elliptic boundary value problems which are the focus of our interest. Finally, using such basis, we show how to transform the problem into an equivalent l_2-problem.

2.1. Problem setting.

Let Ω be a bounded domain in $\mathbb{R}^d$, $d = 1, 2, 3$, with the Lipschitz boundary $\partial\Omega$. Further, let $H := H(\Omega)$ be a Hilbert space and $\mathcal{A} :\ H \to H'$ an elliptic operator of the second order, where H' is a dual to H and $H \hookrightarrow L^2(\Omega) \hookrightarrow H'$. Due to simplicity of our presentation we will be concerned with the following abstract pure Dirichlet boundary value problem: for given $f \in H' := H^{-1}(\Omega)$ find $u \in H := H_0^1(\Omega)$ such that

$$\begin{cases} \mathcal{A}u = f & \text{in } \Omega, \\ \quad u = 0 & \text{on } \partial\Omega. \end{cases} \tag{2.1}$$

The corresponding weak form to (2.1) is

$$\begin{cases} \text{Find } u \in H \text{ such that} \\ a(u, v) = \langle f, v \rangle \quad \forall v \in H, \end{cases} \tag{2.2}$$

where $a(.\,,.) :\ H \times H \to \mathbb{R}$ is the continuous bilinear form defined by $a(u, v) := \langle \mathcal{A}u, v \rangle$ and $\langle .\,,. \rangle$ denotes the standard duality pairing between H and H'.

We assume that problem (2.2) is well-posed, i.e., there are positive finite constants $c_{\mathcal{A}}$, $C_{\mathcal{A}}$ such that $c_{\mathcal{A}}\|v\|_H \leq \|\mathcal{A}v\|_{H'} \leq C_{\mathcal{A}}\|v\|_H,\ v \in H$. This immediately implies the existence of a unique solution to (2.2).

2.2. General requirements on the wavelet bases. We are interested in solving problem (2.2) using the *adaptive wavelet element approach*, i.e., a Galerkin method with a suitably chosen wavelet basis for H in $\boldsymbol{\psi} = \{\psi_\lambda :\ \lambda \in \mathcal{J}\}$, where $\mathcal{J}$ is an infinite index set, and an adaptive procedure for the basis functions selection. In the following, we list the relevant properties of the wavelet basis $\boldsymbol{\psi}$.

For any $v \in L^2(\Omega)$ we have a *unique* expansion $v = \sum_{\lambda\in\mathcal{J}} v_\lambda\psi_\lambda = \mathbf{v}^T\boldsymbol{\psi}$ which induces an isomorphism between H and l_2. More precisely, there is a scaling diagonal matrix $\mathbb{D} = \operatorname{diag}(d_\lambda :\ \lambda \in \mathcal{J})$ and positive constants c_ψ, C_ψ such that $c_\psi\|\mathbf{v}\|_{l_2} \leq \|\mathbf{v}^T\mathbb{D}^{-1}\boldsymbol{\psi}\|_H \leq C_\psi\|\mathbf{v}\|_{l_2}$. For example, when H is the Sobolev space $H^t(I)$ with $I = [0,1]$, a canonical choice of d_λ is $d_\lambda = 2^{t|\lambda|}$; in general for $H = H^1(\Omega)$ we choose $d_\lambda = \sqrt{a(\psi_\lambda,\psi_\lambda)}$ [19, 21, 27, 28]. We assume that basis functions have *vanishing moments* and are *locally supported* with $\operatorname{diam}(\operatorname{supp}\psi_\lambda) \sim 2^{-|\lambda|}$, where $\lambda \in \mathcal{J}$ and $|\lambda|$ is the level of the basis function ψ_λ (see [20, 26, 10]).

2.3. An equivalent l_2-problem. Given a basis with the mentioned properties, we can transform problem (2.2) over a function space H into an algebraic system of equations over the corresponding sequence space. More precisely, we obtain the following equivalent l_2-problem: $\mathbb{A}\mathbf{x} = \mathbf{f}$, where $\mathbb{A} = \mathbb{D}^{-1}\langle\mathcal{A}\boldsymbol{\psi},\boldsymbol{\psi}\rangle\mathbb{D}^{-1} = \left(d_\lambda^{-1}d_\mu^{-1}\langle\mathcal{A}\psi_\mu,\psi_\lambda\rangle\right)_{\lambda,\mu\in\mathcal{J}}$ is an infinite matrix and $\mathbf{f} = \mathbb{D}^{-1}\langle f,\boldsymbol{\psi}\rangle = \left(d_\lambda^{-1}\langle f,\psi_\lambda\rangle\right)_{\lambda\in\mathcal{J}}$ is an infinite vector. It is well known [12, 13, 16, 18] that if $u = \mathbf{x}^T\mathbb{D}^{-1}\boldsymbol{\psi}$ is the scaled wavelet representation of the solution to (2.2), the problem $\mathbb{A}\mathbf{x} = \mathbf{f}$ is an equivalent problem in l_2 to $\mathcal{A}u = f$. In addition, there are positive constants $c_{\mathbb{A}}$, $C_{\mathbb{A}}$ such that $c_{\mathbb{A}}\|\mathbf{v}\|_{l_2} \leq \|\mathbb{A}\mathbf{v}\|_{l_2} \leq C_{\mathbb{A}}\|\mathbf{v}\|_{l_2}$, $\mathbf{v} \in l_2$, with lower and upper estimates $c_{\mathbb{A}} \geq c_\psi^2 c_{\mathcal{A}}$ and $C_{\mathbb{A}} \leq C_\psi^2 C_{\mathcal{A}}$, respectively.

Our infinitely dimensional problem $\mathbb{A}\mathbf{x} = \mathbf{f}$ can be now approximated by a finite-dimensional approximation

$$\mathbb{A}_{\Lambda\times\Lambda}\mathbf{x}_\Lambda = \mathbf{f}_\Lambda, \tag{2.3}$$

where $\mathbf{f}_\Lambda = \left(d_\lambda^{-1}\langle f,\psi_\lambda\rangle\right)_{\lambda\in\Lambda}$, $\mathbb{A}_{\Lambda\times\Lambda} = \left(d_\lambda^{-1}d_\mu^{-1}\langle\mathcal{A}\psi_\mu,\psi_\lambda\rangle\right)_{\lambda,\mu\in\Lambda}$, $\Lambda \subset \mathcal{J}$, and the approximation of the solution u is given by

$$u_\Lambda = \sum_{\mu\in\Lambda} d_\mu^{-1}x_\mu\psi_\mu = \mathbf{x}_\Lambda^T\mathbb{D}_\Lambda^{-1}\boldsymbol{\psi}_\Lambda. \tag{2.4}$$

With $\mathbb{A}$ positive definite, all finite sections $\mathbb{A}_{\Lambda\times\Lambda}$ have $\|\mathbb{A}_{\Lambda\times\Lambda}\|_2 \sim 1$, $\|\mathbb{A}_{\Lambda\times\Lambda}^{-1}\|_2 \sim 1$, and $1 \sim \operatorname{cond}_2(\mathbb{A}_{\Lambda\times\Lambda}) \leq \operatorname{cond}_2(\mathbb{A}) < \infty$ [19, 21, 27].

3. An adaptive WEM algorithm.

In this section, we present our adaptive algorithm; particular settings of the parameters are presented in section 4.

We start initializing all parameters of the algorithm and choosing an appropriate solver of the resulting algebraic systems. The algorithm consists of an outer adaptive iterative process, where all iterations consist of several steps. At iteration n, we have three sets of basis functions, whose sets of indices will be denoted by Λ_n, $\tilde{\Lambda}_n$, and $\overline{\Lambda}_n$. More precisely, Λ_n is the set of indices of the active basis functions, whereas $\tilde{\Lambda}_n$ consists of indices corresponding to the functions that can be activated in this iteration. In the definition of this set one can include essential information on the data of the problem to deal with particular situations. Finally, $\overline{\Lambda}_n$ is the set of indices of basis functions after a thresholding step in which we remove from Λ_n the indices of those basis functions whose coefficients are less significant. We name the sets Λ_n, $\tilde{\Lambda}_n$ and the corresponding functions *active* and *activable*, respectively.

The first step of each iteration consists in enlarging the set $\overline{\Lambda}_{n-1}$ to get Λ_n. In the first iteration $n = 0$ of the adaptive procedure, we add the indices of all scaling functions at the starting scaling level to $\overline{\Lambda}_{-1} = \emptyset$. We obtain the active set Λ_0. In the next iterations, we use the set $\tilde{\Lambda}_{n-1}$ which is constructed by the indices of suitably chosen basis functions whose supports interact with the supports of the functions in $\overline{\Lambda}_{n-1}$. We compute the coefficients of the residual function corresponding to the solution of the previous outer iteration with respect to the dual basis functions whose primal functions are in $\tilde{\Lambda}_{n-1}$. Then, to get Λ_n, we add to $\overline{\Lambda}_{n-1}$ a set of suitably chosen indices $\delta\Lambda_{n-1} \subset \tilde{\Lambda}_{n-1}$ chosen in order to catch as much as possible the bulk of the residuum with the minimum number of new basis functions. We use all active functions corresponding to $\Lambda_n = \overline{\Lambda}_{n-1} \cup \delta\Lambda_{n-1}$ to assemble the stiffness matrix $\mathbb{A}_{\Lambda_n \times \Lambda_n}$ and the right-hand side vector $\mathbf{f}_{\Lambda_n}$. Using a chosen iterative method for the inner iterations, we improve the current approximation of the solution coefficients $\mathbf{x}_{\Lambda_n}$ by approximately solving the corresponding linear algebraic system $\mathbb{A}_{\Lambda_n \times \Lambda_n}\mathbf{x}_{\Lambda_n} = \mathbf{f}_{\Lambda_n}$. Further, we remove the indices of all negligible functions corresponding to the smallest (less significant) absolute entries of the coefficient vector $\mathbf{x}_{\Lambda_n}$ from Λ_n to keep $\overline{\Lambda}_n$ as small as possible. We check an appropriate stopping criterion for the outer iteration in terms of the reduction of the residuum, and if it is not satisfied, we repeat all of the above-mentioned steps. Otherwise, we accept the actual approximation of the coefficient vector as the final solution and stop iterations.

3.1. Computation of $\mathbb{A}_{\boldsymbol{\Lambda}\times\boldsymbol{\Lambda}}$ and $\mathbf{f}_{\boldsymbol{\Lambda}}$. When we have chosen our finite set of indices Λ the solution of the linear system (2.3) gives the approximation u_Λ of the solution u by (2.4). An efficient approximation of the integrals involved in the definitions of the elements $a_{\lambda,\mu}$ of the stiffness matrix and f_λ of the right-hand side vector plays a fundamental role [8, 7]. In this subsection, we recall some results of [8] which are independent of the algorithm presented there for the approximation of the integrals. We introduce a relative precision $\mathrm{tol}_{int} < 1$ in computing the elements of the stiffness matrix and the right-hand side vector. Let us denote by $\mathbb{A}^*_{\Lambda\times\Lambda}$ and $\mathbf{f}^*_\Lambda$ the numerical approximations of the stiffness matrix and the right-hand side vector, respectively. Set $\mathbb{E}_\Lambda = \mathbb{A}_{\Lambda\times\Lambda} - \mathbb{A}^*_{\Lambda\times\Lambda}$, $\mathbf{h}_\Lambda = \mathbf{f}_\Lambda - \mathbf{f}^*_\Lambda$ and define

$$i(\lambda,\mu) = \begin{cases} 1 & \text{if } |\operatorname{supp}\psi_\lambda \cap \operatorname{supp}\psi_\mu| > 0, \\ 0 & \text{otherwise.} \end{cases}$$

PROPOSITION 3.1. *Let us fix, independently of λ and μ, a nonincreasing $l^1(\mathbb{N})$-sequence $\delta = (\delta_l)_{l\in\mathbb{N}}$ with strictly positive elements, whose $l^1(\mathbb{N})$-norm is close to one. Let $\mathbb{E}_\Lambda = (e_{\lambda,\mu})_{\lambda,\mu\in\Lambda}$ satisfy*

$$|e_{\lambda,\mu}| \precsim \mathrm{tol}_{int}\, i(\lambda,\mu)\delta_{||\lambda|-|\mu||}2^{-d/2||\lambda|-|\mu||}\|\mathbb{A}_{\Lambda\times\Lambda}\|_2. \tag{3.1}$$

Then $\|\mathbb{E}_\Lambda\|_2 \precsim \mathrm{tol}_{int}\|\mathbb{A}_{\Lambda\times\Lambda}\|_2$.

