SIAM J. SCI. COMPUT.
Vol. 43, No. 3, pp. A2211–A2241

QUASI-OPTIMAL MESH SEQUENCE CONSTRUCTION THROUGH SMOOTHED ADAPTIVE FINITE ELEMENT METHODS*

ORNELA MULITA†, STEFANO GIANI‡, AND LUCA HELTAI§

Abstract. We propose a new algorithm for adaptive finite element methods (AFEMs) based on smoothing iterations (S-AFEM), for linear, second-order, elliptic partial differential equations (PDEs). The algorithm is inspired by the ascending phase of the V-cycle multigrid method: we replace accurate algebraic solutions in intermediate cycles of the classical AFEM with the application of a prolongation step, followed by the application of a smoother. Even though these intermediate solutions are far from the exact algebraic solutions, their a posteriori error estimation produces a refinement pattern that is substantially equivalent to the one that would be generated by classical AFEM, at a considerable fraction of the computational cost. We provide a qualitative analysis of how the error propagates throughout the algorithm, and we present a series of numerical experiments that highlight the efficiency and the computational speedup of S-AFEM.

Key words. adaptive mesh refinement, finite element method, second-order elliptic PDEs, a posteriori error analysis, inexact algebraic solution, iterative solvers, smoothing iterations, grid construction

AMS subject classifications. 65N15, 65N22, 65N30, 65N50, 65N55

DOI. 10.1137/19M1262097

1. Introduction.

The efficient numerical simulation of complex real-world phenomena requires the use of computationally affordable discrete models. The adaptive finite element method (AFEM) is one such scheme for the numerical solution of partial differential equations (PDEs) in computational sciences and engineering. In finite element simulations (FEM), the domain of the PDE is discretized into a large set of small and simple domains (the cells or elements) depending on a size parameter $h > 0$, and the PDE is transformed into an algebraic system of equations. Rigorous analysis of the numerical method allows one to estimate the discretization error both a priori (giving global bounds on the total discretization error that depend on a global size parameter h), and a posteriori (providing a local distribution of the error on the discretized mesh in terms of known quantities). Classical AFEM consists of successive loops of the steps $\mathcal{S}olve \longrightarrow \mathcal{E}stimate \longrightarrow \mathcal{M}ark \longrightarrow \mathcal{R}efine$ to decrease the total discretization error, by repeating the FEM solution process ($\mathcal{S}olve$) on a mesh that has been refined ($\mathcal{R}efine$) where the a posteriori analysis ($\mathcal{E}stimate$) has shown that the error is larger ($\mathcal{M}ark$).

Intermediate solution steps are *instrumental* for the construction of the finally

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 17, 2019; accepted for publication (in revised form) March 30, 2021; published electronically June 17, 2021.

https://doi.org/10.1137/19M1262097

Funding: The work of the third author was partially supported by the Italian Ministry of Instruction, University and Research (MIUR), under the 2017 PRIN project NA-FROMPDEs MIUR PE1, "Numerical Analysis for Full and Reduced Order Methods for the Efficient and Accurate Solution of Complex Systems Governed by Partial Differential Equations."

†Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany (ornela.mulita@math.hu-berlin.de).

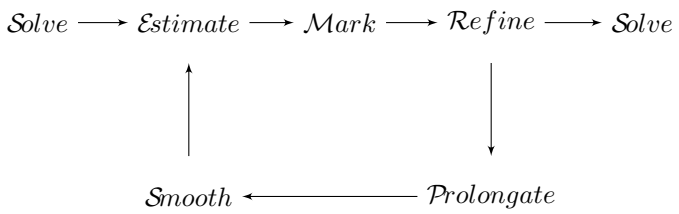
‡School of Engineering and Computer Sciences, Durham University, Stockton Road, Durham, DH1 3LE, UK (stefano.giani@durham.ac.uk).

§Scuola Internazionale Superiore di Studi Avanzati, Via Bonomea 265, 34136 Trieste, Italy (luca.heltai@sissa.it).

adapted grid and play no role in the final solution, which is the only one that is retained for analysis and processing.

In this work we present a simple yet effective algorithm to reduce the overall computational cost of the AFEM algorithm, by providing a fast procedure for the construction of quasi-optimal mesh sequences that do not require the exact solution of the algebraic problem in the intermediate loops of AFEM.

We refer to this new algorithm as the *smoothed adaptive finite element method* (S-AFEM): the $\mathcal{S}olve$ step of AFEM in all intermediate loops is replaced by the application of a prolongation step ($\mathcal{P}rolongate$), followed by the application of a smoother ($\mathcal{S}mooth$):



S-AFEM takes its inspiration from the ascending phase of the V-cycle multigrid method where a sequence of prolongation and smoothing steps is applied to what is considered an algebraically exact solution at the coarsest level. In the multigrid literature, this procedure is used to transfer the low frequency information contained in the coarse solution to a finer—nested—grid, where some steps of a smoothing iteration are applied in order to improve the accuracy of the solution in the high frequency range (see, for example, the classical references [20, 21, 35, 13, 12]). The iteration of this procedure is very effective in providing accurate algebraic solutions in $O(N)$ time, where N is the dimension of the final algebraic system. Even a small number of smoothing iterations is sufficient to eliminate the high frequency error, while the prolongation from coarser grids guarantees the convergence in the low frequency regime, resulting in an overall accurate solution also when local refinement is present (see, for example, [22]).

The classical AFEM algorithm generates nested grids and subspaces,[1] but the construction of the next (unknown) grid in the sequence still requires an exact algebraic solution on the current grid to trigger the $\mathcal{E}stimate$-$\mathcal{M}ark$-$\mathcal{R}efine$ steps. In this paper we show, however, that in many practical situations it is not necessary to use a fully resolved solution in the intermediate steps in order to obtain a good refinement pattern: numerical evidence shows that it is sufficient for the high frequencies of the error to be dumped in order to identify the next grid in the sequence through the $\mathcal{E}stimate$-$\mathcal{M}ark$ steps. In this context, the construction of a grid with excellent approximation properties may require as little as a single ascending step of a V-cycle multigrid method.

It is still not clear how to explain rigorously why the sequence of meshes constructed with S-AFEM is close to the one obtained by classical AFEM. In this work we examine the numerical behavior of S-AFEM in a family of linear second-order elliptic problems and provide a qualitative analysis, based on classical results from the AFEM and multigrid literature, to point the reader towards currently open questions, and towards possible paths to complete the analysis.

[1] During AFEM, the grids remain nested if no de-refinement occurs. In the following we will work under this assumption.

In particular, we consider conforming discretizations of a Poisson problem, and of a class of drift-diffusion problems in 2D and 3D, and we show that

- ⋆ the a posteriori error estimator applied to the outcome of a single ascending phase of the V-cycle multigrid method triggers a $\mathcal{M}ark$ step where the refinement pattern is substantially equivalent to the one that would be generated by a classical $\mathcal{S}olve$ step, at a considerable fraction of the computational cost;
- ⋆ even if the final grid is not exactly identical to the one that would be obtained with the classical AFEM, the accuracy of the final solution is comparable in most cases;
- ⋆ the S-AFEM algorithm is robust with respect to different smoothers, and with respect to different discretization degrees.

This article is organized as follows: we start by describing the general S-AFEM algorithm in section 2, where the main algorithm is exposed. The connection with the multilevel framework and with classical a posteriori error analysis is made in section 3, where a qualitative analysis of S-AFEM is presented. Section 4 is dedicated to the numerical validation of the algorithm and presents a detailed campaign of simulations that shows when the S-AFEM algorithm can be used successfully. Finally, in section 5 we provide some conclusions and perspectives for future works.

2. The S-AFEM algorithm. We consider a class of linear elliptic, second-order, boundary value problems (BVPs), whose variational formulation reads as follows: seek $u \in V$ s.t. $\mathcal{A}u = f$ in V under suitable boundary conditions, where $(V, \|\bullet\|)$ is a normed Hilbert space defined over a Lipschitz bounded domain Ω, the linear operator $\mathcal{A} : V \to V^\star$ is a second-order elliptic operator, and $f \in V^\star$ is a given datum. The finite element method provides numerical solutions to the above problem in a finite dimensional solution space $V_h \subset V$, typically made up by continuous and piecewise polynomial functions, and transforms the continuous problem above into a discrete model of type $\mathcal{A}_h u_h = f_h$ in V_h under suitable boundary conditions, where, e.g., $\mathcal{A}_h = \mathcal{A}\,|_{V_h}$. The overall procedure leads to the solution of a (potentially very large) linear algebraic system of equations of type $A\mathbf{u} = \mathbf{f}$ in $\mathbb{R}^N$, where $N = \dim(V_h)$.

Given an initial (coarse) triangulation $\mathcal{T}_1$, we consider a (a priori unknown) nested sequence of shape regular triangulations $\mathcal{T}_k$ for $k = 1, \dots, \bar{k}$, which induces a nested sequence of finite element spaces

$$V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}} \equiv V_h, \tag{2.1}$$

on which we define standard prolongation operators, considering the canonical embedding $i_k^{k+1} : V_k \hookrightarrow V_{k+1}$ that embeds functions $u_k \in V_k$ in the space V_{k+1}. We denote by $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$ the corresponding discrete matrices, and we let $N_k := \dim(V_k)$ for $k = 1, \dots, \bar{k}$.

The sequence of grids and the solution on the finest grid are computed with the S-AFEM algorithm, defined in Algorithm 2.1.

The motivation behind the strategy at the base of S-AFEM is the numerical observation that classical residual-based a posteriori error estimators [34] used in the $\mathcal{E}stimate$ step are mostly insensitive to low frequencies in the solution, as shown in Figure 2.1 for a benchmark example. Their application to very inaccurate approximate solutions in the intermediate loops—only capturing high frequency oscillations through a smoother—produces an equally good grid refinement pattern at each step of the S-AFEM algorithm, with an accuracy on the final approximation step that is

Algorithm 2.1 S-AFEM algorithm.

Starting from an initial coarse mesh $\mathcal{T}_1$, solve $A_1\mathbf{u}_1 = \mathbf{f}_1$ in $\mathbb{R}^{N_1}$ to high accuracy and generate $\mathbf{u}_1$. Then, do steps $1.-4.$ for $k = 2, \dots, \bar{k}-1$ or until the criterion is met.

1. *Smooth*: Compute ℓ smoothing iterations on the discrete system $A_k\mathbf{u}_k = \mathbf{f}_k$, with initial guess $\mathbf{u}_k^{(0)} := I_{k-1}^k \mathbf{u}_{k-1}^{(\ell)}$, which produce $\mathbf{u}_k^{(\ell)} \in \mathbb{R}^{N_k}$ (take $\mathbf{u}_1^{(\ell)} = \mathbf{u}_1$).
2. *Estimate*: Compute estimators $\eta_T(u_k^\ell)$ for all elements $T \in \mathcal{T}_k$.
3. *Mark*: Choose a set of cells to refine $\mathcal{M}_k \subset \mathcal{T}_k$ based on $\eta_T(u_k^\ell)$.
4. *Refine*: Generate new mesh $\mathcal{T}_{k+1}$ by refinement of the cells in $\mathcal{M}_k$.

Step $k = \bar{k}$: Solve the discrete system $A_{\bar{k}}\mathbf{u}_{\bar{k}} = \mathbf{f}_{\bar{k}}$ to the desired algebraic accuracy.
Output: sequence of meshes $\mathcal{T}_k$, smoothed approximations $\mathbf{u}_k^\ell$, estimators $\eta(u_k^\ell)$, and final adapted-approximation $\mathbf{u}_{\bar{k}}^\ell$ such that $\|\mathbf{e}_{\bar{k}}\| \le tol$.

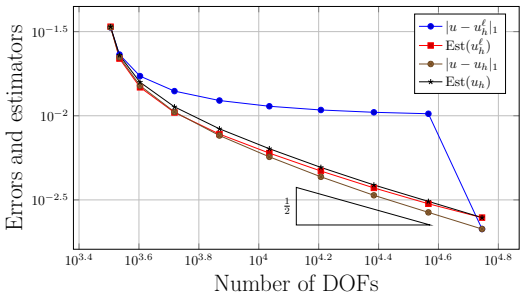


FIG. 2.1. *The values of the total error energy norm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and Est(u_h)) and S-AFEM with $\ell = 2$ smoothing iterations ($|u-u_h^\ell|_1$ and Est(u_h^ℓ)) for a classical Poisson problem on an L-shaped domain in 2D. The first and last loops are solved exactly by both methods.*

comparable to that obtained with the classical AFEM algorithm, at a fraction of the computational cost.

