SISSA
Scuola
Internazionale
Superiore di
Studi Avanzati

Mathematics Area – PhD course in Mathematical Analysis, Modelling, and Applications

Smoothed Adaptive Finite Element Methods

Candidate: Ornela Mulita **Advisor**: Luca Heltai

Academic year 2018-19

Il presente lavoro costituisce la tesi presentata da Ornela Mulita, sotto la direzione di ricerca del Prof. Luca Heltai, al fine di ottenere l'attestato di ricerca post-universitaria *Doctor Philosophiæ* presso la SISSA, Curriculum in Analisi Matematica, Modelli Applicazioni. Ai sensi dell'art. 1, comma 4, dello Statuto della SISSA pubblicato sulla G.U. no. 36 del 13.02.2012, il predetto attestato è equipollente al titolo di *Dottore di Ricerca in Matematica*.

Trieste, Anno Accademico 2018/2019

Acknowledgements

Foremost, my utmost gratitude goes to my supervisor Prof. Luca Heltai for the support, guidance and his valuable advice during my Ph.D studies. I would like to thank him for the passion and enthusiasm he always shared with me, and for giving me the moral support and freedom to guide me towards independence.

Besides my supervisor, I would like to express my sincere gratitude to Prof. Stefano Giani for his guidance, knowledge and for the useful discussions we shared during our collaboration period in Durham.

I acknowledge SISSA, wherein I had the opportunity to attend several high-level courses and seminars. More importantly, SISSA gifted me with many good colleagues, among which I cannot forget Noè Caruso, Zakia Zainib, Enrico Pasqualetto, Danka Lucic, Emanuele Tasso, Giulia Vescovo, Filippo Riva, Flavia Sarcangelo, Maicol Caponi, Giovanni Alzetta. Most of our professional relationships quickly bloomed into beautiful and sincere friendships. The colleagues I met at SISSA allowed me to experience a warm, encouraging and inspirational work environment where we cared for each other and helped and supported each other, and I feel blessed for this experience.

My deepest gratitude and love to my family –my parents Reis and Kadena, and my sisters Megi and Reina– for always believing in me and supporting my choices unconditionally. Their spiritual presence is endless source of energy for me.

During my years of Ph.D, I've been surrounded by the warmth and support of my closest friends –Paola, Ali, Valentina, Veronica, Carolina, Sofia– who fill my life with joy and beautiful experiences.

Contents

Chapter 1

Introduction and Scope

One of the most challenging issues of numerical mathematics of all times can be formulated as follows: *how can we compute a discrete approximating solution to a given continuous problem with the best accuracy and with a minimal computational cost?* When solving numerically Partial Differential Equations (PDEs) through Finite Element (FEM) schemes, one of the most successful ways to balance computational costs and accuracy is represented by *adaptive mesh-refining techniques*. In attempt of *capturing the "essence"* in the various processes during mesh-refining techniques, this thesis proposes and analyses the Smoothed-Adaptive Finite Element Method (S-AFEM), a new algorithm for Adaptive Finite Element Method (AFEM) which takes its inspiration by the ascending phase of the *V-cycle multigrid (MG) method*.

The overarching goal is to provide rigorous algebraic error analysis, a posteriori error analysis and numerical validation to prove that S-AFEM drastically improves the computational costs of the classical AFEM, by maintaining almost the same accuracy of the final solution.

Characterized by '...a great geometric flexibility, practical implementation and powerful and elegant theory' ([73]), the Finite Element Method (FEM) represents one of the most prominent techniques for the numerical solution of PDEs. In FEM simulations, the domain of a PDE is discretised into a large set of small and simple subdomains (the cells or elements) depending on a size parameter $h > 0$. Typical shapes that are used for the discretisation are triangles, quadrilaterals, tetrahedrons, or hexahedrons. The solution space is constructed by gluing together simpler finite dimensional spaces, defined on a piecewise manner on each cell, and the original problem is solved on this simpler, finite dimensional space, transforming the original PDE into an algebraic system of equations.

Finite Element Analysis (FEA) finds its roots in 1943 in a paper by Courant (see [48]), who used the Ritz method of numerical analysis and minimization of variational calculus (see also the previous work [82]). Meanwhile and independently, the concept of FEM originated during the 1940s from engineers that studied stresses in complex airframe structures. Subsequently, the mathematical foundation was laid down in the mid-1950s with the papers of [88], [4] and [8]. The term "Finite Element" was coined by Clough ([45, 47, 46]). Moving on in

history, in the early 1960s engineers used the method to approximate solutions not only of problems in stress analysis, but also in fluid flow, heat transfer, and other areas. Most commercial FEM software packages originated in the 1970s and the first book on FEM by Zienkiewicz and Cheung was published in 1967 ([95]), followed by the book [86], that laid solid grounds for future development in FEM. Since then, the field of applications has widened steadily and nowadays encompasses a vast diversity of problems in nonlinear solid mechanics, fluid/structure interactions, turbulent flows in industrial or geophysical settings, multicomponent reactive flows, mass transfer in porous media, viscoelastic flows in medical sciences, electromagnetism, wave scattering problems, and option pricing ([52]).

Numerous commercial and academic codes based on the finite element method have been developed over the years, however, in the finite element numerical solution of practical problems of physics or engineering such as, e.g., computational fluid dynamics, elasticity, or semi-conductor device simulation, we often encounter the difficulty that the overall accuracy of the numerical approximation is deteriorated by local singularities arising, e.g., from re-entrant corners, interior or boundary layers, or sharp shock-like fronts ([91]). An obvious remedy is to refine the discretisation near the critical regions, i.e., to consider a bigger number of smaller elements where the solution is less regular. This is the core of the Adaptive Finite Element Method (AFEM), which is a numerical scheme that automatically and iteratively adapts the finite element space until a sufficiently accurate approximate solution is found ([6]).

A few questions closely related to each other arise naturally in the adaptivity context. How to identify the regions with more irregular behaviour? How to obtain a good balance between the refined and unrefined regions such that the overall accuracy is optimal? How to obtain reliable estimates of the accuracy of the computed numerical solution? Can we think of any "intelligent" shortcuts that guarantee the same accuracy of the solution, at a fraction of the computational cost? While former questions have been widely and successfully treated in the literature, in this thesis work we provide a novel and successful method to address the latest.

Classical a priori error estimation theory provides little such information, because it involves the exact solution and the estimates are of asymptotic nature. We briefly discuss the main related results in Chapter 2, Section 2.1. The missing link is given by *a posteriori error estimators*, which hinge exclusively on accessible data, i.e. they extract information from the given problem and the approximate solution, without invoking the exact solution. These are computable quantities that can be used to assess the approximation quality and improve it adaptively. In the 1980s and 1990s a great deal of effort was devoted to the design of a posteriori error estimators, following the pioneering work [9]. Since then, a lot of work has been devoted to them. We refer to [90], [10] and [2] for an overview of the state-of-the-art. We define them and introduce their basic properties in Section 2.3 of Chapter 2. Our focus is on residual-based a posteriori estimators, which are derived by the *residual functional* and provide an equivalent measure

of the discretisation error in a suitable norm.

Let us now briefly explain the AFEM scheme.

The Adaptive Finite Element Method (AFEM) can be represented by successive loops of the steps

$$\mathcal{S}olve \longrightarrow \mathcal{E}stimate \longrightarrow \mathcal{M}ark \longrightarrow \mathcal{R}efine \tag{1.1}$$

to decrease the total discretisation error, by repeating the FEM solution process (step $\mathcal{S}olve$) on a mesh that has been refined (step $\mathcal{R}efine$) on the areas where the a posteriori error analysis has shown that the error is larger (steps $\mathcal{E}stimate$ and $\mathcal{M}ark$) (see, e.g., [39]). Despite their practical success, adaptive processes have been shown to converge, and to exhibit optimal complexity, only recently. A brief literature review of AFEM is presented in Chapter 2, Section 2.3.

The mathematical framework of our work is as follows. We consider linear second-order elliptic boundary value problems (BVPs) whose variational formulation reads: seek $u \in V$ s.t.

$$\mathcal{A}u = f \ \text{ in } V, \tag{1.2}$$

under suitable boundary conditions, where $(V, \|\cdot\|)$ 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. FEM transforms the continuous problem (1.2) in a discrete model of type

$$\mathcal{A}_h u_h = f_h \ \text{ in } V_h, \tag{1.3}$$

where an example of $\mathcal{A}_h$ is given by the restriction of the continuous operator $\mathcal{A}_h := \mathcal{A}\,|_{V_h}$, and where $V_h \subset V$ is the finite dimensional solution space, typically made up by continuous and piecewise polynomial functions. Given $N = \dim(V_h)$, the overall procedure leads to the resolution of a (potentially very large) linear algebraic system of equations of type

$$A\mathbf{u} = \mathbf{f} \ \text{ in } \mathbb{R}^N. \tag{1.4}$$

When solving real-world practical applications, the main difficulty we have to face is that exact (or even near-to-exact) solutions of the algebraic problem (1.4) cannot be computed, we only have available an approximation u_h^c that we obtain in a computer. The total error is therefore written as the sum of two contributions

$$\underbrace{u - u_h^c}_{\text{total error}} = \underbrace{(u - u_h)}_{\text{discretisation error}} + \underbrace{(u_h - u_h^c)}_{\text{algebraic error}}. \tag{1.5}$$

The algebraic error may have a significant effect on the computed approximation, and the solution of the algebraic problem has to be considered an *indivisible part* of the *overall solution process*. This issue is *unavoidably* reflected in adaptive mesh-refining procedures. The common practical assumption in computational

sciences and engineering community has been that in step $\mathcal{S}olve$, one obtains the *exact* solution of the algebraic system (1.4), therefore to replace u_h by u_h^c in the expression of the error estimator during the module $\mathcal{E}stimate$. However, numerical roundoff (cf., e.g., [85]) and the need for solving real world large-scale problems conflict with this assumption and this procedure leads to the urgent challenges that the derivation and application of the a posteriori error estimates should resolve

1. The derivation and the construction of the a posteriori estimator should be done on the available inexact approximation u_h^c. The biggest problem here is that u_h^c does not satisfy the *Galerkin property* ([33]), that is the fundamental property under which classical residual-based a posteriori estimates are derived.

2. The estimation of the total error (1.5) should incorporate the algebraic error $u_h - u_h^c$.

The question of stopping criteria for iterative PDE solvers that account for inexactness of the algebraic approximations is nowadays becoming a widely addressed topic [64, 6, 76]. The main focus of the existing literature concerns ways to estimate the algebraic error and introduce stopping criteria by highlighting the interplay between the discretisation and algebraic error (see, e.g., [6, 74, 75, 76]).

The overarching goal of this thesis work is to exploit and reveal the other part of the coin: rough approximate solutions, with large algebraic error, may still offer large computational savings when used in the correct way. Through our project we successfully push the boundaries to explore this new different direction by introducing and analysing a new algorithm that reduces the overall computational cost of the AFEM algorithm, by providing a fast procedure for the construction of a quasi-optimal mesh sequence which does not require the exact solution of the algebraic problem in the intermediate cycles.

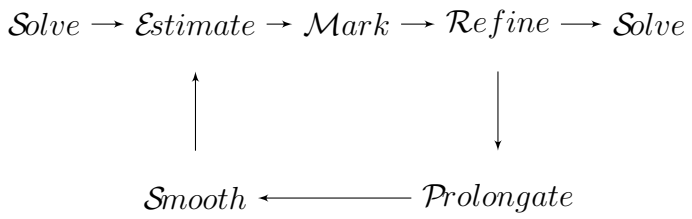
We propose the Smoothed Adaptive Finite Element algorithm (S-AFEM) which takes its inspiration from the ascending phase of the V-cycle multigrid method (see [61, 21]), where a sequence of prolongation and smoothing steps is applied to what is considered an algebraically exact solution at the coarsest level. In MG methods, the prolongation 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. This procedure is based on the principle that 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.

The main difference between the ascending phase of the V-cycle multigrid method and AFEM is that in AFEM the next grid in the sequence is unknown, and requires an exact algebraic solution on the current grid to trigger the $\mathcal{E}stimate$-$\mathcal{M}ark$-$\mathcal{R}efine$ steps. On the other hand, the exact algebraic solutions in the

intermediate cycles are instrumental to the construction of the final grid, and find no other use in the final computations.

Our strategy consists in

1. replacing accurate algebraic solutions in intermediate cycles of the classical AFEM with the application of a prolongation step, followed by a fixed number of few smoothing steps (say three or five)
2. solving exactly the linear algebraic system derived from the discrete problem on the coarsest level and on the finest level
3. executing the $\mathcal{E}stimate$ and $\mathcal{R}efine$ steps on the approximate solutions derived in the intermediate steps.



Even though these intermediate solutions are far from the exact algebraic solution, we will show that their a posteriori error estimation produces a refinement pattern that is substantially equivalent to the one that would be generated by classical AFEM, leading to the same set of cells marked for refinement, at a considerable fraction of the computational cost.

Let us now briefly describe the structure of the thesis and present a panoramic view of the original contributions that we tried to give in this field.

1.1 Structure of the thesis

In Chapter 2 we briefly present the main features of the Finite Element Method (FEM) and of the Adaptive Finite Element Method (AFEM) for the solution of second-order elliptic boundary value problems (BVPs), encompassing their design, basic properties and classical results. The discussion in this chapter includes the literature review of both methods. In Section 2.1 we briefly recall some basic definitions of Sobolev spaces and follow by introducing the Ritz-Galerkin method for the discretisation of symmetric elliptic PDEs of order $2m$, with emphasis on FEM, whose definition arises naturally as a particular class of Ritz-Galerkin methods. A priori error estimates are presented. We will restrict ourselves to Poisson's equation with homogeneous Dirichlet boundary conditions as a model problem, to which we dedicate Section 2.2. Finally, we present a posteriori error estimators and discuss the main features of AFEM in Section 2.3.

We follow with a presentation of the Finite Element Multigrid method in Section 3.1 and in Section 3.2 of Chapter 3. Smoothed-Multilevel Methods can

be introduced as a natural remedy to restore and improve the performance of basic relaxation schemes. The remainder of the chapter is dedicated to the discussion and analysis of Smoothed-Multilevel Methods. Precisely, we consider a generic multilevel context, where a nested sequence of finite element spaces $V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}}$ is given. The repetition of the FEM procedure for any level $k = 1, 2, \ldots, \bar{k}$ gives rise to the associated linear algebraic systems $A_k \mathbf{u}_k = \mathbf{f}_k$ in $\mathbb{R}^{N_k}$, where $N_k = dim(V_k)$. For the algebraic resolution of these systems we apply successive prolongation ($\mathcal{P}rolongate$) and smoothing ($\mathcal{S}mooth$) steps and we prove a series of theorems and results that rigorously estimate the algebraic error propagation between different nested levels following [72]. In particular, for the module $\mathcal{S}mooth$, we consider Richardson smoothing iterations (see e.g., [61, 80]), which we analyse in Subsection 3.3.1.

Chapter 4 extends the a posteriori error analysis previously presented in Section 2.3 to the case when inexact approximations and the algebraic error are taken into consideration. Our attention in Section 4.1 is devoted to the main issues that a posteriori error analysis accounting for the algebraic error has to deal with. In Section 4.2, we are going to prove a bound on the estimator for a generic function in terms of the estimator for the Galerkin solution and the corresponding algebraic error. Finally, in Section 4.3 we slightly touch upon an ongoing work that aims to quantify qualitatively different contributions of different eigenfunctions in the expression of the error estimator.

We devote Chapter 5 to the introduction and description of S-AFEM. We start by providing some motivation through some empirical numerical evidence that justifies the use of S-AFEM, and then connect it to the theoretical results on the error propagation that we proved in Section 3.4. In Section 5.3 we provide numerical validation for our method presenting two-dimensional examples in Subsection 5.3.1 and three-dimensional examples Subsection 5.3.2. We show a comparison of the computational cost associated to the classical AFEM and to the smoothed AFEM for the presented examples in Subsection 5.3.3.

In Section 5.4 we present different variants of our algorithm S-AFEM, where different smoothers are considered for the intermediate cycles, (respectively Richardson iteration, the CG method, and the GMRES method) and we investigate the accuracy and of S-AFEM for high order finite element discretisations. We provide several numerical evidences that S-AFEM turns out to be a good strategy also for higher order finite element discretisations, and one could use directly the CG method (or, alternatively,the GMRES method) as smoothers for the intermediate iterations. Our numerical evidences show that two smoothing iterations are enough for the two dimensional case, and around five smoothing iterations are enough for the three dimensional case, independently on the polynomial degree of the finite element approximation.

In Section 5.5 we show that S-AFEM strategy (for different FEM degrees) works well also for more complex and non symmetric problems, by providing and example of a transport-diffusion two-dimensional problem. Finally, Section 5.6 briefly describes the main conclusions of the thesis.

1.2 Main results and contributions

S-AFEM is an entirely novel idea that originated during this thesis project. Its novelty and originality is given not only by its unique strategy and analysis, but also by an approach to adaptive algorithms with an awareness that is missing in the current literature. Our work has lead us to some fascinating conclusions and beautiful discoveries. The key results and contributions in this dissertation are presented in Chapter 3 through Chapter 5 and include the following.

- One of the key findings of this research program has been 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, justifying the formalisation of a new *Smoothed Adaptive Finite Element* algorithm (S-AFEM), where the exact algebraic solution in intermediate steps is replaced by the application of a prolongation step ($\mathcal{P}rolongate$), followed by a fixed number of smoothing steps ($\mathcal{S}mooth$). The principal motivation is that we've found out that classical a posteriori error estimators are not sensitive to low frequencies in the solution, and that their application to very inaccurate approximate solutions in intermediate cycles – only capturing high frequency oscillations – would produce an equally good grid refinement pattern.

- We have introduced and analysed the Smoothed-Multilevel Methods, where, in the context of a nested sequence of finite element spaces corresponding to nested grids, we solve exactly on the coarsest grid (reaching convergence in all components), and then perform a sequence of prolongations followed by a fixed number of smoothing steps, to improve convergence in the finer grids, under the assumptions that lower frequencies have already been taken care of in the previous levels. This is the core idea behind S-AFEM. We prove a series of theorems and results that rigorously estimate the algebraic error propagation between different nested levels, which shows that the algebraic error is made up by small contributions given by the accumulation of low frequency terms, which have in general a smaller influence on the estimator.

- A parallel exciting challenge of our research has been to show how the algebraic error derived in Theorem 3.4.4 relates to the $\mathcal{E}stimate$ phase of AFEM. This is found in discussion in Section 4.2 and in Theorem 4.2.2, which proves an upper bound on the estimator for generic functions of the finite element space in terms of the estimator for the Galerkin approximation and the algebraic error, up to some oscillating terms of the data.

- We provide several numerical evidences for the case of Poisson's equation in two and three dimensions with different data realized using a custom `C++` code based on the `deal.II` library ([13, 3]), and on the `deal2lkit` library ([81]) that show that S-AFEM algorithm is competitive in cost and accuracy. For three dimensional problems, the speedup in the intermediate steps is in the hundreds, and 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 solutions is comparable.

- Another very interesting conclusion has been that only a few smoothing iterations, *only three*, were enough for the estimator to produce the same set of marked cells for refinement at each cycle. We discuss this theory and provide numerical evidences in Chapter 4 and Chapter 5.

- Another key conclusion been that different smoothers (for instance Richardson iteration, the CG method and the GMRES method) work equally well as smoother candidates for the intermediate levels. A particular attention has to be paid to the Richardson iteration when used as a smoother, that has to come along with the investigation and the subsequent choice of the optimal relaxation parameter.

- As another relevant conclusion, we show that S-AFEM turns out to be a good method not only for piecewise bi-linear FEM discretisations, but also for high order FEM discretisations, for instance when we take the FEM polynomial degree $deg = 2, 3, 4, 5$.

- Finally, we provide a large variety of experiments to validate numerically our strategy, that include classical two and three dimensional problems that are used to benchmark adaptive finite elements (for instance the peak problem and the corner problem), but also a two dimensional non symmetric problem of diffusion-transport type. In all results of this large variety of numerical experiments, the accuracy of the final approximation generated by S-AFEM is almost the same to the one that would be generated by classical AFEM, at a fraction of the computational cost, making S-AFEM a highly valuable algorithm for many practical and realistic applications.

- Motivated by the theoretical results and the numerical evidence, 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 with a chosen smoother. 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.

Finally, some interesting questions for future investigations are as follows.

- Although we have evidences that this technique works well for more complex second-order elliptic problems, such as non symmetric problems of transport-diffusion type, there are theoretical results that need to be developed to validate theoretically the strategy also for these more complex situations.

- An ongoing and future project introduces local smoothing in the parts of the domain in need for refinement to further improve the speedup of the algorithm.

List of included papers:

(I) O. Mulita, S. Giani,and L. Heltai, *Quasi-optimal mesh sequence construction through Smoothed Adaptive Finite Element Methods*, 2019, submitted to SIAM Journal on Scientific Computing, arXiv:1905.06924.

(II) O. Mulita, *Dominating terms in the edge contribution of classical residual based a posteriori error estimators*, 2019, *in progress.*

The new results contained in the above articles are distributed along the thesis according to the following scheme:

– Algorithm 4, Subsection 3.3.2, Section 3.4, 3.4.1, Definition 3.4.1, Theorem 3.4.2, Theorem 3.4.3, Subsection 3.4.2, Assumption 3.4.1, 3.4.4, Remark 3.4.1.

– Section 4.2, Theorem 4.2.1, Remark 4.2.2, Section 4.3.

– Section 5.1, Section 5.2, Algorithm 5, Numerical Evidences in Subsection 5.3.1 and in Subsection 5.3.2, Subsection 5.3.3.

– Section 5.4 and all numerical evidences included in the respective subsections.

– Section 5.5 and all numerical evidences included.

Chapter 2

Prolegomena

For a solid understanding of the basics behind this thesis project, we dedicate this chapter to the basics of the Finite Element Method (FEM) and of the Adaptive Finite Element Method (AFEM) for the solution of second-order elliptic boundary value problems (BVPs). Classical a priori error estimates described in Section 2.1 yield useful information on the asymptotic error behaviour. The price to pay for this information is in terms of regularity conditions of the solution, which are unfortunately not satisfied in the presence of singularities as introduced above. These considerations highlight the need for an error estimator which can be extracted in *a posteriori* fashion from the computed numerical solution and the given data of the problem. In Section 2.2 we introduce our model problem and in Section 2.3 we discuss a posteriori error estimators, as well as mesh refining strategies.

2.1 The Finite Element Method for Linear Elliptic second-order Boundary Value Problems

Let Ω be a open bounded domain in $\mathbb{R}^d$, where $d = 1, 2, 3$, with piecewise smooth boundary $\Gamma := \partial\Omega$. For a positive integer m, the Sobolev space $H^m(\Omega)$ ([1]) is the space of square integrable functions whose weak derivative up to order m is also integrable. $H^m(\Omega)$ is a Hilbert space equipped with the norm $\|u\|_{H^m(\Omega)}$ defined by

$$\|u\|_{H^m(\Omega)} = \left(\sum_{|\alpha| \leq m} \left\| \frac{\partial^\alpha u}{\partial x^\alpha} \right\|^2_{L_2(\Omega)} \right)^{1/2} . \tag{2.1}$$

We will also consider the seminorm

$$|u|_{H^m(\Omega)} = \left(\sum_{|\alpha| = m} \left\| \frac{\partial^\alpha u}{\partial x^\alpha} \right\|^2_{L_2(\Omega)} \right)^{1/2} . \tag{2.2}$$

Using Sobolev spaces we can represent a large class of symmetric elliptic BVPs of order $2m$ in the abstract form:

Find $u \in V \subset H^m(\Omega)$, where V is a closed subspace of $H^m(\Omega)$, such that

$$a(u,v) = l(v), \ \ \forall\, v \in V, \tag{2.3}$$

where $l : V \to \mathbb{R}$ is a bounded linear functional on V and $a(\cdot,\cdot)$ is a symmetric bilinear form that is bounded

$$|a(u,v)| \leq C_1{}^1 \|u\|_{H^m(\Omega)} \|v\|_{H^m(\Omega)}, \ \ \forall u, v \in V, \tag{2.4}$$

and V-elliptic

$$|a(v,v)| \geq C_2 \|v\|^2_{H^m(\Omega)}, \ \ \forall v \in V. \tag{2.5}$$

For symmetric problems, conditions (2.4) and (2.5) imply that the bilinear form $a(\cdot,\cdot)$ defines an inner product on V that induces the norm $\|\cdot\|_a := (a(\cdot,\cdot))^{1/2}$, which is equivalent to the Sobolev norm $\|\cdot\|_{H^m(\Omega)}$. The Riesz Representation Theorem ([78]) and conditions (2.4) and (2.5) guarantee existence and uniqueness of the solution of (2.3).

The Ritz-Galerkin method for the discretisation of (2.3) reads:

Given $l \in V^\star$, find $\tilde{u} \in \tilde{V}$ such that

$$a(\tilde{u},\tilde{v}) = l(\tilde{v}), \ \ \forall \tilde{v} \in \tilde{V}, \tag{2.6}$$

where $\tilde{V}$ is a finite dimensional subspace of V. The approximate solution $\tilde{u}$ is called a Galerkin solution.

By subtracting (2.6) from (2.3) applied to functions that lie in $\tilde{V}$, we get the Galerkin orthogonality property

$$a(u-\tilde{u},\tilde{v}) = 0, \ \ \forall \tilde{v} \in \tilde{V}, \tag{2.7}$$

which implies that

$$\|u-\tilde{u}\|_a = \inf_{\tilde{v}\in\tilde{V}} \|u-\tilde{v}\|_a. \tag{2.8}$$

By combining together (2.8), (2.4) and (2.5) we get Cea's Lemma

$$\|u-\tilde{u}\|_{H^m(\Omega)} \leq \frac{C_1}{C_2} \inf_{\tilde{v}\in\tilde{V}} \|u-\tilde{v}\|_{H^m(\Omega)}, \tag{2.9}$$

that is, the error of the Galerkin solution is quasi-optimal in the Sobolev norm. The solution of the approximation problem (2.9) depends on the regularity of the exact solution u and the nature of the space $\tilde{V}$.

We consider the finite element spaces which are introduced and described in Subsection 2.1.

We briefly recall the main regularity results. In particular, if the boundary $\partial\Omega$ is smooth and the homogeneous boundary conditions are also smooth, then the

[1] We use C, with or without subscript, to represent a generic positive constant that takes different values in different situations.

solution of the elliptic boundary problem (2.3) obeys the classical *Shift Theorem* ([60, 34]) , which states that if the right-hand side of the equation belongs to the Sobolev space $H^l(\Omega)$, then the solution of a $2m$th-order elliptic boundary problem belongs to the Sobolev space $H^{2m+l}(\Omega)$. This theorem does not hold for domains with piecewise smooth boundary in general. For instance, it does not hold true when the types of boundary condition change abruptly.

For two dimensional problems, the vertices of Ω and the points where the boundary condition changes type are singular points. Away from these singular points, the Shift Theorem is valid. For a $2m^{th}$ order problem, the regular part of the solution belongs to the Sobolev space $H^{2m+k}(\Omega)$ if the right-hand side function belongs to $H^k(\Omega)$, and the singular part of the solution is a linear combination of special functions with less regularity. The situation in three dimensions is more complicated because of the presence of edge singularities, vertex singularities and edge-vertex singularities and it remains an active area of research ([34]).

Finite Element Methods. A successful and widely used class of Ritz-Galerkin methods is represented by the Finite Element Methods (FEMs). In FEM, the domain of a partial differential equation (PDE) is discretized in a large set of small and simple domains (the cells or elements) depending on a size parameter $h > 0$. Typical shapes that are used for the discretisation are triangles, quadrilaterals, tetrahedrons, or hexahedrons. The solution space $\tilde{V} \subset V$ is constructed by gluing together simpler finite dimensional spaces, defined on a piecewise manner on each cell, and the original problem is solved on this simpler, finite dimensional space, transforming the original PDE into an algebraic system of equations, see, e.g., some of the numerous books dedicated to FEM [11, 43, 96, 42, 33, 52, 63, 34].

We recall the formal definition of finite element, according to [43], [33]. A d-dimensional finite element is a triple $(T, \mathbb{P}_T, \mathbb{N}_T)$, where T is a closed bounded subset of $\mathbb{R}^d$ with nonempty interior and piecewise smooth boundary, $\mathbb{P}_T$ is a finite dimensional vector space of functions defined on T, and $\mathbb{N}_T$) is a basis of the dual space $\mathbb{P}'_T$. Functions in $\mathbb{P}_T$ are called shape functions, while functionals in $\mathbb{N}_T$ are called nodal variables or degrees of freedom. Typically, the space $\mathbb{P}_T$ is taken as $\mathbb{P}^k_T,\ k > 0$: the space of polynomials of order $k > 0$ defined over T.

In this work, we restrict to considering polyhedral domains. We introduce the basic notions of *partition* and *triangulation* of the computational domain Ω.

A *partition* $\mathcal{P}$ of Ω is a collection of subdomains of Ω such that

$(\mathcal{P}_1)$ $\bar{\Omega} = \bigcup_{T\in\mathcal{P}} \bar{T}$

$(\mathcal{P}_2)$ each element T is a non-empty, open subset of Ω

$(\mathcal{P}_3)$ $T \cap T' = \emptyset$, if $T, T' \in \mathcal{P}$ and $T \neq T'$

$(\mathcal{P}_4)$ each element T is a polyhedron

The subdomains T are usually called *cells* or *elements*. We will consider a family of *triangulations* $\mathcal{T}_h$ $h > 0$ of Ω, which are partitions satisfying the following condition

- *Admissibility*: any two subdomains T, T' in $\mathcal{P}$ are either disjoint, or have an edge/face in common or share a vertex.

The notions of partition and triangulation are different for $d > 1$, while for $d = 1$ every partition is a triangulation. We will consider triangulations consisting of triangles or convex quadrilaterals in two dimensions and tetrahedrons or convex hexahedrons in three dimensions. The shape regularity of the subdomains is measured by the aspect ratio. For triangles (or tetrahedrons), the aspect ratio is measured by the parameter γ_T

$$\gamma_T := \frac{h_T}{\rho_T}, \tag{2.10}$$

where h_T is the diameter of T and ρ_T is the diameter of the largest ball inscribed into T. For convex quadrilaterals (or hexahedrons), the aspect ratio is measured by γ_T defined in (2.10) and by the parameter

$$\sigma_T := \max\left\{ \frac{|e_1|}{|e_2|} : e_1, e_2 \text{ any two edges of } T \right\}. \tag{2.11}$$

We will refer to the number $\max(\gamma_T, \sigma_T)$ as the aspect ratio of the convex quadrilateral (hexahedron). We will consider family of triangulations that satisfy the following condition:

- *Shape regularity:* there exists a positive constant $\gamma_{\mathcal{T}} < \infty$ that bounds the aspect ratios of all subdomains.

Next, we define the finite element approximation spaces. Let $\mathcal{T}$ be a triangulation of Ω, and a finite element $(\bar{T}, \mathbb{P}_{\bar{T}}, \mathbb{N}_{\bar{T}})$ be associated with each subdomain $T \in \mathcal{T}$. We define the corresponding finite element space to be

$$\begin{aligned} FE_{\mathcal{T}} = \{ & v \in L_2(\Omega) : v\,|_{\bar{T}} \in \mathcal{T} \ \ \forall T \in \mathcal{T}, \\ & \text{and } v\,|_T \text{ and } v\,|_{T'} \text{ share the same nodal values on } \bar{T} \cap \bar{T}' \ \} \end{aligned} \tag{2.12}$$

A $\mathcal{C}^r$ finite element space is a finite element space $FE_{\mathcal{T}} \subset \mathcal{C}^r(\Omega)$. In this case, it is automatically a subspace of the Sobolev space $H^{r+1}(\Omega)$ and therefore appropriate for elliptic boundary value problems of order $2(r+1)$.

There are two types of error estimation procedures available for the discretisation error $u - \tilde{u}$. A priori error estimators provide information on the asymptotic behavior of the discretisation errors but are not designed to give an actual error estimate for a given triangulation. In contrast, a posteriori error estimators employ the finite element solution itself to derive estimates of the actual solution errors. They are also used to steer adaptive schemes where either the mesh is locally refined (h-version) or the polynomial degree is raised (p-method) ([58]).

Let $\mathcal{T}$ be a triangulation of Ω. Consider a finite element space $FE_{\mathcal{T}}$ and a finite element $(\bar{T}, \mathbb{P}_{\bar{T}}, \mathbb{N}_{\bar{T}})$ be associated with each subdomain $T \in \mathcal{T}$. Assume

that the resulting finite element space is a subspace of $\mathcal{C}^{m-1}(\bar{\Omega}) \subset H^m(\Omega)$. By imposing appropriate boundary conditions, we can obtain a subspace $V_{\mathcal{T}}$ of $FE_{\mathcal{T}}$ such that $V_{\mathcal{T}} \subset V$.