$\mathbb{A}$ being positive definite, we simply have $\|\mathbb{E}_\Lambda\|_2 \precsim \mathrm{tol}_{int}$.

PROPOSITION 3.2. *Let us fix, independently of λ, a nonincreasing $l^2(\mathbb{N})$-sequence $\delta = (\delta_l)_{l\in\mathbb{N}}$ with strictly positive elements, whose $l^2(\mathbb{N})$-norm is close to one. Let $\mathbf{h}_\Lambda = (h_\lambda)_{\lambda\in\Lambda}$ satisfy*

$$|h_\lambda| \precsim \mathrm{tol}_{int}\delta_{|\lambda|}2^{-d/2|\lambda|}\|\mathbf{f}_\Lambda\|_2. \tag{3.2}$$

Then $\|\mathbf{h}_\Lambda\|_2 \lesssim \mathrm{tol}_{int}\|\mathbf{f}_\Lambda\|_2$.

In each iteration n we verify (3.2) with the approximation of $\|\mathbf{f}_\Lambda\|_2$ given by $\|\mathbf{f}_{\Lambda_n}\|_2$. In our adaptive algorithm we usually form a sequence of enlarging sets Λ_n, $n = 0, 1, \ldots$, so $\|\mathbf{f}_{\Lambda_n}\|_2$ is increasing and we do not need to recompute any previously computed elements of $\mathbf{f}_{\Lambda_k}$, $k = 0, \ldots, n-1$.

Under the assumptions of the previous propositions and for our finite-dimensional l_2-problem we can conclude that there is $\mathrm{tol}_{int_0} > 0$ such that for all $0 < \mathrm{tol}_{int} \leq \mathrm{tol}_{int_0}$ the following estimate holds:

$$\|\mathbf{x}_\Lambda - \mathbf{x}^*_\Lambda\|_2 \lesssim \mathrm{tol}_{int}\|\mathbf{x}_\Lambda\|_2,$$

where $\mathbf{x}^*_\Lambda$ is the coefficient vector computed from the approximated linear system $\mathbb{A}^*_{\Lambda\times\Lambda}\mathbf{x}^*_\Lambda = \mathbf{f}^*_\Lambda$ and $u^*_\Lambda = (\mathbf{x}^*_\Lambda)^T\mathbb{D}^{-1}_\Lambda\boldsymbol{\psi}_\Lambda$ is the corresponding approximate solution.

In the following it will be useful to evaluate the effect of the approximation in computing the integrals on the norm of the residuum. Let us define $\mathcal{J}_n = \Lambda_n \cup \tilde{\Lambda}_n$, $\mathbf{r}_{\mathcal{J}_n} = \mathbf{f}_{\mathcal{J}_n} - \mathbb{A}_{\mathcal{J}_n\times\Lambda_n}\mathbf{x}_{\Lambda_n}$, and $\mathbf{r}^*_{\mathcal{J}_n} = \mathbf{f}^*_{\mathcal{J}_n} - \mathbb{A}^*_{\mathcal{J}_n\times\Lambda_n}\mathbf{x}^*_{\Lambda_n}$. According to the results of the previous propositions and properties such as the well conditioning and stability of the l_2-problem, we have

$$\|\mathbf{r}_{\mathcal{J}_n} - \mathbf{r}^*_{\mathcal{J}_n}\|_2 \lesssim \mathrm{tol}_{int}\|\mathbf{f}_{\mathcal{J}_n}\|_2.$$

In our algorithm we want to find a solution $\mathbf{x}_\Lambda$ which satisfies $\|\mathbf{f}_{\mathcal{J}} - \mathbb{A}_{\mathcal{J}\times\Lambda}\mathbf{x}_\Lambda\|_{l_2} \leq \varepsilon\|\mathbf{f}_{\mathcal{J}}\|_{l_2}$. Therefore we propose to choose the parameter tol_{int} such that $\mathrm{tol}_{int}\|\mathbf{f}_{\mathcal{J}_n}\|_2$ is negligible with respect to $\varepsilon\|\mathbf{f}_{\mathcal{J}_n}\|_2$, i.e., $\mathrm{tol}_{int} \ll \varepsilon$.

Different approaches for computing entries of stiffness matrix and right-hand side can be found in [2, 3].

3.2. Initialization part. We choose an appropriate solver of the resulting linear algebraic systems, such as *steepest descent, CG, BiCG, GMRES*, and set its stopping tolerance tol_{grad}, which is an upper bound for the ratio between the final algebraic residuum and the algebraic residuum corresponding to the initial guess of the current outer iteration (see (3.3)). The choice of tol_{int} has to be done in such a way that the approximation of the integrals gives a negligible contribution in the approximation of the bulk of the residuum. Further, we set all remaining parameters: relative activating and thresholding tolerances tol_{on} and tol_{off} involved in adding and removing indices of basis functions to and from the active set Λ, respectively. Finally, we set the relative target accuracy ε of the adaptive process, i.e., the final reduction factor of the total residuum.

3.3. Activating new basis functions. In each iteration $n > 0$ we want to catch and reduce the bulk of the residuum $\mathbf{r} = \mathbf{f}_{\mathcal{J}} - \mathbb{A}_{\mathcal{J}\times\overline{\Lambda}_{n-1}}\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}$ and improve the solution with its components corresponding to the most significant entries of the residuum. In practice, we choose those significant components of the residuum whose indices belong to the activable set $\tilde{\Lambda}_{n-1}$. This procedure is realized by sorting the absolute values of the entries of the vector $\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}} = \tilde{\mathbf{f}}_{\tilde{\Lambda}_{n-1}} - \mathbb{A}_{\tilde{\Lambda}_{n-1}\times\overline{\Lambda}_{n-1}}\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}$ in decreasing order, summing successively the squares of the largest ones and adding the corresponding indices to $\delta\Lambda_{n-1}$, until the target percentage of $\|\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}\|^2_2$ is reached, i.e., while $\sum_{\lambda\in\delta\Lambda_{n-1}}|\tilde{r}_\lambda|^2 \leq \mathrm{tol}^2_{on}\|\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}\|^2_2$. Then we set $\Lambda_n = \overline{\Lambda}_{n-1} \cup \delta\Lambda_{n-1}$ and extend the old approximation of the coefficient vector $\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}$ by zeros at the positions corresponding to indices $\lambda \in \delta\Lambda_{n-1}$ to get the initial coefficient vector $\mathbf{x}^{\mathrm{init}}_{\Lambda_n}$. In the first outer iteration $n = 0$ we have $\overline{\Lambda}_{-1} = \emptyset$, $\overline{\mathbf{x}}_{\overline{\Lambda}_{-1}} = 0$ and $\delta\Lambda_{-1}$ is the set of indices of all scaling functions at the starting level j_0 greater than or equal to the minimum construction level of basis functions.

FUNCTION $[\Lambda_n, \delta\Lambda_{n-1}] =$ Activate$(\overline{\Lambda}_{n-1}, \tilde{\Lambda}_{n-1}, \tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}, \mathrm{tol}_{on})$.

(i) *Set* $N := \mathrm{card}\,\tilde{\Lambda}_{n-1}$ *and sort the absolute values of the entries of* $\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}$ *in decreasing order. The vector* $\lambda^* := \lambda^*(\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}) = (\lambda_1^*, \lambda_2^*, \dots, \lambda_N^*)$ *is the vector of permuted indices which gives the decreasing rearrangement* $\tilde{\mathbf{r}}^*_{\tilde{\Lambda}_{n-1}} = (|\tilde{r}_{\lambda_1^*}|, |\tilde{r}_{\lambda_2^*}|, \dots, |\tilde{r}_{\lambda_N^*}|)$ *of* $\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}$*; then compute* $\|\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}\|_2^2$.

(ii) *For* $k = 1, 2, \dots,$ *compute the sum* $\sum_{l=1}^{k} |\tilde{r}_{\lambda_l^*}|^2$ *in order to find the biggest* k *such that this sum is less than or equal to* $\mathrm{tol}_{on}^2 \|\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}\|_2^2$. *For the resulting* k, *set* $\delta\Lambda_{n-1} := \{\lambda_l^*;\ l = 1, \dots, k\}$ *and* $\Lambda_n := \overline{\Lambda}_{n-1} \cup \delta\Lambda_{n-1}$.

FUNCTION $\mathbf{x}^{\mathrm{init}}_{\Lambda_n} =$ Extend$(\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}, \overline{\Lambda}_{n-1}, \delta\Lambda_{n-1})$.

(i) *Define* $\mathbf{x}^{\mathrm{init}}_{\Lambda_n}$ *keeping the coefficients* $\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}$ *in the positions corresponding to* $\overline{\Lambda}_{n-1}$ *and set to zero the coefficients corresponding to the added basis functions* ψ_μ, $\mu \in \delta\Lambda_{n-1}$.

3.4. Assembling $\mathbf{f}_{\boldsymbol{\Lambda}_n}$ and $\mathbb{A}_{\boldsymbol{\Lambda}_n \times \boldsymbol{\Lambda}_n}$.

For a given Λ_n we need to assemble the right-hand side vector $\mathbf{f}_{\Lambda_n}$ and the stiffness matrix $\mathbb{A}_{\Lambda_n \times \Lambda_n}$. In the computation of $a_{\lambda,\mu}$ and f_λ we use numerical quadrature formulas that allow us to control the approximation of the elements as in Propositions 3.1 and 3.2.

Computing the entries of $\mathbf{f}_{\boldsymbol{\Lambda}_n}$. We form a sequence $\{\mathbf{f}^{(J)}_{\Lambda_n}\}_{J\in\mathbb{N}}$ of approximations to $\mathbf{f}_{\Lambda_n}$ with increasing accuracy to satisfy the test criterion (3.2):

1. *Fix a level* $J > \max|\lambda| + 2$ *and compute* $\mathbf{f}^{(J)}_{\Lambda_n}$ *using the Simpson composite formula on the corresponding grid.*
2. *Using the same evaluations of the right-hand side, compute* $\mathbf{f}^{(J-1)}_{\Lambda_n}$ *at the subgrid of the level* $J - 1$.
3. *Set* $\mathbf{h}^{(J)}_{\Lambda_n} := \mathbf{f}^{(J)}_{\Lambda_n} - \mathbf{f}^{(J-1)}_{\Lambda_n}$ *and* $\mathbf{f}^{(J+1)}_{\Lambda_n} := \mathbf{f}^{(J)}_{\Lambda_n}$.
4. *For all entries* $h^{(J)}_\lambda$ *of* $\mathbf{h}^{(J)}_{\Lambda_n}$, *do*
   - *if* $|h^{(J)}_\lambda| > \mathrm{tol}_{int}\delta_{|\lambda|}2^{-d/2|\lambda|}\|\mathbf{f}^{(J)}_{\Lambda_n}\|_2$, *then recompute the element* $f^{(J+1)}_\lambda$ *of* $\mathbf{f}^{(J+1)}_{\Lambda_n}$ *using grid nodes of the level* $J + 1$.
5. *If the previous test criterion is not satisfied for all entries* $h^{(J)}_\lambda$, $\lambda \in \Lambda_n$, *then set* $J = J + 1$ *and go back to the point* 3. *Otherwise output* $\mathbf{f}_{\Lambda_n} := \mathbf{f}^{(J+1)}_{\Lambda_n}$.

We set $\delta_{|\lambda|} = 1/|\lambda|$ in (3.2), and the level J is chosen in such a way that we are able to detect all singularities in f.

Remark 3.3. The choice $J > \max|\lambda| + 2$ in the previous algorithm is not optimal. Nevertheless, in many applications in which the maximal level needed by the approximation process is not too large, the evaluation of the right-hand side on a fine fixed grid leads to a faster approximation of the integrals than an adaptive method like the one presented for the entries of the stiffness matrix. An algorithm similar to the one proposed for the stiffness matrix can be applied for the right-hand side too.

Computing the entries of $\mathbb{A}_{\boldsymbol{\Lambda}_n \times \boldsymbol{\Lambda}_n}$. In this case, we apply a composite Gauss quadrature formula to compute $a_{\lambda,\mu}$. To satisfy (3.1) for all entries we can use a modification of the above-mentioned adaptive scheme applying a sequence of Gauss quadrature formulas on each dyadic cube in the intersection of the supports of test and trial functions [5] to satisfy the test criterion (3.1), where $\delta_{||\lambda|-|\mu||} = 1/(||\lambda|-|\mu||^2)$. In this approach only dyadic cubes of level $\max\{|\lambda|, |\mu|\}$ are involved in the computation of the integrals.