In the S-AFEM algorithm, we capture the smoothest (i.e., less oscillatory) part of the discrete approximation in the first loop ($k = 1$) by solving the discrete system exactly on the coarsest level. As the mesh is locally refined from one level to the other, we increase the higher portion of the spectrum of the matrix A_k. Thanks to the structure of the refinement in typical finite element methods, mostly high frequencies are added to the system, while low frequencies are substantially left unaltered. Even though there are cases where low frequency components are added to the solution on the whole mesh when performing local refinement (a phenomenon usually referred to as the pollution effect), several local refinements are required for the effect to show up, suggesting that this phenomenon may be neglected in the context of S-AFEM, where we use the term "low frequencies" to indicate loosely functions that can be captured by coarse meshes and the term "high frequency" to indicate functions that can be captured by fine meshes.

Even though the distance between the algebraic approximation u_h^ℓ (coming from the S-AFEM algorithm at step k) and the exact solution u drifts away during the various steps of the algorithm (curve $|u-u_h^\ell|_1$ in Figure 2.1), the error estimator evaluated on u_h^ℓ remains substantially attached to the error estimator evaluated on an algebraically exact solution u_h computed on the same mesh (comparison between curves Est(u_h^ℓ) and Est(u_h) in Figure 2.1).

The first and last loops of the S-AFEM algorithm coincide with those of the classical AFEM algorithm. In intermediate S-AFEM loops, however, the solution u_h^ℓ is far from the exact algebraic solution u_h. These intermediate solutions serve solely the construction of the final grid and find no other use in the final computations; therefore,

their inexactness is irrelevant, provided that the final adapted grid provides a good approximation. Their role is *instrumental* in triggering the $\mathcal{E}stimate-\mathcal{M}ark-\mathcal{R}efine$ steps.

In our setting, intermediate steps are only computed through a fixed number of smoothing iterations, and have nonnegligible algebraic errors. This is in contrast with the common practical assumption made in AFEM, where it is assumed that the $\mathcal{S}olve$ step produces the *exact* solution of the algebraic system. Recent developments dedicated a great deal of effort to accounting for inexactness of the algebraic approximations and introducing stopping criteria based on the interplay between discretization and algebraic computation in adaptive FEM. Among others, we mention the seminal contributions [10, 23, 4, 5, 31, 26, 30, 29, 17, 28, 25, 16].

Nevertheless, most of this literature focuses on ways to estimate the algebraic error, without really exploiting the other side of the coin: inexact approximate solutions, with large algebraic error, offer large computational savings when used in the correct way. S-AFEM provides a good strategy to exploit this fact.

3. Qualitative analysis of S-AFEM. To fix the ideas, in this section we apply S-AFEM to a model Poisson problem with Richardson iteration as a smoother and discuss the interplay between the algebraic solution in intermediate steps and the classical a posteriori error estimator theory.

A larger selection of problem types and smoother algorithms is tested in section 4, where we compare the application of a fixed number of Richardson iterations or of the conjugate gradient (CG) method for symmetric systems, and of the generalized minimal residual (GMRES) method for nonsymmetric systems coming from the discretization of drift-diffusion problems.

Let $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ be a bounded, polygonal domain (an open and connected set with polygonal boundary) with Lebesgue and Sobolev spaces $L^2(\Omega)$ and $H_0^1(\Omega)$. We look for the solution $u \in H_0^1(\Omega)$ such that

$$-\Delta u = f \text{ in } \Omega \text{ and } u = 0 \text{ on } \partial\Omega, \tag{3.1}$$

where $f \in L^2(\Omega)$ is a given source term. We use standard notation for norms and scalar products in Sobolev spaces (cf. [1]): for $u \in H_0^1(\Omega)$ we write $|u|_1 := (\int_\Omega |\nabla u|^2)^{1/2}$ and denote by $(\cdot,\cdot)$ the $L^2(\Omega)$- scalar product with corresponding norm $\|\cdot\|$. The weak form of (3.1) is to find $u \in H_0^1(\Omega)$ s.t.

$$(\nabla u, \nabla v) = (f, v) \ \forall v \in H_0^1(\Omega). \tag{3.2}$$

We consider a shape regular family of triangulations $\{\mathcal{T}_h\}_h$ of Ω in the sense of Ciarlet [15], depending on a parameter $h > 0$ with shape regularity parameter $C_{\mathcal{T}_h}$ consisting of cells T that are convex quadrilaterals in two dimensions and convex hexahedrons in three dimensions.

The set of all edges/faces E of the cells is denoted by $\mathcal{E}_h$, and similarly, $\mathcal{E}_{h,int} := \mathcal{E}_h \setminus \partial\Omega$ is the set of internal edges/faces. We use the Courant finite element space $V_h := \{\varphi \in C^0(\Omega) \text{ s.t. } \varphi|_T \in P^k(T) \ \forall T \in \mathcal{T}_h, \varphi = 0 \text{ on } \partial\Omega\} \subset H_0^1(\Omega)$. The Galerkin solution $u_h \in V_h$ is obtained by solving the discrete system

$$(\nabla u_h, \nabla v_h) = (f, v_h) \ \forall v_h \in V_h. \tag{3.3}$$

In exact arithmetic, the discretization error $e_h := u - u_h$ satisfies the standard orthogonality condition

$$(\nabla(u - u_h), \nabla v_h) = 0 \ \forall v_h \in V_h. \tag{3.4}$$

Let $N = \dim(V_h)$; the discrete system (3.3) leads to a linear algebraic system of type

$$A\mathbf{u} = \mathbf{f} \text{ in } \mathbb{R}^N, \tag{3.5}$$

where A denotes the symmetric positive definite (SPD) stiffness matrix with entries $a_{ij} := (\nabla\varphi_j, \nabla\varphi_i)\ \forall\, i,j = 1,\dots,N$, $\mathbf{u} = [u_1,\dots,u_N]^T$ denotes the coefficients vector in $\mathbb{R}^N$ of the discrete approximation $u_h = \sum_{j=1}^N u_j\varphi_j \in V_h$, and $\mathbf{f} = [f_1,\dots,f_N]^T$ is the vector with entries $f_j = (f,\varphi_j)\ \forall j = 1,..,N$.

In particular, we assume that an initial (coarse) triangulation $\mathcal{T}_1$ is given, and we consider a (a priori unknown) nested sequence of shape regular triangulations $\mathcal{T}_k$ for $k = 1,\dots,\bar{k}$, which induces a nested sequence of finite element spaces $V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}}$. We let $N_k := \dim(V_k)$ for $k = 1,\dots,\bar{k}$. By construction, the inequalities $N_1 < N_2 < \cdots < N_{\bar{k}}$ hold true. The associated discrete systems for each level $k = 1,2,\dots,\bar{k}$ read

$$(\nabla u_k, \nabla v_k) = (f, v_k)\ \ \forall\, v_k \in V_k, \tag{3.6}$$

and they generate linear systems of type

$$A_k\mathbf{u}_k = \mathbf{f}_k \tag{3.7}$$

of respective dimensions N_k.

The sequence of meshes in the classical AFEM algorithm is constructed through the *Solve–Estimate–Mark–Refine* steps, where the *Solve* step should compute an algebraically exact solution of (3.7). In the *Estimate* step, standard residual-based a posteriori error estimators are the most widely used. They were first introduced in the context of FEM by Babuška and Rheinboldt in [8], and they have been thereafter widely studied in the literature (for a review, see [34, 2]).

Their derivation is based on the residual functional associated to the Galerkin solution, which is defined as $\mathcal{R}\{u_h\} : H_0^1(\Omega) \longrightarrow \mathbb{R}$, $\mathcal{R}\{u_h\} := (f,\bullet) - a(u_h,\bullet)$ with corresponding dual norm

$$\|\mathcal{R}\{u_h\}\|_\star := \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{\mathcal{R}\{u_h\}(v)}{|v|_1} = \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{(f,v) - a(u_h,v)}{|v|_1}. \tag{3.8}$$

The identity $|e_h|_1 = \|\mathcal{R}\{u_h\}\|_\star$ leads to reliable and efficient residual-based a posteriori bounds for the discretization error via estimation of the residual function, i.e.,

$$\|e_h\| \le C_{\text{rel}}\eta(u_h) + h.o.t._{\text{rel}} \tag{3.9}$$

and

$$\eta(u_h) \le C_{\text{eff}}\|e_h\| + h.o.t._{\text{eff}}, \tag{3.10}$$

where the multiplicative constants C_{rel} and C_{eff} are independent of the mesh size and h.o.t. denotes oscillations of the right-hand side f, which are generally negligible w.r.t. $\|e_h\|$.

We use standard residual-based a posteriori error estimators which are locally defined through the jump of the gradient of the discrete approximation across the edges/faces E of the cells; i.e., for a given function $v_h \in V_h$, define for $E \in \mathcal{E}_h$ and

$T \in \mathcal{T}_h$

$$
\begin{aligned}
J_E(v_h) &:= h_E^{1/2} \left\| \left[\frac{\partial v_h}{\partial n_E} \right] \right\|_E, \; J_T(v_h) := \sum_{E \in \partial T} J_E(v_h), \\
J(v_h) &:= \left(\sum_{E \in \mathcal{E}_h} J_E(v_h)^2 \right)^{1/2} = \left(\frac{1}{2} \sum_{T \in \mathcal{T}_h} J_T(v_h)^2 \right)^{1/2},
\end{aligned} \tag{3.11}
$$

where $[\bullet]$ indicates the jump of a piecewise continuous function across the edge/face E in normal direction n_E. A classical upper bound on the discretization error using J is given by the estimate

$$
|u - u_h|_1 \leq C^\star (\mathrm{osc}^2 + J^2(u_h))^{1/2}, \tag{3.12}
$$

where the constant $C^\star > 0$ depends on the shape of the triangulation, on Ω, and on Γ, but it is independent of f and of the mesh-sizes h_T, and osc is an oscillatory term (see [14] for the exact definition of osc and for a proof of (3.12)).

In S-AFEM, we do not solve the linear systems (3.7) for $k = 2, \ldots, \bar{k} - 1$, but only apply a smoother (typically, ℓ steps of a smoothing iteration) by taking as an initial guess the prolongation of the approximation from the previous level, obtaining an algebraically inexact approximation u_h^ℓ of u_h in all intermediate steps. In this case, the total error in intermediate steps can be written as the sum of two contributions

$$
\underbrace{u - u_h^\ell}_{\text{total error}} = \underbrace{(u - u_h)}_{\text{discretization error}} + \underbrace{(u_h - u_h^\ell)}_{\text{algebraic error}}. \tag{3.13}
$$

A vast literature is dedicated to the extension of the standard residual-based a posteriori error estimator theory to incorporate in some way the algebraic error deriving from an inexact *Solve* step. We refer the reader to the seminal and investigative paper by Papež and Strakoš [30] and the references therein for various approaches.

In particular, in [30] the authors give a detailed proof of a (worst-case scenario) residual-based upper bound on the energy norm of the total error

$$
|u - v_h|_1^2 \leq 2C^2 (J^2(v_h) + \mathrm{osc}^2) + 2C_{\mathrm{intp}}^2 |u_h - v_h|_1^2 \tag{3.14}
$$

for arbitrary $v_h \in V_h$. In order to provide sharper bounds, one should exploit the fact that v_h is actually not arbitrary, but it originates from the *Smooth* step of S-AFEM; i.e., it is the result of a smoothing iteration.

For simplicity of exposition, in this section we use a fixed number of Richardson iterations as a smoother, but other choices are possible (see, for example, the reviews in [12, 19, 35]).

Given $\omega_k \in \mathbb{R}$ a fixed parameter and $\mathbf{u}_k^{(0)} \in \mathbb{R}^{N_k}$ an initial guess, a Richardson smoothing iteration for (3.7) takes the form

$$
\mathbf{u}_k^{(i+1)} = \mathbf{u}_k^{(i)} + \omega_k (\mathbf{f}_k - A_k \mathbf{u}_k^{(i)}) \text{ for } i = 0, 1, \ldots, \ell. \tag{3.15}
$$

After $i+1$ iterations, the error $\mathbf{e}_k^{(i+1)} := \mathbf{u}_k - \mathbf{u}_k^{(i+1)}$ satisfies the error propagation formula $\mathbf{e}_k^{(i+1)} = M_k \mathbf{e}_k^{(i)} = \cdots = M_k^{i+1} \mathbf{e}_k^{(0)}$, where $M_k := Id_{N_k} - \omega_k A_k$ is the Richardson iteration matrix. When using the Richardson method as an iterative solver, convergence takes place for $0 < \omega_k < 2/\rho(A_k)$, where $\rho(A_k)$ is the spectral

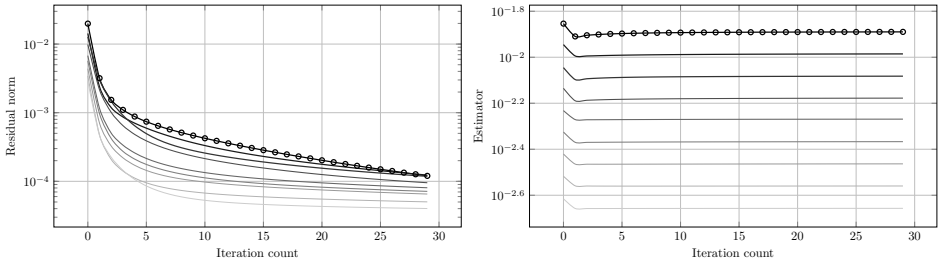


FIG. 3.1. *Algebraic residual ℓ^2-norm (left) and error estimator (right) for intermediate cycles of the classical AFEM algorithm when using Richardson iteration without preconditioner as a solver, with prolongation from the previous solution as starting guess. Darker lines correspond to earlier cycles. Only the first* 30 *iterations are shown.*

radius of A_k. The optimal choice of the parameter ω_k is in this case $\omega_k = 1/\rho(A_k)$ (see, e.g., [20, 32]).