We will focus on second-order problems. Assume that $\Omega \subset \mathbb{R}^2$ and that the right-hand side of the elliptic boundary value is in the Sobolev space $L_2(\Omega)$. Then $u \in H^{1+\alpha(T)}(T)$ for each $T \in \mathcal{T}$, where $\alpha(T) \in (0,1]$ and $\alpha(T) = 1$, for T away from the singular points ([49]). Let $V \subset H^1(\Omega)$ be defined by homogeneous Dirichlet boundary conditions on $\Gamma \subset \partial\Omega$ and assume that $\mathcal{T}$ is a triangulation of Ω such that the resulting finite element space $FE_{\mathcal{T}}$ is a subspace of $C^0(\Omega) \subset H^1(\Omega)$ (this can be achieved, for e.g., by considering Lagrange finite elements ([34]). We take $V_{\mathcal{T}} = V \cap FE_{\mathcal{T}}$. The proofs adopt the use of the nodal interpolation operator ([34]) and corresponding estimates.

The following *a priori* discretisation error estimate holds true

$$\|u - u_{\mathcal{T}}\|_{H^1(\Omega)} \leq C \left(\sum_{T \in \mathcal{T}} (h_T)^{2\alpha(T)} |u|^2_{H^{1+\alpha(T)}(T)} \right)^{1/2}, \tag{2.13}$$

where h_T is the diameter of T and C is a positive constant depending only on the aspect ratios of the cells T.

In particular, if $\{\mathcal{T}_i, i \in I\}$ is a shape regular family of triangulations, and the solution u of (2.3) belongs to the Sobolev space $H^{1+\alpha}(\Omega)$ for some $\alpha \in (0,1]$, then (2.13) implies that

$$\|u - u_{\mathcal{T}_i}\|_{H^1(\Omega)} \leq C h_i^{\alpha} |u|^2_{H^{1+\alpha}(\Omega)}, \tag{2.14}$$

where $h_i = \max_{T \in \mathcal{T}_i} h_T$ is the mesh size of $\mathcal{T}_i$ and C is independent of $i \in I$. Furthermore, the following estimate in the $L_2(\Omega)$-norm holds true

$$\|u - u_{\mathcal{T}_i}\|_{L_2(\Omega)} \leq C h_i^{2\alpha} |u|^2_{H^{1+\alpha}(\Omega)} \tag{2.15}$$

where C is also independent of $i \in I$. The above two dimensional results also hold true for three dimensional elements if the solution $u \in H^{1+\alpha}(\Omega)$ where $1/2 < \alpha \leq 1$, since the nodal interpolation operator is well-defined by the Sobolev embedding theorem [35]. A case where this is verified is when $\Gamma = \partial\Omega$.

2.2 Model Problem

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

$$-\Delta u = f \ \text{ in } \ \Omega \ \text{ and } \ u = 0 \text{ on } \Gamma := \partial\Omega, \tag{2.16}$$

where $f \in L^2(\Omega)$ is a given source term. We use the standard notation for norms and scalar products in Lebesgue and Sobolev spaces (cf. [1]) : for $u \in H_0^1(\Omega)$ and $\omega \subset \Omega$, we write $|u|_{1,\omega} := (\int_\omega |\nabla u|^2)^{1/2}$ and denote by $(\cdot,\cdot)_\omega$ the $L^2(\omega)$- scalar

product with corresponding norm $\|\cdot\|_\omega$. For $\omega = \Omega$, we omit the corresponding subscripts. The weak form of (2.16) is to find $u \in H_0^1(\Omega)$ s.t.

$$(\nabla u, \nabla v) = (f, v), \quad \forall v \in H_0^1(\Omega). \tag{2.17}$$

We consider a shape regular family of triangulations $\{\mathcal{T}_h\}_h$ of Ω depending on a parameter $h > 0$ with shape regularity parameter $C_{\mathcal{T}_h}$. We will consider triangulations consisting of triangles or convex quadrilaterals in two dimensions, and tetrahedrons or convex hexahedrons in three dimensions.

We denote by z the nodes of $\mathcal{T}_h$ (i.e. the vertices of the cells) and by $\mathcal{N}_h$ the set of all nodes, while $\mathcal{N}_{h,int} := \mathcal{N}_h \setminus \Gamma$ denotes the set of the free nodes. The set of all edges/faces E of the cells is denoted by $\mathcal{E}_h$ and similarly, $\mathcal{E}_{h,int} := \mathcal{E}_h \setminus \Gamma$ is the set of internal edges/faces. Let φ_z be the nodal basis function associated to a node $z \in \mathcal{N}_h$ with support ω_z, which is equal to the patch $\omega_z = \cup\{T \in \mathcal{T}_h | z \in T\}$. We use the Courant finite element space $V_h := \text{span}\{\varphi_z | z \in \mathcal{N}_{h,int}\} \subset H_0^1(\Omega)$. The Galerkin solution $u_h \in V_h$ is defined by the discrete system

$$(\nabla u_h, \nabla v_h) = (f, v_h), \ \ \forall v_h \in V_h. \tag{2.18}$$

2.3 Local Mesh Refinement

The Adaptive Finite Element Method (AFEM) consists of successive loops of the steps

$$\mathcal{S}olve \longrightarrow \mathcal{E}stimate \longrightarrow \mathcal{M}ark \longrightarrow \mathcal{R}efine \tag{2.19}$$

to decrease the total discretisation error, by repeating the FEM solution process ($\mathcal{S}olve$) on a mesh that has been refined ($\mathcal{R}efine$) on the areas where the a-posteriori analysis has shown that the error is larger ($\mathcal{E}stimate$ and $\mathcal{M}ark$).

FEM provides numerical solutions to the above problem in the discrete finite dimensional space $V_h \subset V$, and transforms the continuous problem above in a discrete model of type $\mathcal{A}_h u_h = f_h$ in V_h under suitable boundary conditions, where $\mathcal{A}_h = \mathcal{A}\,|_{V_h}$. The overall procedure leads to the resolution of a (potentially very large) linear algebraic system of equations of type $A\mathbf{u} = \mathbf{f}$ in $\mathbb{R}^N$, where $N = \dim(V_h)$. The standard AFEM algorithm (following [39]) can be summarised in the following steps:

This procedure solves for any level $k = 1, 2, \ldots, \bar{k}$ the following discrete problems: seek $u_k \in V_k$ s.t $\mathcal{A}_k u_k = f_k$ in V_k under suitable boundary conditions, where $\mathcal{A}_k : V_k \to V_k^\star$, $\mathcal{A}_k := \mathcal{A}\,|_{V_k}$. The finite element spaces are nested $V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}}$ and the inequality $N_1 < N_2 < \cdots < N_{\bar{k}}$ holds for the relative dimensions of the finite element spaces.

Next, we briefly describe the modules $\mathcal{E}stimate$, $\mathcal{M}ark$ and $\mathcal{R}efine$. We start by providing an insight into classical a posteriori error estimation theory for the module $\mathcal{E}stimate$. Our focus is on residual-based a posteriori error estimators, which were historically defined and derived in terms of the Galerkin approximation.

Algorithm 1: AFEM Algorithm

Input: initial mesh $\mathcal{T}_1$

Loop: for $k = 1, 2, \ldots, \bar{k}$ do steps $1. - 4.$

1. $\mathcal{S}olve$: $A_k \mathbf{u}_k = \mathbf{f}_k$ in $\mathbb{R}^{N_k}$, where $dim(V_k) = N_k$, based on $\mathcal{T}_k$.
2. $\mathcal{E}stimate$: Compute $\eta_T(u_k)$ for all $T \in \mathcal{T}_k$.
3. $\mathcal{M}ark$: Choose set of cells to refine $\mathcal{M}_k \subset \mathcal{T}_k$ based on $\eta_T(u_k)$.
4. $\mathcal{R}efine$: Generate new mesh $\mathcal{T}_{k+1}$ by refinement of the cells in $\mathcal{M}_k$.

Output: nested sequence of meshes $\mathcal{T}_k$, approximations u_k, and local estimators $\eta_T(u_k)$, for $k = 1, \ldots, \bar{k} - 1$, and final problem-adapted approximation $u_{\bar{k}}$.

The concepts of Error Estimator, Efficiency and Reliability. Classical a posteriori error estimation theory focuses on measuring a suitable norm of the discretisation error e_h by providing upper and lower bounds in terms of a posteriori error estimators.

By definition, "regarded as an approximation to an (unknown) suitable norm of the discretisation error $\|e_h\|$, a (computable) quantity $\eta(u_h)$ is called a *posteriori error estimator* if it is a function of the known domain Ω, its boundary Γ, the right-hand side f as well as of the discrete solution u_h, or the underlying triangulation" ([34]).

There are two main requirements that an a posteriori error estimator $\eta(u_h)$ should satisfy, apart from being easy and cheap to compute: it has to be *reliable* in the sense of an upper bound

$$\|e_h\| \leq C_{\text{rel}} \eta(u_h) + h.o.t._{rel}, \tag{2.20}$$

and *efficient* in the sense of a lower bound

$$\eta(u_h) \leq C_{\text{eff}} \|e_h\| + h.o.t._{eff}. \tag{2.21}$$

The multiplicative constants C_{rel} and C_{eff} are independent on the mesh size and h.o.t. refer to high order terms which are due to oscillations of the right-hand side f, and which in generic cases are of magnitudes smaller than $\|e_h\|$.

Residual-based a Posteriori Error Estimators. Standard residual-based a posteriori error estimators are the most widely used for adaptive techniques. They were first introduced in the context of FEM by Babuška and Rheinboldt in [9] and they have been thereafter widely studied in the literature; we refer, e.g., to the books [90] and [2].

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

with corresponding dual norm

$$\|\mathcal{R}\{u_h\}\|_\star := \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{\mathcal{R}\{u_h\}(v)}{|v|_1} = \sup_{v\in H_0^1(\Omega)\setminus\{0\}} \frac{(f,v)-a(u_h,v)}{|v|_1}. \tag{2.22}$$

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. The main tool exploited in the derivation is the Galerkin orthogonality (2.7). We will come back to the a posteriori error estimation theory in Chapter 4.

The module $\mathcal{M}ark$. Given a standard residual based a posteriori estimator expressed as a sum over all elements $T \in \mathcal{T}_k$

$$\eta = \left\{ \sum_{T\in\mathcal{T}_k} \eta_T^2 \right\}^{1/2}, \tag{2.23}$$

the marking strategy is an algorithm that, for any level k selects for refinement the subset of elements

$$\mathcal{M}_k := \{T \in \mathcal{T}_k : \eta_T \geq L\}, \tag{2.24}$$

where L is a *treshold error*. Typical examples for the computation of L are the *maximum criterion*, which is defined as the largest value such that

$$L := \Theta \max\{\eta_T : T \in \mathcal{T}\}, \tag{2.25}$$

or the *bulk criterion* ([50]), where L is the largest value such that

$$\Theta^2 \sum_{T\in\mathcal{T}_k} \eta_T^2 \leq \sum_{T\in\mathcal{M}_k} \eta_T^2. \tag{2.26}$$

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.

"Hanging nodes". After the refinement procedure, consistency of the finite element functions between the refined and the coarse part of the mesh must be ensured. We use the "hanging nodes" technique, which is particularly favorable to deal with all-quadrilateral and all-hexahedral meshes (cf. [13]). Consistence is ensured by adding additional algebraic constraints to the linear system. The only restriction we require mostly for algorithmic reasons is that each face of a cell is divided at most once ("one-irregular" mesh). This can be ensured by refining a few additional cells in each cycle as necessary and has no significant detrimental effects on the complexity of a mesh.

Literature on AFEM. One of the first AFEM analysis was provided by Babuška and Vogelius in [37] for linear symmetric elliptic problems in one dimension. Despite their practical success, adaptive processes have been shown to converge, and to exhibit optimal complexity, only recently and for linear elliptic PDE. The first multidimensional convergence result was given by Dörfler in [50], which introduced the marking criterion and proved linear convergence of the error for some FEM for the Poisson problem up to some tolerance. [69] extended the analysis and included data approximation to prove convergence of a practical adaptive algorithm. The first complexity result was given by Binev, Dehmen, and DeVore in [17], which first proved convergence with optimal rates for the Poisson problem. [84] proved convergence with optimal rates for the adaptive algorithm. [41] included standard newest vertex bisection as mesh refinement into the mathematical analysis. Until then, only variations of FEM for the Poisson model problem with homogeneous Dirichlet boundary conditions were analyzed in the literature.

Independently, [56] and [57] developed the analysis for integral equations and proved convergence with optimal rates for standard boundary element methods (BEMs). [7] proved optimal convergence rates for FEM for the Poisson problem with general boundary conditions. Finally, [55] concluded the theory for general second order linear elliptic PDEs.

The work [39] collects all the mentioned seminal works in a unifying and abstract framework. The work identifies the *axioms of adaptivity* and proves optimal rates for any problem that fits in the abstract setting. The latter covers the existing literature on rate optimality for conforming FEM (also the known results for nonlinear problems) and BEM as well as nonconforming and mixed FEM. With some additional (resp.relaxed) axioms, the abstract framework [39] covers also inexact solvers and other types of error estimators.

In recent years, convergence, convergence rate, and complexity results have been incrementally improved for AFEM applied to second-order elliptic problems for conforming finite element methods. For a detailed description on AFEM we refer to the books [51, 70, 58, 73, 14].

Chapter 3

Multilevel Methods

3.1 Introduction

This chapter presents the essential ideas behind the Finite Element Multigrid Method (cf. Section 3.2). The aim is to present the multigrid (MG) method as a representative example of a larger family of methods, referred to by its chief developer Achi Brandt as *multilevel methods* ([36]). This is also required to understand the motivation and theory behind Smoothed-Multilevel Methods (cf. Section 3.3) and our Smoothed-Adaptive Finite Element Method (cf. Chapter 5). Multigrid techniques exploit discretisations with different mesh sizes of a given problem to obtain optimal convergence from relaxation techniques. For the resolution of elliptic PDEs, the MG method turns out to be very effective in providing accurate algebraic solutions in $\mathcal{O}(N)$ time, where N is the dimension of the corresponding algebraic system.

At the foundation of these techniques is the basic and powerful principle of divide and conquer. Though most relaxation-type iterative processes, such as Gauss-Seidel, Richardson, etc, may converge slowly for typical problems, it can be noticed that the components of the errors (or residuals) in the directions of the eigenvectors of the iteration matrix corresponding to large eigenvalues are damped very rapidly. These eigenvectors are known as the oscillatory modes or high-frequency modes. Other components, associated with low-frequency or smooth modes, are difficult to damp with standard relaxation. This causes the observed slow down of all basic iterative methods. However, many of these modes (say half) are mapped naturally into high-frequency modes on a coarser mesh. Hence the idea of moving to a coarser mesh to eliminate the corresponding error components. The process can obviously be repeated with the help of recursion, using a hierarchy of meshes. Some of the modes which were smooth on the fine grid, become oscillatory. At the same time the oscillatory modes on the fine mesh are no longer represented on the coarse mesh. The iteration fails to make progress on the fine grid when the only components left are those associated with the smooth modes. Multigrid strategies do not attempt to eliminate these components on the fine grid. Instead, they first move down to a coarser grid where smooth modes are translated into oscillatory ones. This is done by going

back and forth between different grids ([80]).

The review material presented in this chapter is based on several sources. Foremost among these are the references [18, 22, 25, 80, 36] and [33]. Nowadays, the body of multigrid literature is vast and continues to grow at an astonishing rate. However, there are several classical text books that we recommend as detailed volumes. In particular, we recommend [36], which provides an excellent and easy to read introduction to the subject. This tutorial includes enough theory to understand how multigrid methods work. Other classical books are [68, 61, 62, 92], [36] and [87].

Early works on multigrid methods date back to the 1960s and include the papers by [19, 53, 54, 12, 65]. However, multigrid methods have seen much of their modern development in the 1970s and early 1980s, essentially under the pioneering work of Brandt ([27, 28, 29]). Brandt played a key role in promoting the use of multigrid methods by establishing their overwhelming superiority over existing techniques for elliptic PDEs and by introducing many new concepts which are now widely use in MG literature ([80]). Algebraic multigrid (AMG) methods were later developed to attempt to obtain similar performance. These methods don't use any partial differential equation nor geometrical problem background to construct the multilevel hierarchy. Instead, they construct their hierarchy of operators directly from the system matrix. In classical AMG, the levels of the hierarchy are simply subsets of unknowns without any geometric interpretation. These methods were introduced in [31] and analyzed in a number of papers (see e.g., [30, 79]). Today MG methods are still among the most efficient techniques available for solving Elliptic PDEs on regularly structured problems.

The remainder of the chapter (cf. Section 3.3 and Section 3.4) is dedicated to the discussion and analysis of Smoothed-Multilevel Methods. Inspired by the ascending phase of the V-cycle multigrid method, the strategy of Smoothed-Multilevel Methods is to solve exactly on a coarse grid (reaching convergence in all components), and then perform a sequence of prolongations followed by a fixed number of smoothing steps, to improve convergence in the finer grids, under the assumptions that lower frequencies have already been taken care of in the previous levels. In Section 3.3 and Section 3.4, we introduce them and provide rigorous algebraic error analysis following [72].

3.2 The Finite Element Multigrid Method

The multigrid method provides an optimal order algorithm for solving elliptic BVPs. The error bounds of the approximate solution obtained from the full multigrid algorithm are comparable to the theoretical bounds of the error in the finite element method (cf. Theorem 3.2.2), while the amount of computational work involved is proportional only to the number of unknowns in the discretised equations (cf. Theorem 3.2.3) ([33]).

Geometric multigrid methods use a hierarchy of mesh levels. On each level, an approximate solver –a so called smoother– is employed, which reduces the error in the high frequency, and then an approximation on a coarser level is used to reduce the error in lower frequencies. This is done down to the coarsest level, where we assume that the solution process is exact and cheap. The method has two main features: smoothing on the current grid and error correction on a coarser grid. The smoothing step has the effect of damping out the oscillatory part of the error. The smooth part of the error can then be accurately corrected on the coarser grid ([33]).

Recall the classical a priori error estimate for piecewise linear finite elements (cf. [33])

$$\|u - u_k\|_{H^s\Omega} \leq Ch_k^{2-s}\|u\|_{H^2\Omega}, \ s = 0, 1 \ , \forall k = 1, 2, \ldots, \tag{3.1}$$

where C is a generic positive constant independent of k. Let $N_k = dim V_k$. The goal of the multigrid method is to calculate $\hat{u}_k \in V_k$ in $\mathcal{O}(N_k)$ operations such that

$$\|u_k - \hat{u}_k\|_{H^s\Omega} \leq Ch_k^{2-s}\|u\|_{H^2\Omega}, \ s = 0, 1 \ , \forall k = 1, 2, \ldots. \tag{3.2}$$

The $\mathcal{O}(N_k)$ operation count means that the multigrid method is asymptotically optimal.

The discussion in this chapter is based on the papers [22, 24] and on the books [80] and [33]. We recommend the classical books [62], [68], and [20] and the survey article [26], and the references therein for the general theory of multigrid methods.

3.2.1 The k^{th} Level Iteration

Following [22, 24], we give a general framework for the development of multilevel algorithms. We start by defining a hierarchy of spaces and the corresponding discrete problems. We then follow by defining the multigrid method through its ingredients and present the algorithm.

Definition 3.2.1. *A **hierarchy of spaces** $\{V_k\}_{1\leq k\leq \bar{k}}$ is a sequence of spaces of the form*

$$V_1 \subset V_2 \subset \cdots \subset V_{\bar{k}}. \tag{3.3}$$

We assume that $V_{\bar{k}}$ is the high resolution space on which we want to solve (2.3), but where the condition number of the matrix $A_{\bar{k}}$ is bad. On the other

end of the spectrum, we assume that the solution of (2.3) on V_1 is easily possible. We restrict the discussion to model problem (2.16) for simplicity. The discrete problems associated with the hierarchy of spaces (3.3) read:

Find $u_k \in V_k$ such that

$$a(u_k, v_k) = (f, v_k), \ \ \forall v_k \in V_k. \tag{3.4}$$

Define the operator $A_k : V_k \longrightarrow V_k^\star$ by

$$(A_k v_k, w_k) = a(v_k, w_k), \ \ \forall v_k, w_k \in V_k. \tag{3.5}$$

In terms of the operator A_k, the discretised equation (3.4) can be written as

$$A_k u_k = f_k, \tag{3.6}$$

where $f_k \in V_k$ satisfies

$$(f_k, v_k) = (f, v_k), \ \ \forall v_k \in V_k. \tag{3.7}$$

Definition 3.2.2. *Given a hierarchy of spaces like in* (3.3)*, a* ***multigrid method*** *consists of the following components:*

1. *A smoother R_k acting on the level space V_k, usually an iterative method like Richardson, Jacobi, Gauß-Seidel or a Schwarz method.*

2. *A coarse grid solver solving the problem on V_1 exactly.*

3. *Transfer operators between the levels V_{k-1} and V_k.*

We refer the reader to [23] for the analysis of smoothers for multigrid algorithms. For standard finite element methods, the transfer operator is typically the embedding operator. The transfer in opposite direction is achieved by the L_2-projection. We recall their definition following [33].

Definition 3.2.3 (Intergrid Transfer Operators). *The* ***coarse-to-fine intergrid transfer operator***

$$I_{k-1}^k : V_{k-1} \longrightarrow V_k \tag{3.8}$$

is taken to be the natural injection. In other words,

$$I_{k-1}^k v = v, \ \forall\, v \in V_{k-1}. \tag{3.9}$$

The ***fine-to-coarse intergrid transfer operator***

$$I_k^{k-1} : V_k \longrightarrow V_{k-1} \tag{3.10}$$

is defined to be the transpose of I_{k-1}^k with respect to the L_2-inner product $(\cdot, \cdot)$. In

other words,

$$(I_k^{k-1}w, v) = (w, I_{k-1}^k v), \quad \forall\, v \in V_{k-1}, \forall\, w \in V_k. \tag{3.11}$$

On a given level V_k, the multigrid method consists of an alternating sequence of smoothing steps and coarse grid corrections, where the latter consist of a projection of the residual to the space V_{k-1} and then recursive application of the same sequence. The standard multigrid algorithm is defined as a process which produces a function $\mathcal{MG}_k(w^{(0)}, f_k) \in V_k$, an improved approximation to the solution $u_k = A_k^{-1} f_k$ of (3.6). Here, k is the grid level, $w^{(0)} \in V_k$ is a given approximation to the solution u_k , and $f_k \in V_k{}^\star$. A standard presentation of this algorithm is given below.

Algorithm 2: The k^{th} Level Iteration of the Multigrid Algorithm.

For $k = 1$, $\mathcal{MG}_1(w^{(0)}, f_1)$ is the solution obtained from a direct method. In other words,

$$\mathcal{MG}_1(w^{(0)}, f_1) = {A_1}^{-1} f_1. \tag{3.12}$$

For $k > 1$, $\mathcal{MG}_k(w^{(0)}, f_k)$ is obtained recursively in three steps.

1. Pre-smoothing step: apply μ steps of a Richardson iteration preconditioned with the smoother R_k:

$$w^{(i+1)} = w^{(i)} + R_k(f_k - A_k w^{(i)}),\ 0 \le i \le \mu - 1. \tag{3.13}$$

2. Coarse grid correction step: let $q^{(0)} \in V_{k-1}$ and $f_{k-1} \in {V_{k-1}}^\star$ such that

$$f_{k-1} = I_k^{k-1}(f_k - A_k w^{(\mu)}),\ q^{(0)} = 0. \tag{3.14}$$

Compute

$$q^{(i)} = \mathcal{MG}_{k-1}(q^{(i-1)}, f_{k-1}),\ 1 \le i \le p. \tag{3.15}$$

Then,

$$w^{(\mu+1)} = w^{(\mu)} + I_{k-1}^k q^{(p)}. \tag{3.16}$$

3. Post-smoothing step: apply ν steps of a Richardson iteration preconditioned with the smoother R_k:

$$w^{(i+1)} = w^{(i)} + R_k(f_k - A_k w^{(i)}),\ \mu + 1 \le i \le \mu + \nu. \tag{3.17}$$

Then, the output of the k-th level iteration is

$$\mathcal{MG}_k(w^{(0)}, f_k) := w^{(\mu+\nu+1)}. \tag{3.18}$$

Remark 3.2.1. *The k^{th} level iteration of the multigrid method has three parameters, the numbers of pre- and post- smoothing steps μ and ν, as well as the number of coarse grid iterations p. The parameters are taken as positive integers and of*

the three, p has a strong impact on the structure of the iteration. It defines what is called the cycle type. For $p = 1$, the method is called a $\mathcal{V}$-cycle method, while for $p = 2$ it is called a $\mathcal{W}$-cycle method. The complexity analysis of the method shows that higher values of p do not lead to efficient algorithms.

The k^{th} level iteration results in a very simple error reduction process. More precisely, we can analyse directly the relation between the initial error $u_k - w^{(0)}$ and the final error $u_k - \mathcal{MG}_k(w^{(0)}, f_k)$ after one application of the multigrid process on the k^{th} subspace in terms of a linear operator E_k (cf (3.20)). The latter is defined in terms of the following orthogonal projection operator.

Definition 3.2.4. *Let $P_k : V \longrightarrow V_k$ be the orthogonal projection with respect to $a(\cdot, \cdot)$. In other words, for any $v \in V$, $P_k v \in V_k$ and*

$$a(v - P_k v, w) = 0, \ \ \forall w \in V_k. \tag{3.19}$$

Definition 3.2.5. *The error operator $E_k : V_k \longrightarrow V_k$ is defined recursively by*

$$\begin{aligned} &E_1 = 0 \\ &E_k = (I - R_k A_k)^\nu [I - (I - E_{k-1}) P_{k-1}]^p (I - R_k A_k)^\mu, \ \ k \geq 1. \end{aligned} \tag{3.20}$$

In other words, E_k relates the final error of the k^{th} level iteration to the initial error by the relation

$$E_k(u_k - w^{(0)}) = u_k - \mathcal{MG}_k(w^{(0)}, f_k). \tag{3.21}$$

Relation (3.21) can be proved by induction. We refer to the paper [24] for a version of the proof.

The multigrid process is often applied repeatedly to develop an iterative method for solving $A_{\bar{k}} u_{\bar{k}} = f_{\bar{k}}$. Given an initial approximation v^0, subsequent approximations are defined by

$$v^{i+1} = \mathcal{MG}_{\bar{k}}(v^i, f), \ \ \forall\, i = 0, 1, \ldots \tag{3.22}$$

From the above discussion, the error $u - v^i$ is given by

$$u - v^i = (E_{\bar{k}})^i (u - v^0). \tag{3.23}$$

Consequently, the multigrid iterative process corresponds to a linear iterative procedure. This can be written equivalently as

$$u^{i+1} = u^i + B_{\bar{k}}(f - A_{\bar{k}} u^i), \tag{3.24}$$

for the operator $B_{\bar{k}} = (I - E_{\bar{k}}) A_{\bar{k}}^{-1}$. In particular, B_k^M satisfies

$$I - B_k A_k = R_k[(I - P_{k-1}) + (I - B_{k-1} A_{k-1}) P_{k-1}] R_k, \tag{3.25}$$

i.e., $E_k = I - B_k A_k$, for $k = 1, \ldots, \bar{k}$. This is an important observation because it allows the use of the multigrid process to define preconditioning operators

$B_{\bar{k}}$. For example, the operator $B_{\bar{k}}$ can be used as a preconditioner with the conjugate gradient method to develop more effective iteration procedures in many applications.

This linear iterative procedure can be thought of as defining an operator $B_{\bar{k}} : V_{\bar{k}} \longrightarrow V_{\bar{k}}$ which approximately inverts $A_{\bar{k}}$. The goal of the multigrid analysis is to provide estimates for either the spectrum of $B_{\bar{k}}A_{\bar{k}}$ or an appropriate norm of $I - B_{\bar{k}}A_{\bar{k}}$.

3.2.2 The Full Multigrid Algorithm

In the application of the k^{th} level iteration to Equation (3.6), we take the initial guess to be the prolongation $I_{k-1}^{k}v_{k-1}$, where v_{k-1} is the approximate solution obtained solving $A_{k-1}u_{k-1} = f_{k-1}$. Then we apply the k^{th} level iteration r times. The full multigrid algorithm therefore consists of the following nested iterations (following [33]) :

Algorithm 3: The Full Multigrid Algorithm.

For $k = 1$, the approximate solution is obtained by applying a direct or iterative method to $A_1u_1 = f_1$.

For $k \geq 2$, the approximate solutions $\hat{u}_k$ are obtained via the iteration

$$\begin{aligned} u_k^{(0)} &= I_{k-1}^{k}\hat{u}_{k-1} \\ u_k^{(l)} &= \mathcal{MG}_k(u_k^{(l-1)}, f_k), \quad 1 \leq l \leq r, \\ \hat{u}_k &= u_k^{(r)}. \end{aligned} \tag{3.26}$$

It turns out that the techniques used for the analysis of the V-cycle and W-cycle, respectively are quite different. We recall the basic convergence and complexity results of the symmetric V-cycle algorithm, corresponding to Algorithm 2 with $p = 1$ and $\mu = \nu$. The following theorems show that the k^{th} level iteration scheme for a V-cycle method has a contraction number bounded away from 1 for every ν. In turn, the convergence of the full multigrid method is a simple consequence of the convergence of the k^{th} level iteration. We recall the following classical results by Braess and Hackbusch (cf. [18]):

Theorem 3.2.1 (Convergence of the kth Level Iteration for the V cycle)**.** *Let ν be the number of smoothing steps. Then,*

$$\|u_k - \mathcal{MG}_k(w^{(0)}, f_k)\|_E \leq \frac{C^\star}{\nu + C^\star}\|u_k - w^{(0)}\|_E \tag{3.27}$$

Hence, the k^{th} level iteration for any ν is a contraction, with the contraction number independent of k.

Next, follows Theorem 3.2.2, which only consideres the piecewise linear case, since higher-order finite element equations would be preconditioned by a low-order solver.

Theorem 3.2.2 (**Full Multigrid Convergence**). *If the k^{th} level iteration is a contraction with a contraction number independent of k and if r in Algorithm 3 is large enough, then there exists a constant $C > 0$ such that*

$$\|u_k - \hat{u}_k\|_E \leq Ch_k|u|_{H^2(\Omega)}. \tag{3.28}$$

We finally turn our attention to the work estimate and recall the following main theorem.

Theorem 3.2.3. *The work involved in the full multigrid algorithm is $\mathcal{O}(N_k)$, where $N_k = dim(V_k)$.*

We refer to [33] for a version of the proof.

For the theory regarding the general case, see [83]. For a detailed theory of the convergence analysis of the multigrid method we refer to [15, 25, 94, 24, 32].

3.3 Smoothed-Multilevel Methods

Multilevel methods can be easily motivated by taking an in-depth look at simple smoothing iterative schemes, such as Richardson, Gauss-Seidel, Jacobi, SOR (cf., eg., [80]). As an example and prototype for all other iterative methods, in this work we only use Richardson iteration as a smoothing iteration, but other choices are possible, see, for example, the review in [93, 59, 21]. In the next subsection, we consider the algebraic resolution of model problem (2.16) and discuss Richardson smoothing iteration, a firm understanding of which is essential for the development of multilevel concepts. With an appreciation of how the conventional methods work and why they fail, Smoothed-Multilevel Methods can be introduced as a natural remedy for restoring and improving the performance of basic relaxation schemes. We discuss them in Section 3.3 and provide rigorous algebraic analysis in Section 3.4.

3.3.1 Smoothing iterations

Let $N_k = \dim(V_k)$, then the discrete systems (3.4) lead to linear algebraic systems of type

$$A_k \mathbf{u}_k = \mathbf{f}_k \text{ in } \mathbb{R}^{N_k}, \tag{3.29}$$

where A_k denotes the symmetric positive definite (SPD) stiffness matrix with entries $a_{ij}^{(k)} := (\nabla\varphi_j^{(k)}, \nabla\varphi_i^{(k)})\ \forall\, i,j = 1,..,N_k$, $\mathbf{u}_k = [u_1, \ldots, u_{N_k}]^T$ denotes the coefficients vector in $\mathbb{R}^{N_k}$ of the discrete approximation $u_k = \sum_{j=1}^{N_k} u_j \varphi_j^{(k)} \in V_k$ and $\mathbf{f}_k = [f_1, \ldots, f_{N_k}]^T$ is the vector with entries $f_j = (f, \varphi_j^{(k)}),\ \forall j = 1, \ldots, N_k$.

Let $\left\{\mathbf{w}_j^{(k)}\right\}_{j=1}^{N_k}$ denote the eigenvectors of A_k, which by the Spectral Theorem form an orthonormal basis of $\mathbb{R}^{N_k}$ (cf., eg., [66]), and let

$$0 < \lambda_1^{(k)} \leq ... \leq \lambda_{N_k}^{(k)}, \tag{3.30}$$

denote the corresponding eigenvalues, ordered non-decreasingly. Eigenvectors corresponding to higher eigenvalues are increasingly oscillatory, i.e., their A_k- norm is larger. This follows trivially from the fact that $\lambda_i^{(k)} = \|\mathbf{w}_i^{(k)}\|_{A_k}^2 / \|\mathbf{w}_i^{(k)}\|_{\ell^2}^2 = \|\mathbf{w}_i^{(k)}\|_{A_k}^2$.