For $n > 0$, the entries $a_{\lambda,\mu}$ and f_λ with $\lambda \in \delta\Lambda_{n-1} \subset \tilde{\Lambda}_{n-1}$ and $\mu \in \overline{\Lambda}_{n-1}$ have been computed and stored in the previous iteration during the computation of the residuum $\tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}$. Thus we compute only the remaining entries.

Function $[\mathbb{A}_{\Lambda_n\times\Lambda_n}, \mathbf{f}_{\Lambda_n}] = \text{Assemble}(\overline{\Lambda}_{n-1}, \delta\Lambda_{n-1})$.

(i) *Compute all nonzero entries* $a_{\lambda,\mu}$*, with* $(\lambda,\mu) \in \{\{\overline{\Lambda}_{n-1} \times \delta\Lambda_{n-1}\} \cup \{\delta\Lambda_{n-1} \times \delta\Lambda_{n-1}\}\}$.

(ii) *Assemble* $\mathbb{A}_{\Lambda_n\times\Lambda_n}$ *and* $\mathbf{f}_{\Lambda_n}$ *using these entries and those previously stored.*

3.5. Iterative improvement of $\mathbf{x}_{\Lambda_n}$. For each outer iteration we successively improve the current approximation of the coefficient vector $\mathbf{x}^{\text{init}}_{\Lambda_n}$ by approximately solving the linear algebraic system $\mathbb{A}_{\Lambda_n\times\Lambda_n}\mathbf{x}_{\Lambda_n} = \mathbf{f}_{\Lambda_n}$ using the chosen iterative method. We stop iterations when the following relative stopping criterion is matched:

$$\|\mathbf{r}_{\Lambda_n}\|_2^2 \le \text{tol}^2_{grad}\|\mathbf{f}_{\Lambda_n} - \mathbb{A}_{\Lambda_n\times\Lambda_n}\mathbf{x}^{\text{init}}_{\Lambda_n}\|_2^2, \tag{3.3}$$

where $\mathbf{r}_{\Lambda_n} = \mathbf{f}_{\Lambda_n} - \mathbb{A}_{\Lambda_n\times\Lambda_n}\mathbf{x}_{\Lambda_n}$ is the residual vector corresponding to the active set Λ_n. The parameter tol_{grad} is a given relative stopping tolerance. The idea of the iterative improvement is to solve in each outer iteration the corresponding algebraic system to a moderate accuracy, in order to avoid oversolving the linear system when the discretization error is still large.

Function $[\mathbf{x}_{\Lambda_n}, \mathbf{r}_{\Lambda_n}] = \text{GradMethod}(\mathbb{A}_{\Lambda_n\times\Lambda_n}, \mathbf{f}_{\Lambda_n}, \mathbf{x}^{\text{init}}_{\Lambda_n}, tol_{grad})$. *The body of this routine depends on the solved problem. When we deal with* symmetric positive definite matrices *we replace* GradMethod *by* steepest descent*,* CG*, etc., and for* nonsymmetric matrices *we replace* GradMethod *by* BiCG*,* GMRES*, etc.*

3.6. Removing negligible basis functions. Some basis functions ψ_λ, $\lambda \in \Lambda_n$, can have negligible energy contributions $|x_\lambda|^2$ to the solution, i.e., their coefficients are less significant so we remove them from Λ_n to get $\overline{\Lambda}_n$, in order to keep Λ_{n+1} as small as possible. The number of basis functions, which we may remove, is related to the total energy $\|\mathbf{x}_{\Lambda_n}|_{\delta\Lambda_{n-1}}\|_2^2$ of newly added functions ψ_μ, $\mu \in \delta\Lambda_{n-1}$. We call this process *thresholding*, and it can be realized by sorting absolute values of the entries of $\mathbf{x}_{\Lambda_n}$ in decreasing order, successively summing (starting from the bottom of the sorted vector) squares of the smallest ones, while this sum is less than $\text{tol}^2_{\mathit{off}}\|\mathbf{x}_{\Lambda_n}|_{\delta\Lambda_{n-1}}\|_2^2$. We remove these entries from $\mathbf{x}_{\Lambda_n}$ and their indices from Λ_n *except those which correspond to the scaling functions.* The particular realization is similar to the routine Activate and appears as follows.

Function $[\overline{\mathbf{x}}_{\overline{\Lambda}_n}, \overline{\Lambda}_n] = \text{Threshold}(\mathbf{x}_{\Lambda_n}, \Lambda_n, \delta\Lambda_{n-1}, \text{tol}_{\mathit{off}})$.

(i) *Set* $N := \text{card}\,\Lambda_n$ *and sort absolute values of the entries of* $\mathbf{x}_{\Lambda_n}$ *into decreasing order, obtaining the permutation vector* $\lambda^* := \lambda^*(\mathbf{x}_{\Lambda_n}) = (\lambda^*_1, \lambda^*_2, \ldots, \lambda^*_N)$ *of indices, which gives the decreasing rearrangement* $\mathbf{x}^*_{\Lambda_n} = (|x_{\lambda^*_1}|, |x_{\lambda^*_2}|, \ldots, |x_{\lambda^*_N}|)$ *of* $\mathbf{x}_{\Lambda_n}$*; then compute* $\|\mathbf{x}_{\Lambda_n}|_{\delta\Lambda_{n-1}}\|_2^2$.

(ii) *For* $k = 1, 2, \ldots,$ *compute the sum* $\sum_{l=1}^{k} |x^*_{\lambda_{N-l+1}}|^2$ *in order to find the biggest* k *such that this sum is less than* $\text{tol}^2_{\mathit{off}}\|\mathbf{x}_{\Lambda_n}|_{\delta\Lambda_{n-1}}\|_2^2$*. For the resulting* k*, set* $\overline{\Lambda}_n := \{\lambda^*_l;\ l = 1, \ldots, N-k\} \cup \{$*all indices* λ^*_m*,* $m > N - k$*, of scaling functions*$\}$ *and set* $\overline{\mathbf{x}}_{\overline{\Lambda}_n} := \mathbf{x}_{\Lambda_n}|_{\overline{\Lambda}_n}$.

3.7. Construction of the activable set $\tilde{\Lambda}_n$. We start with the empty set $\tilde{\Lambda}_n$. For all $\lambda \in \overline{\Lambda}_n$, the indices μ of all functions ψ_μ constructed at the level $|\mu| = |\lambda| + 1$ whose supports have nonempty intersections with the support of ψ_λ and $\mu \notin \overline{\Lambda}_n$

are added to $\tilde{\Lambda}_n$. The number of activable functions ψ_μ, $\mu \in \tilde{\Lambda}_n$, can be many times greater than the number of active functions corresponding to $\overline{\Lambda}_n$. Therefore it is convenient to introduce a spatial truncation rule to keep $\tilde{\Lambda}_n$ small. We use contracted support instead of considering intersections with the whole support of ψ_λ, using a contraction factor c. Typical choices are $c = 1/4$ or $c = 1/8$. Another possibility is to select only adjacent wavelets with respect to the positions of their central dyadic nodes, i.e., 2 functions in one dimension, 8 in two dimensions and 26 in three dimensions.

FUNCTION $\tilde{\Lambda}_n = \mathrm{ConstructActivableSet}(\overline{\Lambda}_n)$.

(i) *Set* $\tilde{\Lambda}_n := \emptyset$.

(ii) *For all* $\lambda \in \overline{\Lambda}_n$ *add to* $\tilde{\Lambda}_n$ *indices* μ *of all functions* ψ_μ *constructed at the level* $|\mu| = |\lambda| + 1$ *such that* $\mu \notin \overline{\Lambda}_n$ *and*

Var. 1. $\operatorname{supp}\psi_\mu$ *has a nonempty intersection with the contraction of* $\operatorname{supp}\psi_\lambda$ *by the factor* $c \in (0, 1]$ *with respect to its barycenter;*

Var. 2. ψ_μ *belongs to adjacent wavelets.*

In this function one can include the "a priori" known information on the most significant basis functions for describing the forcing function and the residuum.

3.8. Assembling the residual vector $\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}$. For given $\tilde{\Lambda}_n$ we assemble the residual vector $\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}$ using the activable functions ψ_λ, $\lambda \in \tilde{\Lambda}_n$. The entries $\tilde{r}_\lambda$ are computed by

$$\tilde{r}_\lambda = \tilde{f}_\lambda - d_\lambda^{-1}\langle \mathcal{A}\overline{u}_{\overline{\Lambda}_n}, \psi_\lambda\rangle \quad \forall \lambda \in \tilde{\Lambda}_n, \tag{3.4}$$

where $\tilde{f}_\lambda = d_\lambda^{-1}\langle f, \psi_\lambda\rangle$ and $\overline{u}_{\overline{\Lambda}_n}$ is given by

$$\overline{u}_{\overline{\Lambda}_n} = \sum_{\mu \in \overline{\Lambda}_n} d_\mu^{-1}\overline{x}_\mu \psi_\mu = \overline{\mathbf{x}}^T_{\overline{\Lambda}_n}\mathbb{D}^{-1}_{\overline{\Lambda}_n}\boldsymbol{\psi}_{\overline{\Lambda}_n}. \tag{3.5}$$

We substitute the expansion (3.5) of $\overline{u}_{\overline{\Lambda}_n}$ into (3.4) and we obtain

$$\tilde{r}_\lambda = \tilde{f}_\lambda - \sum_{\forall \mu \in \overline{\Lambda}_n} d_\lambda^{-1} d_\mu^{-1}\overline{x}_\mu \langle \mathcal{A}\psi_\mu, \psi_\lambda\rangle \quad \forall \lambda \in \tilde{\Lambda}_n,$$

or in matrix form, $\tilde{\mathbf{r}}_{\tilde{\Lambda}_n} = \tilde{\mathbf{f}}_{\tilde{\Lambda}_n} - \mathbb{A}_{\tilde{\Lambda}_n \times \overline{\Lambda}_n}\overline{\mathbf{x}}_{\overline{\Lambda}_n}$.

To compute the entries of $\tilde{\mathbf{f}}_{\tilde{\Lambda}_n}$ and $\mathbb{A}_{\tilde{\Lambda}_n \times \overline{\Lambda}_n}$ we use the same quadrature formulas as for computing the entries of $\mathbf{f}_{\Lambda_n}$ and $\mathbb{A}_{\Lambda_n \times \Lambda_n}$. The resulting routine appears as follows.

FUNCTION $[\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}, \tilde{\mathbf{f}}_{\tilde{\Lambda}_n}] = \mathrm{Residual}(\tilde{\Lambda}_n, \overline{\Lambda}_n, \overline{\mathbf{x}}_{\overline{\Lambda}_n})$.

(i) *Compute and store all entries* $\tilde{f}_\lambda$, $a_{\lambda,\mu}$, *and* $\tilde{r}_\lambda$ *for* $\lambda \in \tilde{\Lambda}_n$ *and* $\mu \in \overline{\Lambda}_n$ *using the above-mentioned formulas.*

3.9. Stopping criterion. A natural stopping criterion of the outer loop is based on the achievement of the sufficient (relative) reduction of the total residuum $\|\mathbf{f}_{\mathcal{J}} - \mathbb{A}_{\mathcal{J}\times\Lambda_n}\mathbf{x}_{\Lambda_n}\|_{l_2}$. If we admit that the sets $\tilde{\Lambda}_n$ catch the bulk of the residuum in each iteration, $\|\mathbf{f}_{\mathcal{J}} - \mathbb{A}_{\mathcal{J}\times\Lambda_n}\mathbf{x}_{\Lambda_n}\|_{l_2}$ is well approximated by $\sqrt{\|\mathbf{r}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}\|_2^2}$ and $\|\mathbf{f}_{\mathcal{J}}\|_{l_2}$ by $\sqrt{\|\mathbf{f}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{f}}_{\tilde{\Lambda}_n}\|_2^2}$ for n sufficiently large. Our final stopping criterion is as follows:

if $(\|\mathbf{r}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}\|_2^2 \le \varepsilon^2\,(\|\mathbf{f}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{f}}_{\tilde{\Lambda}_n}\|_2^2))$,

then $\mathbf{x}(\varepsilon) := \overline{\mathbf{x}}_{\overline{\Lambda}_n}$, $\quad \Lambda(\varepsilon) := \overline{\Lambda}_n$ and stop the outer iterations.