The high frequency components of the error are reduced by a factor which is close to zero, while the low frequency components of the error are left substantially unaltered. The high frequencies are also the ones that have a greater influence on classical a posteriori error estimators, justifying the S-AFEM algorithm.

To show the effect of the smoothing iterations on the a posteriori error estimator, we consider as an example case the Peak problem in two dimensions as described in subsection 4.1, and we apply ten cycles of the classical AFEM algorithm using non-preconditioned Richardson iterations for the algebraic resolution of the system with initial guess given by the prolongation of the previous approximation for each cycle.

In Figure 3.1 we plot the ℓ^2-norm of the residual $\mathbf{r}_k^{(\ell)} := A_k \mathbf{e}_k^{(\ell)}$ and the value of the a posteriori error estimator $J(u_k^\ell)$ for all cycles as the Richardson iteration count ℓ increases from 1 to 30.

The same behavior is present in every refinement cycle: the first few Richardson iterations induce a rapid drop in the residual norm (due to convergence of the highly oscillatory terms in the solution), while the remaining iterations converge very slowly, corresponding to the convergence speed of the low frequencies in the solution (typical of Richardson iterations). The estimator, on the other hand, stagnates after very few Richardson iterations (around two or three), suggesting that $J(u_h^\ell)$ is almost the same as $J(u_h)$ for $\ell \geq 3$, i.e., the error estimator is mainly affected by the highly oscillatory components of the discrete algebraic solution u_h^ℓ.

In this respect, classical results of a posteriori error analysis do not provide sharp bounds for the estimator evaluated on the smoothed algebraic approximation $J(u_h^\ell)$, and this closeness remains an open problem.

One can combine the upper bound (3.12) with a global lower bound [11] of the estimator evaluated on a generic v_h, i.e.,

$$J^2(v_h) \leq C_\star(|u - v_h|_1^2 + \mathrm{osc}^2) \ \forall v_h \in V_h, \tag{3.16}$$

to relate $J(u_h^\ell)$ with $J(u_h)$. However, the result (proved in the following lemma) remains a worst-case estimate (similar to (3.14)) and fails to capture the behavior that we observe, for example, in Figure 2.1, where $J(u_h)$ and $J(u_h^\ell)$ are substantially equivalent.

LEMMA 3.1. *There exist positive constants C_1, C_2, C_3 that only depend on the*

minimum angle of the triangulation, on Ω*, and on* Γ*, and that are independent of* f, u, u_h*, and of the mesh-sizes* h_T *such that*

$$J^2(v_h) \le C_1 J^2(u_h) + C_2 |u_h - v_h|_1^2 + C_3 osc^2 \ \forall v_h \in V_h. \tag{3.17}$$

Proof. For a given function $v_h \in V_h$, we decompose $u - v_h = (u - u_h) + (u_h - v_h)$, and we apply the equality $|u - v_h|_1^2 = |u - u_h|_1^2 + |u_h - v_h|_1^2$ (see, e.g., [24]) to the lower bound (3.16):

$$\begin{aligned} J^2(v_h) &\le C_\star(|u - v_h|_1^2 + \mathrm{osc}^2) \\ &= C_\star(|u - u_h|_1^2 + |u_h - v_h|_1^2 + \mathrm{osc}^2) \\ &\le C_\star(C^{\star 2}(\mathrm{osc}^2 + J^2(u_h)) + |u_h - v_h|_1^2 + \mathrm{osc}^2) \\ &= C_\star C^{\star 2} J^2(u_h) + C_\star |u_h - v_h|_1^2 + C_\star(C^{\star 2} + 1)\mathrm{osc}^2 \\ &= C_1 J^2(u_h) + C_2 |u_h - v_h|_1^2 + C_3 \mathrm{osc}^2, \end{aligned} \tag{3.18}$$

where we have used the upper bound (3.12) in (3.18). ∎

If we apply Lemma 3.1 with $v_h = u_h^\ell$, we obtain an upper bound on $J^2(u_h^\ell)$ in terms of $J^2(u_h)$ and of the algebraic error. A similar result involving the full estimator $\eta(u_h^\ell)$ can be found in [4].

It is still unclear how to improve the upper bound (3.17) to explain why $J(u_h^\ell)$ and $J(u_h)$ are as close as the numerical evidence suggests. What remains to be proved is that a sharper estimate on the constant C_2 of (3.17) may be obtained, that depends on the frequency content of v_h, showing that C_2 is small when $v_h = u_h^\ell$ stems from a smoothing iteration, bringing $J^2(u_h)$ close to $J^2(u_h^\ell)$ even though $|u_h^\ell - u_h|_1^2$ is in fact not small at all.

What can be done, however, is an estimate of the evolution of $|u_h^\ell - u_h|_1$ in the intermediate steps of S-AFEM, exploiting classical results of multigrid analysis. The following theorem provides such result when the smoothing iteration is performed using the Richardson method.

THEOREM 3.2 (algebraic error propagation in S-AFEM). *Let* $\mathbf{e}_k^{(\ell)} := \mathbf{u}_k - \mathbf{u}_k^\ell$ *denote the algebraic error after* ℓ *smoothing iterations at step* k *of S-AFEM for* $k = 2, \dots, \bar{k} - 1$*. Let* $\mathbf{a}_1 = 0$*, and let*

$$\mathbf{a}_{k+1} := \mathbf{u}_{k+1} - I_k^{k+1} \mathbf{u}_k \in \mathbb{R}^{N_{k+1}}, \qquad k = 1, \dots, \bar{k} - 1, \tag{3.19}$$

denote the difference between the exact algebraic solution $\mathbf{u}_{k+1}$ *at level* $k+1$ *and the prolongation to level* $k+1$ *of the exact algebraic solution* $\mathbf{u}_k$ *at level* k*. Then, the following error propagation formula holds true:*

$$\mathbf{e}_{k+1}^{(\ell)} = {M_{k+1}}^\ell (\mathbf{a}_{k+1} + I_k^{k+1} \mathbf{e}_k^{(\ell)}) \ \text{ for } \ k = 1, \dots, \bar{k} - 1. \tag{3.20}$$

Proof. In the *Prolongate* step of S-AFEM, the outcome $\mathbf{u}_k^{(\ell)}$ of the *Smooth* procedure at step k is prolongated to step $k+1$ and used as an initial guess in the Richardson iteration at step $k+1$. We can write

$$\begin{aligned} \mathbf{u}_{k+1}^{(0)} &= I_k^{k+1} \mathbf{u}_k^{(\ell)} \\ &= I_k^{k+1} \mathbf{u}_k - I_k^{k+1} \mathbf{e}_k^{(\ell)} \end{aligned} \tag{3.21}$$

and therefore express the initial error at step $k+1$ as

$$\begin{aligned} \mathbf{e}_{k+1}^{(0)} &= \mathbf{u}_{k+1} - \mathbf{u}_{k+1}^{(0)} \\ &= \mathbf{u}_{k+1} - I_k^{k+1}\mathbf{u}_k + I_k^{k+1}\mathbf{e}_k^{(\ell)} \\ &= \mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}. \end{aligned} \tag{3.22}$$

By applying directly the property of the error propagation formula for Richardson iterations, we obtain the final algebraic error at level $k+1$:

$$\begin{aligned} \mathbf{e}_{k+1}^{(\ell)} &= {M_{k+1}}^{\ell}\mathbf{e}_{k+1}^{(0)} \\ &= {M_{k+1}}^{\ell}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(\ell)}), \end{aligned} \tag{3.23}$$

which proves the recursive formula. ∎

Theorem 3.2 shows that the nature of the algebraic error in S-AFEM is the result of ℓ smoothing iterations applied to a vector that accumulates the (smoothed) prolongation of the exact algebraic solution coming from step zero.

The rationale behind S-AFEM is then that in the first step we perform a full $\mathcal{S}olve$, resulting in a negligible algebraic error, and the only components of the error that we are introducing when prolongating from step k to step $k+1$ are high frequency errors (introduced by local refinement). These, however, are reduced very quickly by ℓ steps of smoothing iterations.

The residual algebraic error that persists as S-AFEM proceeds (clearly visible in Figure 2.1) seems not to have a detrimental effect on $J(u_h^\ell)$. Such an algebraic error is probably confined on medium to low frequencies and shows no noticeable effect on the $\mathcal{E}stimate$ and $\mathcal{M}ark$ steps of S-AFEM.

Although the value we plot in Figure 3.1 for the estimator is a global one and gives no information on the distribution of the local estimator on the grid, it is a good hint that the overall behavior of such a distribution will not be changing too much after the first few Richardson iterations. We show some numerical evidence that this is actually the case in the numerical validation provided in section 4.

To summarize the idea behind S-AFEM, we argue that in the intermediate AFEM cycles it is not necessary to solve exactly the discrete system. What matters instead is to capture accurately the highly oscillatory components of the discrete approximation. Low frequency components *may* have an influence on the error estimator; however, this is mostly a *global* influence that has a small effect on the cells that will actually be marked for refinement in the $\mathcal{M}ark$ step. As an example, consider Figure 4.11, where a Peak problem in 3D is solved using both AFEM and S-AFEM. The estimator evaluated on u_h^ℓ in this case is farther away from the one evaluated on u_h w.r.t. the same problem in two dimensions, but the convergence rate of the solution obtained with the sequence of grids constructed with S-AFEM is still the optimal one and is comparable to the one obtained with AFEM at a fraction of the computational cost.

4. Numerical validation. We consider a class of drift-diffusion problems of the following form:

$$\begin{aligned} -\Delta u + \boldsymbol{\beta}\cdot\nabla u &= f \quad \text{in } \Omega, \\ u &= u_g \quad \text{on } \partial\Omega \end{aligned} \tag{4.1}$$

in two and three dimensions.

We compare the classical AFEM algorithm with S-AFEM based on three different smoothing strategies: Richardson iterations (Richardson), conjugate gradient iterations (CG), and generalized minimal residual iterations (GMRES) in the symmetric case ($\boldsymbol{\beta} = 0$), and GMRES alone in the nonsymmetric case ($\boldsymbol{\beta} \neq 0$), and for different discretization degrees.

We test two classical experiments used to benchmark AFEMs when $\boldsymbol{\beta}$ is zero, and a simple drift-diffusion problem with constant transport term $\boldsymbol{\beta}$ to test S-AFEM in general drift-diffusion problems.

The numerical results presented in this paper were realized using a custom `C++` code based on the `deal.II` library [3, 6, 7] and on the `deal2lkit` library [33]. The open-source code is available on a public repository at https://github.com/luca-heltai/sa-pinvit. In the $\mathcal{M}ark$ step, we use the classical Dörfler marking strategy [18]: for any level k we mark for refinement the subset of elements

$$\mathcal{M}_k := \{T \in \mathcal{T}_k : \eta_T \geq L\}, \tag{4.2}$$

where L is a *threshold error*, defined as the largest value such that

$$\Theta \sum_{T \in \mathcal{T}_k} \eta_T^2 \leq \sum_{T \in \mathcal{M}_k} \eta_T^2. \tag{4.3}$$

The parameter Θ is such that $0 \leq \Theta \leq 1$, where $\Theta = 1$ corresponds to an almost uniform refinement, while $\Theta = 0$ corresponds to no refinement. In our numerical tests, unless otherwise stated, we set $\Theta = 0.3$. The refinement strategy that we adopt in this work is based on the use of hanging nodes [9].

The results presented in this section are a subset of the full campaign of simulations presented in [27].

4.1. Two-dimensional examples: Pure diffusion, bilinear case, Richardson smoother.

Smooth domain, peak right-hand side. The first example we consider consists in solving the model problem with no transport term ($\boldsymbol{\beta} = 0$) on a square domain, with a custom forcing term that contains a peak in a specified point in the domain, forcing the exact solution to be

$$u(x, y) = x(x-1)y(y-1)e^{-100\left((x-0.5)^2+(y-0.117)^2\right)}, \tag{4.4}$$

as shown in Figure 4.1.

L-shaped domain, smooth right-hand side. In the second 2D test case in pure diffusion, we consider an L-shaped domain, i.e., a square where the upper right corner is removed, and the reentrant corner coincides with the origin. No forcing term is added to the problem, but the boundary conditions are set so that the following exact solution is obtained (when expressed in polar coordinates):

$$u(r, \theta) = r^{2/3} \sin\left(\frac{2\theta + 5\pi}{3}\right), \tag{4.5}$$

as shown in Figure 4.2.