Next, we define the notions of smoothing iteration and Richardson iteration. For simplicity of the exposure, we omit the index k, whenever we're referring to the linear systems (3.29), and consider the generic form

$$A\mathbf{u} = \mathbf{f} \text{ in } \mathbb{R}^N. \tag{3.31}$$

Definition 3.3.1. ***(Smoother vector)*** *Consider two generic vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, which can be uniquely decomposed as $\mathbf{a} = \sum_{i=1}^N a_i \mathbf{w}_i$ and $\mathbf{b} = \sum_{i=1}^N b_i \mathbf{w}_i$. We say that $\mathbf{b}$ is smoother than $\mathbf{a}$ if $\|\mathbf{b}\| \leq \|\mathbf{a}\|$ for a suitable norm $\|\cdot\|$, and its components along the most oscillatory eigenvectors are smaller. By convention we define as "oscillatory" the components from $N/2$ onwards. Then $\mathbf{b}$ is smoother*

than $\mathbf{a}$ *if* $b_i \leq a_i$ *for* $i \in (N/2, N]$.

Definition 3.3.2. ***(Smoothing Iteration)*** *Given an initial guess* $\mathbf{u}^{(0)} \in \mathbb{R}^N$, *consider the classical linear iteration for the resolution of* (3.31) *of the form*

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + R(\mathbf{f} - A\mathbf{u}^{(i)}) \text{ for } i = 0, 1, \ldots, \tag{3.32}$$

with some nonsingular matrix R. *Let* $\mathbf{u}$ *denote the exact solution of* (3.31), *we denote by* $\mathbf{e}^{(i)} := \mathbf{u} - \mathbf{u}^{(i)}$ *the error after* i *iterations. We say that* (3.32) *is a smoothing iteration if* $(I - RA)\mathbf{e}^{(i)}$ *is smoother than* $\mathbf{e}^{(i)}$ *for any* i.

The matrix $I - RA$ is called the *iteration matrix* and it is generally denoted by $M := I - RA$. From (3.32) it is immediate that

$$\mathbf{e}^{(i+1)} = \mathbf{e}^{(i)} - R(\mathbf{f} - A\mathbf{u}^{(i)}) = \mathbf{e}^{(i)} - RA\mathbf{e}^{(i)} = M\mathbf{e}^{(i)} = \cdots = M^{i+1}\mathbf{e}^{(0)}. \tag{3.33}$$

Definition 3.3.2 and equation (3.33) imply that, for smoothing iterations, the iteration matrix M has a "smoothing" effect on the error, by dumping the highly oscillatory components of the error.

Definition 3.3.3. ***(Richardson Iteration)*** *Given a fixed parameter* $\omega \in \mathbb{R}$ *and an initial guess* $\mathbf{u}^{(0)}$, *Richardson iteration for the resolution of* (3.31) *takes the form:*

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \omega(\mathbf{f} - A\mathbf{u}^{(i)}) \text{ for } i = 0, 1, \ldots \tag{3.34}$$

Richardson iteration can also be written as

$$\mathbf{u}^{(i+1)} = (I - \omega A)\mathbf{u}^{(i)} + \omega\mathbf{f} \text{ for } i = 0, 1, \ldots \tag{3.35}$$

Remark 3.3.1. *Richardson iteration* (3.34) *is of type* (3.32) *where the matrix* R *is given by* ωI. *The optimal choice for the parameter* ω *is* $\omega = 1/\gamma$, *where* γ *is a damping parameter of the same order as the spectral radius of* A

$$\rho(A) := \max\{\lambda_i | 1 \leq i \leq N\}. \tag{3.36}$$

In practical situations, $\gamma = \rho(A)$ *or* $\gamma \geq \rho(A)$ *(see, eg., [61, 80]).*

Remark 3.3.2. *Richardson iteration* (3.34) *is a smoothing iteration. Consider the errors after respectively* i, *and* $i + 1$ *Richardson iterations*

$$\mathbf{e}^{(i)} = \sum_{j=1}^{N} c_j^{(i)} \mathbf{w}_j \text{ and } \mathbf{e}^{(i+1)} = \sum_{j=1}^{N} c_j^{(i+1)} \mathbf{w}_j \tag{3.37}$$

for some coefficients $\{c_j^{(i)}\}_{j=1}^N$ *and* $\{c_j^{(i+1)}\}_{j=1}^N$. *Observe that by construction,* M *and* A *share the same eigenvectors* $\{\mathbf{w}_i\}_{i=1}^N$, *and we can easily derive from the error propagation formula* (3.33) *that*

$$c_j^{(i+1)} = \theta_j c_j^{(i)} \qquad \forall j = 1, \ldots, N, \tag{3.38}$$

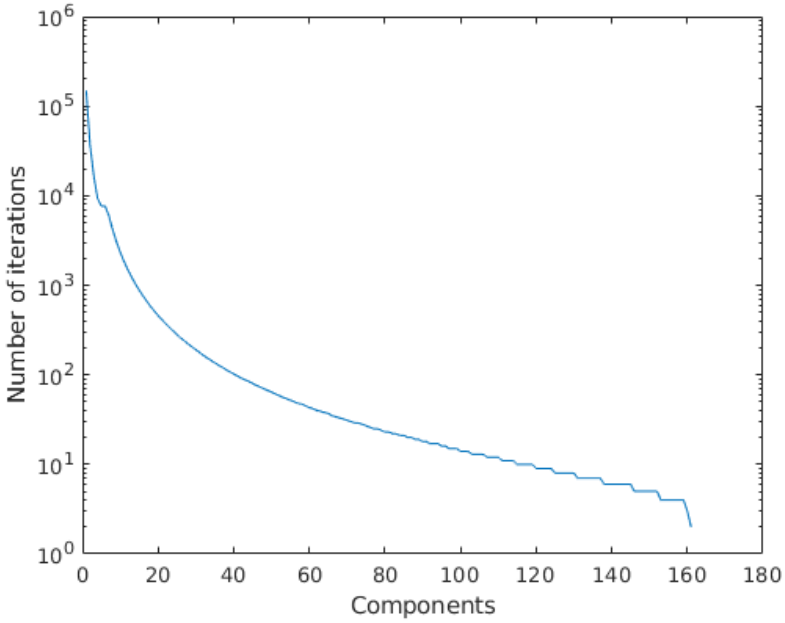


Figure 3.1: Number of required iterations to bring the error for different components below 1e-8 on a one-dimensional problem with 161 uniformly distributed degrees of freedom.

where $\theta_j := 1 - \lambda_j/\gamma$ *is the j-th eigenvalue of the iteration matrix* M*, and it represents the reduction factor associated to the error component in the direction of* $\mathbf{w}_j$*. Notice that the definition of* θ_j *is independent on the iteration step* i*. For the practical choice* $\gamma = 1/\rho(A)$*,*

$$\theta_1 = 1 - \frac{\lambda_1}{\lambda_N} \approx 1 \quad \text{and} \quad \theta_N = 1 - \frac{\lambda_N}{\lambda_N} = 0. \tag{3.39}$$

This implies that after a single Richardson iteration

$$c_1^{(i+1)} = \theta_1 c_1^{(i+1)} \approx c_1^{(i+1)} \quad \text{and} \quad c_N^{(i+1)} = \theta_N c_N^{(i)} = 0, \tag{3.40}$$

i.e. the slowest converging component corresponds to the smallest eigenvalue λ_1*, while the fastest converging component corresponds to the largest eigenvalue* λ_N*.*

In general, the reduction factor for the Richardson iteration is "close to zero" for components corresponding to large eigenvalues and close to one for components corresponding to small eigenvalues. After a single Richardson iteration, the high oscillatory components will have been strongly reduced.

Figure 3.1 shows in a practical example the number of required iterations to bring the error in each component below 10^{-8} for a one-dimensional model problem with 161 uniformly distributed degrees of freedom, showing how higher frequencies require a smaller number of iterations.

This characteristic of Richardson iteration makes it a good smoother candidate for many multilevel algorithms, where one solves exactly on a coarse grid (reaching convergence in all components), and then performs a sequence of prolongations

followed by a fixed number of smoothing steps, to improve convergence in the finer grids, under the assumptions that lower frequencies have already been taken care of in the previous levels.

This is achieved by considering the canonical embedding operator which is represented by the coarse to fine operator (3.8). We still denote by $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$ the corresponding discrete linear operator. Notice that its matrix representation won't be the identity matrix, since we're using different basis functions in V_k and in V_{k+1}. As an example, consider linear finite element functions. These are uniquely determined by their values in the nodes. For nodes that exist both in $\mathcal{T}_k$ and $\mathcal{T}_{k+1}$, the value at those nodes can be determined in $\mathcal{T}_k$ and it remains the same. For the nodes in $\mathcal{T}_{k+1}$ that are not in $\mathcal{T}_k$, their values are determined by linear interpolation.

We will refer to multilevel algorithms that adopt the above procedure of resolution as smoothed-multilevel methods.

Algorithm 4: The Smoothed-Multilevel Algorithm.

For $k = 1$, $\hat{u}_1 = A_1^{-1} f_1$.

For $k \geq 2$, the approximate solutions $\hat{u}_k$ are obtained iteratively from

$$\begin{aligned} u_k^{(0)} &= I_{k-1}^k \hat{u}_{k-1} \\ u_k^{(i)} &= u_k^{(i-1)} + \omega^{(k)}(f_k - A_k u_k^{(i-1)}), \quad 1 \leq i \leq l, \\ \hat{u}_k &= u_k^{(l)}. \end{aligned} \tag{3.41}$$

In order to understand the principle behind many multilevel algorithms, and behind our S-AFEM algorithm (cf. Chapter 5), in [72] we provide a discussion on a simple one-dimensional example, which we present in the next subsection.

3.3.2 Smoothed-Multilevel Methods

Consider model problem (2.16) with constant function $f = 1$ on the right-hand side. We solve (without preconditioner) using either the CG method or Richardson iterations, on a sequence of uniformly refined grids.

We set a stopping tolerance of 10^{-6}, and fix a maximum number of iterations to $1,000,000$. Moving from one level to the next, the mesh is globally refined, doubling the number of cells of the grids.

Tables 3.1 and 3.2 show a comparison between the number of iterations required to reach convergence when we apply the two iterative methods with the initial guess set to zero, or to the prolongation of the solution from the previous cycle.

Surprisingly, the CG method does not seem to gain any advantage from the prolongation step. On the other hand, Richardson iteration shows a dramatic decrease in the number of required iterations when we use the prolongation of the previous solution as initial guess for the iterative procedure.

After the first few levels, Richardson iteration becomes even faster than CG, thanks to its spectral behaviour. The convergence on the coarsest levels captures the less oscillatory part of the solution. Their prolongation allows the iterative solver to start from an already good approximation of the solution in its low frequency part. Intuitively, the prolongation operation substantially leaves unaltered the low frequencies of the previous mesh. By applying smoothing iterations, we're converging towards the solution in the highest frequencies.

Level	DoF	Iterations CG	Iterations CGProl
2	41	20	20
3	81	40	40
4	161	80	80
5	321	160	160
6	641	320	320
7	1281	640	640
8	2561	1280	1280

Table 3.1: Comparison of the number of iterations between CG without prolongation and CG with prolongation with stopping tolerance of 10^{-6} and maximum number of iterations $1,000,000$.

Level	DoF	Iterations Richardson	Iterations RichardsonProl
2	41	5523	1333
3	81	20321	1072
4	161	74115	119
5	321	267713	9
6	641	955805	2
7	1281	******	2
8	2561	******	2

Table 3.2: Comparison of the number of iterations between Richardson and Richardson with prolongation with stopping tolerance of 10^{-6} and maximum number of iterations $1,000,000$.

The CG method, on the other hand, is a projection method of *Krylov subspace* type. The main idea of the projection process in general is to find an approximate solution of system (3.31), where the dimension N is possibly very large, by solving at each step a system of much smaller dimensionality, which is obtained by projecting the original system (3.31) onto a suitable subspace of $\mathbb{R}^N$ ([67]). Specifically, the approximate solution at iterative step m is searched in $\mathbf{x}_0 + \mathcal{K}_m(\mathbf{r}_0, A)$, where $\mathbf{x}_0$ is a given initial guess and $\mathcal{K}_m(\mathbf{r}_0, A) \subset \mathbb{R}^N$ is the *search space* given by the *Krylov subspace* of dimension $m \ll N$ generated by A and $\mathbf{r}_0$ and defined as

$$\mathcal{K}_m(r_0, A) := \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{m-1}\mathbf{r}_0\}, \tag{3.42}$$

where $r_0 := \mathbf{f} - A\mathbf{x}_0$ is the initial residual. Krylov subspaces form a nested

sequence of subspaces.

Definition (3.42) implies that the error at step n can be written as

$$\mathbf{e}_n = p_n(A)\mathbf{e}_0, \tag{3.43}$$

where $p_n \in \mathbb{P}_n^{(0,1)} := \{p \in \mathbb{P}_n, \text{ s.t. } p(0) = 1\}$.

In particular, the *optimality property* of CG (cf. eg. [67]) implies that

$$\|\mathbf{e}_n\|_A^2 = \min_{p \in \mathbb{P}_n^{(0,1)}} \|p(A)\mathbf{e}_0\|_A^2. \tag{3.44}$$

In each iteration, the conjugate gradient method improves the convergence in all error components relying on the optimality property (3.44), instead of capturing only the high oscillatory ones. A particular characteristic is that they *save all information along the way*, i.e., they use at any given iteration the information computed in all previous iterations.

On the other hand, by their nature, smoothing iterations combine aspects of the underlying PDE and the corresponding finite element discretisation. Despite being far less competitive as solvers for large systems in general, smoothing iterations turn out to be very useful in our context, similarly to what happens in multigrid methods: they use the spectral decomposition of M and exploit the strong relation between eigenfunctions of the iteration matrix M and the underlying mesh in order to take advantage of coarser meshes.

3.4 Algebraic Error Analysis for Smoothed-Multilevel Methods

In this section we analyse the algebraic error propagation in smoothed-multilevel methods following [72]. We first provide a one step error propagation recursive formula, and afterwards we provide a compact error propagation formula after introducing the Frequency-Coupling and Smoothing (FCS) Matrices. Finally, we provide the algebraic error analysis under the assumption that the prolongation operator preserves low frequencies from the previous level.

3.4.1 Error propagation

Theorem 3.4.1 (**Error propagation** ([72])). *Let $\mathbf{e}_k^{(l)}$ and $\mathbf{e}_{k+1}^{(l)}$ denote the algebraic errors after l smoothing iterations respectively at step k and $k+1$, for $k = 1, \ldots, \bar{k} - 1$. Let*

$$\mathbf{a}_{k+1} := \mathbf{u}_{k+1} - I_k^{k+1}\mathbf{u}_k \in \mathbb{R}^{N_{k+1}} \tag{3.45}$$

denote the difference between the exact algebraic solution $\mathbf{u}_{k+1}$ at level $k+1$ and the prolongation of the exact algebraic solution $\mathbf{u}_k$ from the previous level k to the current level $k+1$, for $k = 1, \ldots, \bar{k} - 1$. Then, the following error propagation

recursive formula holds true

$$\mathbf{e}_{k+1}^{(l)} = M_{k+1}{}^{l}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(l)}), \quad for \;\; k = 1, \ldots, \bar{k} - 1. \tag{3.46}$$

Proof. Let $\mathbf{e}_1 = \mathbf{u}_1 - \mathbf{u}_1^c$ be the error after the first cycle $k = 1$, where $\mathbf{u}_1^c$ is the numerical computed approximation. After prolongating $\mathbf{u}_1^c$ to the next level $k = 2$, there is an initial error

$$\begin{aligned} \mathbf{e}_2^{(0)} &= \mathbf{u}_2 - I_1^2\mathbf{u}_1^c \\ &= \mathbf{u}_2 - I_1^2\mathbf{u}_1 + I_1^2\mathbf{e}_1 \\ &= \mathbf{a}_2 + I_1^2\mathbf{e}_1. \end{aligned} \tag{3.47}$$

After l smoothing iterations there is a smoothed approximation $\mathbf{u}_2^{(l)}$ produced and the final error is given by

$$\begin{aligned} \mathbf{e}_2^{(l)} &= M_2{}^{l}\mathbf{e}_2^{(0)} \\ &= M_2{}^{l}\mathbf{a}_2 + M_2{}^{l}I_1^2\mathbf{e}_1. \end{aligned} \tag{3.48}$$

Let now $k = 2, 3, \ldots, \bar{k}-1$ be generic. We prolongate the smoothed approximation $\mathbf{u}_k^{(l)} = \mathbf{u}_k - \mathbf{e}_k^{(l)}$ from step k to obtain the initial guess for step $k + 1$

$$\begin{aligned} \mathbf{u}_{k+1}^{(0)} &= I_k^{k+1}\mathbf{u}_k^{(l)} \\ &= I_k^{k+1}\mathbf{u}_k - I_k^{k+1}\mathbf{e}_k^{(l)}, \end{aligned} \tag{3.49}$$

which produces the initial error

$$\begin{aligned} \mathbf{e}_{k+1}^{(0)} &= \mathbf{u}_{k+1} - \mathbf{u}_{k+1}^{(0)} \\ &= \mathbf{u}_{k+1} - I_k^{k+1}\mathbf{u}_k + I_k^{k+1}\mathbf{e}_k^{(l)} \\ &= \mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(l)}. \end{aligned} \tag{3.50}$$

After l smoothing iterations the final error at step $k + 1$ is

$$\begin{aligned} \mathbf{e}_{k+1}^{(l)} &= M_{k+1}{}^{l}\mathbf{e}_{k+1}^{(0)} \\ &= M_{k+1}{}^{l}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(l)}), \end{aligned} \tag{3.51}$$

which proves the recursive formula. □

Observation 3.4.1. *If we repetitively apply the one-step error propagation equation* (3.51)*, we get a recursion of the type*

$$\begin{aligned} \mathbf{e}_{k+1}^{(l)} &= M_{k+1}{}^{l}(\mathbf{a}_{k+1} + I_k^{k+1}\mathbf{e}_k^{(l)}) \\ &= M_{k+1}{}^{l}(\mathbf{a}_{k+1} + I_k^{k+1}(M_k{}^{l}(\mathbf{a}_k + I_{k-1}^{k}\mathbf{e}_{k-1}^{(l)}))) \\ &\ldots \\ &= M_{k+1}{}^{l}(\mathbf{a}_{k+1} + I_k^{k+1}M_k{}^{l}(\mathbf{a}_k + I_{k-1}^{k}M_{k-1}{}^{l}(\mathbf{a}_{k-1} + \ldots))). \end{aligned} \tag{3.52}$$

By applying all the multiplications extensively we get the following extended error propagation formula for Smoothed-Multilevel Methods

$$\begin{aligned}\mathbf{e}_{k+1}^{(l)} = {M_{k+1}}^{l}\mathbf{a}_{k+1} + {M_{k+1}}^{l}I_k^{k+1}{M_k}^{l}\mathbf{a}_k + {M_{k+1}}^{l}I_k^{k+1}{M_k}^{l}I_{k-1}^{k}{M_{k-1}}^{l}\mathbf{a}_{k-1} + \dots \\ \cdots + {M_{k+1}}^{l}I_k^{k+1}{M_k}^{l}I_{k-1}^{k}{M_{k-1}}^{l}\cdots {M_3}^{l}I_2^3{M_2}^{l}\mathbf{a}_2 \\ + {M_{k+1}}^{l}I_k^{k+1}{M_k}^{l}I_{k-1}^{k}{M_{k-1}}^{l}\cdots {M_3}^{l}I_2^3{M_2}^{l}I_1^2\mathbf{e}_1.\end{aligned} \tag{3.53}$$

Observation 3.4.2. *If we let*

$$\begin{aligned}\mathbf{a} := \mathbf{a}_{k+1} + I_k^{k+1}{M_k}^{l}\mathbf{a}_k + I_k^{k+1}{M_k}^{l}\dots I_{k-1}^{k}{M_{k-1}}^{l}\mathbf{a}_{k-1} \\ \cdots + I_k^{k+1}{M_k}^{l}I_{k-1}^{k}{M^l}_{k-1}\cdots {M_3}^{l}I_2^3{M_2}^{l}I_1^2\mathbf{e}_1,\end{aligned} \tag{3.54}$$

then equation (3.53) *becomes*

$$\mathbf{e}_{k+1}^{(l)} = {M_{k+1}}^{l}\mathbf{a}, \tag{3.55}$$

which means that the algebraic error at any step $k+1$ is the result of l smoothing iterations applied to the vector $\mathbf{a}$ that defines the error accumulated from prolongating the contribution of the algebraic errors coming from all previous steps.

Definition 3.4.1 (**Frequency-Coupling and Smoothing (FCS) Matrices**)**.** *Define the frequency-coupling and smoothing (FCS) matrix*

$$B_{j+1,j} := I_j^{j+1}M_j^l \in \mathbb{R}^{N_{j+1}\times N_j} \text{ for } j = 2,\dots,k. \tag{3.56}$$

and the frequency-coupling and smoothing product (FCSP)

$$\mathbb{B}_{k+1,i} := B_{k+1,k}\dots B_{i+1,i} \in \mathbb{R}^{N_{k+1}\times N_i} \text{ for } i = 2,\dots,k. \tag{3.57}$$

Theorem 3.4.2 (**Error propagation formula for Smoothed-Multilevel Methods** ([72]))**.** *The algebraic error in Smoothed-Multilevel Methods satisfies the following error propagation formula for any step k, for $k = 2,\dots,\bar{k}-1$*

$$\mathbf{e}_{k+1}^{(l)} = {M_{k+1}}^{l}\left(\mathbf{a}_{k+1} + \sum_{j=2}^{k}\mathbb{B}_{k+1,j}\mathbf{a}_j + \mathbb{B}_{k+1,2}I_1^2\mathbf{e}_1\right), \tag{3.58}$$

where the vectors $\mathbf{a}_j$ are defined by (3.45)*.*

Proof. The proof is a trivial consequence of substituting Definition 3.4.1 in the extended error propagation formula (3.53), which gives

$$\begin{aligned}\mathbf{e}_{k+1}^{(l)} &= {M_{k+1}}^{l}(\mathbf{a}_{k+1} + B_{k+1,k}\mathbf{a}_k + \dots \\ &\quad \cdots + B_{k+1,k}B_{k,k-1}\dots B_{3,2}\mathbf{a}_2 + B_{k+1,k}B_{k,k-1} \\ &\quad \dots B_{3,2}I_1^2\mathbf{e}_1) \\ &= {M_{k+1}}^{l}(\mathbf{a}_{k+1} + \mathbb{B}_{k+1,k}\mathbf{a}_k + \mathbb{B}_{k+1,k-1}\mathbf{a}_{k-1} + \\ &\quad \cdots + \mathbb{B}_{k+1,2}\mathbf{a}_2 + \mathbb{B}_{k+1,2}I_1^2\mathbf{e}_1).\end{aligned} \tag{3.59}$$

□

Next, we define the frequency cutoff projection operators, which are a useful tool to analyse the structure of the FCS matrix $B_{j+1,j}$. In Theorem 3.4.3 we provide a decomposition of the FCS matrix in the product of the prolongation matrix I_j^{j+1} with the low frequency cutoff projection operator and another matrix, which has a contraction effect on the norms of the vectors.

Definition 3.4.2 (**Frequency cutoff operators**). *We define the projection operator* $L_j : \mathbb{R}^{N_j} \to \mathbb{R}^{N_j}$, *such that* $\mathbf{v} \mapsto L_j\mathbf{v} := \sum_{i=1}^{N_j/2} v_i \mathbf{w}_i^{(j)}$ *and the projection operator* $H_j : \mathbb{R}^{N_j} \to \mathbb{R}^{N_j}$, *such that* $\mathbf{v} \mapsto H_j\mathbf{v} := \sum_{i=N_j/2+1}^{N_j} v_i \mathbf{w}_i^{(j)}$.

In particular,

$$L_j \oplus H_j = Id_{N_j} \text{ and } \|\mathbf{v}\|^2 = \|L_j\mathbf{v}\|^2 + \|H_j\mathbf{v}\|^2 \ \forall \mathbf{v} \in \mathbb{R}^{N_j}. \tag{3.60}$$

Theorem 3.4.3 (**Structure of the FCS matrix**). *([72]) Let* $j = 2, \dots, k$ *be fixed. The FCS matrix can be decomposed as*

$$B_{j+1,j} = I_j^{j+1} L_j + C_j, \tag{3.61}$$

where the matrix $C_j \in \mathbb{R}^{N_j+1 \times N_j}$ *is defined as*

$$C_j := I_j^{j+1}((M_j{}^l - Id_{N_j})L_j + M_j{}^l H_j) \tag{3.62}$$

and it is such that

$$\|C_j\mathbf{v}\| \leq c\|\mathbf{v}\|, \ \forall \mathbf{v} \in \mathbb{R}^{N_j}, \text{ where } c < 1. \tag{3.63}$$

Proof. We apply definition (3.56) of the FCS matrix and relation (3.60) and we get

$$\begin{aligned} B_{j+1,j} &= I_j^{j+1} M_j{}^l \\ &= I_j^{j+1} M_j{}^l L_j + I_j^{j+1} M_j{}^l H_j \\ &= I_j^{j+1} L_j + (I_j^{j+1} M_j{}^l L_j - I_j^{j+1} L_j + I_j^{j+1} M_j{}^l H_j) \\ &= I_j^{j+1} L_j + C_j, \end{aligned} \tag{3.64}$$

where $C_j := I_j^{j+1}((M_j{}^l - Id_{N_j})L_j + M_j{}^l H_j)$.

Next, in order to prove (3.63), consider $\mathbf{v} \in \mathbb{R}^{N_j}$ and estimate

$$\begin{aligned} \|C_j\mathbf{v}\|^2 &= \|I_j^{j+1}((M_j{}^l - Id_{N_j})L_j + M_j{}^l H_j)\mathbf{v}\|^2 \\ &= \|((M_j{}^l - Id_{N_j})L_j + M_j{}^l H_j)\mathbf{v}\|^2 \\ &\leq (\|(M_j{}^l - Id_{N_j})L_j\mathbf{v}\| + \|M_j{}^l H_j\mathbf{v}\|)^2 \\ &\leq 2(\|(M_j{}^l - Id_{N_j})L_j\mathbf{v}\|^2 + \|M_j{}^l H_j\mathbf{v}\|^2), \end{aligned} \tag{3.65}$$

where we've applied the triangle inequality and the discrete Cauchy-Schwarz inequality.

Consider the first term in the rhs

$$\begin{aligned}\|(M_j{}^l - Id_{N_j})L_j\mathbf{v}\|^2 &= \left\|\sum_{i=1}^{N_j/2}\left((\theta_i^{(j)})^l - 1\right)v_i\mathbf{w}_i^{(j)}\right\|^2 \\ &= \sum_{i=1}^{N_j/2}\left((\theta_i^{(j)})^l - 1\right)^2 v_i^2 \\ &\leq \left((\theta_i^{(N_j/2)})^l - 1\right)^2 \sum_{i=1}^{N_j/2} v_i^2 \\ &= \left((\theta_{(N_j/2)}^{(j)})^l - 1\right)^2 \|L_j\mathbf{v}\|^2.\end{aligned} \tag{3.66}$$

Likewise,

$$\begin{aligned}\|M_j{}^l H_j\mathbf{v}\|^2 &= \left\|\sum_{i=N_j/2+1}^{N_j}(\theta_i^{(j)})^l v_i\mathbf{w}_i^{(j)}\right\|^2 \\ &= \sum_{i=N_j/2+1}^{N_j}(\theta_i^{(j)})^{2l}v_i^2 \\ &\leq (\theta_{N_j/2+1}^{(j)})^{2l}\sum_{i=N_j/2+1}^{N_j} v_i^2 \\ &= (\theta_{N_j/2+1}^{(j)})^{2l}\|H_j\mathbf{v}\|^2.\end{aligned} \tag{3.67}$$

We let $c := 2\max\left\{\left((\theta_{(N_j/2)}^{(j)})^l - 1\right)^2, (\theta_{N_j/2+1}^{(j)})^{2l}\right\} < 1$, we substitute it into (3.65), and we get estimate (3.63). □

3.4.2 Non-interacting Frequency Coupling Hypothesis

In order to give a qualitative interpretation to the error propagation formula, we consider the following (reasonable) Assumption 3.4.1 :

Assumption 3.4.1 (**Non-interacting Frequency Coupling Hypothesis for Smoothed-Multilevel Methods** ([72])). *We assume that*

$$L_j I_{j-1}^j L_{j-1} = I_{j-1}^j L_{j-1}\ \forall j = 1, \dots k, \tag{3.68}$$

i.e. the prolongation operator preserves low frequencies from the previous level.

Finally, we propose the main result of the section, which is given in Theorem 3.4.4 where we derive the propagation formula for the algebraic error under hypothesis (3.68). For the proof, we take advatage of decomposition (3.61) of the FCS matrix to obtain a decompostion for the FCSP (3.57) and we substitute the results to Theorem 3.4.2.

Theorem 3.4.4 (**Error propagation formula for smoothed-multilevel methods under the Non-interacting Frequency Coupling Assumption**). *The algebraic error in smoothed-multilevel methods satisfies the following error propagation formula for any step k, for $k = 2, \dots, \bar{k} - 1$*

$$\mathbf{e}_{k+1}^{(l)} = {M_{k+1}}^{l} \left(\mathbf{a}_{k+1} + \sum_{j=2}^{k} I_j^{k+1} L_j \mathbf{a}_j + I_2^{k+1} L_2 I_1^2 \mathbf{e}_1 + \sum_{j=2}^{k} \mathbb{D}_{k+1,j} \mathbf{a}_j + \mathbb{D}_{k+1,2} I_1^2 \mathbf{e}_1 \right), \quad (3.69)$$

where the matrix $\mathbb{D}_{k,m} \in \mathbb{R}^{N_k \times N_m}$ is such that $\|\mathbb{D}_{k,m} v\| \leq c\|v\|$, $\forall v \in \mathbb{R}^{N_m}$, where $c < 1$, the matrix $I_m^k := I_{k-1}^k \dots I_m^{m+1}$, $\forall k > m+1$ and the vectors $\mathbf{a}_j$ are defined by (3.45), i.e., $\mathbf{a}_j := \mathbf{u}_j - I_{j-1}^j \mathbf{u}_{j-1}$.

Proof. Observing that

$$\mathbb{B}_{k,m} = B_{k,k-1} \mathbb{B}_{k-1,m}, \qquad \forall k > m+1, \quad (3.70)$$

we can recursively apply the decomposition of $B_{k,k-1}$, and, using assumption 3.68, we conclude that

$$\mathbb{B}_{k,m} = I_m^k L_m + \mathbb{D}_{k,m}, \quad (3.71)$$

where the matrix $\mathbb{D}_{k,m}$ is in $\mathbb{R}^{N_k \times N_m}$, and contains all the mixed products. In particular, in every one of these products, there will always be at least one of the matrices C_j for some j, that is,

$$\|\mathbb{D}_{k,m} v\| \leq c\|v\|, \ \forall v \in \mathbb{R}^{N_m}, \quad (3.72)$$

where $c < 1$. We conclude by substituting decompostion (3.71) to the error propagation formula (3.58) in Theorem 3.4.2. □

Remark 3.4.1. *Theorem 3.4.4 quantifies the algebraic error that is accumulated after $k+1$ cycles in smoothed-multilevel methods, under the assumption that low frequencies are preserved by the prolongation operator. The smoothing matrix at cycle $k+1$ is responsible for dumping the most oscillatory part of this error. There is a contribution given by the accumulation of all low frequency-parts of the errors of all previous cycles (c.f. second and third term in the summation in the rhs of (3.69)), which is expected to be "small", since low frequencies of the exact algebraic solution at a mesh are close to the low frequencies of the exact algebraic solution at the successive mesh. Finally there is a last type of contribution, which is given by mixed products (cf. fourth and fifth term in the summation in the rhs of (3.69)), which is also "small" due to $c < 1$.*

Notice that Assumption 3.4.1 is only useful to distinguish between high and low frequency parts in the error propagation formula, but it is not essential for

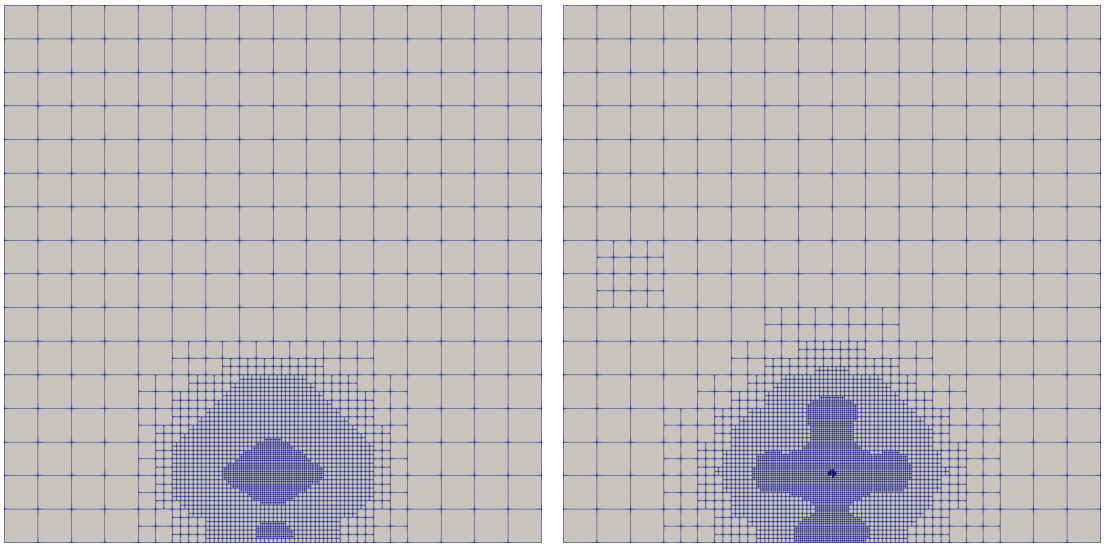


Figure 3.2: An example of grid refinement that does not satisfy Assumption 3.4.1

its proof, given in Theorem 3.4.2. The assumption is useful to identify qualitatively how the error propagates between successive refinement levels, and can be interpreted as a condition on the distribution of the degrees of freedoms between grids on different levels. In particular, it implies that all low eigenfunctions of the space V_{j-1}, in particular those corresponding to the first $N_{j-1}/2$ eigenvalues, can be represented *exactly* by low frequencies of V_j, i.e., they should be representable as linear combinations of the first $N_j/2$ eigenfunctions of V_j.