3.10. ADWEM solver. In this subsection, we summarize the above-mentioned ingredients of the algorithm by the calling sequence of the functions corresponding to the previous different steps.

ADWEM SOLVER. *Set all above-mentioned parameters of the algorithm and define* $\Lambda_{-1} = \overline{\Lambda}_{-1} = \emptyset$, $\overline{\mathbf{x}}_{\overline{\Lambda}_{-1}} = 0$, $\delta\Lambda_{-1} = \{$*indices of all scaling functions*$\}$.

```
For $n = 0, 1, \ldots$ do
    if $(n = 0)$ then $\Lambda_0 = \delta\Lambda_{-1}$, else $[\Lambda_n, \delta\Lambda_{n-1}] = \mathrm{Activate}(\overline{\Lambda}_{n-1}, \tilde{\Lambda}_{n-1}, \tilde{\mathbf{r}}_{\tilde{\Lambda}_{n-1}}, \mathrm{tol}_{on})$;
    $\mathbf{x}^{\mathrm{init}}_{\Lambda_n} = \mathrm{Extend}(\overline{\mathbf{x}}_{\overline{\Lambda}_{n-1}}, \overline{\Lambda}_{n-1}, \delta\Lambda_{n-1})$;
    $[\mathbb{A}_{\Lambda_n \times \Lambda_n}, \mathbf{f}_{\Lambda_n}] = \mathrm{Assemble}(\overline{\Lambda}_{n-1}, \delta\Lambda_{n-1})$;
    $[\mathbf{x}_{\Lambda_n}, \mathbf{r}_{\Lambda_n}] = \mathrm{GradMethod}(\mathbb{A}_{\Lambda_n \times \Lambda_n}, \mathbf{f}_{\Lambda_n}, \mathbf{x}^{\mathrm{init}}_{\Lambda_n}, \mathrm{tol}_{grad})$, with the stopping criterion:   if $(\|\mathbf{r}_{\Lambda_n}\|_2^2 \leq \mathrm{tol}^2_{grad} \|\mathbf{f}_{\Lambda_n} - \mathbb{A}_{\Lambda_n \times \Lambda_n} \mathbf{x}^{\mathrm{init}}_{\Lambda_n}\|_2^2)$ then STOP;
    $[\overline{\mathbf{x}}_{\overline{\Lambda}_n}, \overline{\Lambda}_n] = \mathrm{Threshold}(\mathbf{x}_{\Lambda_n}, \Lambda_n, \delta\Lambda_{n-1}, \mathrm{tol}_{off})$;
    $\tilde{\Lambda}_n = \mathrm{ConstructActivableSet}(\overline{\Lambda}_n)$;
    $[\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}, \tilde{\mathbf{f}}_{\tilde{\Lambda}_n}] = \mathrm{Residual}(\tilde{\Lambda}_n, \overline{\Lambda}_n, \overline{\mathbf{x}}_{\overline{\Lambda}_n})$;
    if $\|\mathbf{r}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{r}}_{\tilde{\Lambda}_n}\|_2^2 \leq \varepsilon^2 (\|\mathbf{f}_{\Lambda_n}\|_2^2 + \|\tilde{\mathbf{f}}_{\tilde{\Lambda}_n}\|_2^2)$ then $\mathbf{x}(\varepsilon) := \overline{\mathbf{x}}_{\overline{\Lambda}_n}$, $\Lambda(\varepsilon) := \overline{\Lambda}_n$ and STOP;
end.
```

Remark 3.4. In the previous description we use a fixed tolerance tol_{grad}. Nevertheless it is possible to use a changing tolerance. We propose the following rule for choosing tol_{grad} at each iteration: given tol_{grad_0}, define

$$\mathrm{tol}_{grad_n} := \min\left\{ \mathrm{tol}_{grad_0}, \mathcal{C} \frac{\|\mathbf{f}_{\Lambda_{n-1}} - \mathbb{A}_{\Lambda_{n-1} \times \Lambda_{n-1}} \mathbf{x}_{\Lambda_{n-1}}\|_2}{\|\mathbf{f}_{\Lambda_n} - \mathbb{A}_{\Lambda_n \times \Lambda_n} \mathbf{x}^{\mathrm{init}}_{\Lambda_n}\|_2} \right\}, \qquad \mathcal{C} \in (0,1).$$

For this choice, the residuum norm $\|\mathbf{r}_{\Lambda_n}\|_2$ decreases minimally with the rate $\mathcal{C}$ with increasing outer iteration n.

4. Numerical examples.

In this section, we illustrate the behavior of the introduced adaptive algorithm. In Example 1, the computational domain Ω is a curved quadrilateral and the forcing function f induces in the solution two Gaussian peaks of different heights, slopes, and positions (see Figures 4.1 and 4.2). We expect the adaptive algorithm to feel at first the higher peak and, after reaching a good approximation of this peak, the algorithm should reveal the smaller one. For this case, we illustrate sensitivity to the ADWEM parameters tol_{off}, tol_{on}, and the support contraction c. In Example 2, we take the same geometry and split it into two subdomains with different mappings. The particular geometries are depicted in Figures 4.4 and 4.10, respectively. With these examples we want to investigate the effects of mapping and matching on the adaptive method.

The remaining two examples correspond to an L-shaped domain. These cases are interesting because f is everywhere infinitely differentiable and a singularity of u is generated by the shape of the domain (see Figures 4.14 and 4.15). The last example is aimed at showing the difference in terms of rates of convergence between the linear and the nonlinear approximations, respectively, depending on the Sobolev and the Besov regularities. Further numerical results and comparisons can be found in [6].

In our numerical experiments we use piecewise linear and piecewise quadratic biorthogonal B-splines, with $L = 2$, $\tilde{L} = 2$ and $L = 3$, $\tilde{L} = 5$, respectively. We denote by L and $\tilde{L}$ the exactness order of the primal and dual basis, respectively, i.e., polynomials up to the degree $L - 1$ and $\tilde{L} - 1$ are reproduced exactly [26]. If

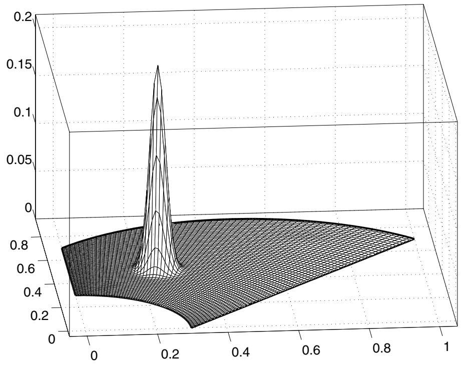

FIG. 4.1. *Example* 1. *Exact solution u_{ex} (linear scale).*

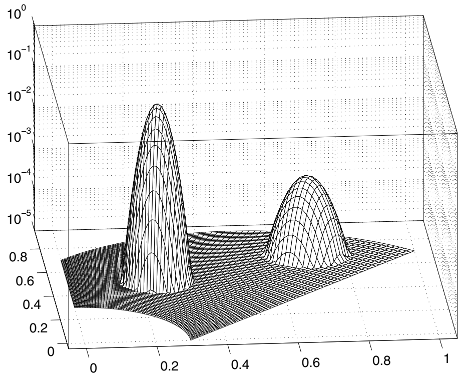

FIG. 4.2. *Example* 1. *Exact solution u_{ex} (log scale).*

not specified, the starting scaling level j_0 is equal to 4. The operator we consider is elliptic and self-adjoint; therefore we replace GradMethod by CG. We set the stopping tolerance $\mathrm{tol}_{grad_0} = 1e-4$ and $\mathcal{C} = 0.9$. Other parameters of the adaptive algorithm are the relative activating and thresholding tolerances $\mathrm{tol}_{on} = 0.99$ and $\mathrm{tol}_{off} = 1e-3$ and the relative target accuracy ε of the adaptive process ($\varepsilon = 1e-3$ for Examples 1 and 2 and $\varepsilon = 5e-2$ for Examples 3 and 4). In computing the entries $a_{\lambda,\mu}$, f_λ we use the relative tolerance $\mathrm{tol}_{int} = 1e-6$. We set $c = 0.125$ in the relation that selects the activable functions (see the routine ConstructActivableSet).

Example 1. Let $\Omega \subset \mathbb{R}^2$ be the curved quadrilateral depicted in Figure 4.4. The corresponding mapping $F : \hat{\Omega} \to \Omega$ is chosen in such a way that it maps the reference unit square $\hat{\Omega}$ to the set

$$\Omega = \left\{(x,y) \in \mathbb{R}^2 \ : x > 0, y > x - 0.3, x^2 + y^2 > 0.09, (x-0.5)^2 + (y-0.2)^2 < 0.5\right\}.$$

Let us consider the problem

$$\begin{cases} -\triangle u = f & \text{in } \Omega, \\ \quad u = 0 & \text{on } \partial\Omega \end{cases} \tag{4.1}$$

with $f = -\triangle u_{ex}$ and

$$u_{ex}(x,y) = x(y-x+0.3)(0.5-(x-0.5)^2-(y-0.2)^2)(x^2+y^2-0.09)\ g,$$

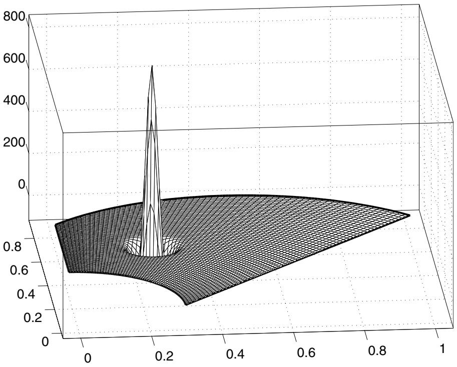

FIG. 4.3. *Example* 1. *Right-hand side f (linear scale).*

TABLE 4.1
Example 1. *ADWEM parameters sensitivity* ($\mathrm{tol}_{off} = 0.001$, $c = 0.125$).

	$\mathrm{tol}_{on} = 0.7$		$\mathrm{tol}_{on} = 0.9$		$\mathrm{tol}_{on} = 0.99$	
Outer iter. n	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$
0	225	6.274e-1	225	6.274e-1	225	6.274e-1
1	226	5.288e-1	230	3.804e-1	241	3.374e-1
2	229	4.015e-1	250	2.170e-1	298	1.722e-1
3	234	3.135e-1	328	1.213e-1	497	8.676e-2
4	245	2.323e-1	567	6.764e-2	1245	4.289e-2
5	268	1.726e-1	1278	3.769e-2	4103	1.953e-2
6	313	1.301e-1	3353	2.024e-2	15161	3.463e-3

TABLE 4.2
Example 1. *ADWEM parameters sensitivity* ($\mathrm{tol}_{off} = 0.01$, $c = 0.125$).

	$\mathrm{tol}_{on} = 0.7$		$\mathrm{tol}_{on} = 0.9$		$\mathrm{tol}_{on} = 0.99$	
Outer iter. n	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$
0	225	6.274e-1	225	6.274e-1	225	6.274e-1
1	226	5.288e-1	230	3.804e-1	241	3.374e-1
2	229	4.015e-1	250	2.170e-1	298	1.722e-1
3	234	3.135e-1	328	1.213e-1	497	8.676e-2
4	245	2.323e-1	567	6.764e-2	1244	4.289e-2
5	268	1.726e-1	1278	3.769e-2	4101	1.954e-2
6	313	1.301e-1	3353	2.024e-2	15152	3.462e-3

where $g = 20e^{-1000r_1^2} + 0.02e^{-300r_2^2}$ with $r_1^2 = (x - 0.25)^2 + (y - 0.5)^2$ and $r_2^2 = (x - 0.7)^2 + (y - 0.7)^2$. The plots of f in the linear scale and u_{ex} in the linear and logarithmic scales are plotted in Figures 4.3, 4.1, and 4.2, respectively.

Sensitivity of ADWEM algorithm with respect to the parameters tol_{off}, tol_{on}, and support contraction c considered are shown in Tables 4.1–4.4. As expected, larger values of tol_{on} imply a faster activation of new basis functions and a parallel reduction of the error. On the other hand, for steady problems the thresholding step is in practice less important, and this can be observed from the smooth dependence of the number of basis functions and of the error on the parameter tol_{off}. The number of active functions $N = \#\Lambda_n$ grows slowly as c increases, since the parameter c mainly affects the number of activable basis functions at each iteration than the number of active basis functions.