In both cases, we apply ten cycles of classical AFEM and of S-AFEM, respectively. For the AFEM algorithm, we use the CG method as an iterative solver, with an

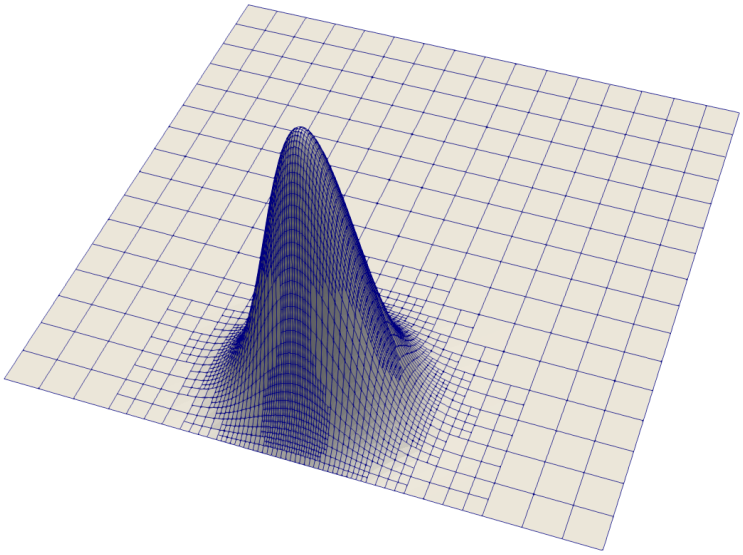


FIG. 4.1. *Solution to the Peak problem in* 2*D* (4.4).

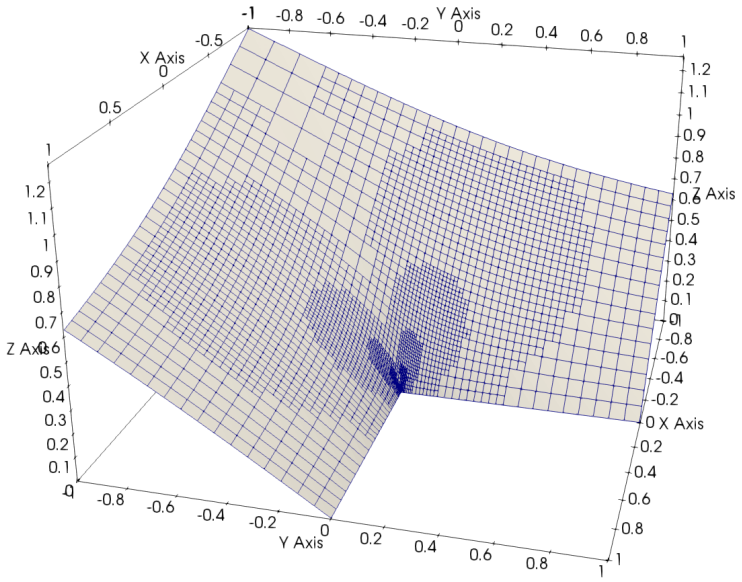


FIG. 4.2. *Solution to the L-shaped domain problem in* 2*D* (4.5).

algebraic multigrid preconditioner (AMG), and we iterate until the ℓ^2-norm of the residual is below a tolerance of 10^{-12} for each cycle. For S-AFEM, we modify the intermediate cycles, and we only apply three Richardson iterations. For reference, we report a comparison between the cells marked for refinement by AFEM and S-AFEM after four cycles for the Peak problem in Figure 4.3 and after nine cycles for the L-shaped domain problem in Figure 4.4.

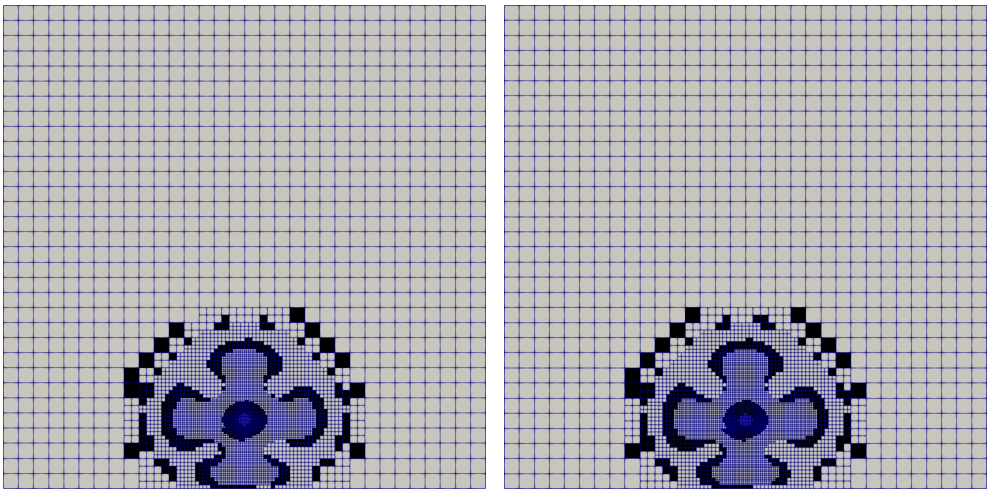


FIG. 4.3. *Comparison between the cells marked for refinement in AFEM (left) and S-AFEM (right) after* 9 *cycles for the Peak problem in* 2*D.*

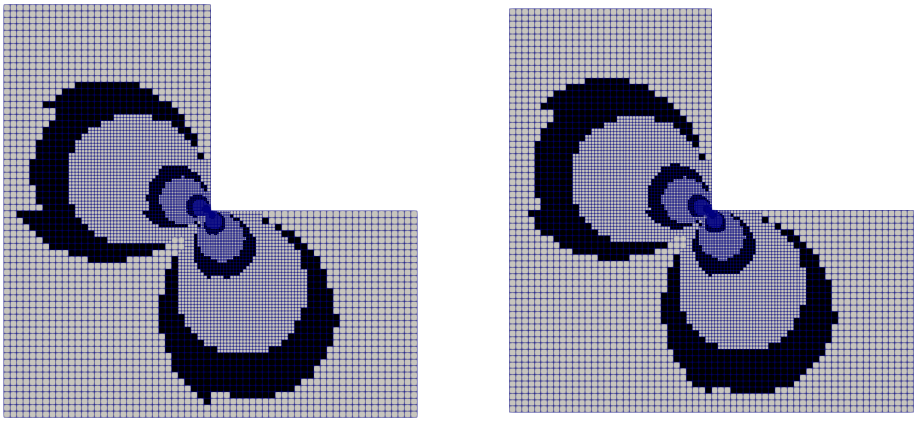


FIG. 4.4. *Comparison between the cells marked for refinement in AFEM (left) and S-AFEM (right) after* 5 *cycles for the L-shaped domain problem in* 2*D.*

In both cases, the sequence of generated grids by AFEM and S-AFEM is similar (although not identical), and the accuracy of the final solution is comparable.

In Figures 4.5 and 4.7 we compare the values of the global estimators $J(u_h)$ and $J(u_h^\ell)$ and of the H_1 seminorm of the total errors for each cycle for the Peak

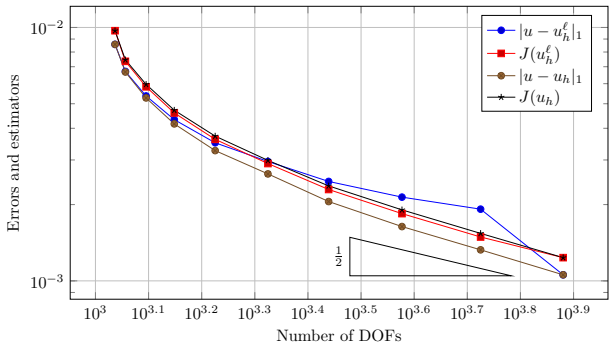


FIG. 4.5. *Values of the total error in the H^1 seminorm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Peak problem in 2D.*

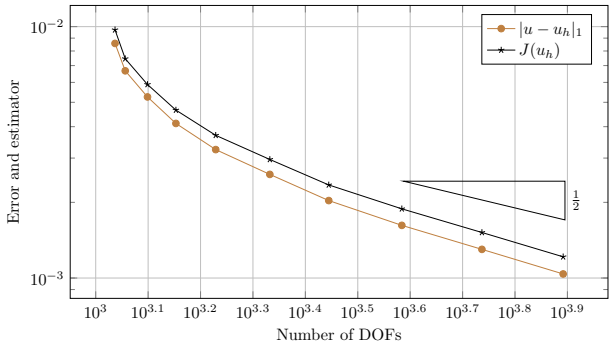


FIG. 4.6. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Peak problem in 2D, using classical AFEM.*

problem, and for the L-shaped domain problem, respectively, when using S-AFEM. For reference, Figures 4.6 and 4.8 show the error and the estimator in the classical AFEM algorithm for the two examples. Notice that the first step of AFEM and the first step of S-AFEM are the same. The last step in the S-AFEM case shows comparable results as in the AFEM algorithm for both examples.

Notice that in S-AFEM the value of the global estimator is almost the same as the one that would be obtained by solving using CG preconditioned with AMG ($J(u_h)$ in Figures 4.5 and 4.7), showing that in the 2D setting the error estimator (3.11) is mainly affected by the high frequencies of the discrete solution, which are well captured with just a few Richardson iterations. On the other hand, the total error increases in the intermediate cycles, due to the algebraic error that has been accumulated by applying smoothing iterations instead of solving the algebraic problem until convergence, as quantified by Theorem 3.2. This error measures the distance between the exact algebraic solution and the smooth nonoscillatory components of the approximate solution that are not captured by Richardson iteration and have little or no influence on the error estimator, and therefore on the generated grid. After ten cycles, we solve the algebraic problem until convergence using CG and AMG, as in the first cycle, and we obtain a solution whose error is controlled by the estimator, as

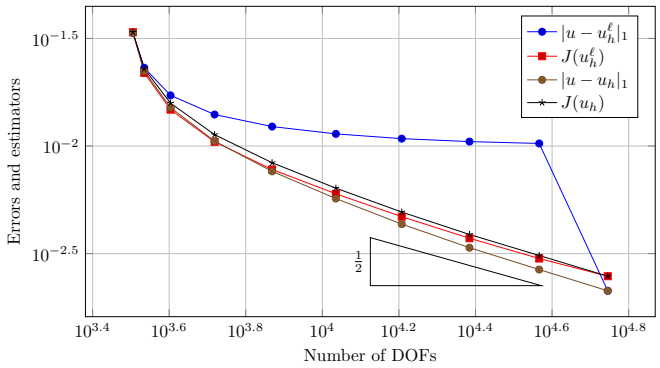


FIG. 4.7. *Values of the total error H^1 seminorm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the L-shaped domain problem in 2D.*

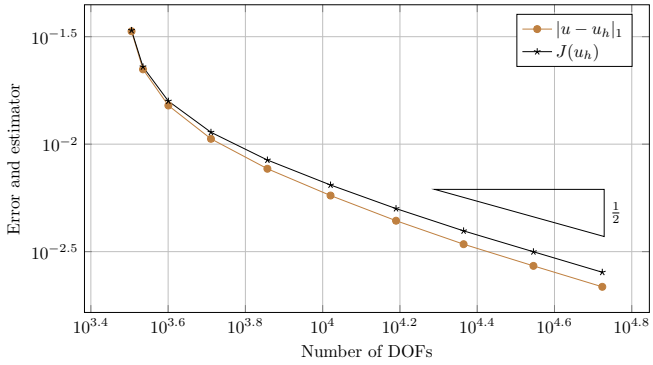


FIG. 4.8. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the L-shaped domain problem in 2D, using classical AFEM.*

expected.

4.2. Three-dimensional examples: Pure diffusion, bilinear case, Richardson smoother.

Smooth domain, peak right-hand side. The first 3D test case that we propose is a model problem on a cube domain, where the forcing term contains a peak in a specified point that forces the exact solution to be given by

$$u(x,y,z) = x(x-1)y(y-1)z(z-1)e^{-100\left((x-0.5)^2+(y-0.117)^2+(z-0.331)^2\right)}, \tag{4.6}$$

as shown in Figure 4.9.