In general, it is not trivial to verify this assumption for practical applications. In fact, for finite element methods of elliptic equations with variable coefficients based on general triangulations, the eigenvalue and eigenvectors of the stiffness matrix are not easy to find out. Indeed it is even harder than solving the linear algebraic equation (cf. eg., the many references [93], [26], [62], [21]). However, it is safe to state that local refinement in finite element simulations introduces more frequencies in the higher part of the spectrum, perturbing only slightly the lowest part of the spectrum.

Assumption 3.4.1 may also be satisfied only approximately. In this case, Theorem 3.4.4 should be modified to take this into account. The main statement would still remain valid, but we would also have higher order error terms appearing in the error propagation formula (3.69), due to the inexactness of the low frequency prolongations.

Figure 3.2 provides an example of grid refinement that may be troublesome for the above hypothesis: when passing from the left grid to the right one, we are introducing some low-frequency terms (in the middle left side of the mesh), that may invalidate the assumption. Notice that, from the practical point of view, the assumption is verified for most refinements that do not add too many degrees of freedom between refinement levels.

When this occurs, it may be necessary to use higher frequencies to describe the low modes of the previous grid, but these high frequencies would be damped

very quickly by the smoothing steps nonetheless, thanks to the presence of the matrix M_{k+1}^l in front of the propagation formula (3.69).

Chapter 4

A Posteriori Error Analysis with algebraic error

This chapter is dedicated to a posteriori error analysis with algebraic error. The purpose is to expand the classical theory on residual-based a posteriori error estimators that we introduced in Section 2.3, and which are historically defined and derived in terms of the Galerkin solution.

We dedicate Section 4.1 to discuss the need to account for inexact algebraic approximations. Our attention is devoted to the main issues that a posteriori error analysis accounting for the algebraic error has to deal with. More specifically, the derivation and the construction of the a posteriori estimator should be done on the available inexact approximation, instead of the Galerkin solution and the algebraic error should be incorporated in the estimate.

In Section 4.2, we are going to prove a bound on the estimator for a generic function in terms of the estimator for the Galerkin solution and the corresponding algebraic error, following [72].

Finally, in Section 4.3 we slightly touch upon a still ongoing work in [71] that quantifies qualitatively different contributions of different eigenfunctions in the expression of the error estimator.

4.1 A Posteriori Error Estimates with Algebraic Error

When solving real-world practical applications, the main difficulty one has to face is that exact (or even near-to-exact) solutions of the algebraic problem associated to finite element problems cannot be computed. The approximation u_h^c that one obtains in a computer, does not satisfy the Galerkin property (2.7). The total error can be written as the sum of two contributions

$$\underbrace{u - u_h^c}_{\text{total error}} = \underbrace{(u - u_h)}_{\text{discretisation error}} + \underbrace{(u_h - u_h^c)}_{\text{algebraic error}} . \tag{4.1}$$

The algebraic error may have a significant effect on the computed approximation, and the solution of the algebraic problem has to be considered an indivisible part of the overall solution process ([76]).

This issue is reflected in adaptive mesh refining procedures. The common practice in computational sciences and engineering community has been to replace u_h by u_h^c in the expression of the error estimator η during the module $\mathcal{E}stimate$. As a result, there are some urgent challenges that the derivation and application of the a posteriori error estimates should resolve

1. The derivation and the construction of the a posteriori estimator should be done on the available inexact approximation u_h^c.

2. The estimation of the total error $u - u_h^c$ should incorporate the algebraic error $u_h - u_h^c$.

The classical reliability and efficiency bounds (2.20) and (2.21) have to be consequently revised and extended to take into account the above points.

A vast literature proposes the use of standard residual-based a posteriori error estimator on the discretisation error as a basic building block and extends it, using various heuristics arguments, to incorporate the algebraic error. We refer to the seminal and investigative paper by Papež and Strakoš [75] and the references therein for various approaches.

Residual-based a posteriori error estimates for the total error for the model problem have been pubblished in [16], [5] and [75].

In [75], the authors give the detailed proof of the residual-based upper bound on the energy norm of the total error

$$|u - v_h|_1^2 \leq 2C^2(J^2(v_h) + osc^2) + 2C_{intp}^2|u_h - v_h|_1^2, \tag{4.2}$$

for $v_h \in V_h$, with the positive multiplicative factors C and C_{intp} that are independent of u, u_h and h, but depend on the shape regularity of the triangulation. The term accounting for the algebraic error is scaled by a multiplicative factor C_{intp} that was introduced in [40]. It represents however a worst case scenario that can lead to an overestimation and it is in general not easy to estimate.

As one of the main still unresolved challenges, the authors point out that a tight upper bound of $|u_h - u_h^c|_1$ is not available yet. The use of finite precision arithmetic and the neglection of roundoff errors may therefore lead to inaccurate and unrealistic estimates.

Last but not least, the authors emphasize that since a lot of methodological difficulties are there already for a simple model problem such as the Poisson problem, it is not clear whether the extension of the estimator to incorporate the algebraic error for more complicated model problems could lead to further complications.

The above discussed issues make the application of the residual-based error estimator for the mesh refinement adaptivity in presence of algebraic errors an open problem, as claimed in [75]. Moreover, when considering h-adaptive algorithms,

another difficulty is added: in the bound (4.2) the algebraic error is estimated globally and its local contributions cannot guarantee an indication of the spatial distribution of the discretisation error over the domain (cf. [67] and [74]). In this regard, there have been recently developed flux reconstruction methodologies that introduce robust stopping criteria and balance the algebraic and discretisation error; we refer to the work by [76] and to the references therein for a more elaborated insight on the topic.

4.2 An upper bound on the Error Estimator applied to generic functions

We recall standard upper bounds on the discretisation error and lower bounds on the total error (see [38] and [16]), and we prove an upper bound on the estimator defined for a generic finite element function $v_h \in V_h$, in terms of the estimator defined for the Galerkin solution and the algebraic error. Let h_T=diam(T) for $T \in \mathcal{T}_h$, h_z=diam(ω_z) for $z \in \mathcal{N}_{h,int}$, and h_E=diam(E) for $E \in \mathcal{E}_h$. Consider the mean value operator $\pi_{\omega_z} : L^1(\Omega) \longrightarrow \mathbb{R}$, $\pi_{\omega_z}(f) := \int_{\omega_z} f/|\omega_z|$.

For a given $z \in \mathcal{N}_h$, define an oscillation term

$$\text{osc}_z := |\omega_z|^{1/2}\|f - \pi_{\omega_z} f\|_{\omega_z}, \ \text{osc} := \left(\sum_{z\in\mathcal{N}_h} osc_z^2\right)^{1/2}. \tag{4.3}$$

For a given function $v_h \in V_h$, define for $E \in \mathcal{E}_h$ and $T \in \mathcal{T}_h$

$$\begin{aligned} J_E(v_h) &:= h_E^{1/2} \left\| \left[\frac{\partial v_h}{\partial n_E}\right] \right\|_E, \ J_T(v_h) := \sum_{E\in\partial T} J_E(v_h), \\ J(v_h) &:= \left(\sum_{E\in\mathcal{E}_h} J_E(v_h)^2\right)^{1/2} = \left(\frac{1}{2}\sum_{T\in\mathcal{T}_h} J_T(v_h)^2\right)^{1/2}, \end{aligned} \tag{4.4}$$

where $[\cdot]$ is the standard notation for denoting the jump of a piecewise continuous function across the edge/face E in normal direction n_E and where we've taken into consideration that when summing overall the elements each edge/face is counted twice.

Lemma 4.2.1 recalls the classical upper bound on the discretization error, which is stated and proved in [38].

Lemma 4.2.1 (**Upper bound on the discretization error** ([72]).)**.** *There exists a constant $C^\star > 0$ which depends on the shape of the triangulation, on Ω, on Γ, and which is independent of f and of the mesh-sizes h_T such that*

$$|u - u_h|_1 \leq C^\star (osc^2 + J^2(u_h))^{1/2}. \tag{4.5}$$

The a posteriori residual-based estimator in the rhs of (4.5) is made up by an oscillating-contribution (volume-contribution) that measures the variations of the

rhs function f from its mean value $\pi_{\omega_z}(f)$ on each patch ω_z, and by an edge/face-contribution that measures the jump of the gradient of the Galerkin solution across the inner edges/faces. Notice that the global upper estimate (4.5) is made up by local cell-wise estimations.

Remark 4.2.1. *The proof of* (4.5) *is based on a quasi-interpolation operator that was first introduced in [38]. It represents a modification of the classical quasi-interpolation operator due to Clément ([44]) in the setting of a partition of the unity, which has the effect that the volume contribution term* (4.3) *in the a posteriori residual based estimate* (4.5) *is smaller compared to the one in the standard estimate according to [90], [2]. The edge/face-contribution* (4.4) *dominates the residual based standard a posteriori estimates for affine finite element approximations ([38], [40]), and if the right-hand-side f is smooth, a Poincaré inequality shows that the oscillating term* (4.3) *is of higher order ([38]).*

In [16], the authors use standard bubble-function techniques of [89] to prove a global lower bound on the $|\cdot|_1$-norm distance between the true solution $u \in H_0^1(\Omega)$ and a generic function $v_h \in V_h$.

Lemma 4.2.2 (Lower bound on the total error)**.** *There exists a constant $C_\star > 0$ which only depends on the minimum angle of the triangulation, on Ω, on Γ, and which is independent of f, u, u_h and of the mesh-sizes h_T such that*

$$J^2(v_h) \leq C_\star(|u - v_h|_1^2 + osc^2) \ \forall v_h \in V_h. \tag{4.6}$$

Now we can use Lemma 4.2.1 and 4.2.1 to prove our main result for this section.

Theorem 4.2.1. *There exist positive constants C_1, C_2, C_3 that only depend on the minimum angle of the triangulation, on Ω, on Γ, and which are independent of f, u, u_h and of the mesh-sizes h_T such that*

$$J^2(v_h) \leq C_1 J^2(u_h) + C_2|u_h - v_h|_1^2 + C_3 osc^2 \ \forall v_h \in V_h. \tag{4.7}$$

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. [67]) to the lower bound (4.6)

$$\begin{aligned} J^2(v_h) &\leq C_\star(|u - v_h|_1^2 + osc^2) \\ &= C_\star(|u - u_h|_1^2 + |u_h - v_h|_1^2 + osc^2) \\ &\leq C_\star(C^{\star 2}(osc^2 + J^2(u_h)) + |u_h - v_h|_1^2 + osc^2) \\ &= C_\star C^{\star 2} J^2(u_h) + C_\star|u_h - v_h|_1^2 + C_\star(C^{\star 2} + 1)osc^2 \\ &= C_1 J^2(u_h) + C_2|u_h - v_h|_1^2 + C_3 osc^2, \end{aligned} \tag{4.8}$$

where we have used the upper bound (4.5) in (4.8). □

Remark 4.2.2. *Theorem 4.2.1 gives an upper bound on $J^2(v_h)$ where v_h is a generic function (for instance, accounting for inexact approximations) in terms of $J^2(u_h)$, the square energy norm of the algebraic error, which is equal to $|u_h - v_h|_1^2$ and oscillation terms which only depend on the triangulation and the data, but are independent of u_h and v_h.*

A related result is found in the paper [5], where the authors set the stopping criterion for the CG method by using a residual-based error estimator in the context of elliptic self-adjoint problems. They provide an upper bound on $\eta(v_h)$ in terms of $\eta(w_h)$ and $|v_h - w_h|_1$, where v_h and w_h are generic functions in V_h. However, their proof proceeds differently, and it is based on the use of the full a-posteriori error estimator, while here we prove that a similar result holds also for the case where only $J(v_h)$ is used, i.e., when only face terms are considered in the estimator.

This result, together with Theorem 3.4.4, gives us a sound theoretical basis for a smoothed AFEM algorithm, where the algebraic error $|u_h - v_h|_1^2$ in the intermediate steps is given explicitly by the error propagation formula 3.69.

More specifically, in the context of Smoothed-Multilevel Methods, let V_{k+1} be the finite element space after $k+1$ cycles and let $v_{k+1} := u_{k+1}^l$ denote the finite element function, which has discrete corresponding vector $\mathbf{u}_{k+1}^l$. Then, it can be easily proved that $|u_{k+1} - u_{k+1}^l|_1^2$ is exactly the A_{k+1}-square energy norm $\|\mathbf{u}_{k+1} - \mathbf{u}_{k+1}^l\|_{A_{k+1}}^2 = \|\mathbf{e}_{k+1}^{(l)}\|_{A_{k+1}}^2$, which has been quantified rigorously in Theorem 3.4.4.

4.3 Dominating terms in the expression of the edge contribution of the estimator

In this section we briefly comment and report some ideas that are an ongoing work in [71]. Let us –for the reader's sake– rewrite the expression of the edge contribution in the expression of the residual based error estimator for a given function $v_h \in V_h$, for $E \in \mathcal{E}_h$ and $T \in \mathcal{T}_h$

$$\begin{aligned} J_E(v_h) &:= h_E^{1/2} \left\| \left[\frac{\partial v_h}{\partial n_E} \right] \right\|_E, \;\; J_T(v_h) := \sum_{E \in \partial T} J_E(v_h), \\ J(v_h) &:= \left(\sum_{E \in \mathcal{E}_h} J_E(v_h)^2 \right)^{1/2} = \left(\frac{1}{2} \sum_{T \in \mathcal{T}_h} J_T(v_h)^2 \right)^{1/2}. \end{aligned} \tag{4.9}$$

Motivated by the numerical evidences in Chapter 5, we want to investigate qualitatively the action of the jump operator on eigenfunctions of different frequencies.

Let w_i be an eigenfunction of the eigenvalue problem corresponding to Problem 2.18, i.e.

$$(\nabla w_i, \nabla v_h) = \lambda_i (w_i, v_h), \;\; \forall v_h \in V_h, \; \forall i = 1, \ldots, N. \tag{4.10}$$

- We want to investigate the action of the estimator on an eigenfunction. In particular, we expect to obtain that

$$J^2(w_i) \approx \lambda_i^2 w_i^2. \tag{4.11}$$

  This result would nicely fit with the observed numerical results in Section 5 that the error estimator is small for small eigenvalues and big for big eigenvalues.

- From the operatorial point of view, we are interested in investigating the relationship between the A^TA-norm of $\mathbf{u}_h$ and $J(u_h)^2$. Again, we suspect that

$$\|\mathbf{u}_h\|_{A^TA} \approx J^2(u_h). \tag{4.12}$$

However, this is still an ongoing work and more specific details followed by the proofs will be provided in [71].

Chapter 5

Smoothed Adaptive Finite Element Method (S-AFEM)

In this chapter, we introduce and describe the Smoothed AFEM algorithm (S-AFEM). To fix the ideas, we provide a small discussion with some empirical numerical evidence that justifies the use of S-AFEM in Section 5.1, which is explained in detail in Section 5.2. The discussion and results in this chapter are taken by [72].

For completeness, we report here the error propagation formula (3.69):

$$\mathbf{e}_{k+1}^{(l)} = {M_{k+1}}^{l} \left(\mathbf{a}_{k+1} + \sum_{j=2}^{k} I_j^{k+1} L_j \mathbf{a}_j + I_2^{k+1} L_2 I_1^2 \mathbf{e}_1 + \sum_{j=2}^{k} \mathbb{D}_{k+1,j} \mathbf{a}_j + \mathbb{D}_{k+1,2} I_1^2 \mathbf{e}_1 \right).$$

We observe that the presence of M_{k+1}^l in front of the error expression guarantees that high frequencies would be damped very quickly by the use of Richardson smoothing. On the other hand, the largest part of the low frequency error is given by the term

$$I_2^{k+1} L_2 I_1^2 \mathbf{e}_1,$$

and by the accumulation of the error in low frequency that is due to the difference between the exat algebraic solutions in the different levels

$$\sum_{j=2}^{k} I_j^{k+1} L_j \mathbf{a}_j.$$

Of all terms, the ones that contain $\mathbf{e}_1$ could be controlled easily (and in a computationally inexpensive way), by ensuring that first iteration of AFEM is solved accurately, i.e., considering $\mathbf{e}_1 = 0$.

In particular, it is still acceptable to have a large difference between u_h and u_h^l (implying a large a posteriori error estimate on the algebraic approximation $J(v_h)$), provided that this difference is *roughly* equally distributed over the grid,

since a (almost) constant difference between $\eta_T(u_h)$ and $\eta_T(v_h)$ for all T would result in (almost) the same cells marked for refinement.

5.1 Some Numerical Evidences

To fix the ideas, we consider the Peak Problem in two dimensions as described in Subsection 5.3.1, and we apply ten cycles of the classical AFEM Algorithm 1.1 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. We use standard residual-based a posteriori error estimators (4.4) which are locally defined through the jump of the gradient of the discrete approximation across the edges/faces E of the cells (cf. Section 2.3). In Figure 5.1 we plot the ℓ^2-norm of the residual $\mathbf{r}_k^{(l)} := A_k \mathbf{e}_k^{(l)}$ and the value of the estimator $\eta(u_k^l)$ for all cycles as the Richardson iteration count l increases from 1 to 30. The same behaviour is present in every refinement cycle. The residual norm shows two distinguishable speeds of convergence. The first few iterations induce a rapid drop in the residual norm (due to convergence of the highly oscillatory terms in the solution), while the second part of the iterations converge very slowly, corresponding to the convergence speed of the low frequency in the solution. The estimator, on the other hand, stagnates after very few Richardson iterations (around two or three). In other words, $J(u_h^l)$ is almost the same as $J(u_h)$ for $l \geq 3$, empirically suggesting that the error estimator (4.4) is mainly affected by the highly oscillatory components of the discrete algebraic solution u_h^l, and that the estimate provided by Theorem 4.2.1 may be improved by exploiting the structure of the algebraic iterative solution in Richardson iteration provided by Theorem 3.4.4.

Although the value we plot in Figure 5.1 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 behaviour of such 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 Section 5.3.

Motivated by these numerical evidences and by the earlier observations, 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.

5.2 Smoothed Adaptive Finite Element Algorithm

We present the *Smoothed Adaptive Finite Element* algorithm (S-AFEM), where the exact algebraic solution in intermediate steps is replaced by the application

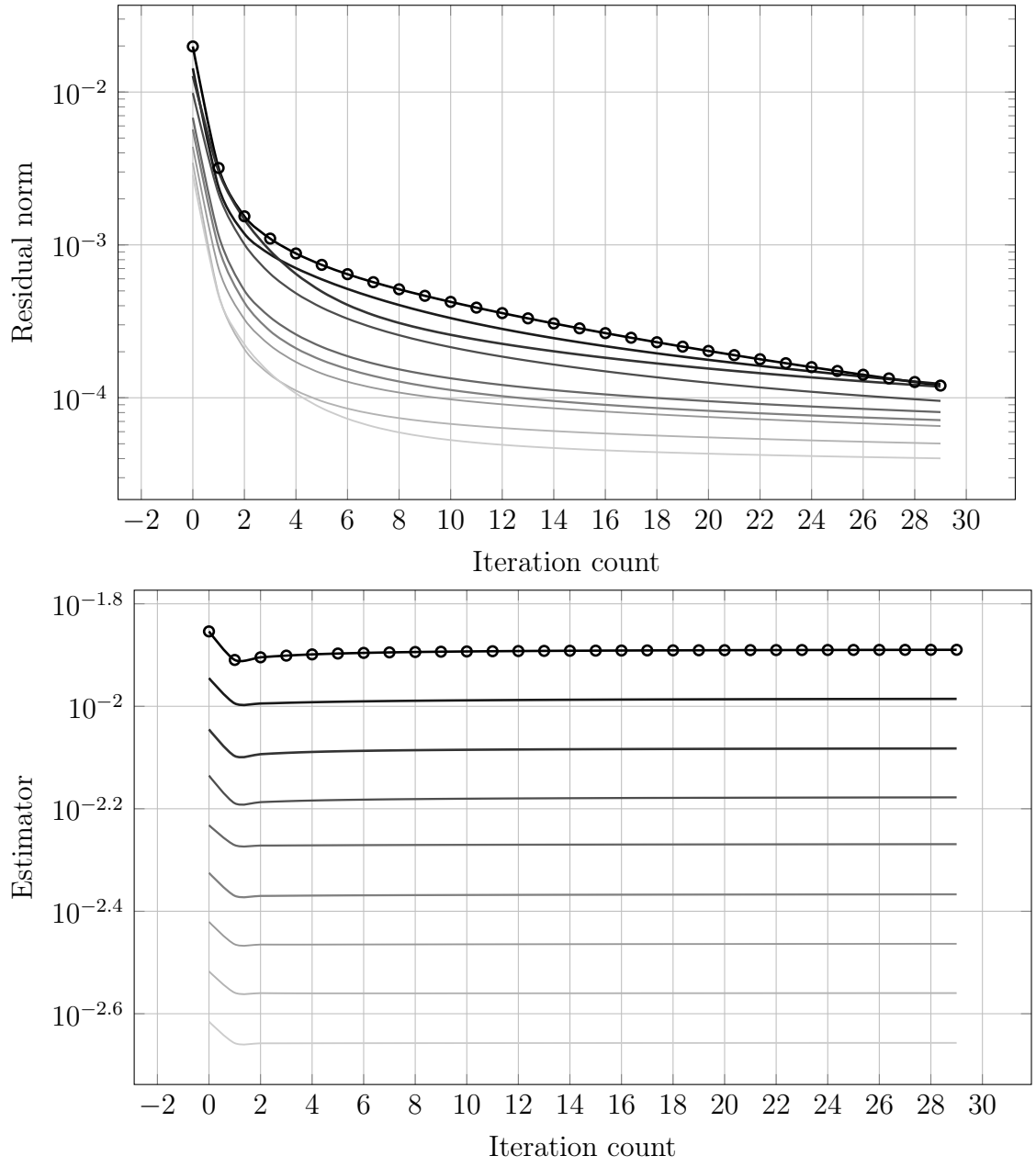
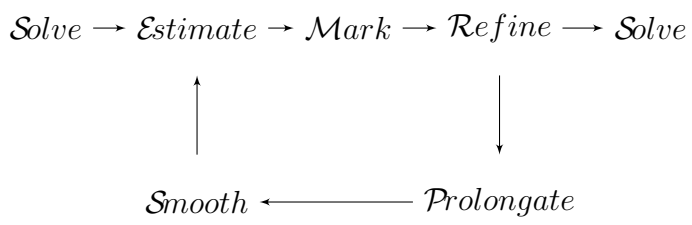


Figure 5.1: Residual norm (top) and Estimator (bottom) for intermediate cycles of the classical AFEM algorithm when using Richardson iteration without preconditioner as a solver, with prolongation from the previous solution as starting guess. Darker lines correspond to earlier cycles. Only the first 30 iterations are shown.

of a prolongation step ($\mathcal{P}rolongate$), followed by a smoothing step ($\mathcal{S}mooth$):



The first and last steps of the S-AFEM algorithm coincide with a classical AFEM. In the intermediate steps, however, the solution of the algebraic problem is replaced by a prolongation step ($\mathcal{P}rolongate$) followed by a fixed number of smoothing iterations ($\mathcal{S}mooth$).

The strategy consists precisely in

1. solving exactly the linear algebraic system derived from the discrete problem on the coarsest level $k = 1$ and on the finest level $k = \bar{k}$;

2. applying a few smoothing iterations on the linear algebraic system in the intermediate levels $k = 2, \dots, \bar{k} - 1$ by using the prolongation of the approximation from the previous level as an initial guess;

3. executing the $\mathcal{E}stimate$ and $\mathcal{R}efine$ steps on the approximate solutions for $k = 2, \dots, \bar{k} - 1$.

In particular, we propose Algorithm 5.

Algorithm 5: Smoothed-adaptive mesh-refining algorithm ([72]) .

Input: initial mesh $\mathcal{T}_1$
Step k=1: Solve $\mathcal{A}_1 u_1 = f_1$ based on $\mathcal{T}_1$.
Loop: for $k = 2, \dots, \bar{k} - 1$ do steps $1 - 4$

1. $\mathcal{S}mooth$: Compute l smoothing iterations on the discrete system $A_k \mathbf{u}_k = \mathbf{f}_k$, with initial guess $\mathbf{u}_k^{(0)} := I_{k-1}^k \mathbf{u}_{k-1}^{(l)}$, which produce $\mathbf{u}_k^{(l)} \in \mathbb{R}^{N_k}$ with corresponding $u_k^l \in V_k$.

2. $\mathcal{E}stimate$: Compute $\eta_T^2(u_k^l)$ for all $T \in \mathcal{T}_k$.

3. $\mathcal{M}ark$: Choose set of cells to refine $\mathcal{M}_k \subset \mathcal{T}_k$ based on $\eta_T^2(u_k^l)$.

4. $\mathcal{R}efine$: Generate new mesh $\mathcal{T}_{k+1}$ by refinement of the cells in $\mathcal{M}_k$.

Step $k = \bar{k}$: Solve the discrete system $A_{\bar{k}} \mathbf{u}_{\bar{k}} = \mathbf{f}_{\bar{k}}$.
Output: sequence of meshes $\mathcal{T}_k$, smoothed approximations u_k^l, and estimators $\eta(u_k^l)$, final adapted-approximation $u_{\bar{k}}^l$.

In step $k = 1$, we capture the smoothest (i.e. less oscillatory) part of the discrete approximation 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. This is formalized by the Non-interacting Frequency Coupling Hypothesis for smoothed-multilevel methods (3.68).

The advantage of S-AFEM is that, on one hand, we save a substantial amount of computational time that would be needed for the algebraic solution in the intermediate steps, and on the other hand we obtain roughly the same mesh-sequence, hence the same refinement at each step, with an accuracy on the final approximation step that is comparable to the classical AFEM algorithm, at a fraction of the computational cost.

5.3 Numerical validation

The numerical results presented in this work were realized using a custom `C++` code based on the `deal.II` library [13, 3], and on the `deal2lkit` library [81]. We consider two classical experiments used to benchmark adaptive finite element methods. In our implementation we use the classical marking strategy following [50], which we described in Section 2.3. 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" (see [13] for a detailed discussion on the implementation details).

5.3.1 Two-dimensional examples

Smooth domain, peak right hand side, two dimensions. The first example we consider consists in solving the model problem 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

$$u(x,y) = x(x-1)y(y-1)e^{-100\left((x-0.5)^2+(y-0.117)^2\right)}, \tag{5.1}$$

(see Figure 5.2).

Fichera corner domain, smooth right hand side, two dimensions. In the second two-dimensional test case, we consider a Fichera corner 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)

$$u(r,\theta) = r^{2/3}\sin\left(\frac{2\theta+5\pi}{3}\right), \tag{5.2}$$

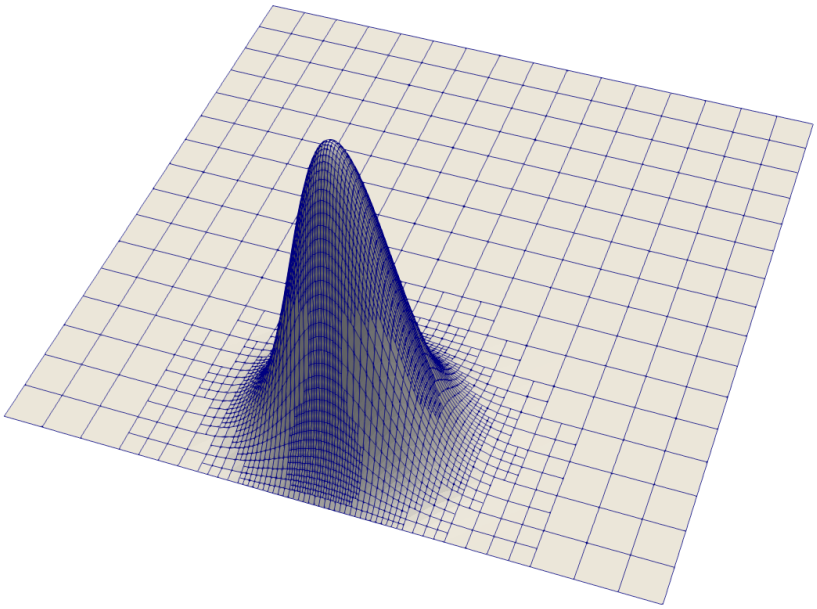


Figure 5.2: Solution to peak problem (5.1) in two dimensions.

as shown in Figure 5.3.

In all cases, we apply ten cycles of classical AFEM and of S-AFEM. For the AFEM algorithm, we use the CG method as iterative solver, with an 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 cylces 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 two-dimensional Fichera corner problem in Figure 5.4, and after nine cycles for the two-dimensional peak problem in Figure 5.7. In both cases, the set of marked cells, although different in some areas, produces a refined grid that is very similar between the classical AFEM and the S-AFEM, and where the accuracy of the final solution is comparable.

In Figures 5.5 and 5.8 we compare the values of the global estimators $J(u_h)$ and $J(u_h^l)$ and of the H_1 semi-norm of the total errors for each cycle for the two-dimensional peak problem, and for the two-dimensional Fichera corner problem when using S-AFEM. For reference, Figures 5.6 and 5.9 show the error and the estimator in the classical AFEM algorithm for the two examples. Notice that the first step of AFEM and 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 of the one that would be obtained by solving using CG and AMG ($J(u_h)$ in Figures 5.5 and 5.8), showing that in the two dimensional setting the error estimator (4.4) 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

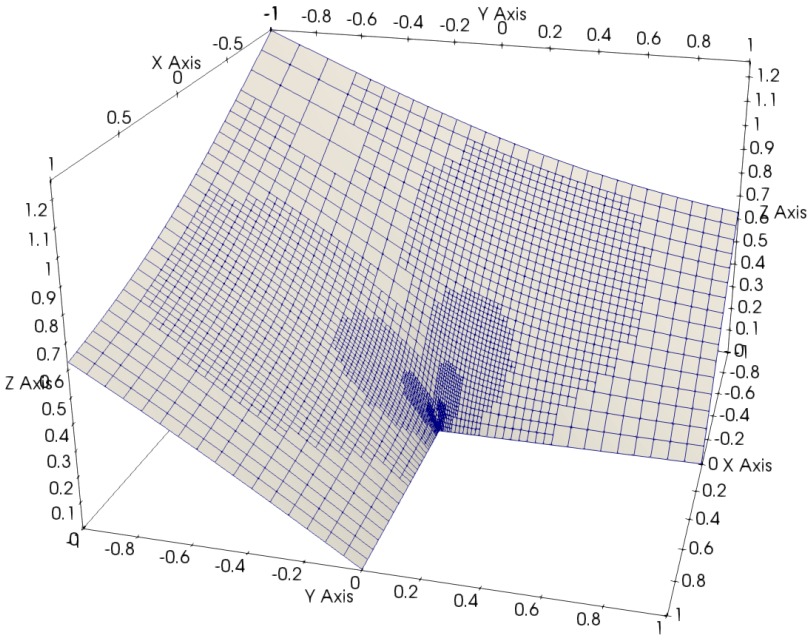


Figure 5.3: Solution to the Fichera corner problem (5.2) in two dimensions.

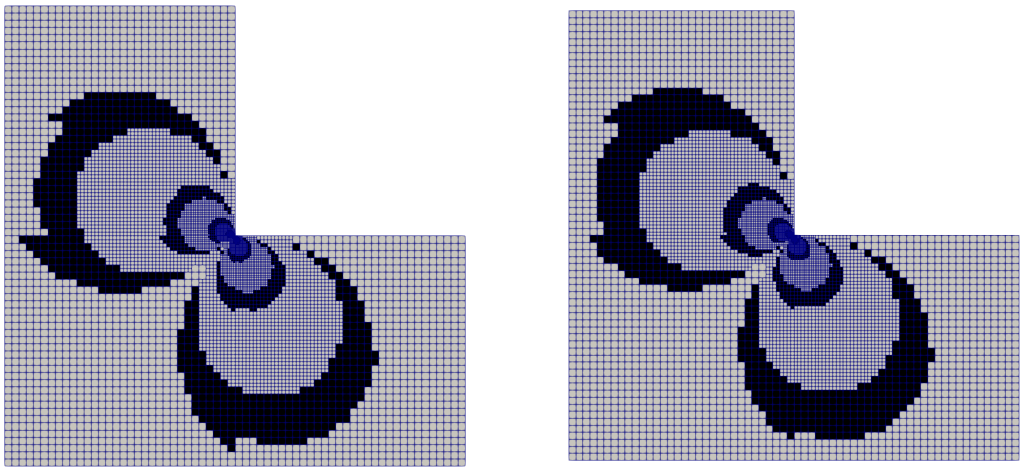


Figure 5.4: Comparison between the cells marked for refinement in AFEM (on the left) and S-AFEM (on the right) after 5 cycles.