TABLE 4.3
Example 1. *ADWEM parameters sensitivity* ($\mathrm{tol}_{\mathit{off}} = 0.1$, $c = 0.125$).

	$\mathrm{tol}_{on} = 0.7$		$\mathrm{tol}_{on} = 0.9$		$\mathrm{tol}_{on} = 0.99$	
Outer iter. n	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$
0	225	6.274e-1	225	6.274e-1	225	6.274e-1
1	226	5.288e-1	230	3.804e-1	237	3.393e-1
2	229	4.015e-1	250	2.170e-1	287	1.736e-1
3	234	3.135e-1	327	1.215e-1	470	8.790e-2
4	245	2.323e-1	559	6.787e-2	1159	4.355e-2
5	268	1.726e-1	1248	3.821e-2	3733	2.006e-2
6	313	1.301e-1	3226	2.074e-2	13599	4.303e-3

TABLE 4.4
Example 1. *ADWEM parameters sensitivity* ($\mathrm{tol}_{\mathit{off}} = 0.001$, $\mathrm{tol}_{on} = 0.99$).

	$c = 0.125$		$c = 0.5$		$c = 1.0$	
Outer iter. n	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$	$\#\Lambda_n$	$\delta_{H^1(\Omega)}$
0	225	6.274e-1	225	6.274e-1	225	6.274e-1
1	241	3.374e-1	241	3.374e-1	241	3.374e-1
2	298	1.722e-1	299	1.721e-1	299	1.721e-1
3	497	8.676e-2	499	8.669e-2	500	8.667e-2
4	1245	4.289e-2	1271	4.272e-2	1282	4.265e-2
5	4103	1.953e-2	4239	1.938e-2	4324	1.930e-2
6	15161	3.463e-3	15756	3.170e-3	16205	2.969e-3

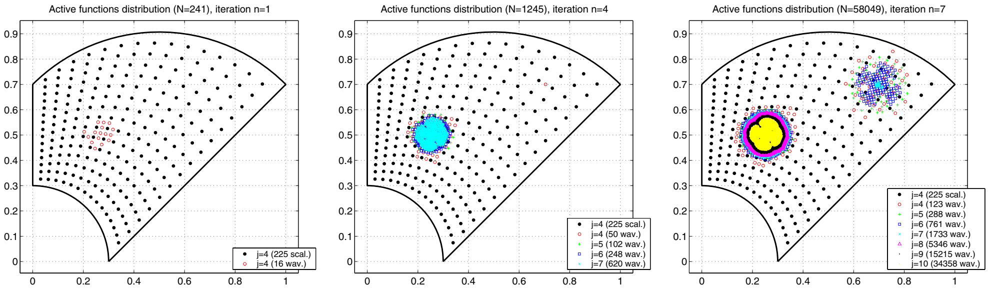

FIG. 4.4. *Example* 1. *Distribution of active functions,* $n = 1, 4, 7$, $N = \#\Lambda_n$.

In Figure 4.4, we successively plot the distributions of active basis functions in outer iterations $n = 1, 4, 7$ for the choice $L = 2$, $\tilde{L} = 2$. We observe that the adaptive algorithm recognizes at first the deficiency corresponding to the higher peak and activates new basis functions near its center. When the adaptive process reaches a good approximation of the higher peak (iteration 4) it reveals another deficiency corresponding to the smaller peak and starts to add new basis functions there as well. The approximate solution u_{Λ_n} and the difference $|u_{\Lambda_n} - u_{ex}|$ are depicted in the logarithmic scale in Figures 4.5 and 4.6, respectively, for $n = 1, 4, 7$. The terms involved in the stopping criterion of the adaptive algorithm are depicted in Figure 4.7.

Finally, we plot the relative errors

$$\delta_{L^2} = \frac{\|u_{\Lambda_n} - u_{ex}\|_{L^2(\Omega)}}{\|u_{ex}\|_{L^2(\Omega)}}, \quad \delta_{H^1} = \frac{\|u_{\Lambda_n} - u_{ex}\|_{H^1(\Omega)}}{\|u_{ex}\|_{H^1(\Omega)}}$$

of the approximate solution u_{Λ_n} in Figure 4.8 with respect to the number of active functions $N = \#\Lambda_n$. The above-mentioned L^2- and H^1-norms are evaluated using

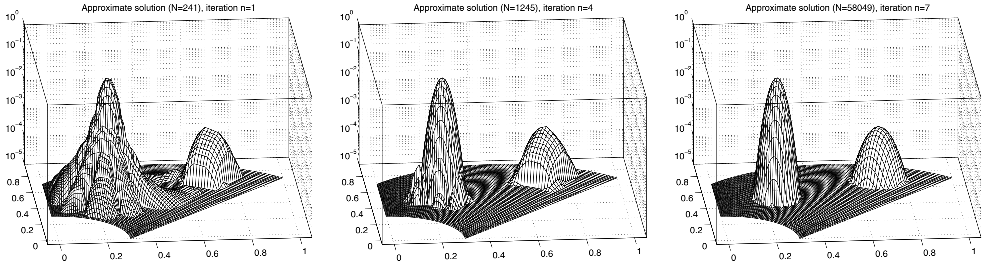

FIG. 4.5. *Example* 1. *Computed solution* u_{Λ_n}*,* $n = 1, 4, 7$*,* $N = \#\Lambda_n$.

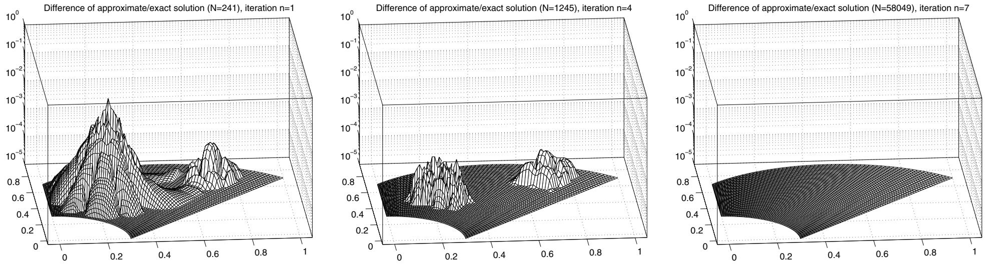

FIG. 4.6. *Example* 1. *Error of the computed solution* $|u_{\Lambda_n} - u_{ex}|$*,* $n = 1, 4, 7$*,* $N = \#\Lambda_n$.

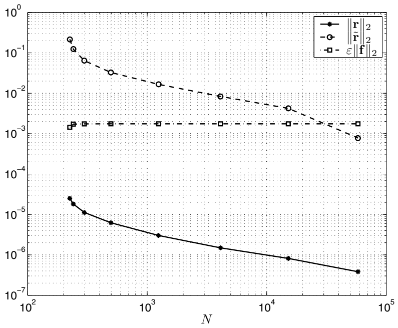

FIG. 4.7. *Example* 1. *Terms in the stopping criterion.*

the Simpson composite formula at the fixed very fine grid of level 13. For comparison purposes we depict the relative errors of the best N-term approximations in $L^2(\Omega)$ and $H^1(\Omega)$ computed using the wavelet coefficients in the final iteration n as the wavelet coefficients of the exact solution. The numerical values of the relative errors δ_{L^2} and δ_{H^1} are listed in Table 4.5. In this table we report also the number of iterations of the CG iterative solver (CG iters.) and the number of active basis functions ($\#\Lambda_n$). We remark that the number of CG iterations is not growing with the number of basis functions and its levels. This is due to the good conditioning of the matrix $\mathbb{A}$ in all the outer iterations. In Figures 4.8 and 4.9 and in Tables 4.5 and 4.6, we report the behavior of the relative errors using linear and quadratic basis functions.

The rates of convergence α of the least squares approximations cN^{α} of the relative errors with respect to $N = \#\Lambda_n$ are given in Table 4.7 for both linear and quadratic basis functions using all iterations reported in Tables 4.5 and 4.6. The

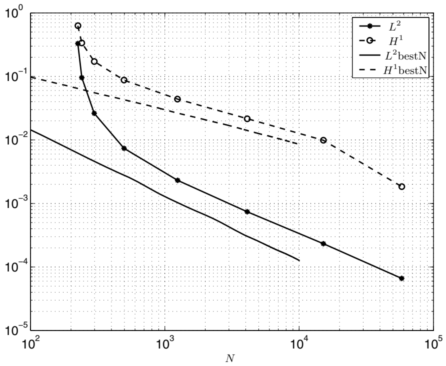

FIG. 4.8. *Example* 1. *Relative errors $L = 2$, $\tilde{L} = 2$.*

TABLE 4.5
Example 1. *Case $L = 2$, $\tilde{L} = 2$. Behavior of the number of inner iterations, the number of active basis functions, and the relative errors in the outer iteration.*

Outer iter. n	CG iters.	$\#\Lambda_n$	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	25	225	3.292e-1	6.274e-1
1	17	241	9.576e-2	3.376e-1
2	15	298	2.626e-2	1.725e-1
3	13	497	7.374e-3	8.725e-2
4	15	1245	2.314e-3	4.388e-2
5	13	4103	7.411e-4	2.161e-2
6	12	15161	2.324e-4	9.872e-3
7	14	58049	6.603e-5	1.846e-3

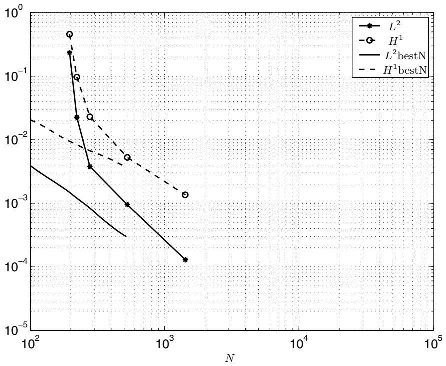

FIG. 4.9. *Example* 1. *Relative errors $L = 3$, $\tilde{L} = 5$.*

theoretical rates of convergence are determined by the polynomial degree of the used basis functions [23].

Example 2. In this case, we split the domain Ω of the previous example into two parts, Ω_1 (lower) and Ω_2 (upper), as shown in Figure 4.10. The mapping consists of two components which map the reference unit square $\hat{\Omega}$ to Ω_1 and Ω_2, respectively. We remark that the splitting intersects one of the Gaussian peaks.

In Figure 4.10, we successively plot the distributions of active basis functions in outer iterations $n = 1, 4, 7$ ($L = 2$, $\tilde{L} = 2$). The behavior of the adaptive algorithm

TABLE 4.6
Example 1. *Case $L = 3$, $\tilde{L} = 5$. Behavior of the number of inner iterations, the number of active basis functions, and the relative errors in the outer iteration.*

Outer iter. n	CG iters.	$\#\Lambda_n$	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	33	196	2.343e-1	4.574e-1
1	15	222	2.251e-2	9.697e-2
2	18	278	3.773e-3	2.310e-2
3	21	527	9.514e-4	5.268e-3
4	31	1427	1.288e-4	1.353e-3

TABLE 4.7
Example 1. *Comparison of the rates of convergence.*

	Theoretical uniform (L^2)	Computed adaptive (L^2)	Theoretical uniform (H^1)	Computed adaptive (H^1)
$L = 2$, $\tilde{L} = 2$	-1	-1.346	$-1/2$	-0.902
$L = 3$, $\tilde{L} = 5$	$-3/2$	-2.059	-1	-1.702

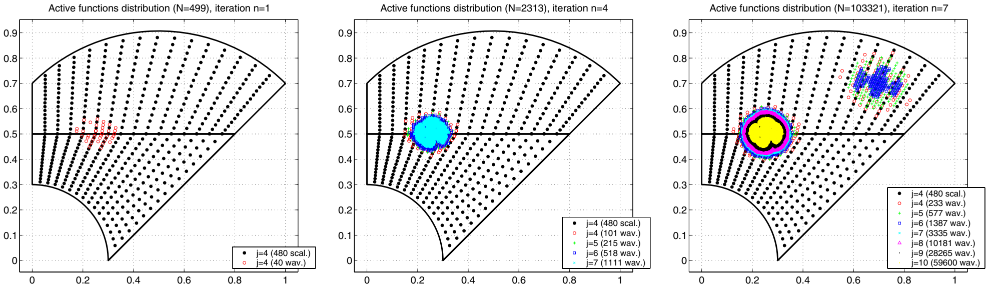

FIG. 4.10. *Example* 2. *Distribution of active functions, $n = 1, 4, 7$, $N = \#\Lambda_n$.*

is similar to the previous case, but we have approximately twice as many candidates for activation. This results from the splitting of Ω into two subdomains Ω_1 and Ω_2 and the use of the same starting scaling level $j_0 = 4$ as in the previous examples in both subdomains. Therefore the algorithm activates more basis functions in the equivalent outer iterations for the same fixed activating and thresholding tolerances than in Example 1 (compare Tables 4.5 and 4.8). The terms involved in the stopping criterion of the adaptive algorithm are depicted in Figure 4.11. Finally, we plot the relative errors $\delta_{L^2(\Omega)}$ and $\delta_{H^1(\Omega)}$ of the approximate solution u_{Λ_n} in Figures 4.12 and 4.13 for $L = 2$, $\tilde{L} = 2$ and $L = 3$, $\tilde{L} = 5$, respectively, together with the relative errors of the best N-term approximations in $L^2(\Omega)$ and $H^1(\Omega)$. The numerical values are listed for $L = 2$, $\tilde{L} = 2$ in Table 4.8. The rates of convergence of the least squares approximations of the errors with respect to N are given in Table 4.9 for both linear and quadratic basis functions.