Fichera corner domain, smooth right-hand side. In the second 3D example, we consider the classic Fichera corner domain, i.e., a cube where the upper right corner is removed and the reentrant corner coincides with the origin. We set the exact solution to be

$$u(r,\theta,\phi) = r^{1/2}, \tag{4.7}$$

and we add a forcing term that induces the above exact solution as shown in Fig-

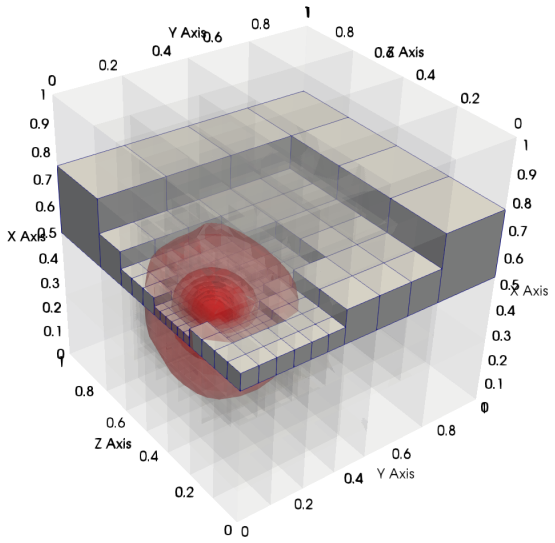


FIG. 4.9. *Solution to the Peak problem* (4.6) *in* 3*D*.

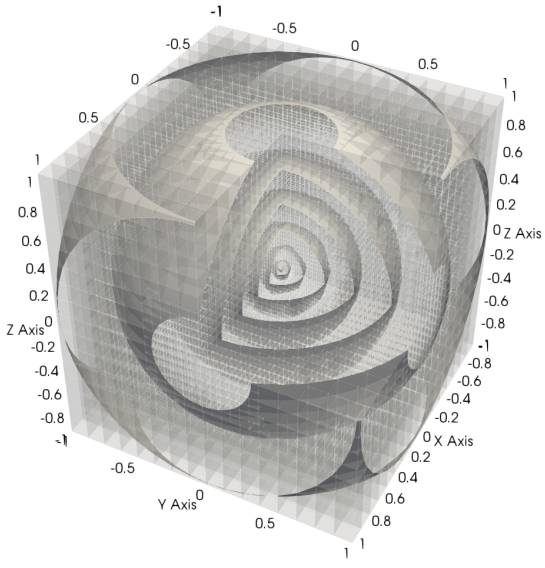


FIG. 4.10. *Solution to the Fichera domain problem* (4.7) *in* 3*D*.

ure 4.10.

In both examples, the estimator applied to the algebraic solution after three smoothing steps (see Figures 4.11 and 4.13) seems to be more sensitive to the low frequency content of u_h^ℓ.

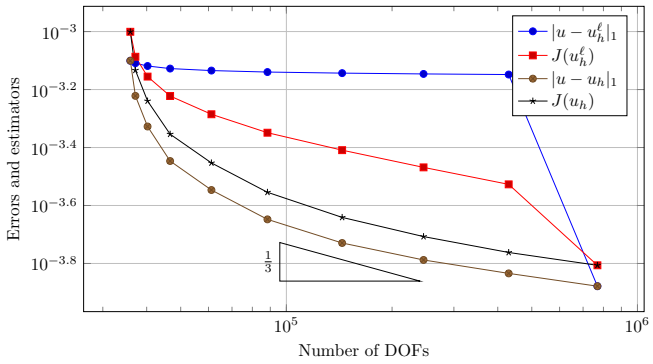


FIG. 4.11. *Values of the total error H^1 seminorm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Peak problem in 3D.*

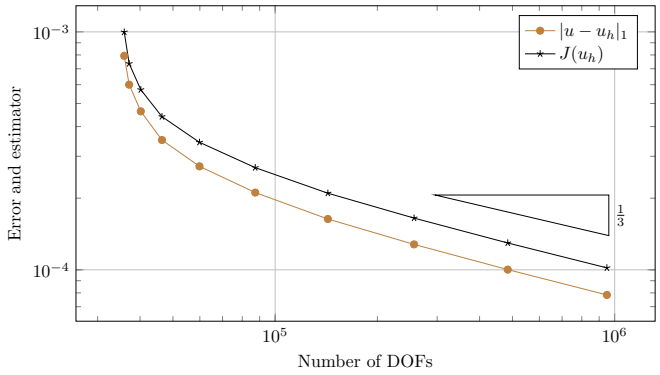


FIG. 4.12. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Peak problem in 3D, using classical AFEM.*

For reference, Figures 4.12 and 4.14 show the error and the estimator in the classical AFEM algorithm for the two examples. In the 3D case Lemma 3.1 and Theorem 3.2 provide a sharper estimate, and we do not observe the same behavior as in the 2D case (i.e., $J(u_h^\ell)$ does not seem to remain close to $J(u_h)$). Nonetheless, the difference in accuracy at the final step between AFEM and S-AFEM is negligible also in this case, showing that the differences in the refinement patterns between AFEM and S-AFEM remain small and do not hinder the final accuracy.

4.3. Robustness with respect to approximation degree, smoothing algorithms, and number of smoothing steps. We now consider different variants of our S-AFEM algorithm, where we apply a different number of smoothing iterations, and different smoother types in the intermediate steps (respectively, Richardson iteration, CG, and GMRES smoothers), for high order finite element discretizations of the pure diffusion case.

We apply both AFEM and S-AFEM to the 2D Corner problem (4.5), and we show a comparison for different fixed FEM degrees, as $deg = 1, 2, 3$, and for different choices of smoothers for the intermediate cycles, respectively, Richardson iteration, the CG method, and the GMRES method. For all cases, we plot the value of the error estimator J and the value of the $|\cdot|_1$ seminorm of the total error as the number

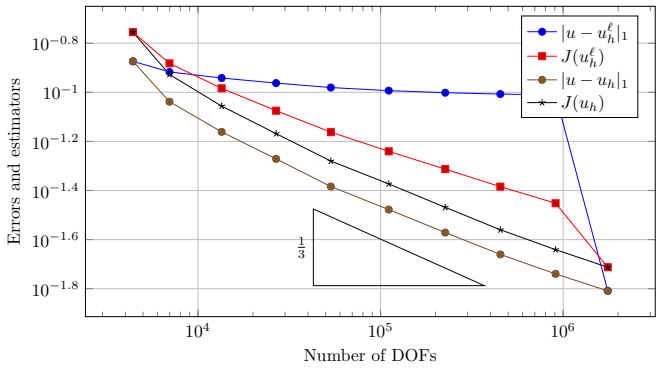


FIG. 4.13. *Values of the total error H^1 seminorm and of the error estimator for each loop of the classical AFEM ($|u-u_h|_1$ and $J(u_h)$) and S-AFEM with $\ell = 3$ smoothing iterations ($|u-u_h^\ell|_1$ and $J(u_h^\ell)$) for the Fichera corner domain problem in 3D.*

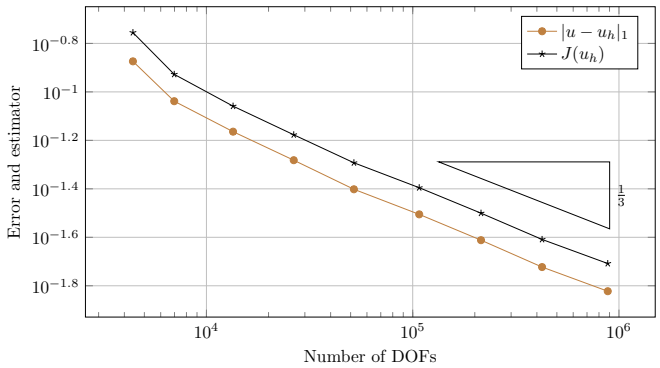


FIG. 4.14. *Values of the total error $|u-u_h|_1$ and error estimator $J(u_h)$ for the Fichera corner problem in 3D, using classical AFEM.*

of smoothing iterations ℓ increases from 1 to 5 in Figures 4.15–4.23.

For bilinear finite elements (cf. Figures 4.15–4.17), all considered smoothers (Richardson, CG, and GMRES) turned out to be good. In all cases, the estimator $J(u_h^\ell)$ with $\ell = 1, 3, 5$ exhibits the same behavior (same order of convergence) of the estimator $J(u_h)$, showing that one or two smoothing iterations would be enough for the intermediate cycles. When we look at the total error, the CG behaves better, leading to less error accumulated at the intermediate levels as shown in Figure 4.16, while Richardson behaves the worst (cf. Figure 4.15). Nevertheless, in all cases the accuracy of the final approximation for the last cycle obtained by S-AFEM is almost the same as the one that is generated by classical AFEM.

As the polynomial degree increases, we observed that the Dörfler marking strategy does not provide good refinement patterns for the different problems, unless a fine tuning is made on the marking parameter. Using the same value for Θ used for degree one, no cells are marked for refinement in higher order finite elements, making the choice for this parameter too problem dependent and polynomial degree dependent. As an alternative marking strategy, we opted for a marking criterion where a fraction of 1/3 of the cells with the largest error indicators are selected for refinement, leading to an increase of the number of degrees of freedom of roughly a factor of two in each

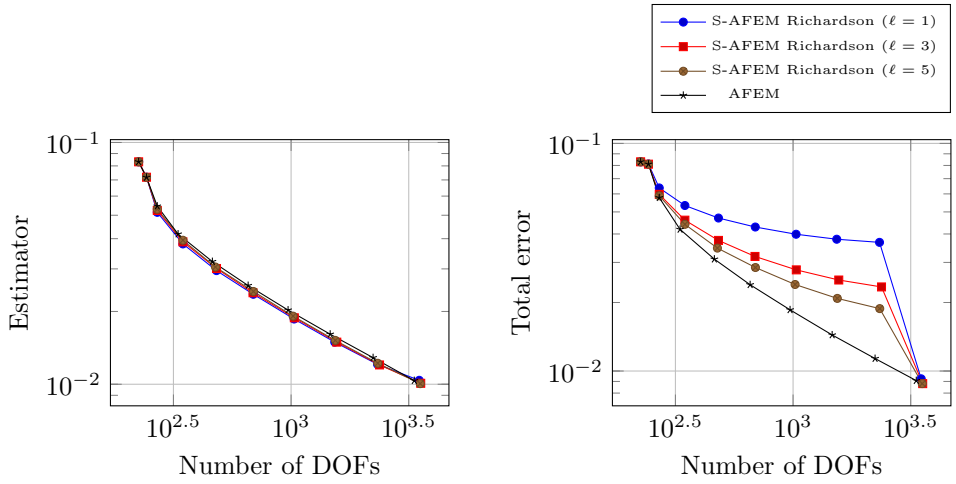


FIG. 4.15. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ semi-norm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 1$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of Richardson method as a smoother. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

refinement cycle, independently of the problem type.

For $deg = 2$, all smoothers work well, exhibiting a quasi-optimal convergence order compared to classical AFEM, and the accuracy of the final approximation is almost the same, as shown in Figures 4.18, 4.19, and 4.20.

For $deg = 3$ and higher (only $deg = 3$ is shown here; see [27] for the full set of simulations), the Richardson iteration turns out to be a bad smoother for S-AFEM, unless further tuning of the relaxation parameter ω is performed. Although $J(u_h^\ell)$ seems to exhibit the same behavior as $J(u_h)$, as the smoothing iteration count ℓ increases, contrarily to what one might expect, the value of the estimator increases with increasing degrees of freedom (Figure 4.21), showing that our selection of ω may not be correct for these problems, and we should estimate in a better way the spectral radius of the final matrix A and modify ω accordingly. In our experience, Richardson's method performs badly for higher order elements.

On the other hand, both the CG method and the GMRES method turn out to be good smoothers for S-AFEM, without the need to tune any parameter, as evidenced in Figures 4.19, 4.20, 4.22, and 4.23. In all cases, in fact, $J(u_h^\ell)$, for $\ell \geq 2$, shows the same optimal convergence rate as the $J(u_h)$ obtained by the classical AFEM, and although the total error at the intermediate cycles is evident, the accuracy of the final approximations is almost the same.