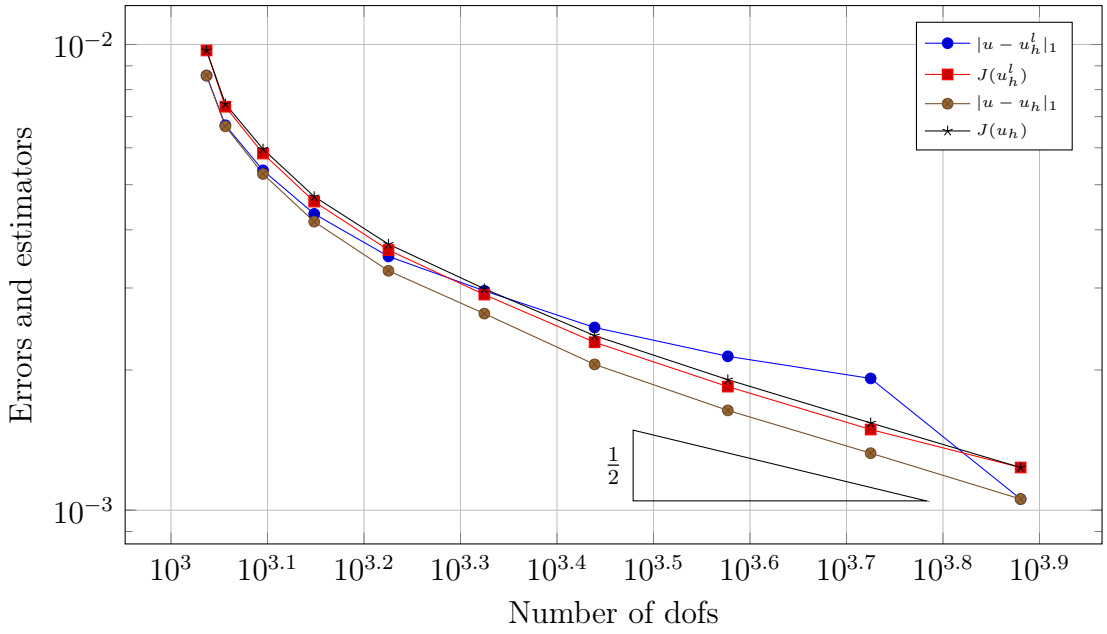


Figure 5.5: True error, algebraic error, and estimator for the Peak problem in two dimensions.

the algebraic problem until convergence, as quantified by Theorem 3.4.4. This error measures the distance between the exact algebraic solution and the smooth non-oscillatory 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 converge using CG and AMG, as in the first cycle, and we obtain a solution whose error is controlled by the estimator, as expected.

5.3.2 Three-dimensional examples

Smooth domain, peak right hand side, three dimensions. The first three-dimensional 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 (see Figure 5.10):

$$u(x,y,z) = x(x-1)y(y-1)z(z-1)e^{-100\left((x-0.5)^2+(y-0.117)^2+(z-0.331)^2\right)}. \tag{5.3}$$

Fichera Corner, smooth right hand side, three dimensions. In the second three-dimensional example, we consider again 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

$$u(r,\theta,\phi) = r^{1/2}, \tag{5.4}$$

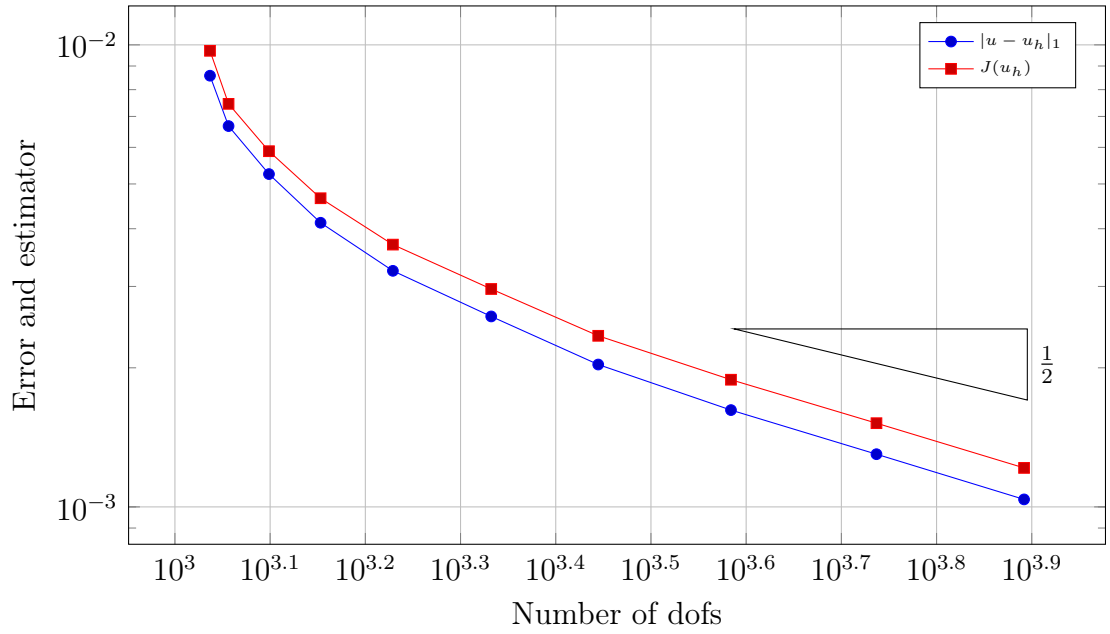


Figure 5.6: Error and estimator for the Peak problem in two dimensions, using classical AFEM.

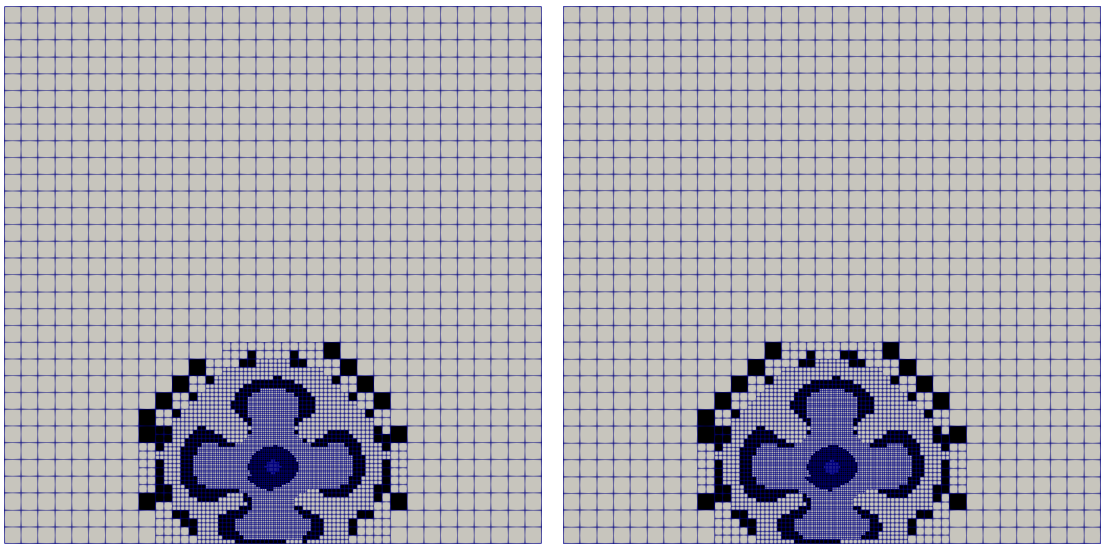


Figure 5.7: Comparison between the cells marked for refinement in AFEM (on the left) and Smoothed-AFEM (on the right) after 9 cicles.

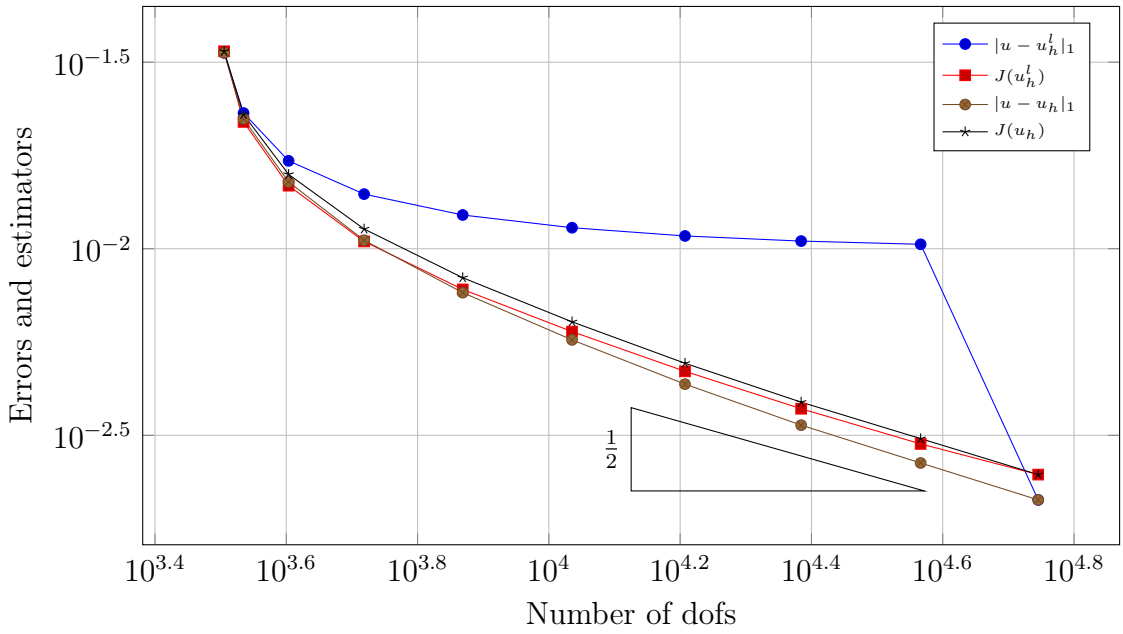


Figure 5.8: True error, algebraic error, and estimator for the Fichera corner in two dimensions.

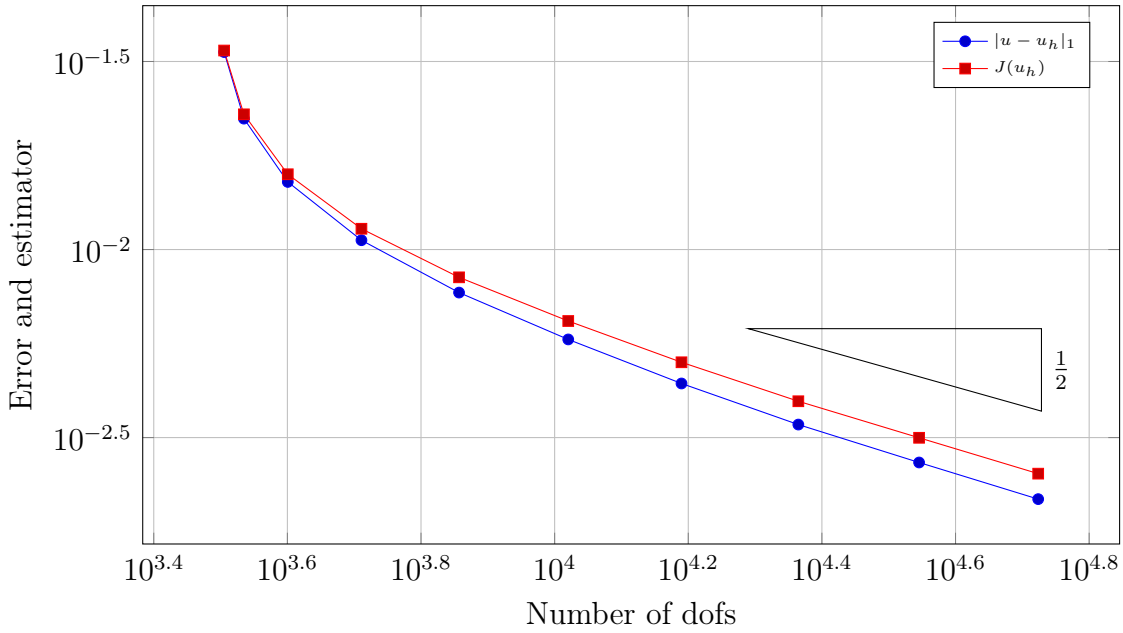


Figure 5.9: Error and estimator for the Fichera corner problem in two dimensions, using classical AFEM.

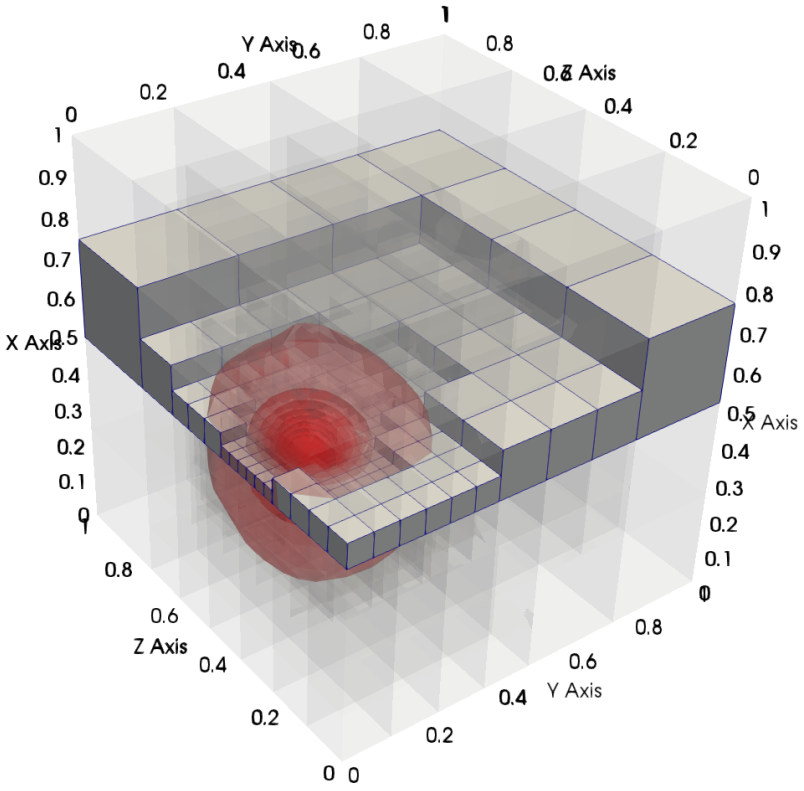


Figure 5.10: Solution to peak problem (5.3) in 3D.

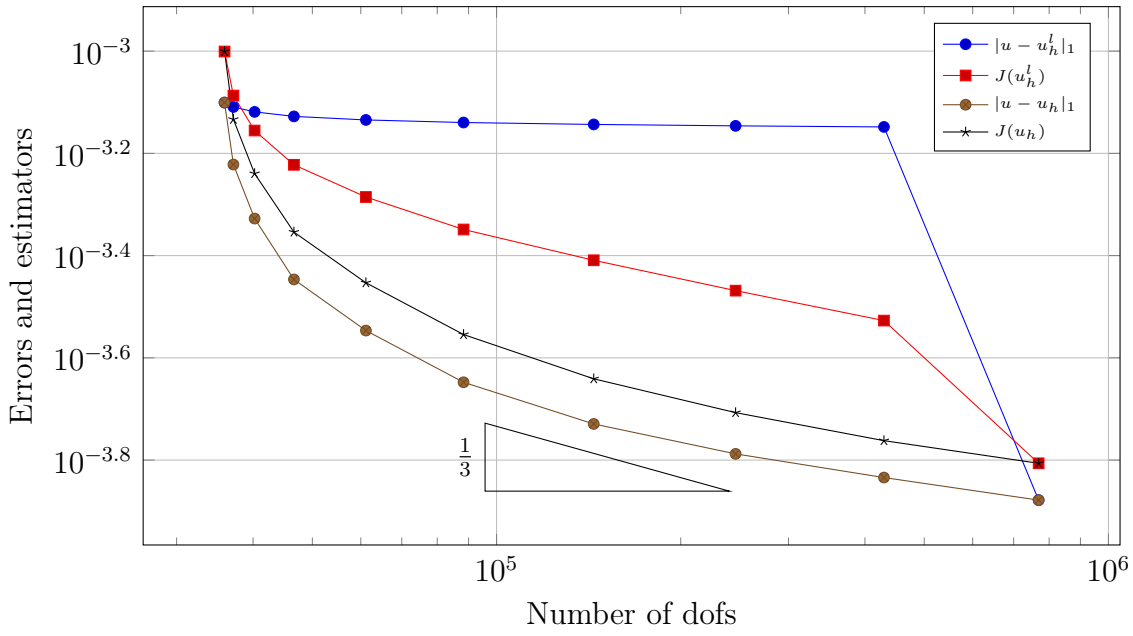


Figure 5.11: True error, algebraic errors, and estimator for the Peak problem in three dimensions, with three smoothing steps.

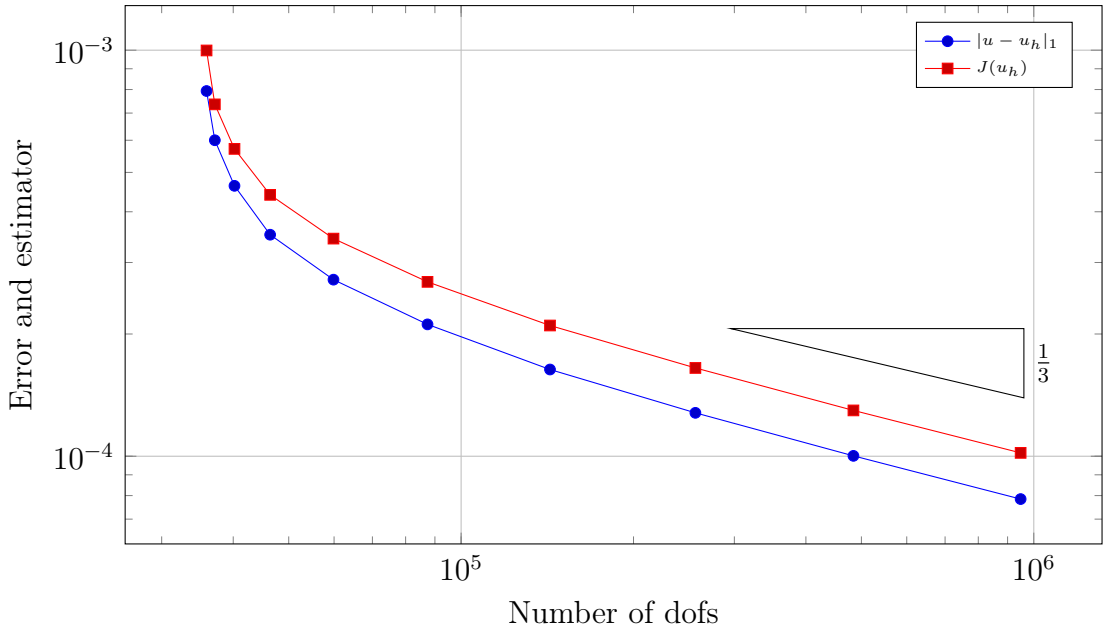


Figure 5.12: Error and estimator for the Peak problem in three dimensions, with classical AFEM.

and we add a forcing term that induces the above exact solution (see Figure 5.13).

In both examples, the estimator applied to the algebraic solution after three smoothing steps (see Figures 5.11 and 5.14) seems to be more sensitive to the low frequency content of u_h^l. In other words, in the three-dimensional case the combination of Theorem 3.4.4 and Theorem 4.2.1 provides a sharper estimate. This may also be related to the fact that the increase on the number of degrees of freedom between successive cycles in the three-dimensional setting is much more severe w.r.t. the two-dimensional case, maybe hindering the non-interacting frequency coupling hypothesis. Nonetheless, the difference in accuracy at the final step between AFEM and S-AFEM is negligible also in the three-dimensional case, showing that the (small) differences in the refinement patterns between AFEM and S-AFEM do not influence the final accuracy.

5.3.3 Computational costs

In the following table we show a comparison of the computational cost associated to the classical AFEM and to the smoothed AFEM, for the four examples we presented in the previous 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 5.1 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 solves (Richardson) in the table), with the corresponding computational cost for standard AFEM (Intermediate solves (CG)

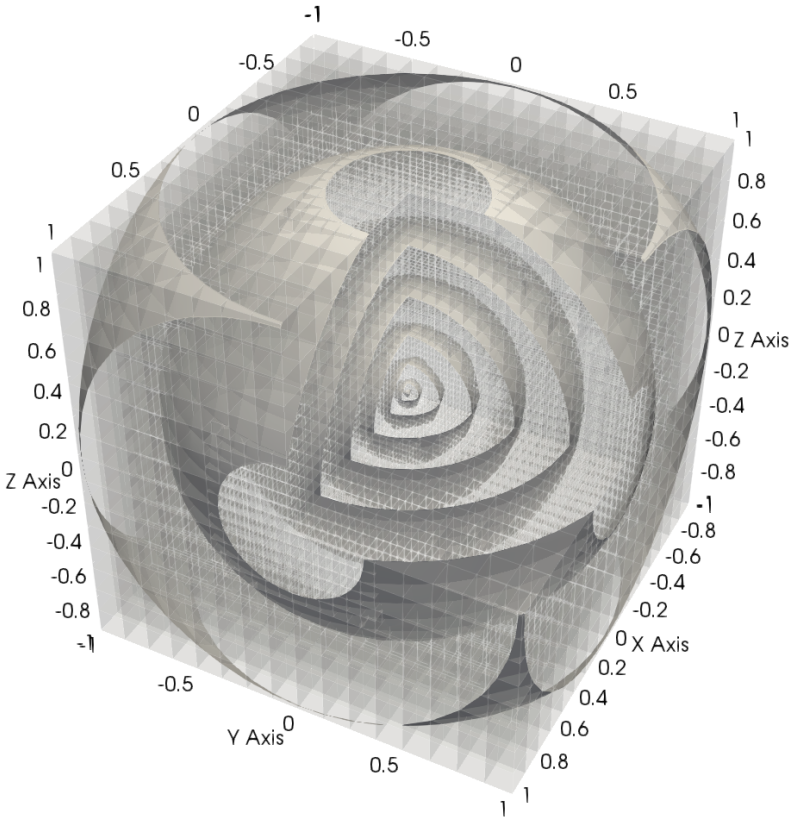


Figure 5.13: Solution to corner domain problem (5.4)

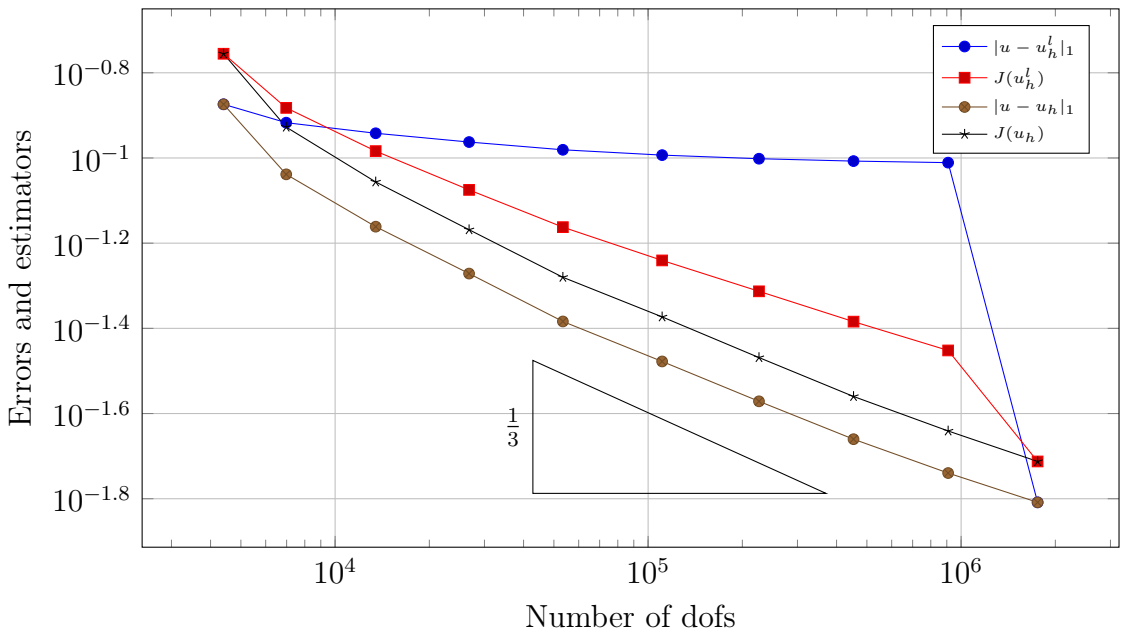


Figure 5.14: True error, algebraic errors, and estimator for the Fichera corner problem in three dimensions, using three smoothing steps.

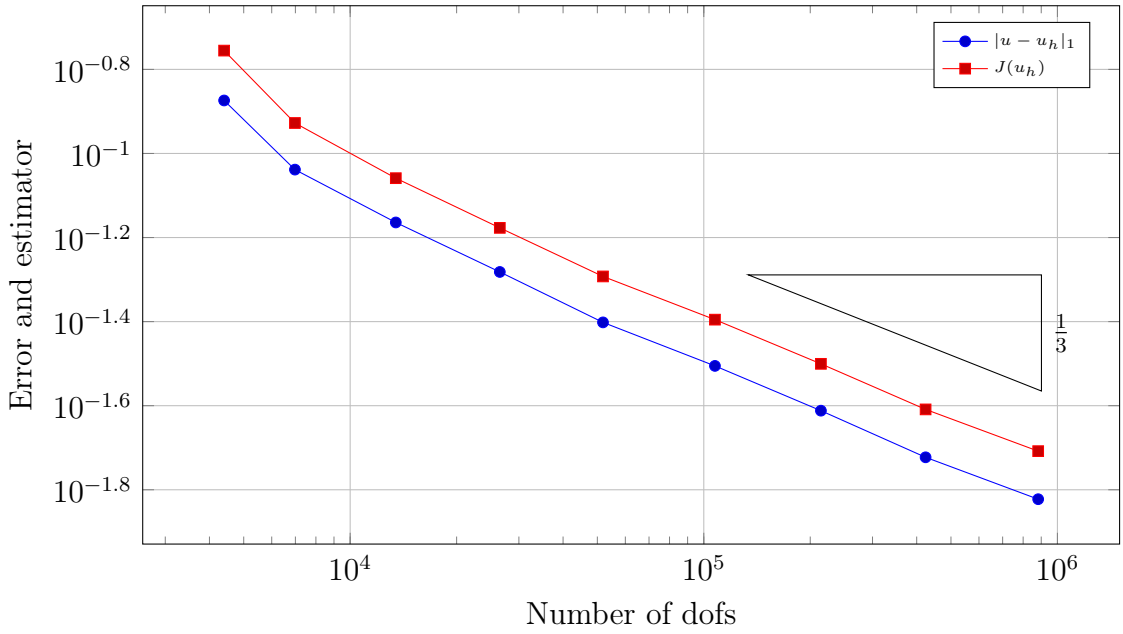


Figure 5.15: Error and estimator for the Fichera corner problem in three dimensions, using classical AFEM.

	Peak 2d	Corner 2d	Peak 3d	Corner 3d
First and last solve	0.0187s	0.0601s	32s	101s
Intermediate solves (CG)	0.0663s	0.219s	76.4s	185s
Intermediate solves (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

Table 5.1: Comparison of the computaional cost of the solution stage for ten cycles of adaptive refinement using classical AFEM and S-AFEM

in the table). The first and last solve 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.

5.4 S-AFEM for high order FEM and with different smoothers.

In this section we consider different variants of our algorithm S-AFEM, where different smoothers are considered for the intermediate cycles, respectively Richardson iteration, the CG method, and the GMRES method. Moreover, we investigate the accuracy and of S-AFEM for high order finite element discretisations and we test the accuracy of our algorithm S-AFEM for our numerical examples in all the above mentioned cases. In conclusion, S-AFEM turns out to be a good strategy also for higher order finite element discretisations, and one could use directly the CG method (or, alternatively,the GMRES method) as smoothers for the intermediate iterations. Our numerical evidences show that two smoothing iterations are enough for the two dimensional case, and around five smoothing for the three dimensional case, independently on the polynomial degree of the finite element approximation.

5.4.1 Two-dimensional examples

We apply both AFEM and S-AFEM to the two-dimensional Corner Problem (5.2) and we show a comparison for different fixed FEM degrees, as $deg = 1, \ldots, 5$, 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 of smoothing iterations l increases from 1 to 5 in Figures 5.16-5.46.

For bi-linear linear finite elements (cf. Figures 5.16-5.21), all considered smoothers (Richardson, CG, and GMRES) turn out to be good smoothers for intermediate levels. In all cases, the estimator $J(u_h^l)$ as $l = 1, \ldots, 5$ exhibits the same behaviour (same order of convergence) of the estimator $J(u_h)$ (see Figures 5.16, 5.18 and 5.20), 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 5.19, while Richardson behaves the worst (cf. Figure 5.17). Nevertheless, in all cases the accuracy of the final approximation for the last cycle obtained by S-AFEM, is almost the same to 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 much 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 refinement cycle, independent on the problem type.

For $deg = 2$, all smoothers work well exhibiting a quasi optimal convergence order compared to classical AFEM, as shown in Figures 5.22, 5.24, and 5.26, and the accuracy of the final approximation is almost the same, as shown in Figures 5.23, 5.25 and 5.27. For $deg = 3, 4$, and 5, Richardson iteration turns out to be a bad smoother for S-AFEM, unless further tuning of the relaxation parameter γ is performed. Although $J(u_h^1)$ seems to exhibit the same behaviour of $J(u_h)$, as the smoothing iteration count l increases, contrarily to what one might expect, the value of the estimator increases with increasing DOFs (see Figures 5.29, 5.30, 5.35, 5.36, 5.41 and 5.42), showing that our selection of γ is not correct for these problems, and we should estimate in a better way the spectral radius of the final matrix A, and modify γ accordingly.

On the other hand, both the CG method and the GMRES method turn out to be good smoothers for our S-AFEM, without the need to tune any parameter, as evidenced in Figures 5.24, 5.26, 5.31, 5.33, 5.37, 5.39, 5.43 and 5.45. In both cases, in fact, $J(u_h^l)$, for $l \geq 2$, 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, as evidenced in Figures 5.25, 5.27, 5.32, 5.34, 5.38, 5.40, 5.44 and 5.46.

Next, we consider the two-dimensional Peak Problem (5.1) to which we apply both AFEM and S-AFEM we provide a comparison for different fixed FEM degrees, as deg $= 1, \ldots, 5$, 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 of smoothing iterations l increases from 1 to 5 in Figures 5.47-5.76.

As a marking strategy, we choose a marking criterion where a fraction of 1/3 of cells with the largest error indicators are selected for refinement.