We remark that the matching induces a faster increase of the number of basis functions caused by the constraint of a minimum construction level and to the larger supports of matched basis functions. These facts do not affect too much the rates of convergence of the ADWEM algorithm because it causes just a shift of the curves (Figures 4.8, 4.9 and 4.12, 4.13).

Example 3 (see [2]). Let $\Omega \subset \mathbb{R}^2$ be the L-shaped domain depicted in Figure 4.16 and let us consider again the problem (4.1) in Ω. Introducing polar coordinates (r, φ),

Table 4.8

Example 2. Case $L = 2$, $\tilde{L} = 2$. Behavior of the number of inner iterations, the number of active basis functions, and the relative errors in the outer iteration.

Outer iter. n	CG iters.	$\#\Lambda_n$	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	53	465	1.834e-1	4.262e-1
1	37	499	9.958e-2	3.146e-1
2	30	607	3.565e-2	1.828e-1
3	37	968	8.829e-3	8.848e-2
4	38	2313	2.341e-3	4.334e-2
5	28	7385	7.523e-4	2.090e-2
6	39	27060	2.229e-4	9.354e-3
7	28	103321	6.594e-5	1.790e-3

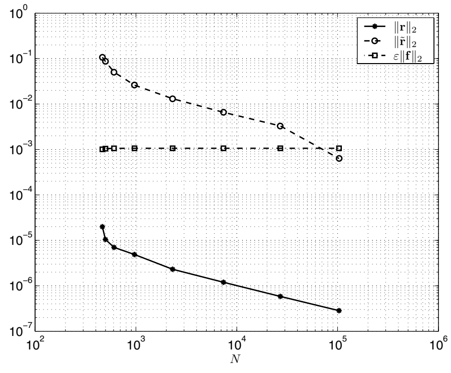

Fig. 4.11. *Example 2. Terms in the stopping criterion.*

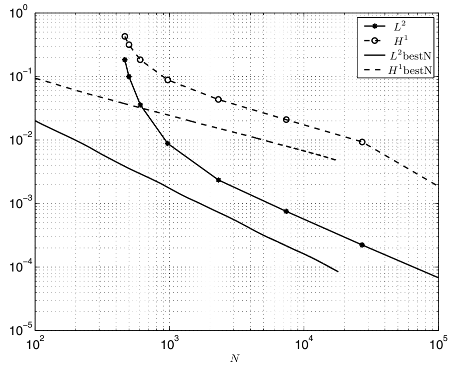

Fig. 4.12. *Example 2. Relative errors $L = 2$, $\tilde{L} = 2$.*

the right-hand side f is defined in such a way that the corresponding exact solution is [2] $u_{ex}(r, \varphi) = h(r)r^{2/3} \sin\left(\frac{2}{3}(\varphi + \frac{\pi}{2})\right)$, where $h \in C^\infty(\Omega)$ is the following truncation function:

$$h(r) = \frac{w(3/4 - r)}{w(r - 1/2) + w(3/4 - r)}, \quad w(r) = \begin{cases} e^{-1/r^2} & \text{if } r > 0, \\ 0 & \text{else.} \end{cases}$$

The plots of the right-hand side f and the exact solution u_{ex} are depicted in Figures 4.14 and 4.15, respectively.

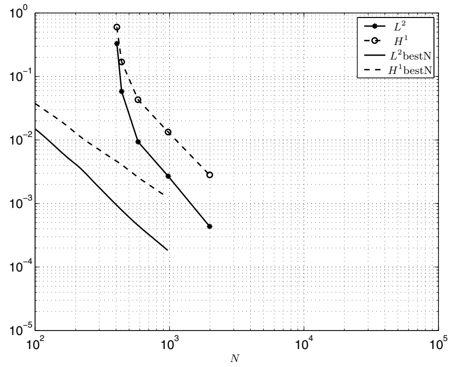

FIG. 4.13. *Example* 2. *Relative errors* $L = 3$, $\tilde{L} = 5$.

TABLE 4.9
Example 2. *Comparison of the rates of convergence.*

	Theoretical uniform (L^2)	Computed adaptive (L^2)	Theoretical uniform (H^1)	Computed adaptive (H^1)
$L = 2$, $\tilde{L} = 2$	-1	-1.374	$-1/2$	-0.911
$L = 3$, $\tilde{L} = 5$	$-3/2$	-2.506	-1	-2.226

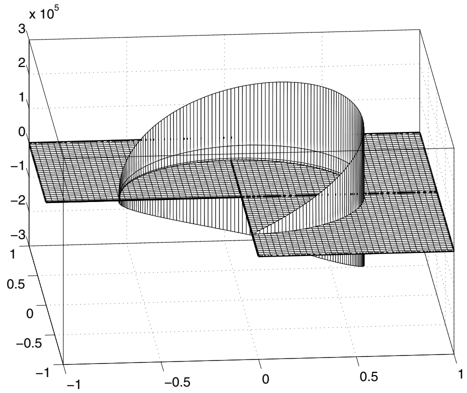

FIG. 4.14. *Example* 3. f *(linear scale).*

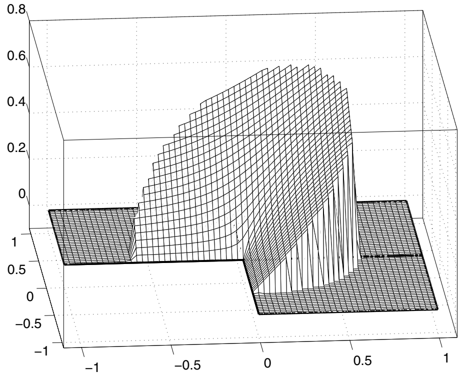

FIG. 4.15. *Example* 3. u_{ex} *(linear scale).*

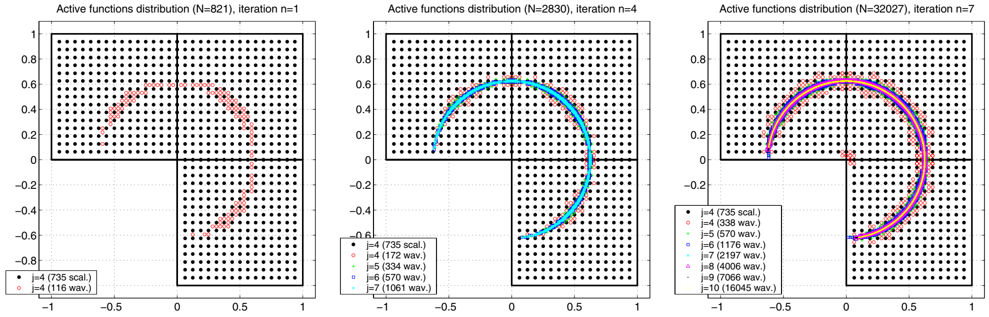

FIG. 4.16. *Example* 3. *Distribution of active functions,* $n = 1, 4, 7$*,* $N = \#\Lambda_n$.

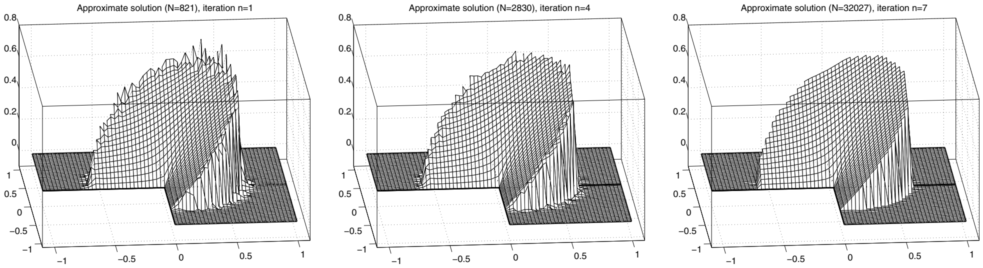

FIG. 4.17. *Example* 3. *Computed solution* u_{Λ_n}*,* $n = 1, 4, 7$*,* $N = \#\Lambda_n$.

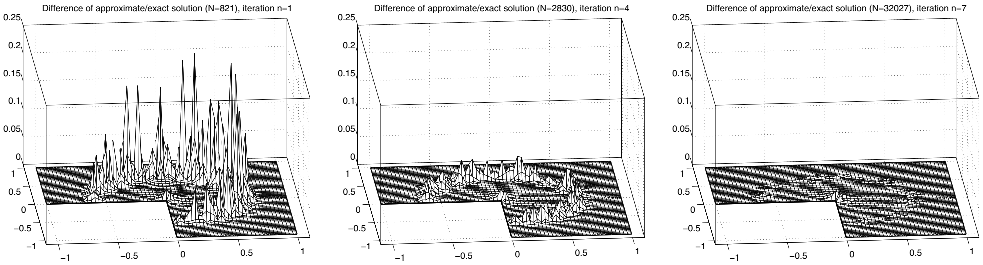

FIG. 4.18. *Example* 3. *Error of the computed solution* $|u_{\Lambda_n} - u_{ex}|$*,* $n = 1, 4, 7$*,* $N = \#\Lambda_n$.

In Figure 4.16, we plot successively the distributions of active basis functions in outer iterations $n = 1, 4, 7$ ($L = 2$, $\tilde{L} = 2$). At first the algorithm recognizes the deficiency corresponding to the right-hand side f and activates new basis functions in the region where f displays strong gradients (see Figure 4.14). In iteration 6, the adaptive process reaches a good approximation of the behavior of f, and it reveals another deficiency around the reentrant corner of the L-shaped domain and starts to add new basis functions also there. The approximate solution u_{Λ_n} and the difference $|u_{\Lambda_n} - u_{ex}|$ are depicted in the linear scale in Figures 4.17 and 4.18, respectively, for $n = 1, 4, 7$. In Figures 4.19, 4.20, and 4.21 we plot the behavior of the relative errors corresponding to linear wavelets with $j_0 = 3$, $j_0 = 4$ and quadratic wavelets with $j_0 = 4$, together with the relative errors of the best N-term approximations in $L^2(\Omega)$ and $H^1(\Omega)$. The numerical values of these relative errors for linear basis functions are listed in Table 4.10 ($j_0 = 4$).

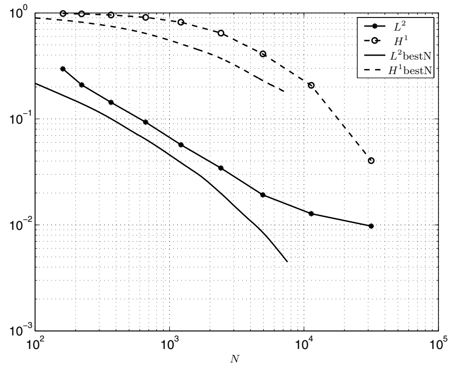

FIG. 4.19. *Example* 3. *Relative errors* $L = 2$, $\tilde{L} = 2$, $j_0 = 3$.

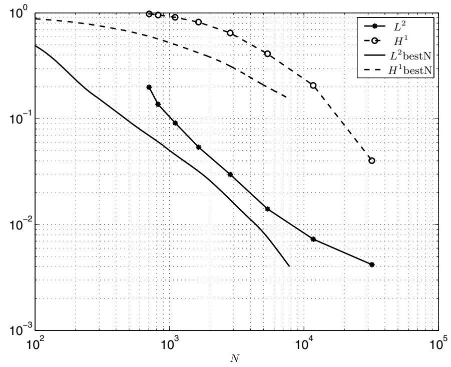

FIG. 4.20. *Example* 3. *Relative errors* $L = 2$, $\tilde{L} = 2$, $j_0 = 4$.