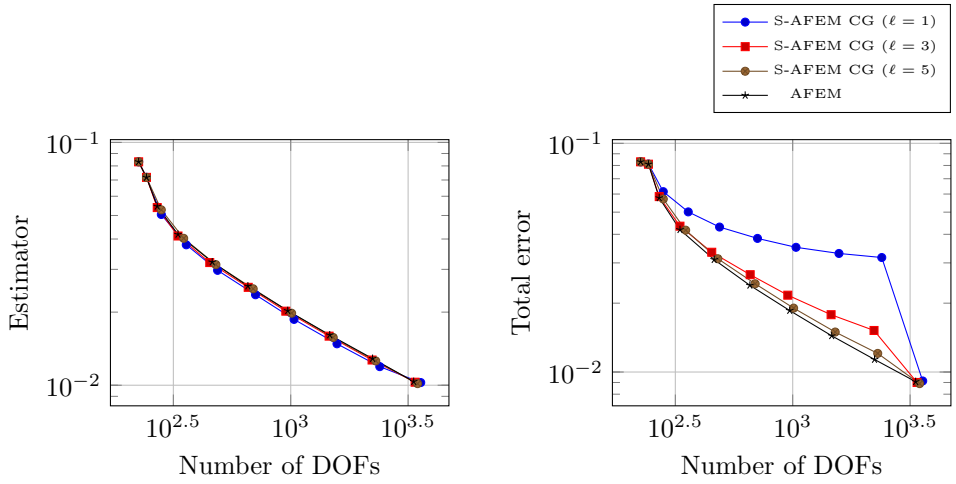


FIG. 4.16. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is deg $= 1$, and we execute* 10 *cycles of AFEM and S-AFEM with $\ell = 1, 3$, and* 5 *iterations of CG method as a smoother. The initial global refinement is* 3 *and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

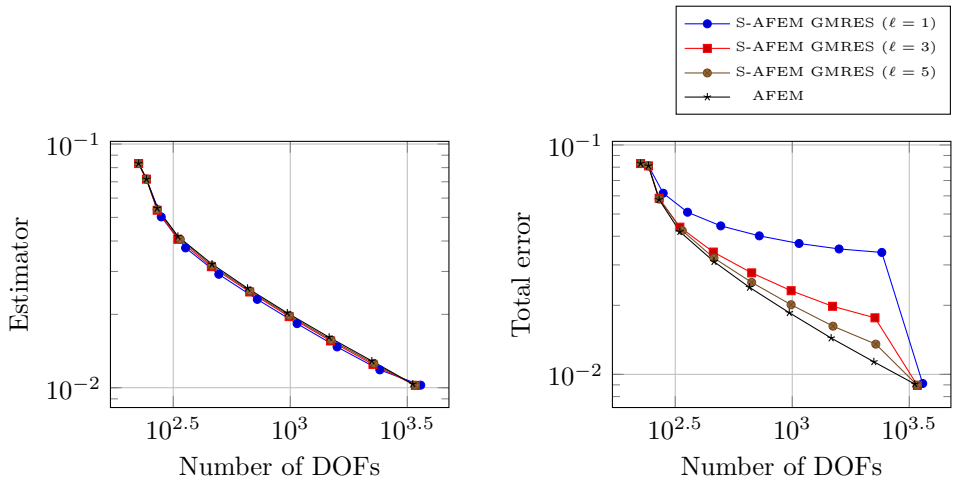


FIG. 4.17. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is deg $= 1$, and we execute* 10 *cycles of AFEM and S-AFEM with $\ell = 1, 3$, and* 5 *iterations of GMRES method as a smoother. The initial global refinement is* 3 *and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

4.4. Two-dimensional drift-diffusion problem. We proceed with testing the accuracy of S-AFEM for a class of drift-diffusion problems where the transport term $\boldsymbol{\beta}$ is nonnegligible. We consider a 2D problem, where $\boldsymbol{\beta} = (\beta, \beta)^T$, and the scalar parameter β takes the values 1, 10, and 50. In particular, we impose boundary conditions and forcing terms so that the exact solution is

$$x + y + \frac{-e^{\beta x} + 1}{e^{\beta} - 1} - \frac{e^{\beta y} - 1}{e^{\beta} - 1}. \tag{4.8}$$

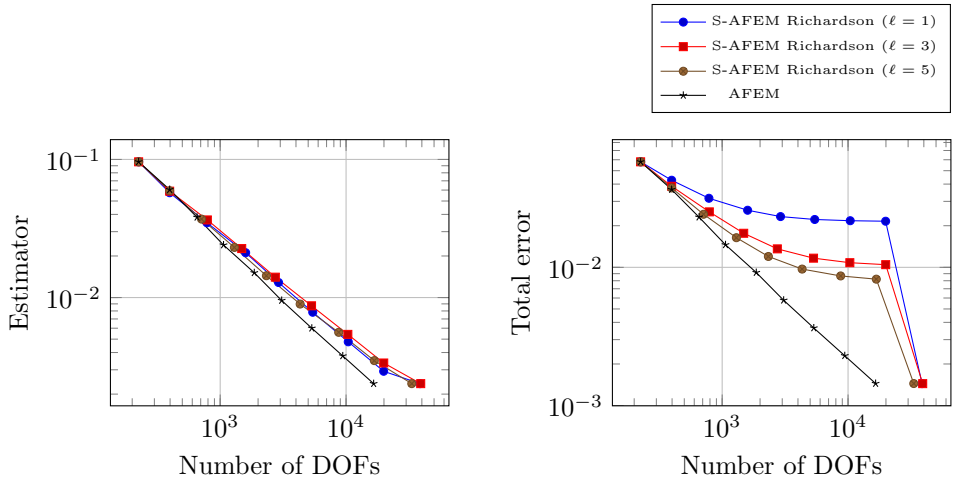


FIG. 4.18. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of Richardson method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

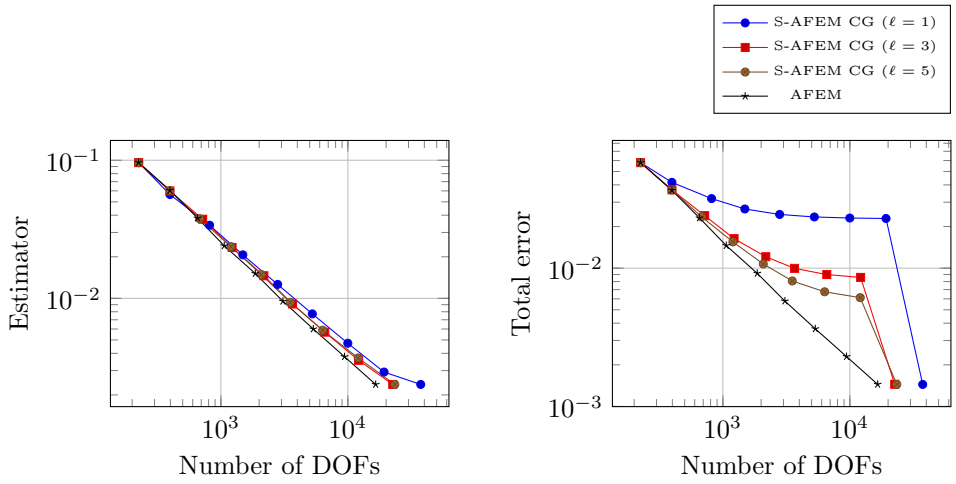


FIG. 4.19. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of CG method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

Since the linear system associated to problem (4.1) when $\boldsymbol{\beta} \neq 0$ is not symmetric, we use as a solver and as a smoother the GMRES method (cf., e.g., [24]). We apply classic AFEM and S-AFEM for both bilinear and higher order finite element discretizations for $deg = 1, 2, 3$. We plot the value of the estimator J and the value of the $|\cdot|_1$ seminorm of the total error, using a fixed number of GMRES iterations as a smoother, with $\ell = 1, 3, 5$.

For all the cases where the transport term $\boldsymbol{\beta} = (1, 1)^T$ is small, the behavior of

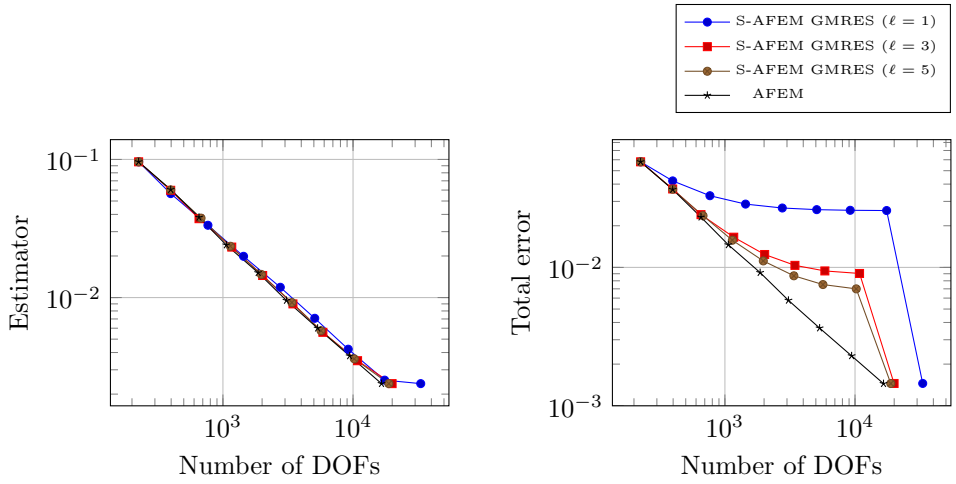


FIG. 4.20. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

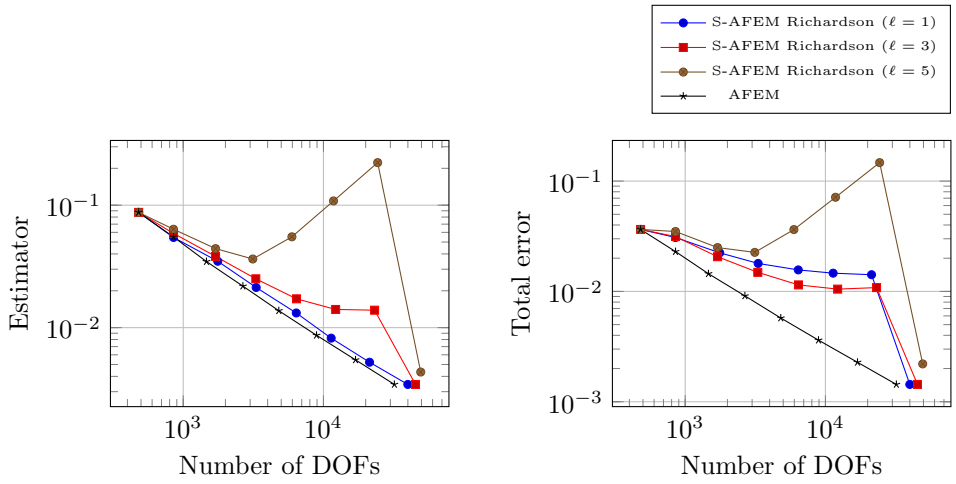


FIG. 4.21. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of Richardson method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

the estimator for S-AFEM is exactly the same as the one given by AFEM, for the case $deg = 1$, as shown in Figure 4.24, while it approaches it as the GMRES smoothing iteration count increases for higher order FEM discretizations (i.e., for $deg \geq 2$), as shown in Figures 4.25 and 4.26. However, in all cases, the accuracy of the final approximations is almost the same.

For the choices of the transport term $\boldsymbol{\beta} = (10, 10)^T$ (corresponding to moderate transport) and $\boldsymbol{\beta} = (50, 50)^T$ (corresponding to large transport), again the behavior

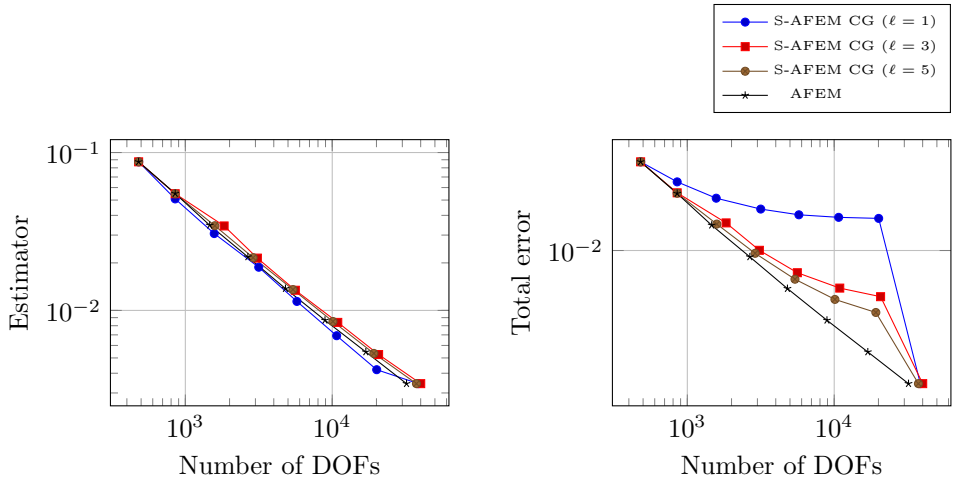


FIG. 4.22. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of CG method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

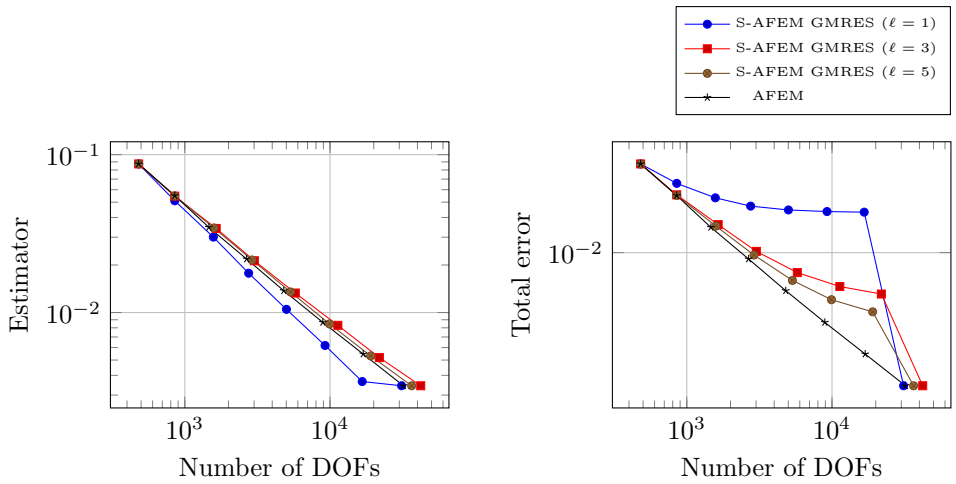


FIG. 4.23. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the corner problem in 2D. The FEM discretization degree is $deg = 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.*

of the estimator for S-AFEM is exactly the same as that given by AFEM for the case $deg = 1$, as shown in Figures 4.27 and 4.30, while it approaches it as the GMRES smoothing iteration count increases for higher order FEM discretizations, as evidenced in Figures 4.28, 4.29, 4.31, and 4.32. In all cases, however, the accuracy of the final approximations is almost the same, showing that S-AFEM turns out to be a good