For piecewise bi-linear finite elements (cf. Figures 5.47-5.52), all considered smoothers (Richardson, CG and GMRES) turn out to be very good smoothers for the intermediate levels. In all cases, the estimator $J(u_h^l)$ as $l = 1, \ldots, 5$ exhibits exactly the same order of convergence of the estimator $J(u_h)$ (see Figures 5.47, 5.49 and 5.51), showing that one or two smoothing iterations would be enough for the intermediate cycles. When we look at the total error, the CG and the GMRES behave better leading to less error accumulated at the intermediate levels, as shown in Figures 5.50 and 5.52, while Richardson behaves less well as

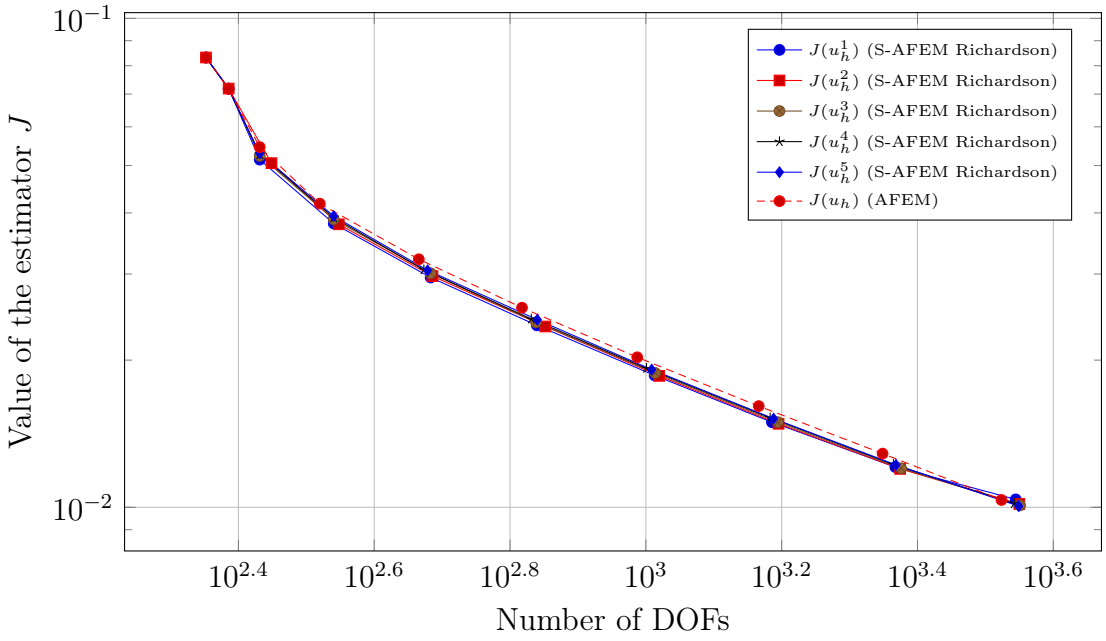


Figure 5.16: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson iterations as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

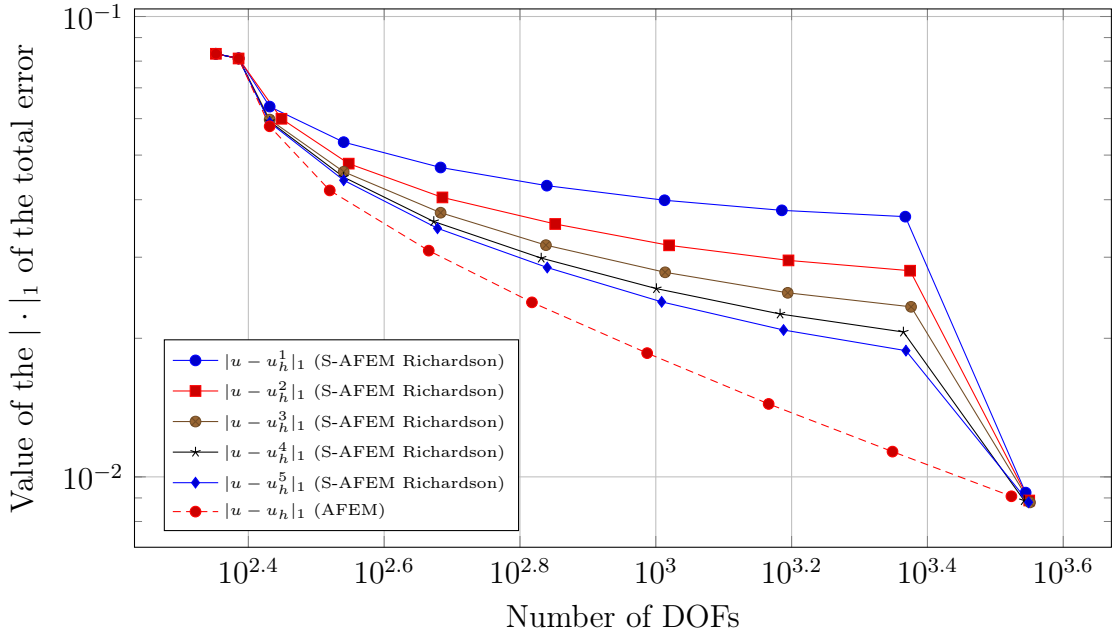


Figure 5.17: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

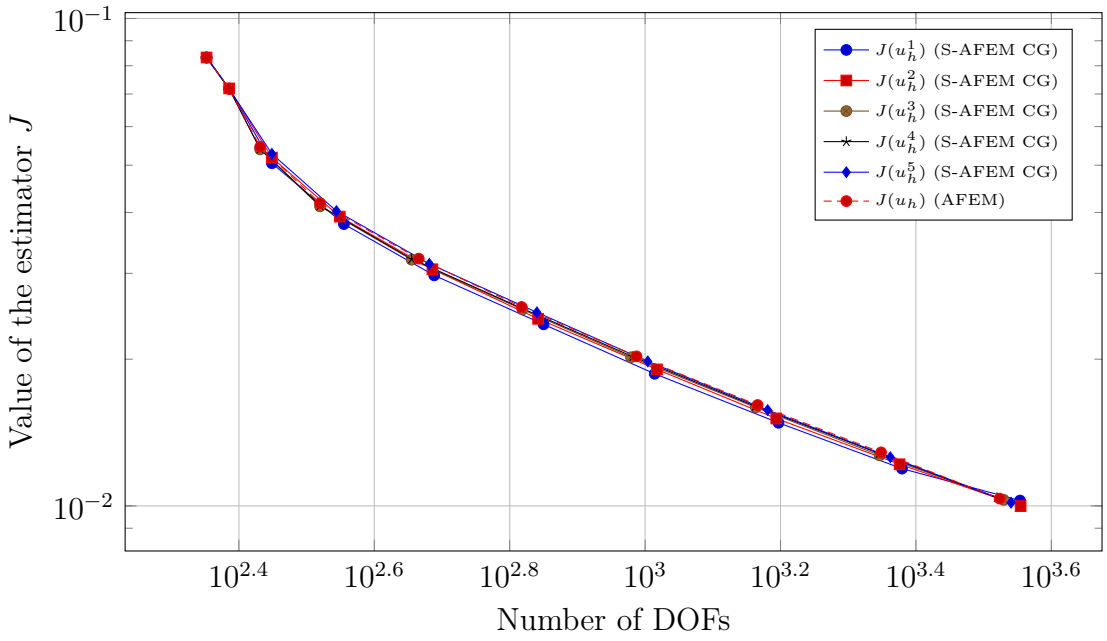


Figure 5.18: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

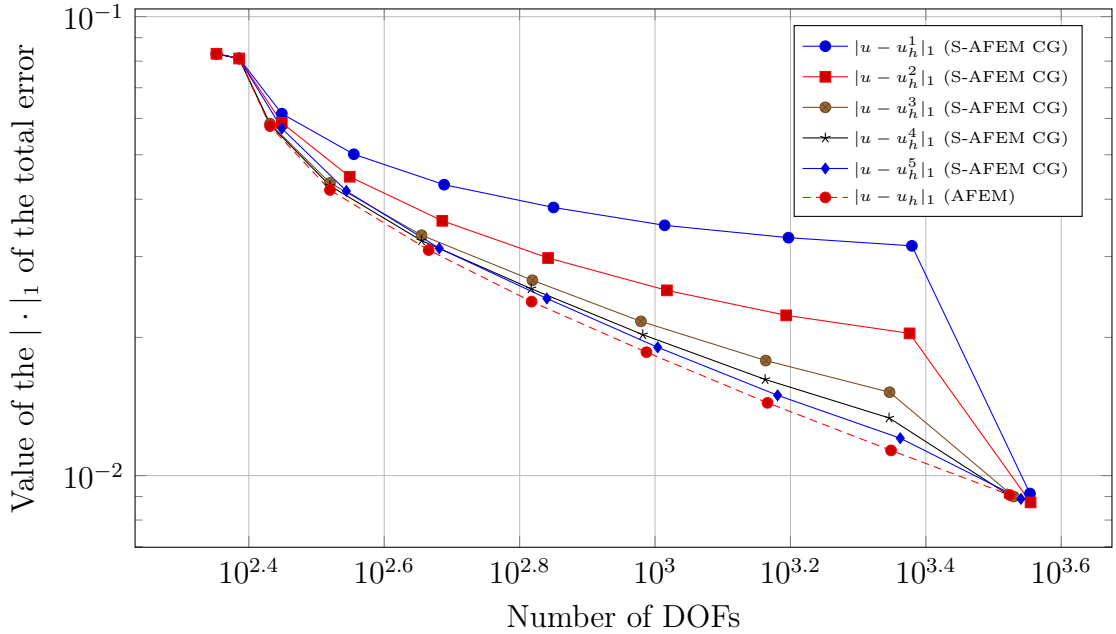


Figure 5.19: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

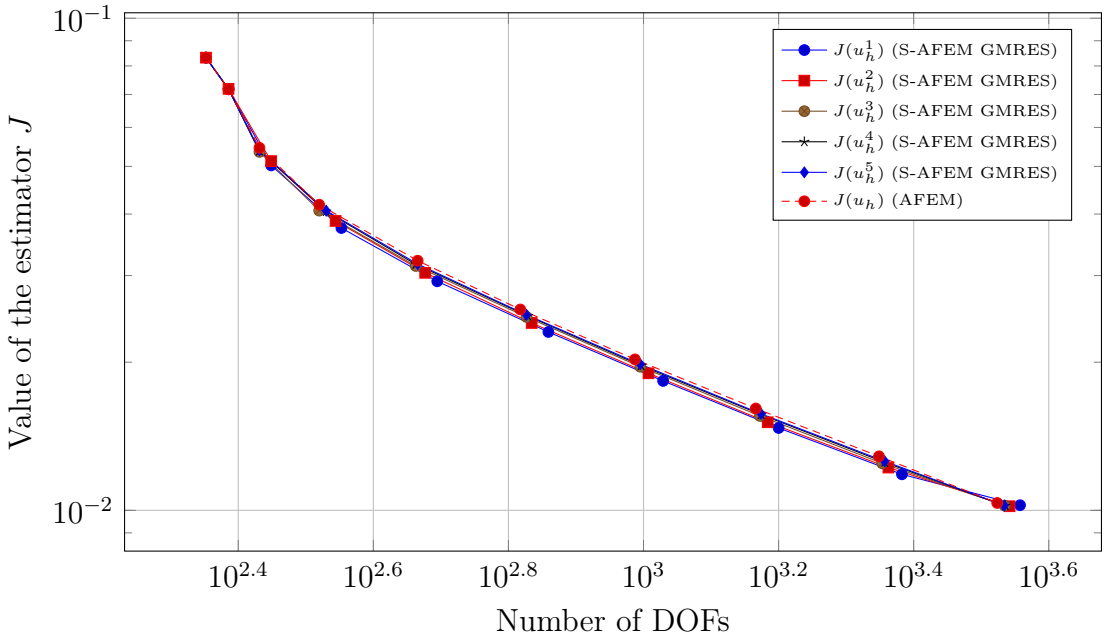


Figure 5.20: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

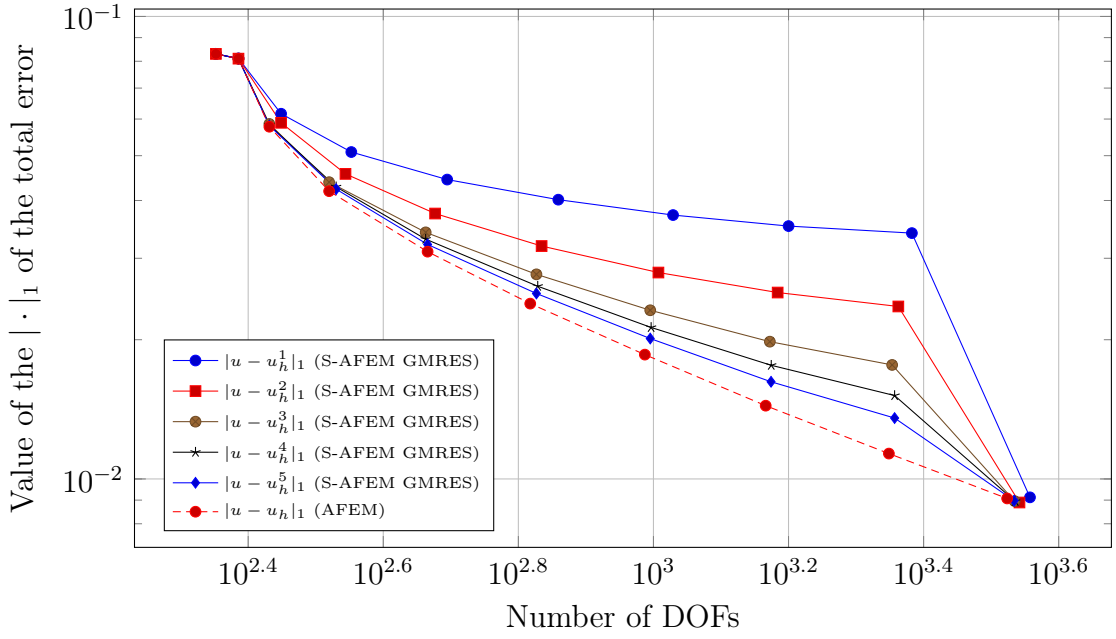


Figure 5.21: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and the Dörfler parameter for the marking of the cells is $\Theta = 0.3$.