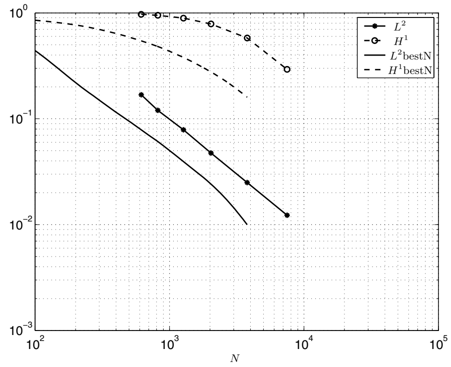

FIG. 4.21. *Example* 3. *Relative errors* $L = 3$, $\tilde{L} = 5$.

The least squares approximations cN^α of the errors in the L^2- and H^1-norms with respect to N are given in Table 4.11. In this case, $f \in C^\infty(\Omega)$ and $u_{ex} \in H^s(\Omega)$ only for $s < 5/3$ due to its singularity around the reentrant corner [25]. Moreover, u_{ex} belongs to the Besov spaces $B^s_{\tau,\tau}(\Omega)$ for all $s > 0$ and $1/\tau = s/2 + 1/2$ [15]. The upper bound in the Sobolev regularity implies an upper bound in the theoretical rate of convergence for uniform refinement, whereas for the unbounded Besov regularity

TABLE 4.10
Example 3. *Case $L = 2$, $\tilde{L} = 2$. Behavior of the number of inner iterations, the number of active basis functions, and the relative errors in the outer iteration.*

Outer iter. n	CG iters.	$\#\Lambda_n$	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	32	705	1.987e-1	9.777e-1
1	25	821	1.369e-1	9.554e-1
2	26	1103	9.119e-2	9.081e-1
3	21	1647	5.390e-2	8.176e-1
4	19	2830	2.976e-2	6.467e-1
5	25	5358	1.405e-2	4.099e-1
6	20	11731	7.284e-3	2.071e-1
7	21	32027	4.191e-3	4.030e-2

TABLE 4.11
Example 3. *$L = 2$, $\tilde{L} = 2$. Comparison of asymptotic constants in the error decay.*

	Theoretical uniform	Computed uniform	Theoretical adaptive	Computed adaptive
L^2	$c_0^u N^{-5/6+\epsilon}$	$3.650N^{-0.408}$	$c_0^a N^{-1}$	$116.734N^{-1.018}$
H^1	$c_1^u N^{-1/3+\epsilon}$	$10.706N^{-0.277}$	$c_1^a N^{-1/2}$	$215.285N^{-0.773}$

TABLE 4.12
Example 4. *$L = 2$, $\tilde{L} = 2$. Comparison of asymptotic constants in the error decay.*

	Theoretical uniform	Computed uniform	Theoretical adaptive	Computed adaptive (1)	Computed adaptive (2)
L^2	$c_0^u N^{-5/6+\epsilon}$	$2.227N^{-0.783}$	$c_0^a N^{-1}$	$17.564N^{-1.131}$	$6.339N^{-0.981}$
H^1	$c_1^u N^{-1/3+\epsilon}$	$2.580N^{-0.456}$	$c_1^a N^{-1/2}$	$6.763N^{-0.636}$	$5.023N^{-0.588}$

we expect a higher theoretical asymptotic rate of convergence for the adaptive case bounded by the order of the polynomial approximation of the basis functions. The theoretically derived rates of convergence for a uniform refinement are $-5/6+\epsilon$ in L^2 and $-1/3+\epsilon$ in H^1 with $\epsilon > 0$.

A similar problem with a L-shape domain with curved boundaries was considered in [6], and results similar to the ones presented above were obtained, meaning that for this kind of problem a nonaffine mapping in the subdomains does not affect the behavior of the method.

Example 4. Let us consider now the same problem as in Example 3, but with the truncation function h defined by

$$h(r) = \frac{w(3/4-r)}{w(r-1/4)+w(3/4-r)}, \quad w(r) = \begin{cases} r^2 & \text{if } r > 0, \\ 0 & \text{else.} \end{cases}$$

In contrast to the solution of Example 3, this solution does not present any sharp structure. Thus both adaptive and uniform refinement methods can reach the asymptotic rates of convergence with relatively few degrees of freedom. For this problem we consider two adaptive solutions with the following refinement parameters: $\text{tol}_{on} = 0.9$, $\text{tol}_{\mathit{off}} = 0.01$ (case 1) and $\text{tol}_{on} = 0.99$, $\text{tol}_{\mathit{off}} = 0.001$ (case 2), respectively, and a uniform refinement with linear basis functions.

In Table 4.12, we report the least squares approximations cN^{α} of the errors in L^2- and H^1-norms. The numerical computations give results very close to the expected theoretical rates of convergence, and in this case the constants of the adaptive case, proportional to the Besov norm of the exact solution, are larger than the constant of the uniform case, proportional to the Sobolev norm of the exact solution.

5. Conclusion.

The origin of wavelets (signal processing, harmonic analysis) is far from the numerical solution of boundary value problems in general bounded domains. Yet, the adaptive wavelets algorithm that we have investigated shows that wavelet expansions can be employed in this task as well, with encouraging results. The proposed algorithm shows good behavior with respect to efficiency and reliability in all considered examples, and also when the discretization of the problem requires mapping and splitting into subdomains. To our knowledge this is the first time that these aspects of an adaptive wavelet Galerkin method have been thoroughly investigated. In our tests, we find that the asymptotic behavior of the error is parallel to that of the best N-term approximation in H^1, although the number of basis functions involved in the solution process depends also on the mapping and the splitting. The comparison of results obtained in Examples 1 and 2 clearly suggests keeping the number of subdomains as small as possible.

The efficiency of a wavelet discretization method with respect to other methods (finite elements, spectral methods) is strongly dependent on the specific data structures and algorithms used to represent the discrete operator equations (matrix against matrix-free, approximation of integrals, etc.). By comparing the number of unknowns needed by some well-assessed adaptive FEMs (mentioned in the appendix) and by our ADWEM method to reach the same H^1-relative error on the proposed examples, we find that our method requires significantly fewer basis functions, denoting a good capability of adaptive wavelets methods to catch and resolve very sharp structures in the solution and singularities. However, by comparing the CPU-time, the FEMs are faster, the assembling of the linear system in our method being still quite slow. The gap in CPU-time between the two methods can be strongly reduced with some suitable choices of the data structures, and many improvements are still possible. We remark that wavelet discretization methods offer other advantages; for example, for systems of equations they allow easy choices of different basis functions for different unknowns without mesh constraint. These advantages can be relevant in some applications.

Appendix. Comparison with adaptive FEMs.

In this appendix, we compare some results achieved by our algorithm ($L = \tilde{L} = 2$, corresponding to linear wavelets) with the results produced using linear finite elements by the well-assessed adaptive finite element code ALBERTA (Adaptive multiLevel finite element toolbox using Bisectioning refinement and Error control by Residual Techniques for scientific Applications) [30, 1]. For adaptive FEMs, as for the adaptive wavelets methods, important results on the convergence and optimality are known [24, 29, 9, 31].

We point out that the comparisons reported here are aimed at drawing indications about viability of the proposed ADWEM approach and not to establish a competition between the software or the implementations of the methods.

A.1. Residual based adaptive FEM algorithms.

In order to discretize problem (2.2), we introduce a regular family of conforming partitions $\{\mathcal{T}_h\}_h$ of the domain $\overline{\Omega}$ into triangles; let us denote by h_T the diameter of the element $T \in \mathcal{T}_h$. The space H in (2.2) is approximated by $V_h = \left\{v_h \in H \cap \mathcal{C}^0(\overline{\Omega}) :\ v_{h|_T} \in \mathbb{P}_1(T) \text{ for all } T \in \mathcal{T}_h\right\}$, where $\mathbb{P}_1(T)$ is the space of polynomials of degree at most 1, on the element $T \in \mathcal{T}_h$. For simplicity we do not consider any approximation of the data. For problem (4.1) an equivalence relation between the true error $\|u - u_h\|_{H^1(\Omega)}$ and a residual-based a posteriori error estimator can be proved [32].

For all triangles $T \in \mathcal{T}_h$ we denote by $\mathcal{E}(T)$ the set of its edges and by $\mathcal{E}_{h,\Omega}$ the set of all edges of the triangulation $\mathcal{T}_h$ inside the domain Ω. For each edge $E \in \mathcal{E}_{h,\Omega}$, $\hat{n}_E$ represents a unit vector which is orthogonal to E and $\omega_E = \bigcup_{\{T'\colon\, E\in\mathcal{E}(T')\}} T'$. Given $E \in \mathcal{E}_{h,\Omega}$ and $\varphi \in \mathrm{L}^2(\,\omega_E\,)$ with $\varphi_{|_{T'}} \in \mathcal{C}^0(\,T'\,)$ for all $T' \in \omega_E$, we denote by $[\![\varphi]\!]_E$ the jump of φ across E in the direction $\hat{n}_E$. Finally, let us define the *equation-residual* $R_{T,h}$ and the *stress-jump* $J_{E,h}$ for this problem,

$$R_{T,h} = -\triangle u_h - f|_T\,, \qquad J_{E,h} = \left[\!\!\left[\frac{\partial u_h}{\partial\, \hat{n}_E} \right]\!\!\right]_E,$$

and the local and global error estimators,

$$\eta^2_{R,T} = C_0^2 \,\|\, R_{T,h}\,\|^2_{L^2(T)} + C_1^2 \sum_{E\in\mathcal{E}(T)\,\cap\,\mathcal{E}_{h,\Omega}} \|\, J_{E,h}\,\|^2_{L^2(E)}, \qquad \eta^2_\Omega = \sum_{T\in\mathcal{T}_h} \eta^2_{R,T}\,.$$

The following equivalence relation between the global error estimator and the true error holds: $c\,\eta^2_\Omega \le \|\, u_h - u\,\|^2_{H^1(\Omega)} \le C\,\eta^2_\Omega$.

We refer to the online manual of the code [1] or to [30] for a detailed description of the adaptive methods implemented in ALBERTA. We are presenting results obtained applying the following strategies: uniform refinement, maximum strategy (MS), equidistribution strategy (ES), and guaranteed error reduction strategy (GERS). We briefly recall these adaptive marking strategies for refinement:

MS: mark T with $\eta_{R,T} > \gamma \max_{T'\in\mathcal{T}_h} \eta_{R,T'}$.
ES: mark T with $\eta_{R,T} > \theta \mathrm{tol}/\sqrt{N_T}$, where N_T is the number of elements in $\mathcal{T}_h$.
GERS: mark $T \in \mathcal{A}$ such that $\sum_{T\in\mathcal{A}} R^2_{T,h} > (1-\theta_*)^2\,\eta^2_\Omega$; see [24, 29] for more details on the choice of the set $\mathcal{A}$.

In all the numerical experiments, coarsening is allowed and we use the following parameters, which correspond, except for polynomial_degree and max_iteration, to the default values [1].

Member	Value	Member	Value
polynomial_degree	1	MS_gamma	0.5
estimator C0	1.0	MS_gamma_c	0.1
estimator C1	1.0	ES_theta	0.9
max_iteration	200	ES_theta_c	0.2
refine_bisection	2	GERS_theta_star	0.6
coarsen_allowed	1	GERS_nu	0.1
coarse_bisections	2	GERS_theta_c	0.1

A.2. Numerical examples. Let us consider Examples 3 and 4 in section 4. We will solve them by the adaptive FEMs implemented in ALBERTA and by uniform refinement. These results should be compared with the corresponding ones from the previous section.

Example 3. The solution is obtained using ALBERTA with tolerance tol $= 4.0$ (chosen to have a final H^1-relative error close to the one in Table 4.10) starting from a coarse grid with $N = 2113$ nodes. In Table A.1 we report the quantities δ_{L^2} and δ_{H^1}; in Table A.2 we report the constant c and the slope α of the least squares approximations cN^α of the relative errors with respect to the number of nodes. From these data we can conclude that the behavior of the relative error on these meshes

TABLE A.1

Example 3. Behavior of the adaptive FEMs, number of nodes, and relative errors in some adaptive iterations.