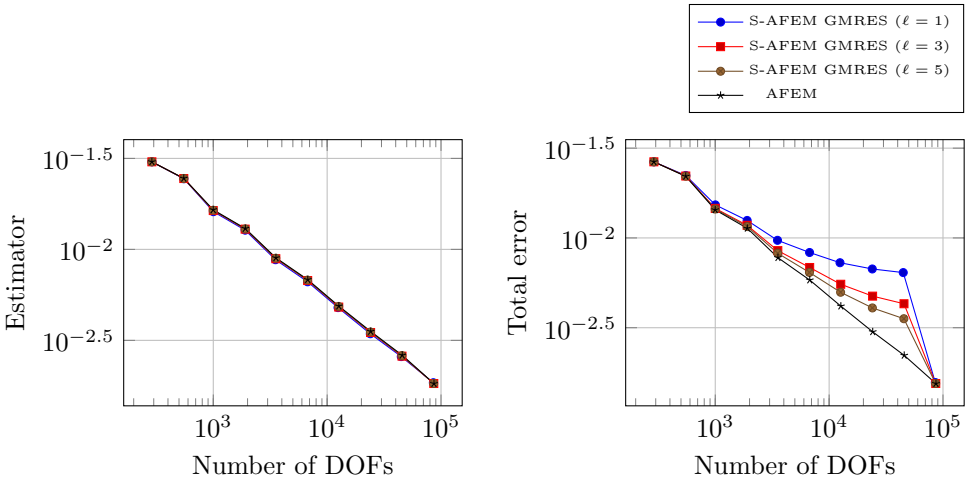


FIG. 4.24. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (1,1)$. The FEM discretization degree is deg $= 1$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

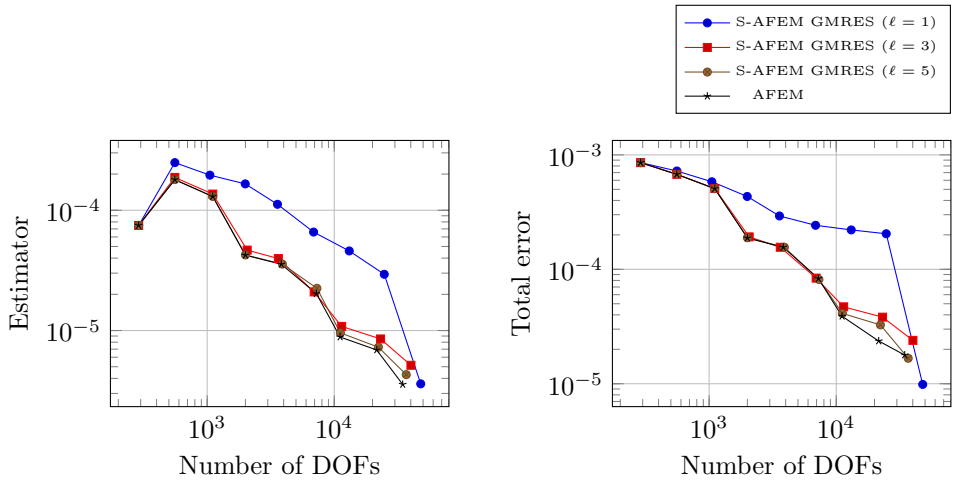


FIG. 4.25. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (1,1)$. The FEM discretization degree is deg $= 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

method also for drift-diffusion problems.

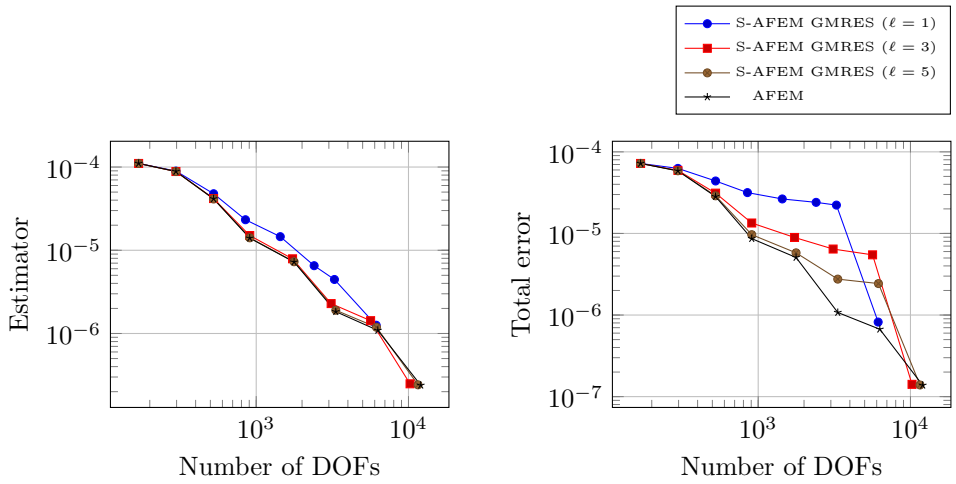


FIG. 4.26. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (1,1)$. The FEM discretization degree is deg $= 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

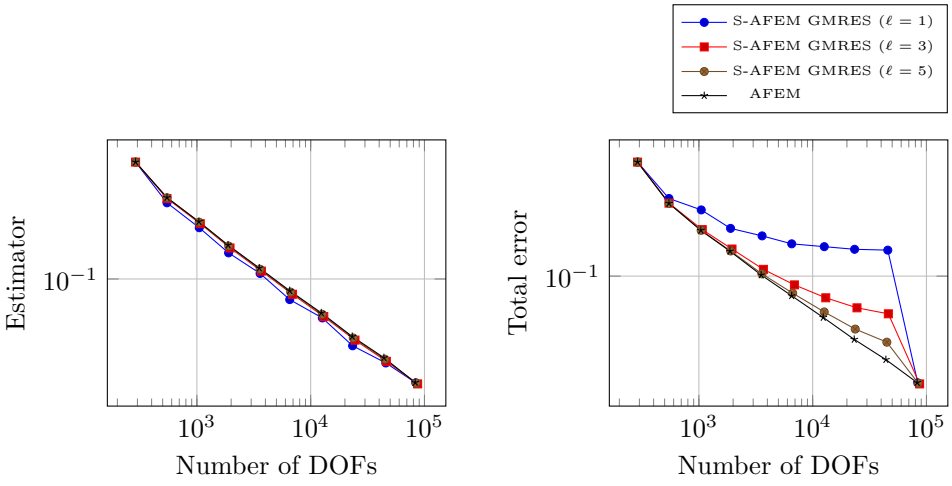


FIG. 4.27. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (10,10)$. The FEM discretization degree is deg $= 1$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

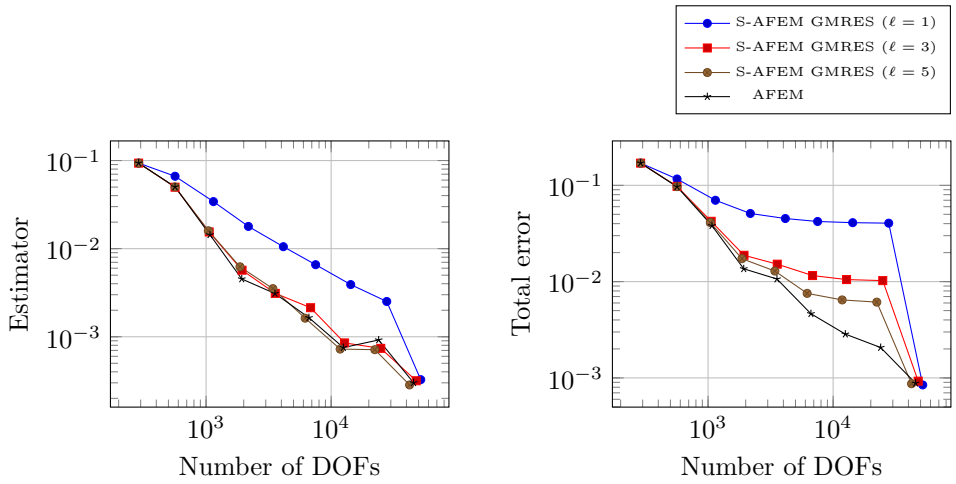


FIG. 4.28. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (10, 10)$. The FEM discretization degree is deg $= 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

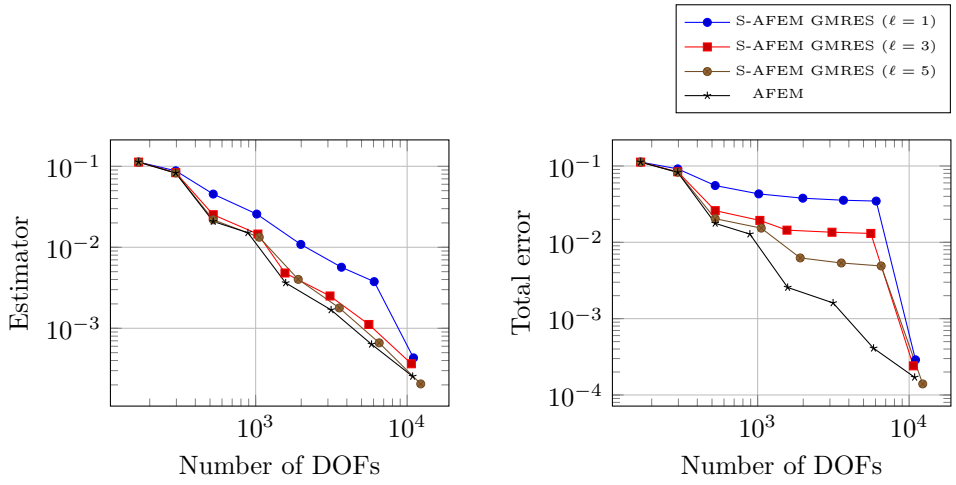


FIG. 4.29. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (10, 10)$. The FEM discretization degree is deg $= 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

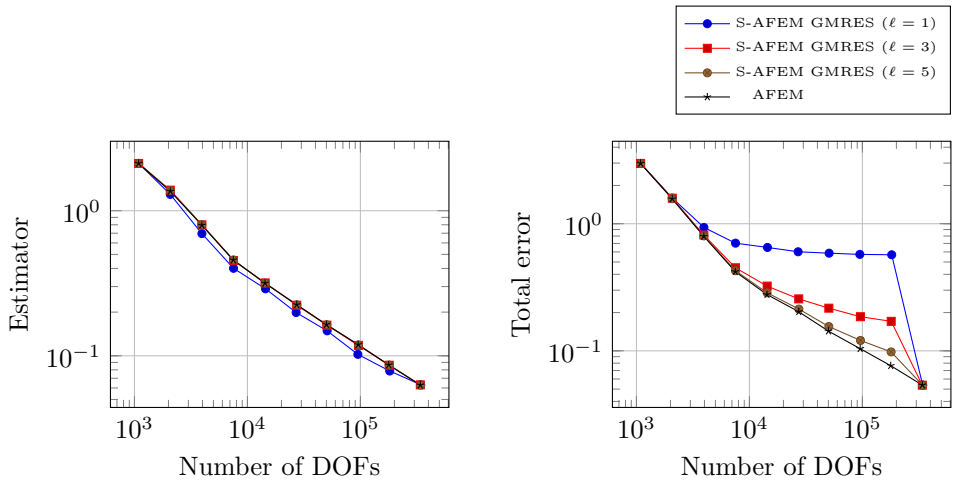


FIG. 4.30. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (50, 50)$. The FEM discretization degree is deg $= 1$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.*