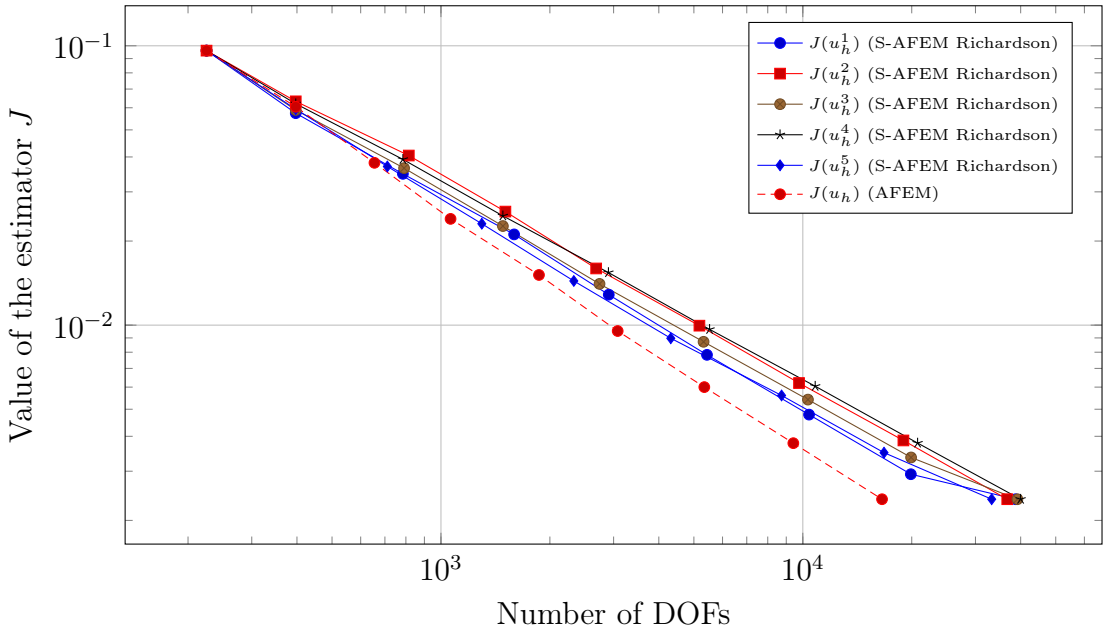


Figure 5.22: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

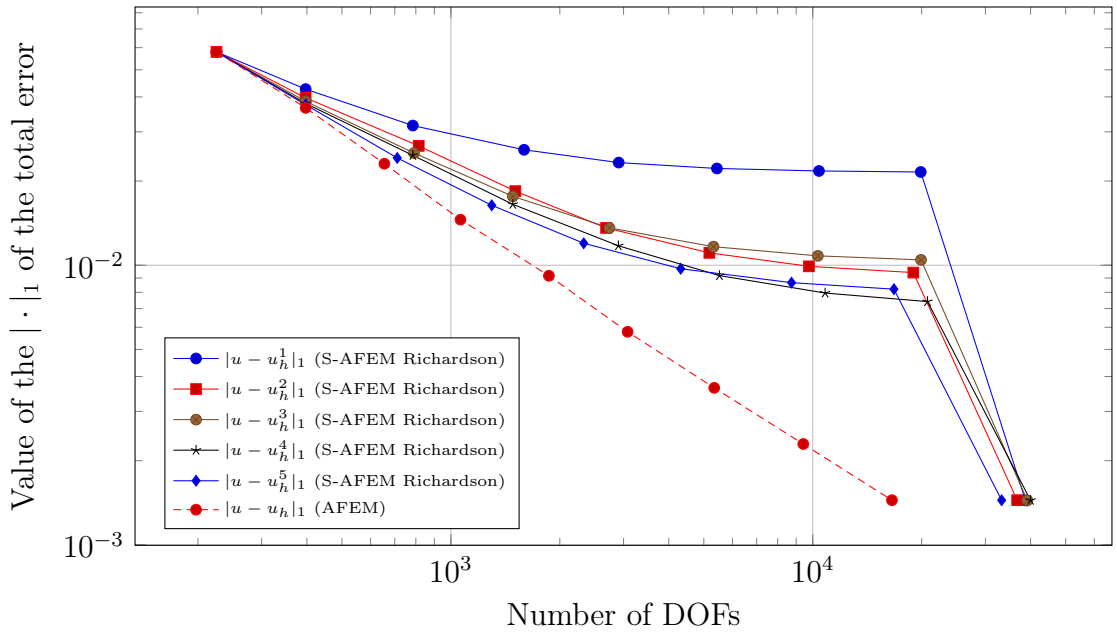


Figure 5.23: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

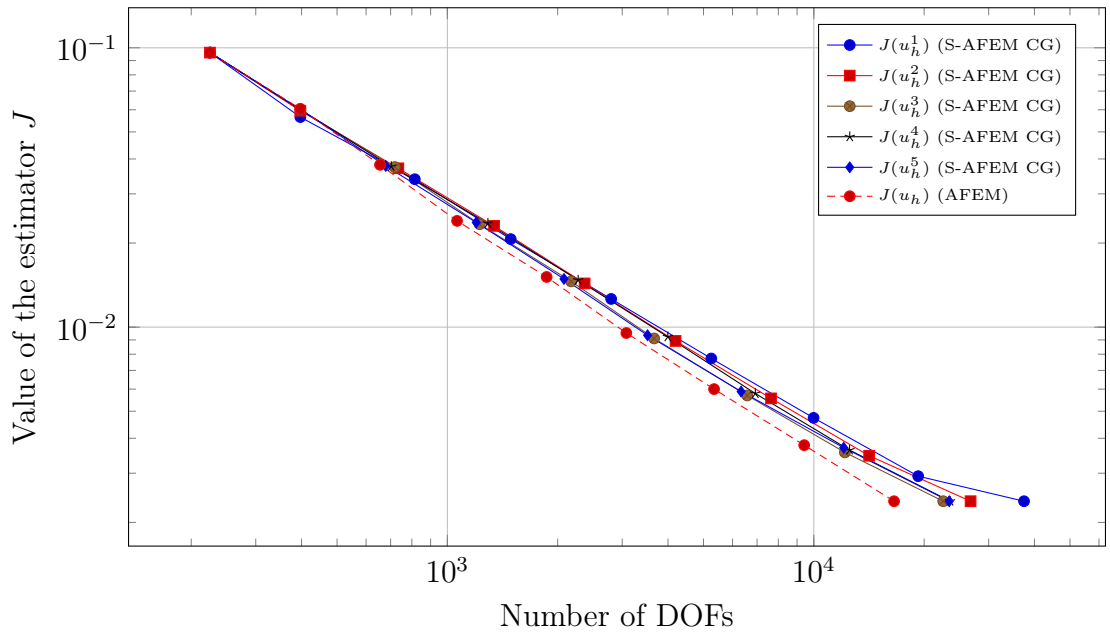


Figure 5.24: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

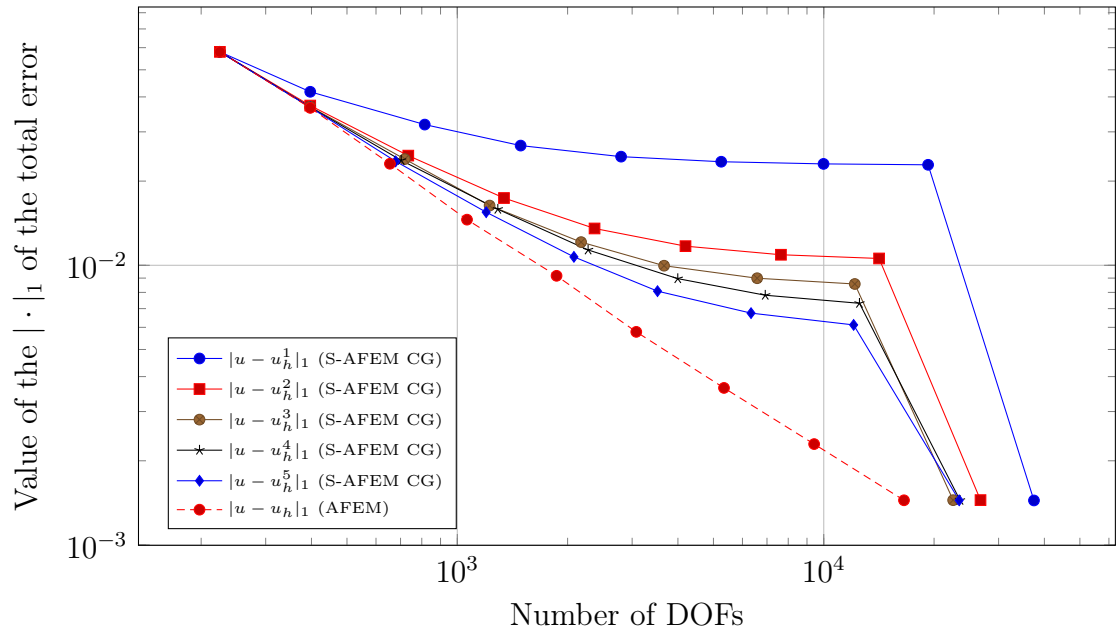


Figure 5.25: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

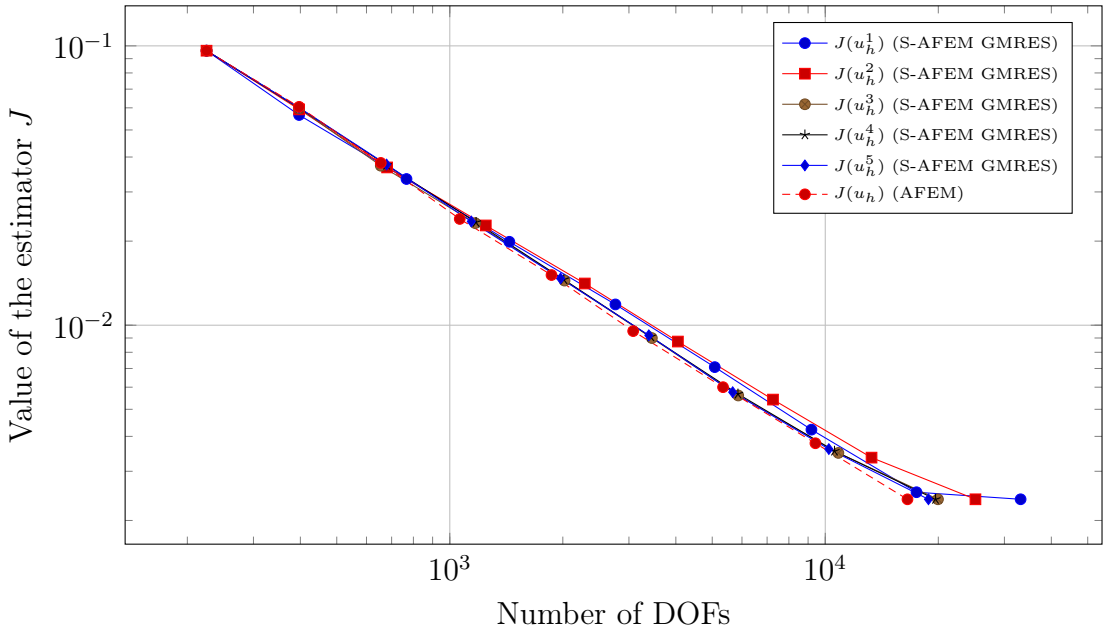


Figure 5.26: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

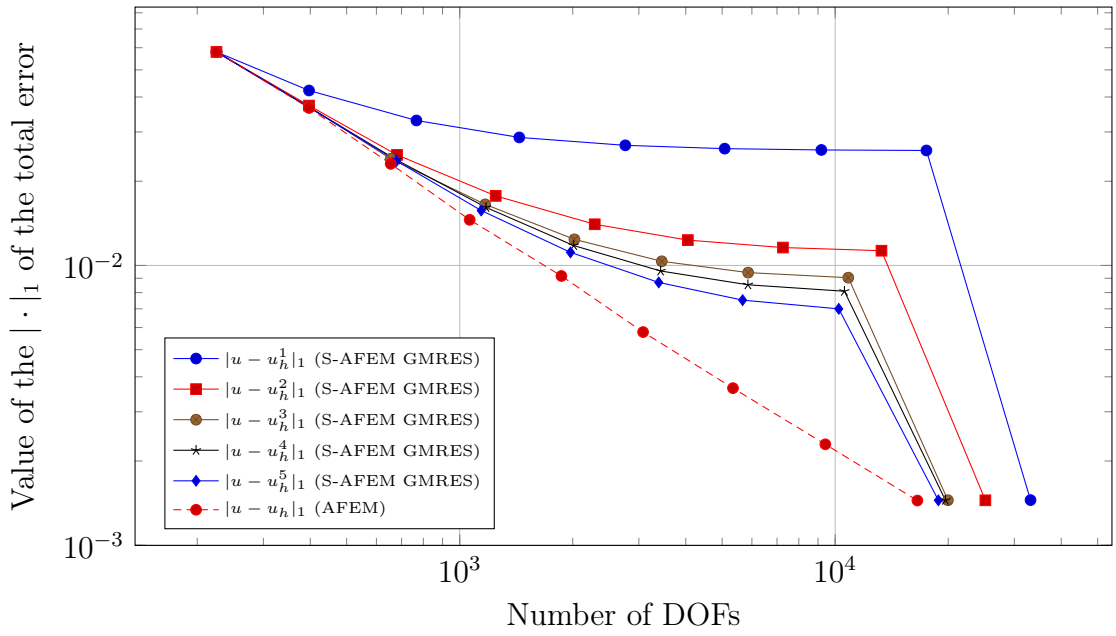


Figure 5.27: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

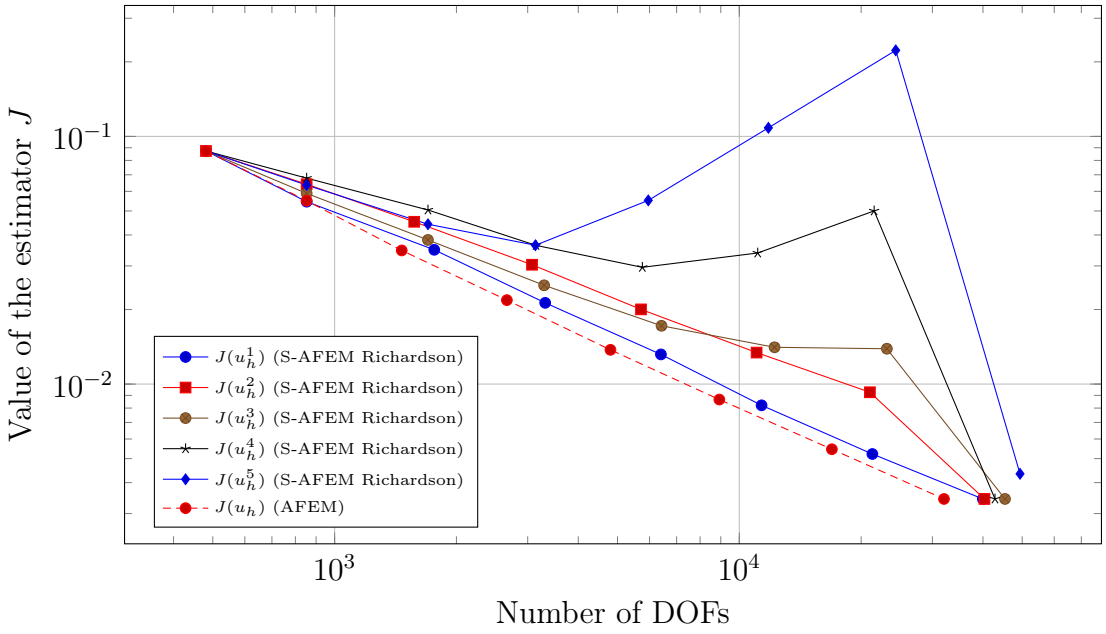


Figure 5.28: corner 2d, deg=3

Figure 5.29: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

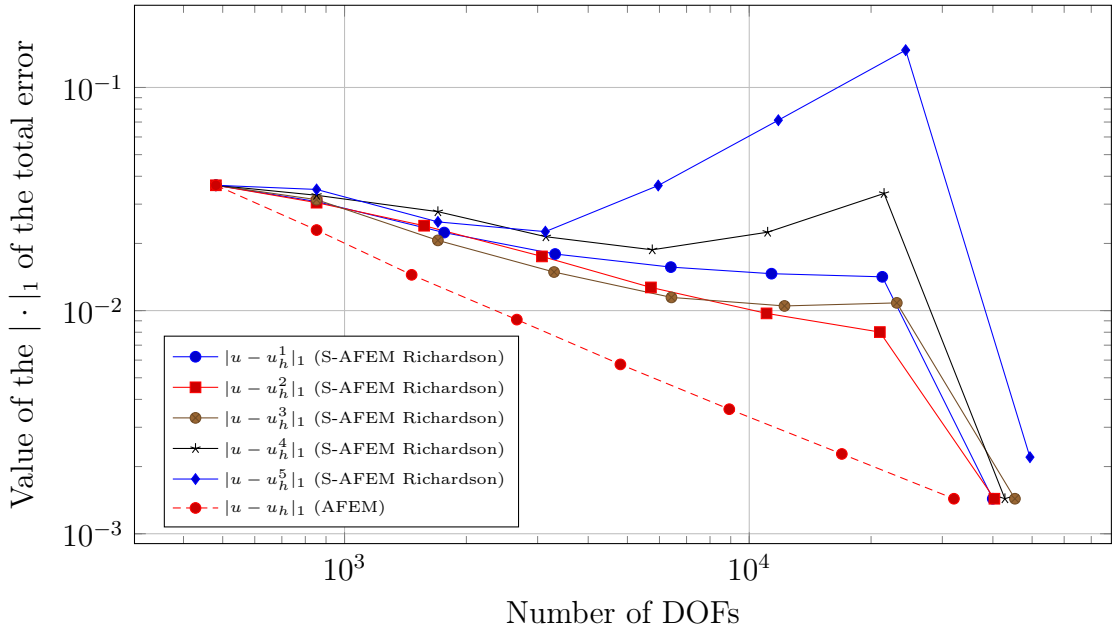


Figure 5.30: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

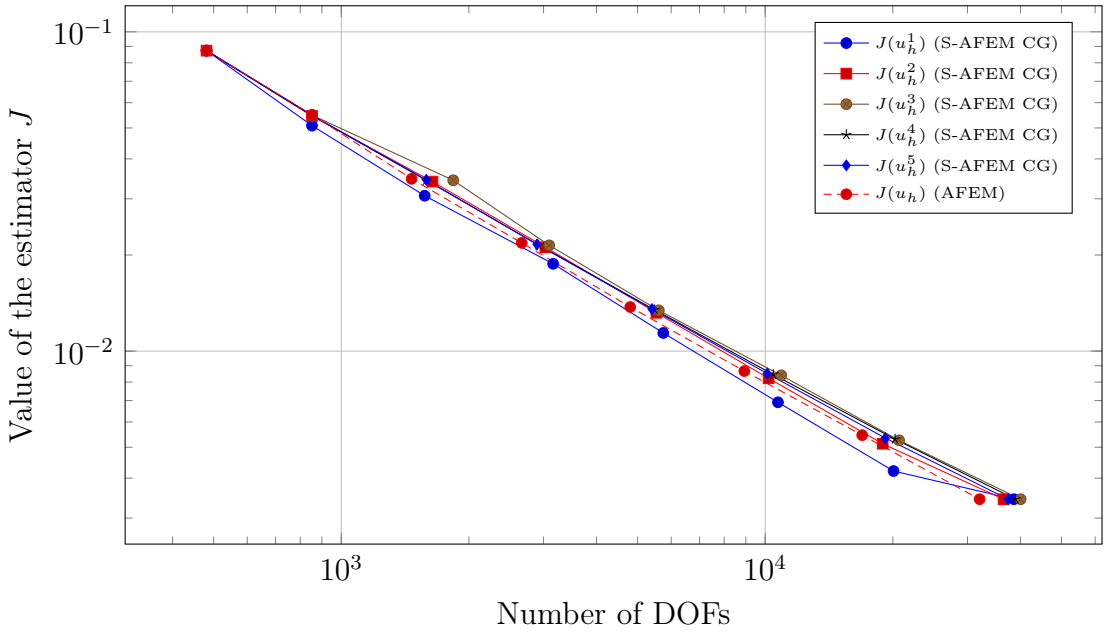


Figure 5.31: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

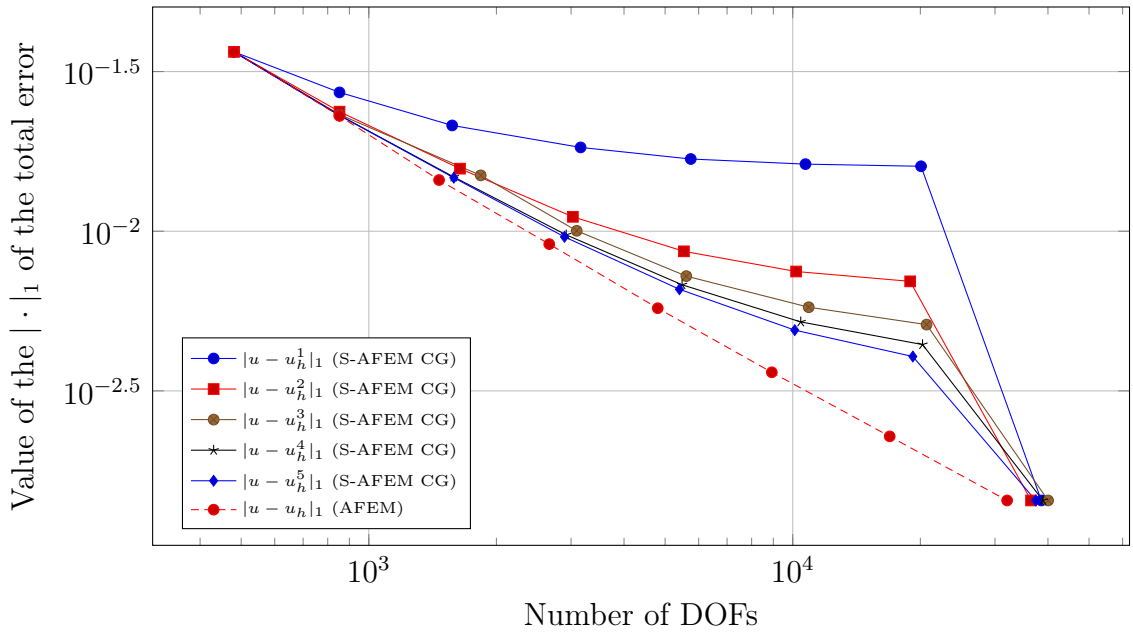


Figure 5.32: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

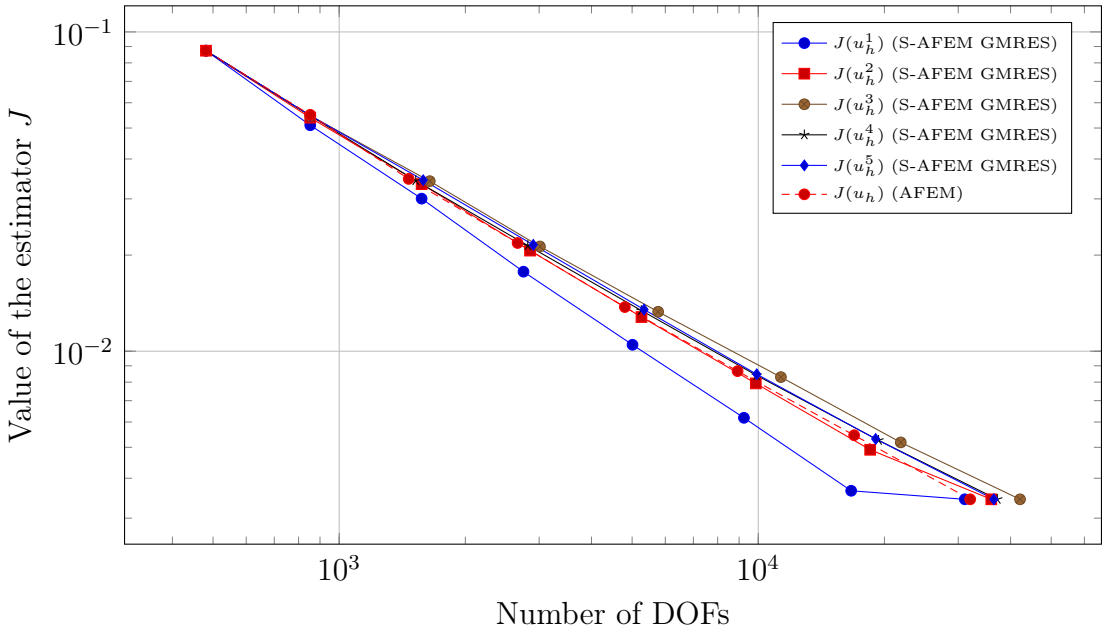


Figure 5.33: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

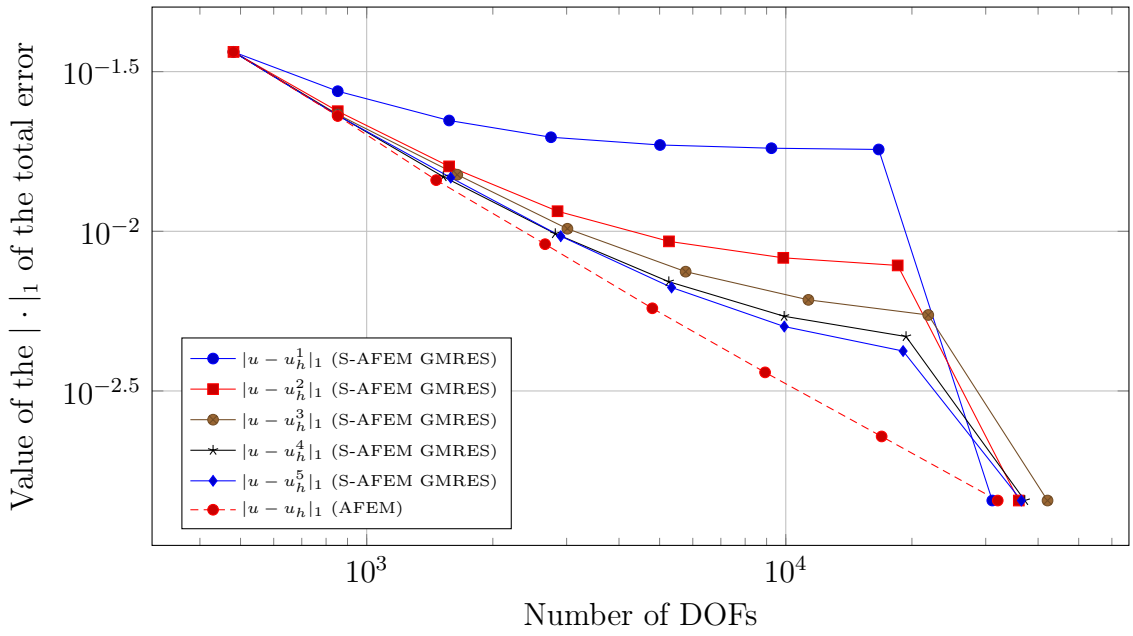


Figure 5.34: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

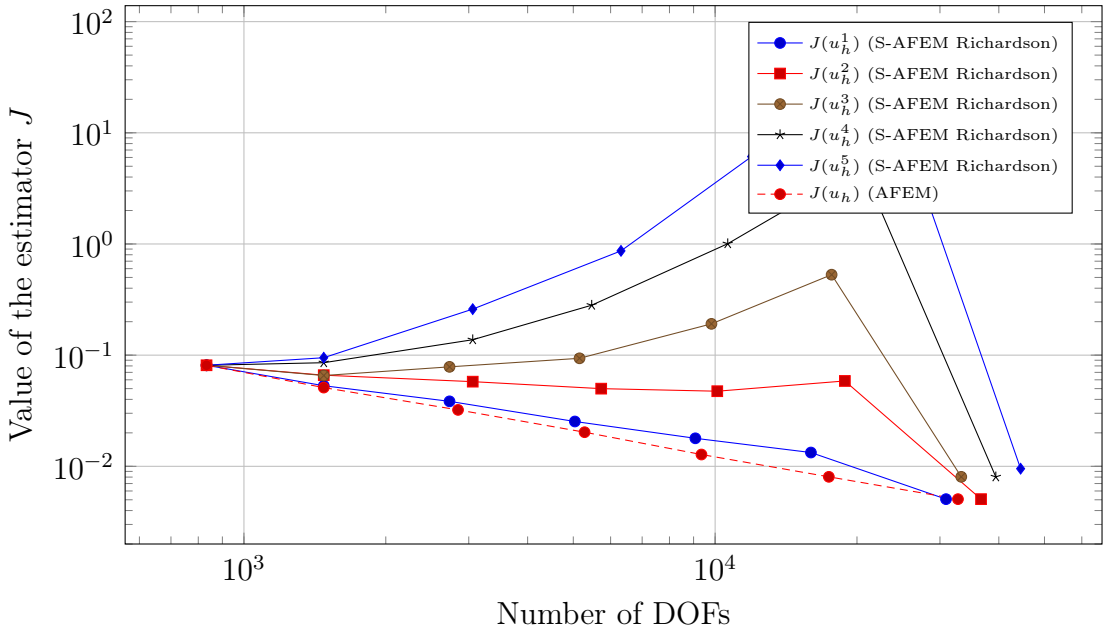


Figure 5.35: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

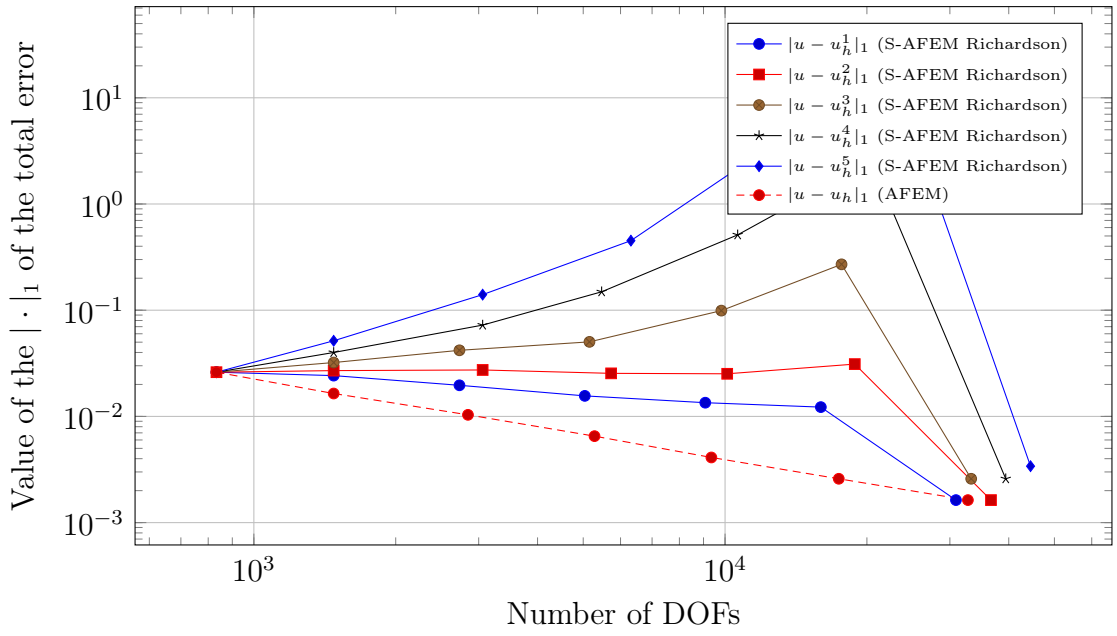


Figure 5.36: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

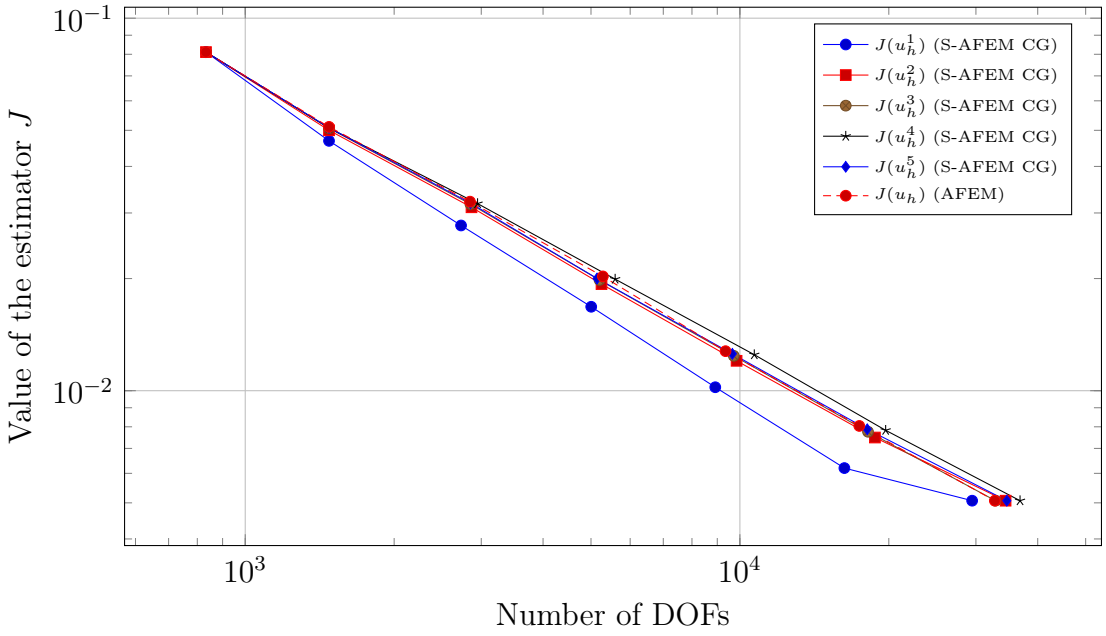


Figure 5.37: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

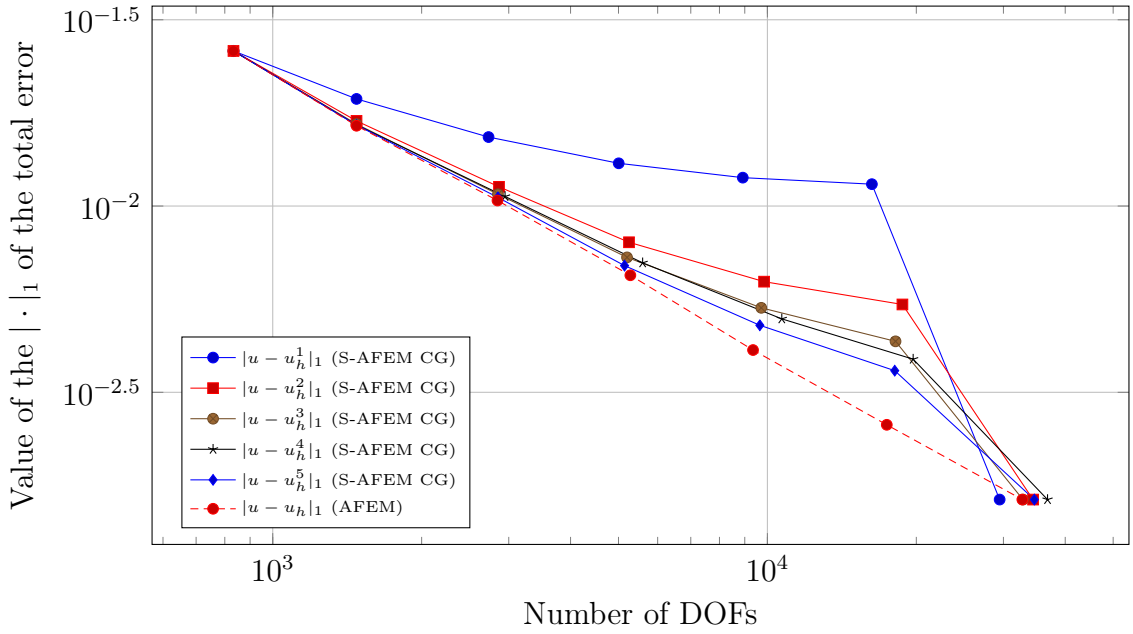


Figure 5.38: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

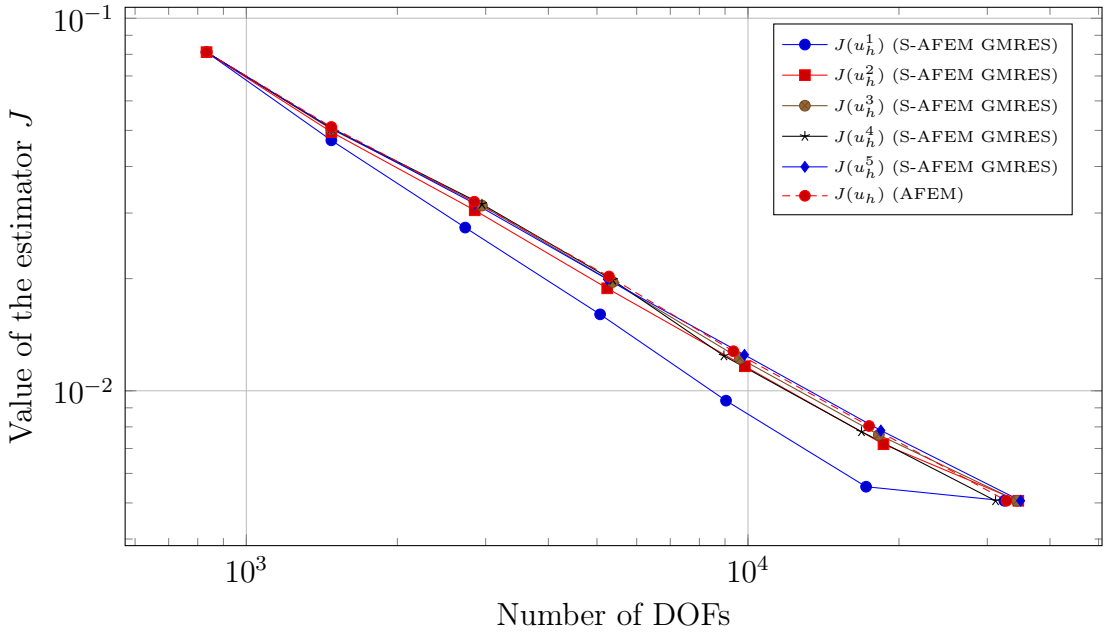


Figure 5.39: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

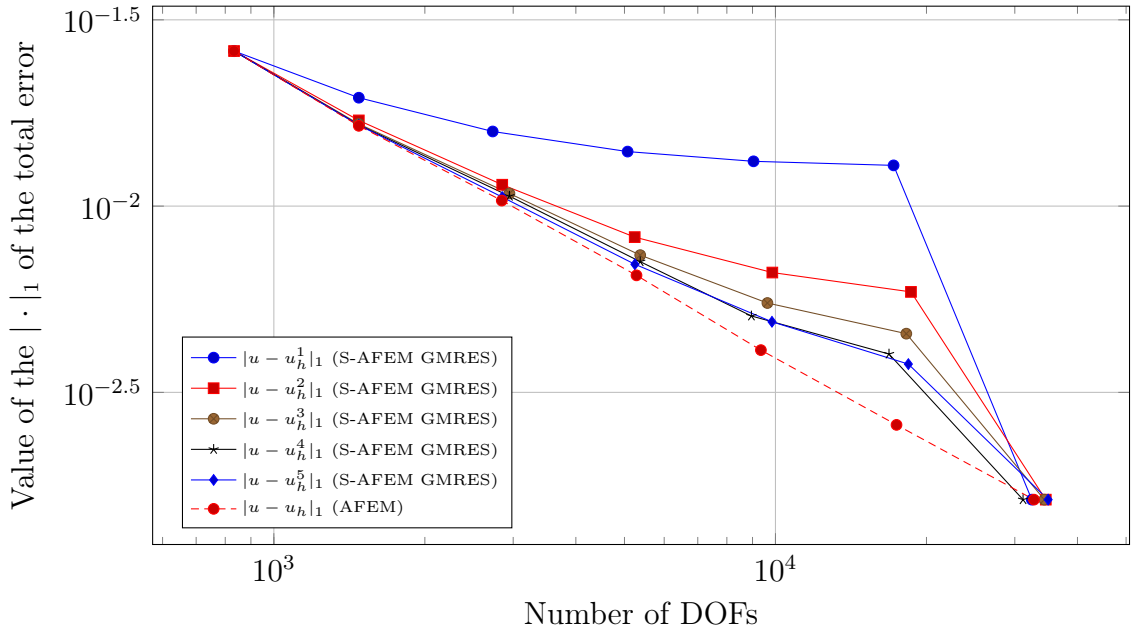


Figure 5.40: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

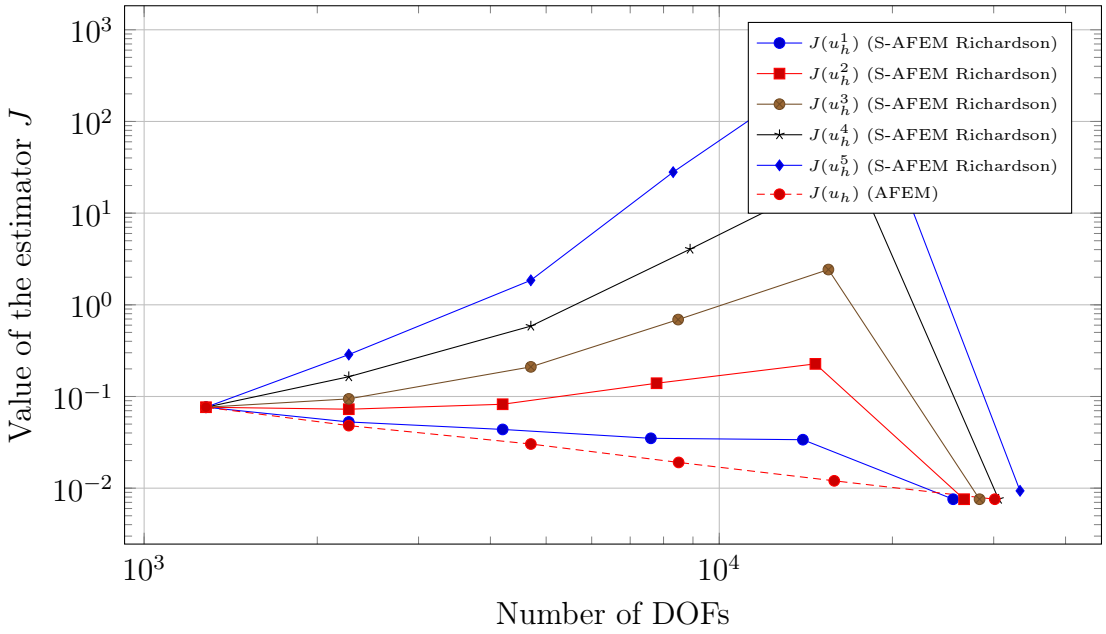


Figure 5.41: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson iteration ad a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

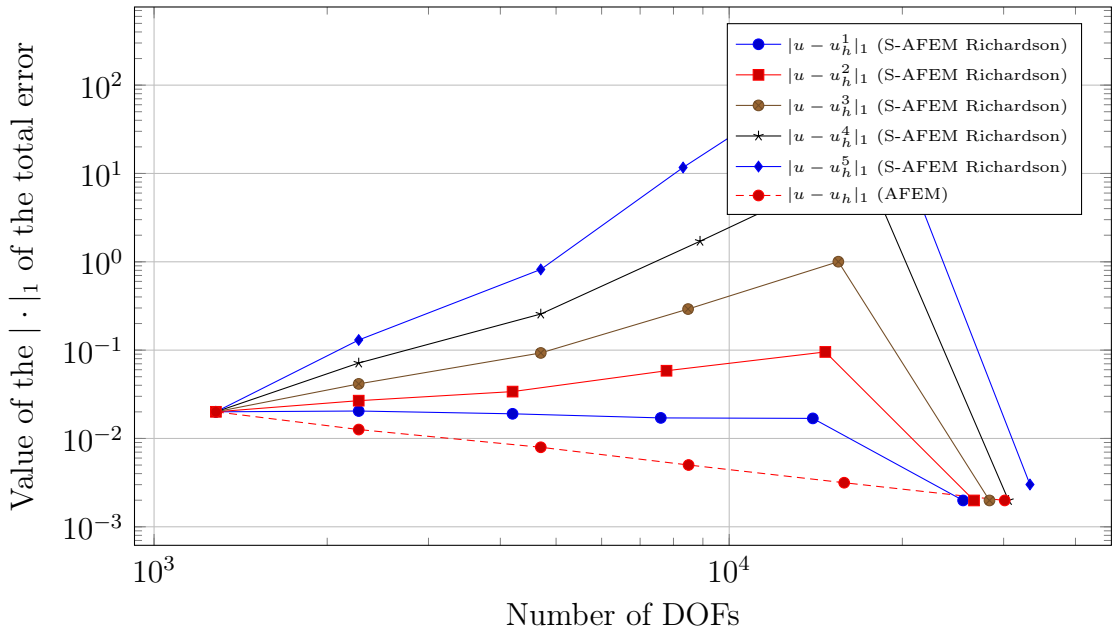


Figure 5.42: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

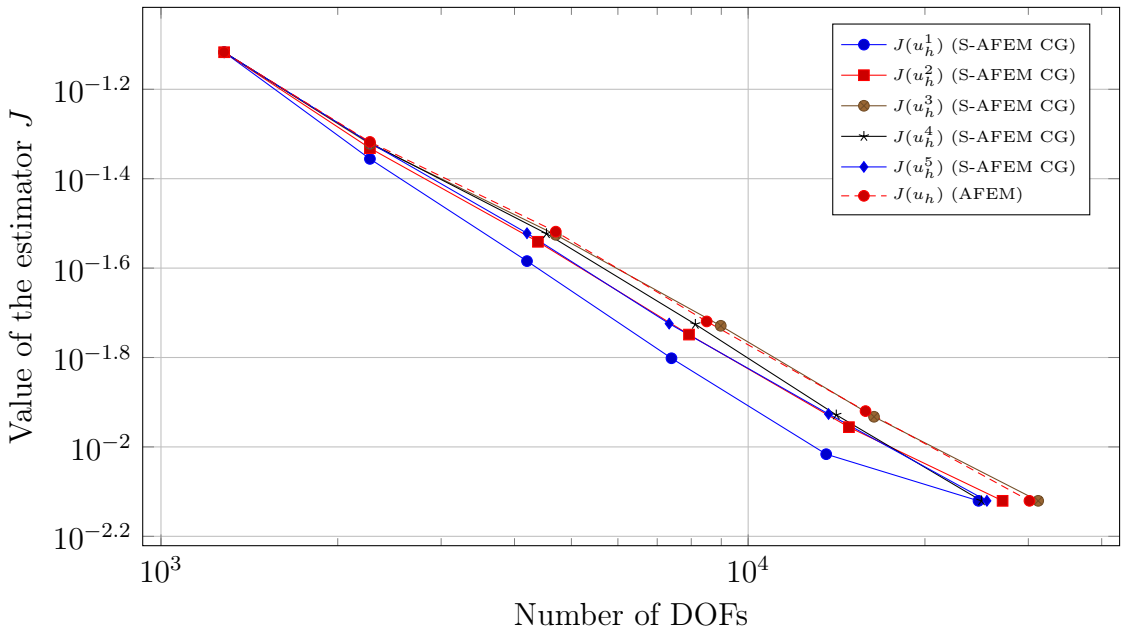


Figure 5.43: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

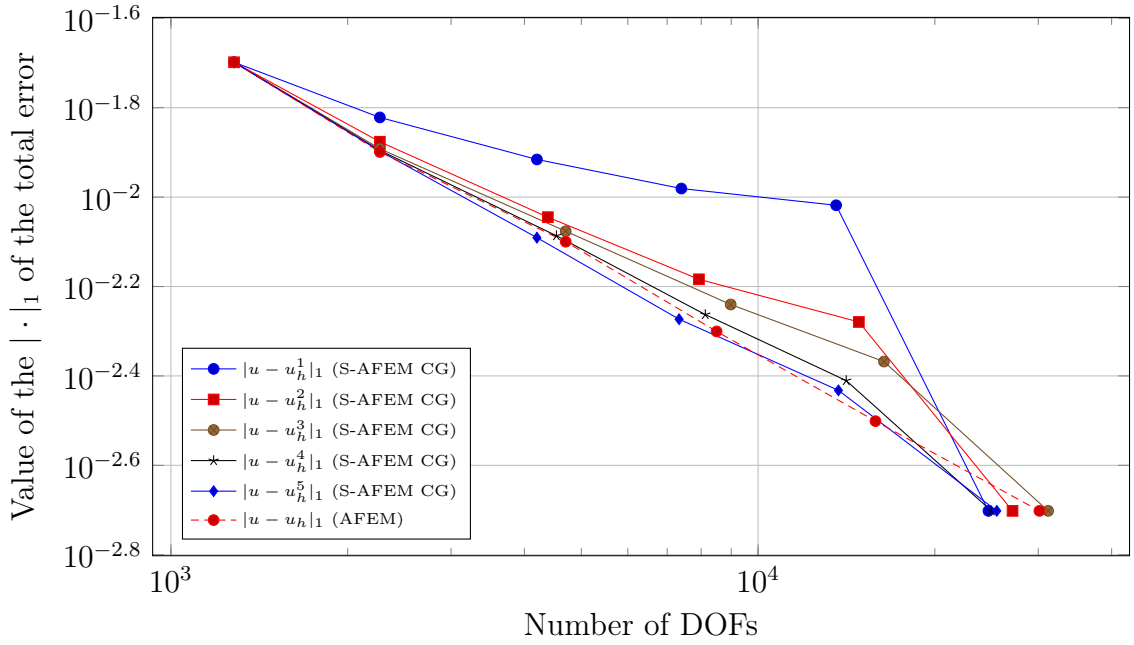


Figure 5.44: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

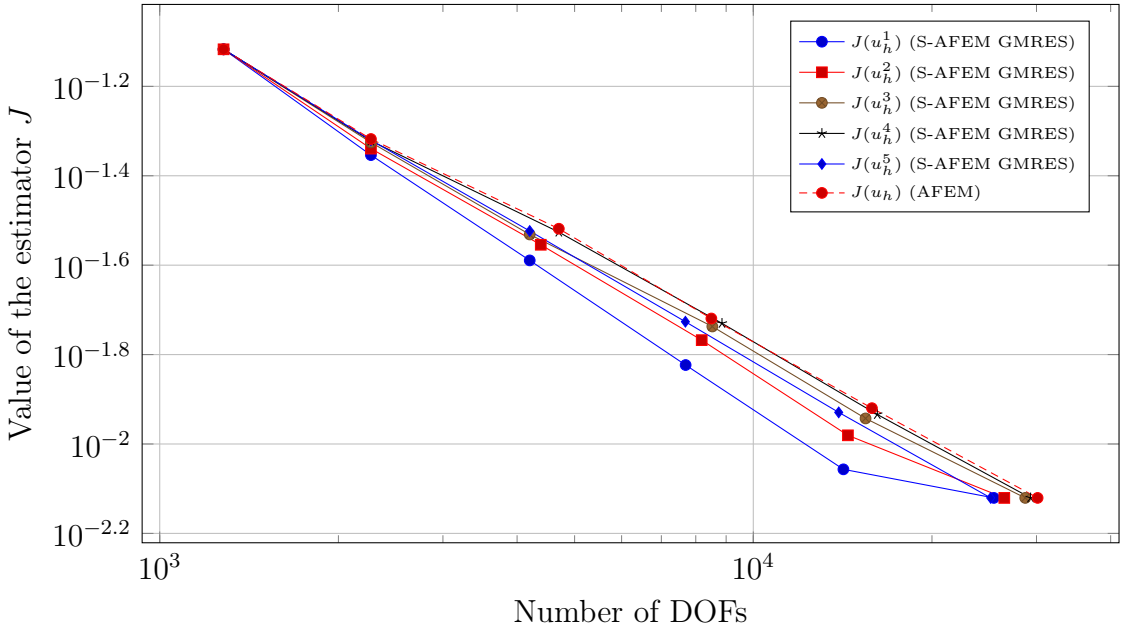


Figure 5.45: Value of the error estimator J for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

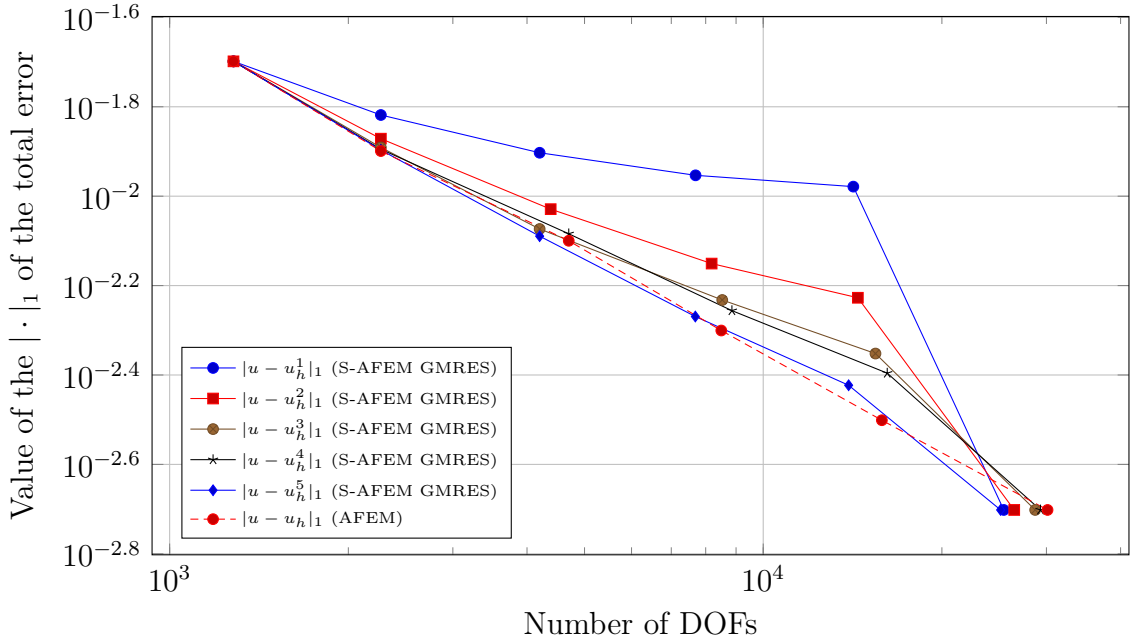


Figure 5.46: Value of the $|\cdot|_1$ of the total error for the corner problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

shown in Figure 5.48. Nevertheless, in all cases the accuracy of the final approximation for the last cycle obtained by S-AFEM, is the same to the one that is generated by classical AFEM.