MS			
Iter.	N	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	2113	8.078e+1	1.302e+1
1	2225	7.277e+1	1.304e+1
2	2337	3.669e+1	6.010e+0
3	2556	2.495e+1	5.710e+0
4	2740	5.934e+1	6.950e+0
5	2698	6.171e+1	7.276e+0
6	3112	2.783e+1	4.193e+0
7	2960	2.289e+1	3.893e+0
8	3259	8.540e+0	3.157e+0
9	3243	9.822e+0	3.299e+0
10	4612	9.427e+0	2.224e+0
11	4381	1.251e+1	2.387e+0
12	4825	3.928e+0	1.649e+0
13	6492	5.164e+0	1.247e+0
14	6894	3.990e+0	1.083e+0
15	8328	2.995e+0	8.471e−1
16	8174	2.841e+0	8.480e−1
17	13092	7.083e−1	4.777e−1
18	19644	6.209e−1	3.338e−1
19	23163	4.434e−1	2.843e−1
20	37020	1.218e−1	1.972e−1
21	53319	3.960e−2	1.527e−1
22	88401	2.313e−2	1.091e−1
23	124269	3.768e−2	9.011e−2
24	234983	1.169e−2	6.331e−2
25	358596	5.962e−3	5.065e−2
26	684219	4.500e−3	3.621e−2

ES			
Iter.	N	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	2113	8.072e+1	1.301e+1
1	8275	2.249e+1	5.370e+0
2	26441	1.684e+1	4.482e+0
3	74534	9.302e+0	1.734e+0
4	98914	1.656e+0	7.136e−1
5	87681	2.430e−1	2.967e−1
6	96674	5.811e−3	1.507e−1
7	224574	2.216e−3	7.641e−2
8	605245	2.851e−3	3.968e−2

GERS			
Iter.	N	$\delta_{L^2(\Omega)}$	$\delta_{H^1(\Omega)}$
0	2113	8.112e+1	1.305e+1
1	2142	5.293e+1	9.553e+0
2	2155	4.849e+1	9.077e+0
3	2195	9.724e+1	1.405e+1
4	2220	7.840e+1	1.329e+1
5	2264	4.305e+1	9.828e+0
6	2368	3.599e+1	6.064e+0
7	2464	6.333e+1	7.928e+0
8	2571	3.220e+1	6.013e+0
9	2699	4.957e+1	7.004e+0
10	3041	3.443e+1	5.664e+0
11	3150	2.478e+1	4.443e+0
12	3560	1.408e+1	3.039e+0
13	4056	2.697e+1	3.304e+0
14	16093	4.259e+0	1.138e+0
15	16744	1.689e+0	9.141e−1
16	17723	1.496e+0	7.300e−1
17	18864	1.874e+0	6.714e−1
18	20824	1.175e+0	5.223e−1
19	22958	1.178e+0	4.425e−1
20	26229	6.394e−1	3.589e−1
21	31278	2.613e−1	2.814e−1
22	36930	2.037e−1	2.306e−1
23	44012	7.848e−2	1.961e−1
24	51949	9.507e−2	1.703e−1
25	69388	6.032e−2	1.350e−1
26	84972	4.188e−2	1.161e−1
27	105211	2.095e−2	1.015e−1
28	130982	2.128e−2	8.935e−2
29	179451	1.380e−2	7.456e−2
30	257101	1.177e−2	6.060e−2
31	337177	7.635e−3	5.243e−2
32	474162	8.456e−3	4.390e−2

is still far from the asymptotic one. Comparing results reported in Table A.1 with the results of Table 4.10 we can clearly see the very effective capability of a wavelets method to catch and resolve sharp structures and singularities. The comparison of the number of basis functions (DOF) between the two methods is not necessarily related to the computational cost. In fact, the task of assembling the linear system is usually more expensive for wavelet codes than for FEMs. Moreover, the state of the art for the finite element codes is widely more assessed than for the few and recent wavelets codes, and many improvements may be possible.

Example 4. The solution is obtained using tolerance tol $= 0.03$, starting from a coarse grid with $N = 2113$ nodes. The final relative errors are $\delta_{L^2(\Omega)} \approx$ 3e−4 and $\delta_{H^1(\Omega)} \approx$ 5e−3 for the three adaptive methods considered. In Table A.3 (compare with Table 4.12), we report the data of the least squares approximations cN^α of

Table A.2
Example 3. Comparison of constants in the error decay.

	Theoretical uniform	Computed uniform
L^2	$c_0^u N^{-5/6+\epsilon}$	$4.482e{+}3N^{-0.628}$
H^1	$c_1^u N^{-1/3+\epsilon}$	$6.291e{+}2N^{-0.503}$

		Theoretical adaptive	Computed adaptive
MS	L^2	$c_0^a N^{-1}$	$3.067e{+}7N^{-1.777}$
	H^1	$c_1^a N^{-1/2}$	$1.221e{+}4N^{-1.008}$
ES	L^2	$c_0^a N^{-1}$	$2.223e{+}9N^{-2.031}$
	H^1	$c_1^a N^{-1/2}$	$1.120e{+}5N^{-1.097}$
GERS	L^2	$c_0^a N^{-1}$	$9.485e{+}7N^{-1.858}$
	H^1	$c_1^a N^{-1/2}$	$3.816e{+}4N^{-1.101}$

Table A.3
Example 4. Comparison of constants in the error decay.

	Theoretical uniform	Computed uniform
L^2	$c_0^u N^{-5/6+\epsilon}$	$4.831N^{-0.752}$
H^1	$c_1^u N^{-1/3+\epsilon}$	$2.972N^{-0.438}$

		Theoretical adaptive	Computed adaptive
MS	L^2	$c_0^a N^{-1}$	$2.864N^{-0.717}$
	H^1	$c_1^a N^{-1/2}$	$6.198N^{-0.551}$
ES	L^2	$c_0^a N^{-1}$	$5.930N^{-0.801}$
	H^1	$c_1^a N^{-1/2}$	$4.201N^{-0.491}$
GERS	L^2	$c_0^a N^{-1}$	$4.267N^{-0.760}$
	H^1	$c_1^a N^{-1/2}$	$6.178N^{-0.552}$

the relative errors in L^2- and H^1-norms obtained by the adaptive algorithms and a uniform refinement. In this case, as for the ADWEM algorithm, we note a very good agreement between the theoretical values and the obtained ones for the H^1-norm [9], while for the L^2-norm we have a small mismatch and the slope is closer to the expected one for a linear approximation technique (uniform refinement) than to the expected one for a nonlinear approximation technique (adaptivity).

Acknowledgments. The authors would like to thank the editor and two anonymous referees for several helpful comments and suggestions which significantly improved the paper. This work was written when the second author was at the Dipartimento di Matematica of Politecnico di Torino.

REFERENCES

[1] *ALBERTA: An Adaptive Hierarchical Finite Element Toolbox*, http://www.alberta-fem.de/.

[2] A. Barinka, T. Barsch, P. Charton, A. Cohen, S. Dahlke, W. Dahmen, and K. Urban, *Adaptive wavelet schemes for elliptic problems—implementation and numerical experiments*, SIAM J. Sci. Comput., 23 (2001), pp. 910–939.

[3] A. Barinka, S. Dahlke, and W. Dahmen, *Adaptive application of operators in standard representation*, Adv. Comput. Math., 24 (2006), pp. 5–34.

[4] S. Berrone and L. Emmel, *A realization of a wavelet Galerkin method on nontrivial domains*, Math. Models Methods Appl. Sci., 12 (2002), pp. 1525–1554.

[5] S. Berrone and L. Emmel, *A Realization of a Wavelet Galerkin Method on Non-trivial Domains*, Preprint 33/2001, Dipartimento di Matematica, Politecnico di Torino, Italy, 2001.

[6] S. Berrone and T. Kozubek, *An Adaptive WEM Algorithm for Solving Elliptic Boundary Value Problems in Fairly General Domains*, Report 38, Dipartimento di Matematica, Politecnico di Torino, Italy, 2004.

[7] S. Berrone and K. Urban, *Adaptive wavelet Galerkin methods on distorted domains: Setup of the algebraic system*, in Curve and Surface Fitting: Saint-Malo 1999, A. Cohen, C. Rabut, and L. L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, pp. 65–74.

[8] S. Bertoluzza, C. Canuto, and K. Urban, *On the adaptive computation of integrals of wavelets*, Appl. Numer. Math. 34 (2000), pp. 13–38.

[9] P. Binev, W. Dahmen, and R. DeVore, *Adaptive finite element methods with convergence rates*, Numer. Math., 97 (2004), pp. 219–268.

[10] C. Canuto, A. Tabacco, and K. Urban, *The wavelet element method.* I. *Construction and analysis*, Appl. Comput. Harmon. Anal., 6 (1999), pp. 1–52.

[11] C. Canuto, A. Tabacco, and K. Urban, *The wavelet element method.* II. *Realization and additional features in* 2*D and* 3*D*, Appl. Comput. Harmon. Anal., 8 (1999), pp. 123–165.

[12] A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods for elliptic operator equations: Convergence rates*, Math. Comp., 70 (2001), pp. 27–75.

[13] A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods.* II. *Beyond the elliptic case*, Found. Comput. Math., 2 (2002), pp. 203–245.

[14] A. Cohen and R. Masson, *Wavelet adaptive method for second order elliptic problems: Boundary conditions and domain decomposition*, Numer. Math., 86 (2000), pp. 193–238.

[15] S. Dahlke, *Besov regularity for elliptic boundary value problems in polygonal domains*, Appl. Math. Lett, 12 (1999), pp. 31–36.

[16] W. Dahmen, *Wavelet and multiscale methods for operator equations*, Acta Numer., 6 (1997), pp. 55–228.

[17] W. Dahmen, *Wavelet methods for PDEs: Some recent developments* J. Comput. Appl. Math., 128 (2001), pp. 133–185.

[18] W. Dahmen, *Multiscale and wavelet methods for operator equations*, in Multiscale Problems and Methods in Numerical Simulations, C. Canuto, ed., Springer-Verlag, Berlin, 2003, pp. 31–96.

[19] W. Dahmen and A. Kunoth, *Multilevel preconditioning*, Numer. Math., 63 (1992), pp. 315–344.

[20] W. Dahmen, A. Kunoth, and K. Urban, *Biorthogonal spline wavelets on the interval-stability and moment conditions*, Appl. Comput. Harmon. Anal., 6 (1999), pp. 132–196.

[21] W. Dahmen, S. Prössdorf, and R. Schneider, *Multiscale methods for pseudo-differential equations on smooth manifolds*, in Proceedings of the International Conference on Wavelets: Theory, Algorithms and Applications, C. K. Chui, L. Montefusco, and L. Puccio, eds., Academic Press, New York, 1994, pp. 385–424.

[22] W. Dahmen and R. Schneider, *Composite wavelet bases for operator equations*, Math. Comp., 68 (1999), pp. 1533–1567.

[23] R. DeVore, *Nonlinear approximation*, Acta Numer., 7 (1998), pp. 51–150.

[24] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124.

[25] P. Grisvard, *Singularities in Boundary Value Problems*, Rech. Math. Appl. 22, Springer-Verlag, Berlin, 1992.

[26] S. Grivet-Talocia and A. Tabacco, *Wavelets on the interval with optimal localization*, Math. Models Methods Appl. Sci., 10 (2000), pp. 441–462.

[27] S. Jaffard, *Wavelets methods for fast resolution of elliptic problems*, SIAM J. Numer. Anal., 29 (1992), pp. 965–986.

[28] A. Kunoth, A.S. Piquemal, and J. Vorloeper, *Multilevel Preconditioners for Discretizations of Elliptic Boundary Value Problems, Experiences with Different Software Packages*, IGPM report 194, RWTH Aachen, Aachen, Germany, 2000.

[29] P. Morin, R. H. Nochetto, and K. G. Siebert, *Data oscillation and convergence of adaptive FEM*, SIAM J. Numer. Anal., 38 (2000), pp. 466–488.

[30] A. Schmidt and K. G. Siebert, *Design of Adaptive Finite Element Software the Finite Element Toolbox ALBERTA*, Lect. Notes Comput. Sci. Eng. 42, Springer-Verlag, Berlin, 2005.

[31] R. Stevenson, *An optimal adaptive finite element method*, SIAM J. Numer. Anal., 42 (2005), pp. 2188–2217.

[32] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, John Wiley & Sons, Chichester, UK, New York, 1996.