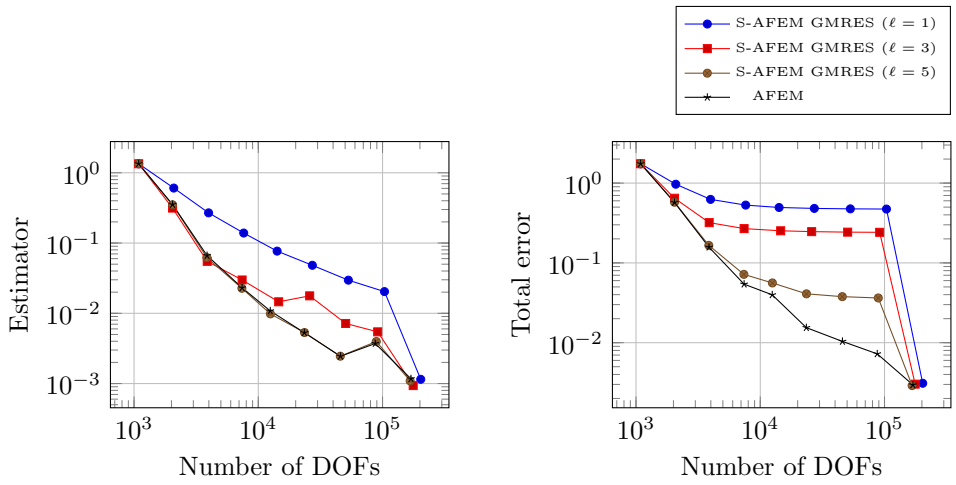


FIG. 4.31. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ seminorm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (50, 50)$. The FEM discretization degree is deg $= 2$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

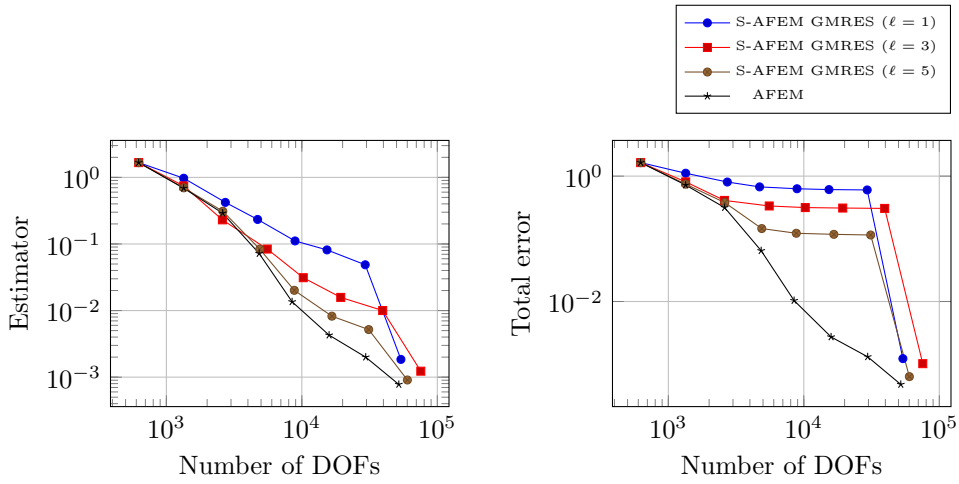


FIG. 4.32. *Value of the error estimator J (left) and of the total error in the $|\cdot|_1$ semi-norm (right) for the drift-diffusion problem in 2D, with transport $\boldsymbol{\beta} = (50, 50)$. The FEM discretization degree is deg $= 3$, and we execute 10 cycles of AFEM and S-AFEM with $\ell = 1, 3$, and 5 iterations of GMRES method as a smoother. The initial global refinement is 3 and we select a fraction of $1/3$ of cells for refinement at each cycle.*

TABLE 4.1
Comparison of the computational cost of the solution stage for ten cycles of adaptive refinement using classical AFEM and S-AFEM on bilinear elements for the pure diffusion case.

	Peak 2D	L-shaped 2D	Peak 3D	Fichera 3D
First and last solve (same for AFEM and S-AFEM)	0.0187s	0.0601s	32s	101s
AFEM Intermediate solves (CG)	0.0663s	0.219s	76.4s	185s
S-AFEM Intermediate smoothing steps (Richardson)	0.005s	0.00892s	0.252s	0.426s
S-AFEM intermediate speedup	13.26	24.6	303.7	434.3
S-AFEM total speedup	3.59	4.045	3.361	2.819

4.5. Computational costs. In Table 4.1 we show a comparison of the computational cost associated to the classical AFEM and to the smoothed AFEM in the pure diffusion case for the first four examples we presented in this section.

The results were obtained on a 2.8 GHz Intel Core i7 with 4 cores and 16GB of RAM, using MPI parallelization on all 4 cores.

Table 4.1 only shows the comparison between AFEM and S-AFEM in the solve phase, where S-AFEM is always faster than AFEM, offering an average speedup of a factor three. In the table we compare the computational cost of all intermediate cycles in S-AFEM ("Intermediate smoothing steps (Richardson)" in the table), with the corresponding computational cost for standard AFEM ("Intermediate solves (CG)" in the table). The first and last solves are the same in the two algorithms and are reported to provide a scaling with respect to the total computational cost of the solution phase in the program. Other phases (like graphical output, mesh setup, assembling setup, and error estimation) are not shown since they are identical in the two algorithms.

5. Conclusions. This work proposes a new smoothed algorithm for adaptive finite element methods (S-AFEM), inspired by multilevel techniques, where the exact algebraic solution in intermediate steps is replaced by the application of a prolongation step, followed by a fixed number of smoothing steps.

The main argument behind the S-AFEM algorithm is that the combined application of the $\mathcal{E}stimate$–$\mathcal{M}ark$ steps of AFEM is largely insensitive to substantial algebraic errors in low frequencies. Indeed, even though the intermediate solutions produced by S-AFEM are far from the exact algebraic solutions, we show that their a posteriori error estimation produces a refinement pattern for each cycle that is substantially equivalent to the one that would be generated by classical AFEM, leading roughly to the same set of cells marked for refinement.

Our strategy is based on solving exactly the problem at the coarsest level and at the finest level, and then applying the $\mathcal{E}stimate$–$\mathcal{M}ark$–$\mathcal{R}efine$ steps directly to the result of a smoother ($\mathcal{S}mooth$) in intermediate levels.

We provide numerical evidence that the S-AFEM strategy is competitive in cost and accuracy by considering some variants of our algorithm, where different smoothers are used in the intermediate cycles (respectively, the Richardson iteration, the CG method, and the GMRES method).

We conclude that, in general, CG and GMRES act as robust smoothers in S-AFEM also for high order approximations, and for nonsymmetric problems, like, for example, drift-diffusion problems with dominant transport.

Our numerical evidence shows that two or three smoothing iterations are enough for the 2D case, while 3D problems require from five up to seven smoothing iterations

in order to produce good final approximations, independently of the polynomial degree of the finite element approximation.

We analyzed the error propagation properties of the S-AFEM algorithm and provided a bound on the a posteriori error estimator applied to the approximated algebraic solution. The results are not sharp, and do not provide a definitive answer on the convergence of the final S-AFEM solution to the AFEM one, but could be used as a ground state for further investigation, which is currently ongoing.

Acknowledgment. The first author is thankful to Durham University for the hospitality during her visiting research there.

REFERENCES

[1] R. A. Adams and J. J. Fournier, *Sobolev Spaces*, 2nd ed., Pure Appl. Math. 140, Elsevier, Academic Press, Amsterdam, 2003.
[2] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, John Wiley & Sons, New York, 2011.
[3] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, *The* `deal.II` *library, version* 9.0, J. Numer. Math., 26 (2018), pp. 173–183, https://doi.org/10.1515/jnma-2018-0054.
[4] M. Arioli, E. H. Georgoulis, and D. Loghin, *Stopping criteria for adaptive finite element solvers*, SIAM J. Sci. Comput., 35 (2013), pp. A1537–A1559, https://doi.org/10.1137/120867421.
[5] M. Arioli, J. Liesen, A. Miçdlar, and Z. Strakoš, *Interplay between discretization and algebraic computation in adaptive numerical solution of elliptic PDE problems*, GAMM-Mitt., 36 (2013), pp. 102–129.
[6] D. Arndt, W. Bangerth, T. C. Clevenger, D. Davydov, M. Fehling, D. Garcia-Sanchez, G. Harper, T. Heister, L. Heltai, M. Kronbichler, R. Maguire Kynch, M. Maier, J. P. Pelteret, B. Turcksin, and D. Wells, *The deal.II library, version* 9.1, J. Numer. Math., 27 (2019), pp. 203–213.
[7] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells, *The deal.II finite element library: Design, features, and insights*, Comput. Math. Appl., 81 (2021), pp. 407–422.
[8] I. Babuška and W. C. Rheinboldt, *A-posteriori error estimates for the finite element method*, Internat. J. Numer. Methods Engrg., 12 (1978), pp. 1597–1615.
[9] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.II: A general purpose object oriented finite element library*, ACM Trans. Math. Software, 33 (2007), 24.
[10] R. Becker, C. Johnson, and R. Rannacher, *Adaptive error control for multigrid finite element*, Computing, 55 (1995), pp. 271–288.
[11] R. Becker and S. Mao, *Convergence and quasi-optimal complexity of a simple adaptive finite element method*, ESAIM Math. Model. Numer. Anal., 43 (2009), pp. 1203–1219.
[12] J. H. Bramble, *Multigrid Methods*, Chapman and Hall/CRC, Boca Raton, FL, 2019.
[13] J. H. Bramble and X. Zhang, *The analysis of multigrid methods*, in Handbook of Numerical Analysis, Vol. VII, Hanb. Numer. Anal. 7, North-Holland, Amsterdam, 2000, pp. 173–415.
[14] C. Carstensen, *Quasi-interpolation and a posteriori error analysis in finite element methods*, ESAIM Math. Model. Numer. Anal., 33 (1999), pp. 1187–1202.
[15] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, Classics Appl. Math. 40, SIAM, Philadelphia, 2002, https://doi.org/10.1137/1.9780898719208.
[16] P. Daniel, A. Ern, and M. Vohralík, *An adaptive hp-refinement strategy with inexact solvers and computable guaranteed bound on the error reduction factor*, Comput. Methods Appl. Mech. Engrg., 359 (2020), 112607.
[17] P. Daniel and M. Vohralík, *Guaranteed Contraction of Adaptive Inexact hp-Refinement Strategies with Realistic Stopping Criteria*, preprint, https://hal.inria.fr/hal-02486433, 2020.
[18] W. Dörfler, *A convergent adaptive algorithm for Poisson's equation*, SIAM J. Numer. Anal., 33 (1996), pp. 1106–1124, https://doi.org/10.1137/0733054.
[19] M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math., 70 (1995), pp. 163–180.
[20] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Appl. Math. Sci. 95,

Springer, New York, 1994.
[21] W. Hackbusch, *Multigrid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer, Berlin, 1985.
[22] B. Janssen and G. Kanschat, *Adaptive multilevel methods with local smoothing for H^1- and H^{curl}-conforming high order finite element methods*, SIAM J. Sci. Comput., 33 (2011), pp. 2095–2114, https://doi.org/10.1137/090778523.
[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, https://doi.org/10.1137/08073706X.
[24] J. Liesen and Z. Strakos, *Krylov Subspace Methods: Principles and Analysis*, Oxford University Press, Oxford, UK, 2013.
[25] G. Mallik, M. Vohralík, and S. Yousef, *Goal-oriented a posteriori error estimation for conforming and nonconforming approximations with inexact solvers*, J. Comput. Appl. Math., 366 (2020), 112367.
[26] A. Miraçi, J. Papež, and M. Vohralík, *A multilevel algebraic error estimator and the corresponding iterative solver with p-robust behavior*, SIAM J. Numer. Anal., 58 (2020), pp. 2856–2884, https://doi.org/10.1137/19M1275929.
[27] O. Mulita, *Smoothed Adaptive Finite Element Methods*, Ph.D. thesis, International School for Advanced Studies, Trieste, Italy, 2019.
[28] J. Papež, J. Liesen, and Z. Strakoš, *Distribution of the discretization and algebraic error in numerical solution of partial differential equations*, Linear Algebra Appl., 449 (2014), pp. 89–114.
[29] J. Papež, U. Rüde, M. Vohralík, and B. Wohlmuth, *Sharp algebraic and total a posteriori error bounds for h and p finite elements via a multilevel approach: Recovering mass balance in any situation*, Comput. Methods Appl. Mech. Engrg., 371 (2020), 113243.
[30] J. Papež and Z. Strakoš, *On a residual-based a posteriori error estimator for the total error*, IMA J. Numer. Anal., 38 (2018), pp. 1164–1184.
[31] J. Papež, Z. Strakoš, and M. Vohralík, *Estimating and localizing the algebraic and total numerical errors using flux reconstructions*, Numer. Math., 138 (2018), pp. 681–721.
[32] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[33] A. Sartori, N. Giuliani, M. Bardelloni, and L. Heltai, *deal2lkit: A toolkit library for high performance programming in deal.II*, SoftwareX, 7 (2018), pp. 318–327, https://doi.org/10.1016/j.softx.2018.09.004.
[34] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley–Teubner, New York, 1996.
[35] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613, https://doi.org/10.1137/1034116.