For $deg = 2$, the CG and the GMRES are better smoothers exhibiting a quasi optimal convergence order compared to classical AFEM, as shown in Figures 5.55 and 5.57, while the Richardson iteration works less well (cf. Fig. 5.53). Nevertheless, the accuracy of the final approximation is almost the same, as shown in Figures 5.54, 5.56 and 5.58.

For $deg = 3, 4$, and 5, similarly to the two-dimensional Corner Problem, Richardson iteration turns out to be a bad smoother for S-AFEM, unless further tuning of the relaxation parameter γ is performed. Here, for all values of smoothing iteration count l, as it increases from 1 to 5, contrarily to what one might expect, the value of the estimator increases with increasing DOFs (see Figures 5.59, 5.60, 5.65, 5.66, 5.71 and 5.72), showing that we should estimate in a better way the spectral radius of the final matrix A, and modify the Richardson relaxation parameter γ accordingly.

As the polynomial degree increases, it can be observed that the presence of the algebraic error seems to influence more the error estimator (cf. Figures 5.55, 5.57, 5.61, 5.63, 5.67, 5.69, 5.73 and 5.75), at any case this has less influence at the final error, leading to an accuracy of the final approximation, which is very close or almost the same to the one that would be generated by classical AFEM (cf. Figures 5.56, 5.58, 5.62, 5.64, 5.68, 5.70, 5.74 and 5.76).

In conclusion, S-AFEM turns out to be a good strategy also for higher order finite element discretisations, and one could use directly the CG method (or, alternatively, the GMRES method) as a smoother for the intermediate iterations. Our numerical evidences show that two smoothing iterations are enough for the two dimensional case, independently on the polynomial degree of the finite element approximation.

5.4.2 Three-dimensional examples

We apply both AFEM and S-AFEM to the three dimensional Corner Problem (5.4) and we show a comparison for different fixed FEM degrees, as deg $= 1, \ldots, 5$, 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 of smoothing iterations $l = 2i - 1$ increases for $i = 1, \ldots, 8$.

As a marking criterion we select a fraction of $1/7$ of cells with the largest error indicator to be marked for refinement.

For piecewise bi-linear finite elements, we observe that when we consider the CG and the GMRES as smoothers for the intermediate cycles, the accuracy of the final approximations generated with S-AFEM is almost the same to the one generated with classical AFEM, as evidenced in Figures 5.80 and 5.82 after the

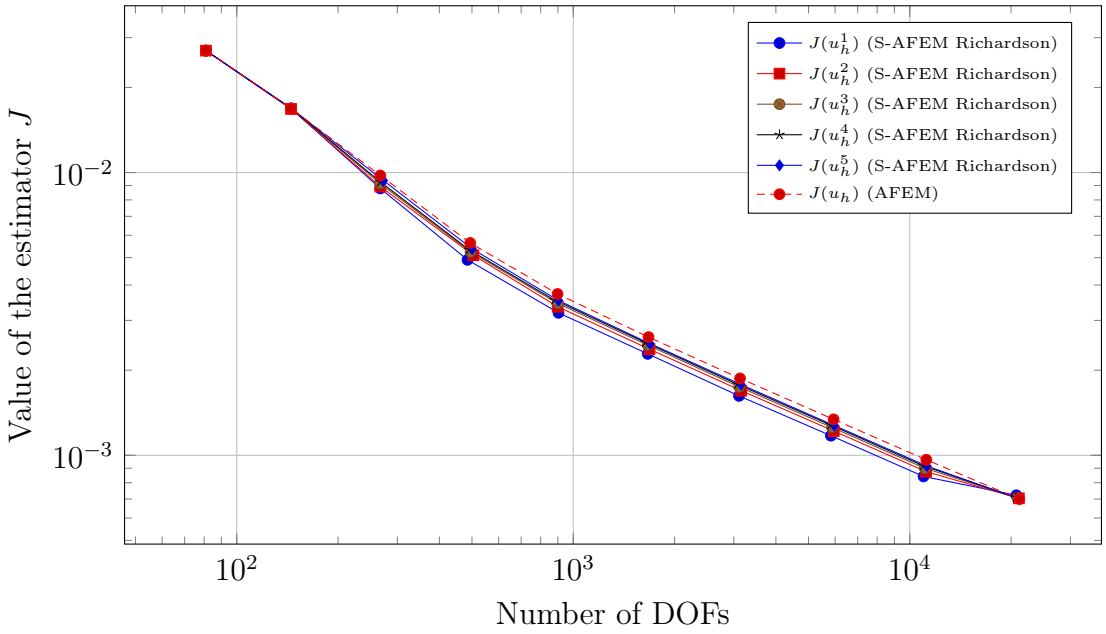


Figure 5.47: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree deg=1, when we apply 10 cycles of AFEM and S-AFEM with Richardson iterations as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

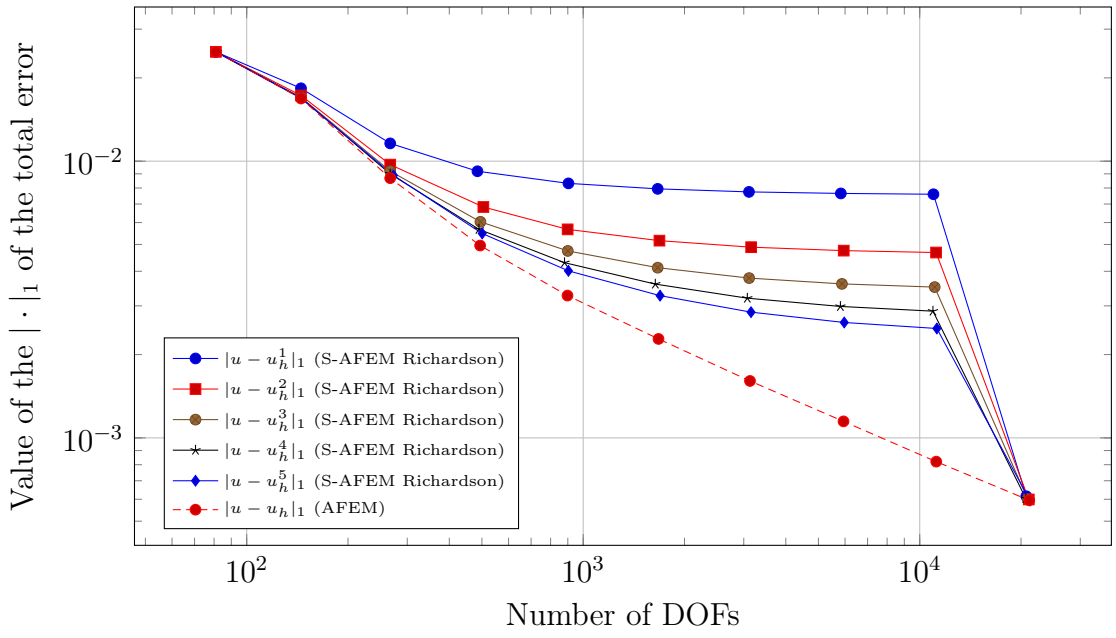


Figure 5.48: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

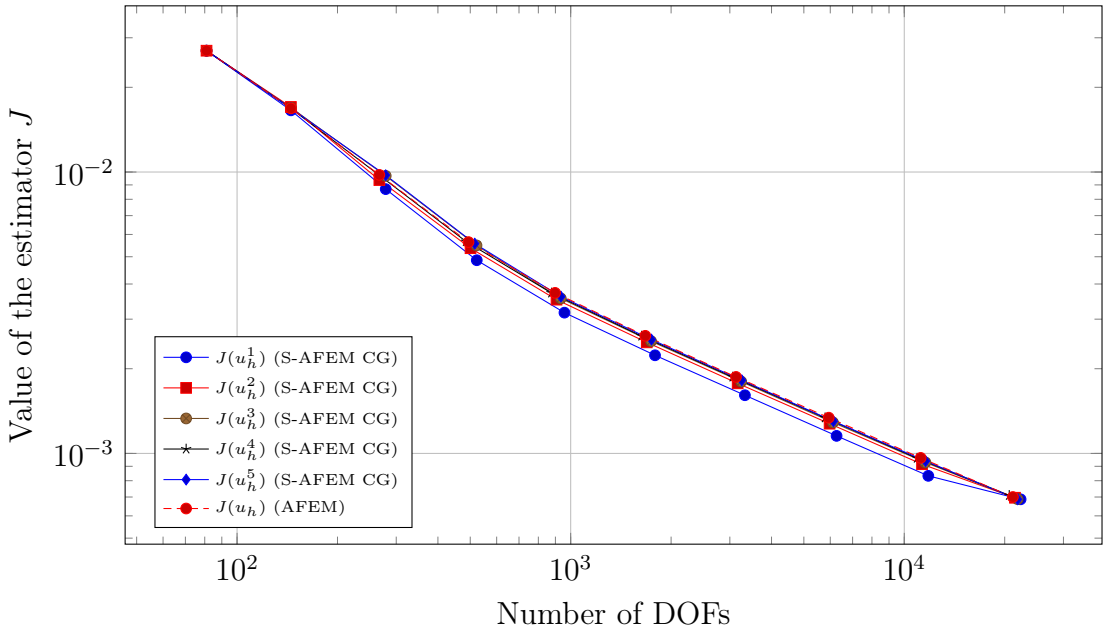


Figure 5.49: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

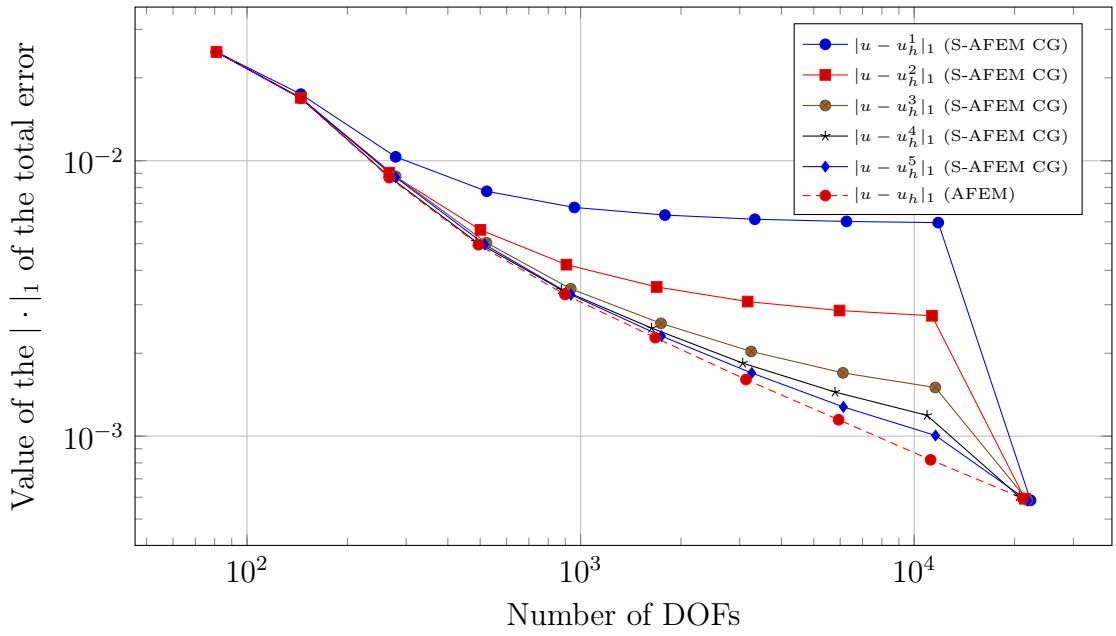


Figure 5.50: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

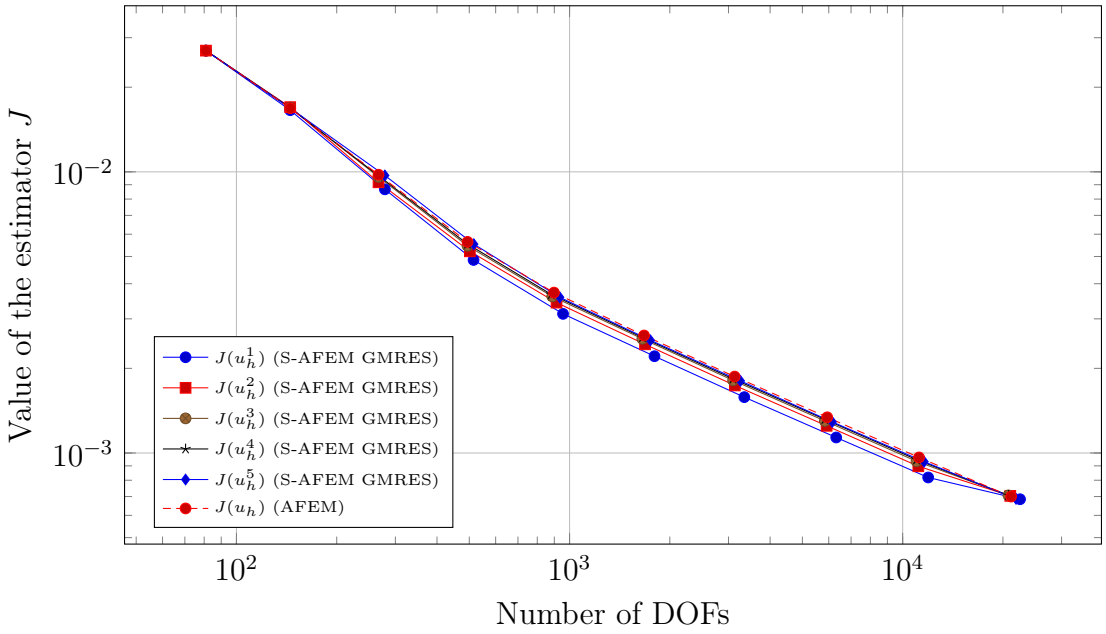


Figure 5.51: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

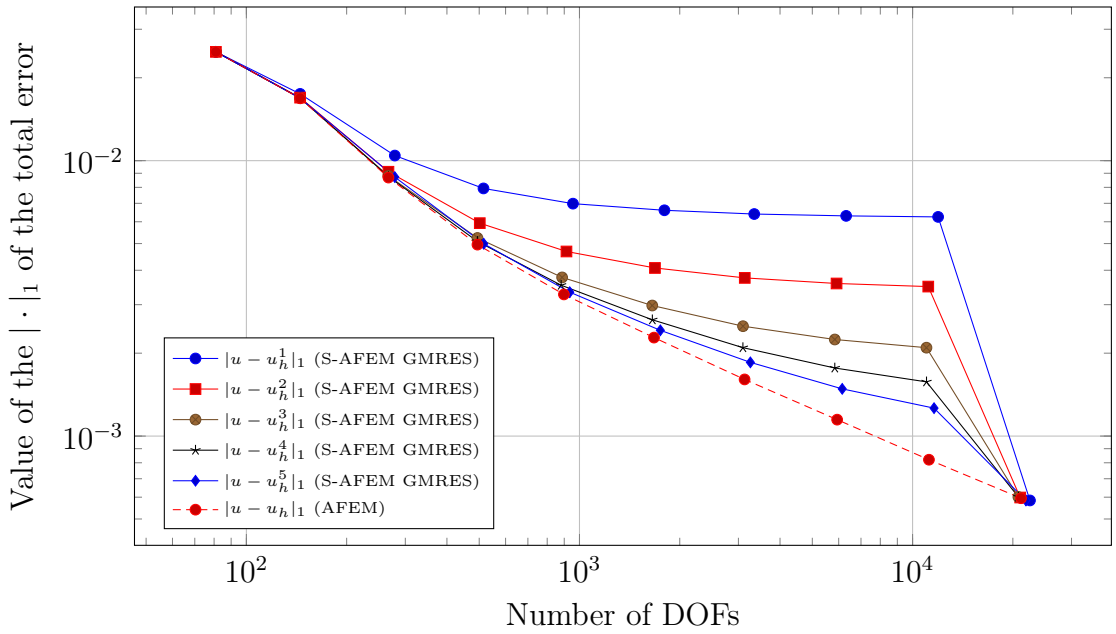


Figure 5.52: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

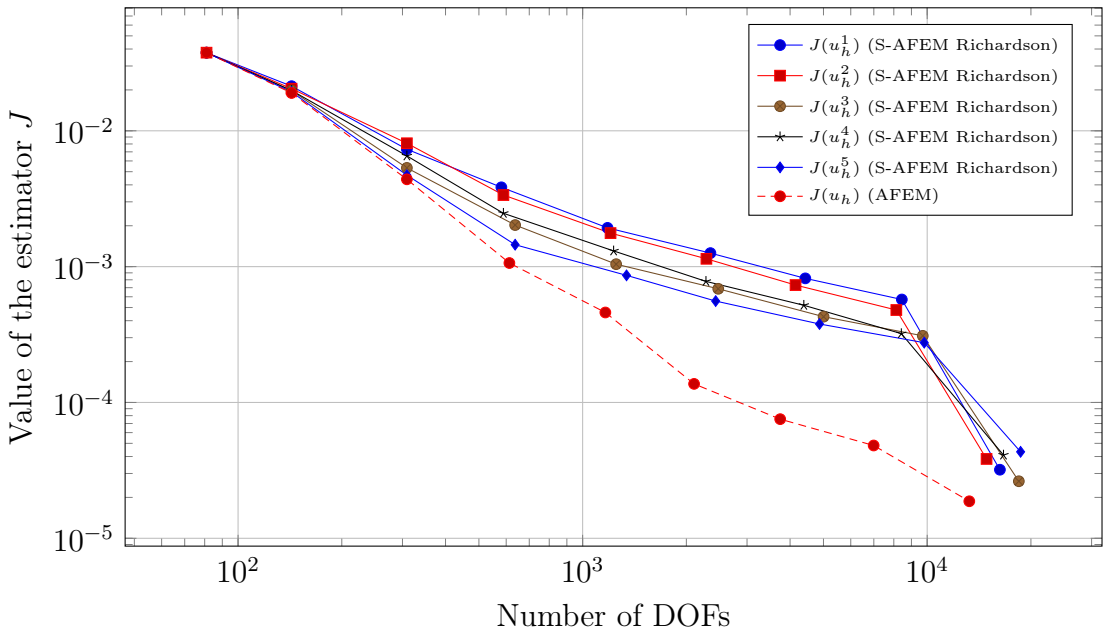


Figure 5.53: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 3 and we select a fraction of 1/3 of cells for refinement at each cycle.

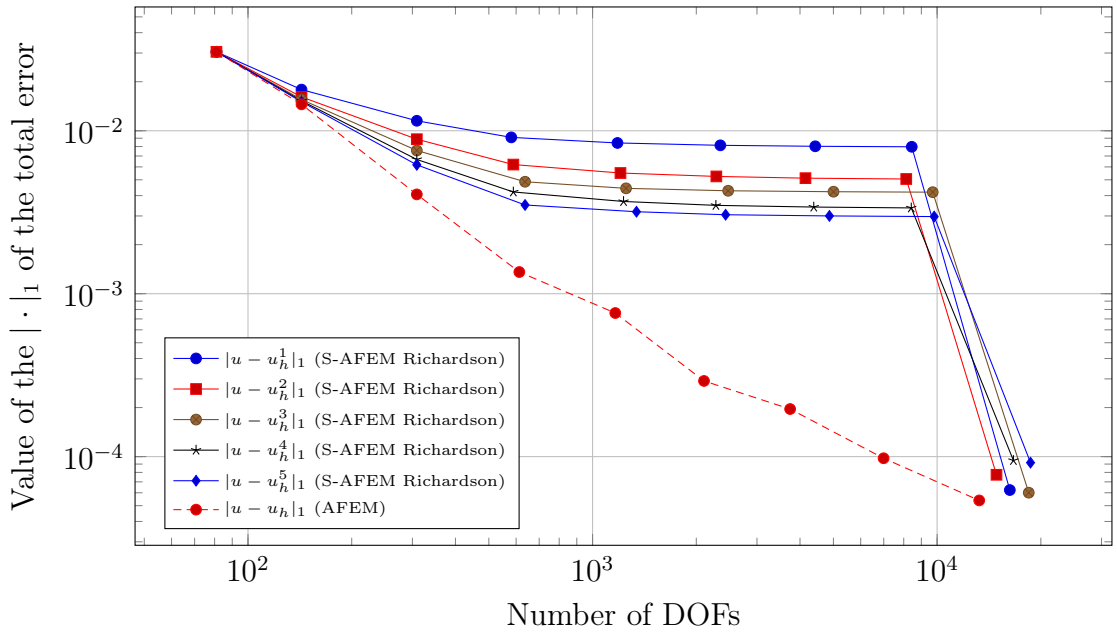


Figure 5.54: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the Richardson as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

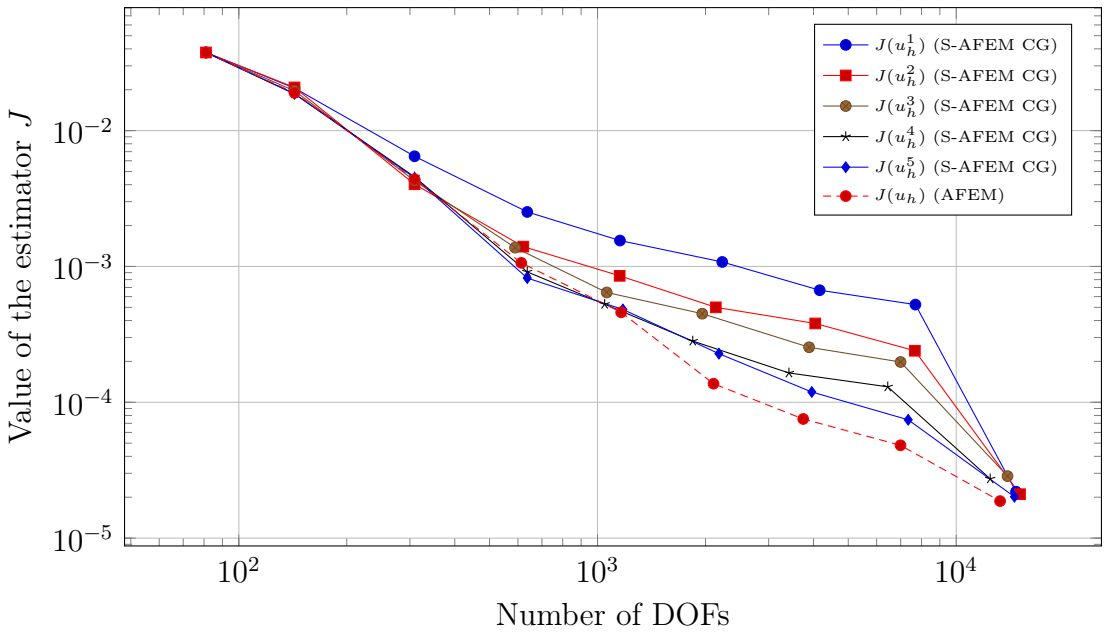


Figure 5.55: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

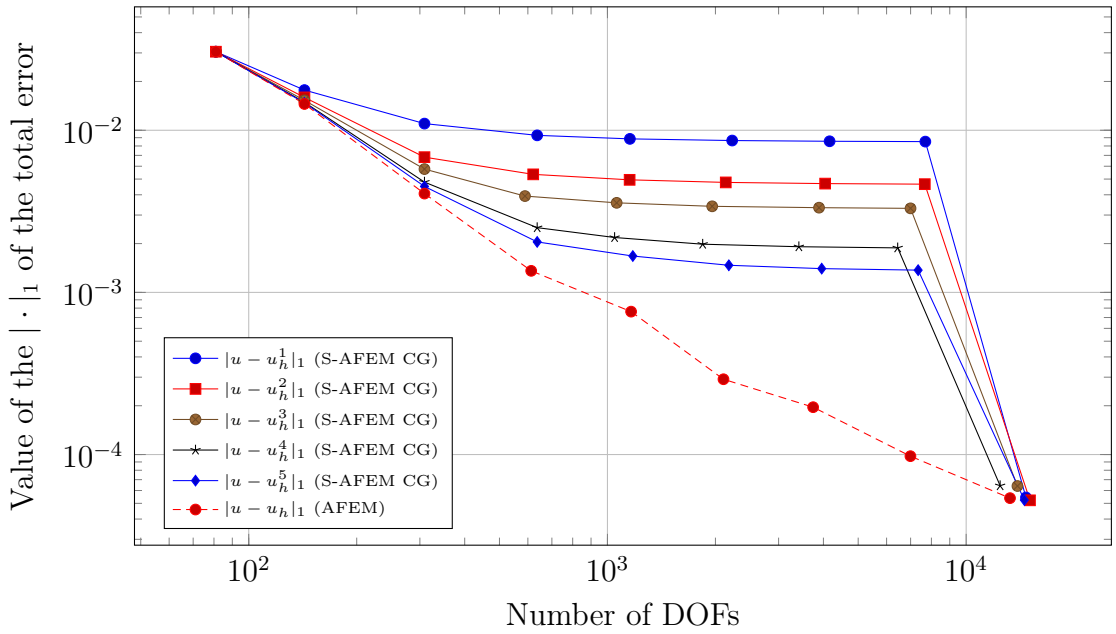


Figure 5.56: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

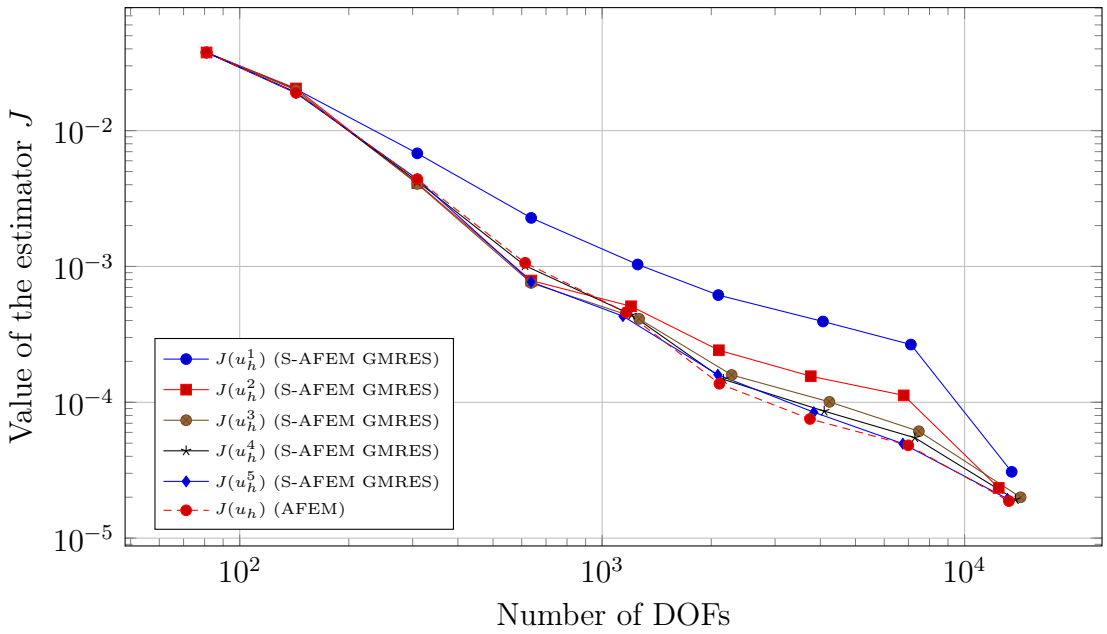


Figure 5.57: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

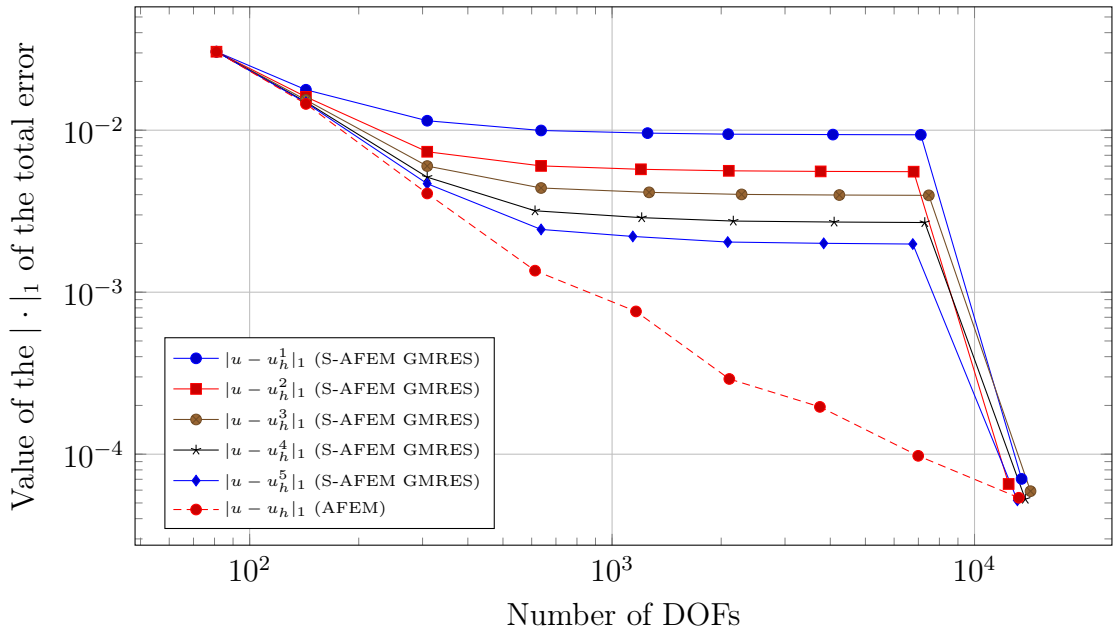


Figure 5.58: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

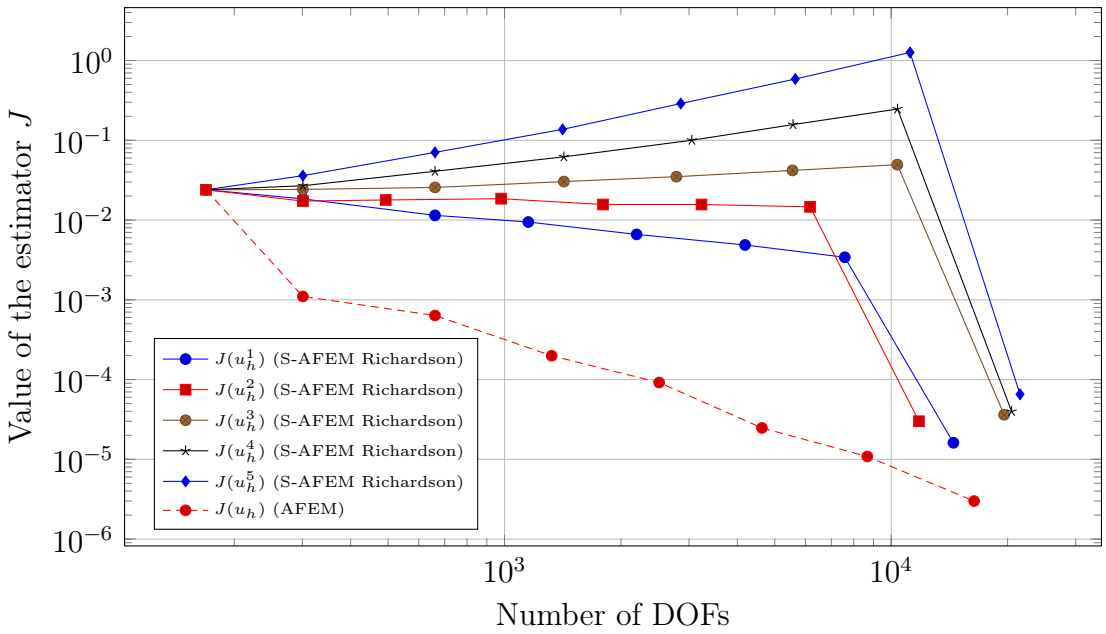


Figure 5.59: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

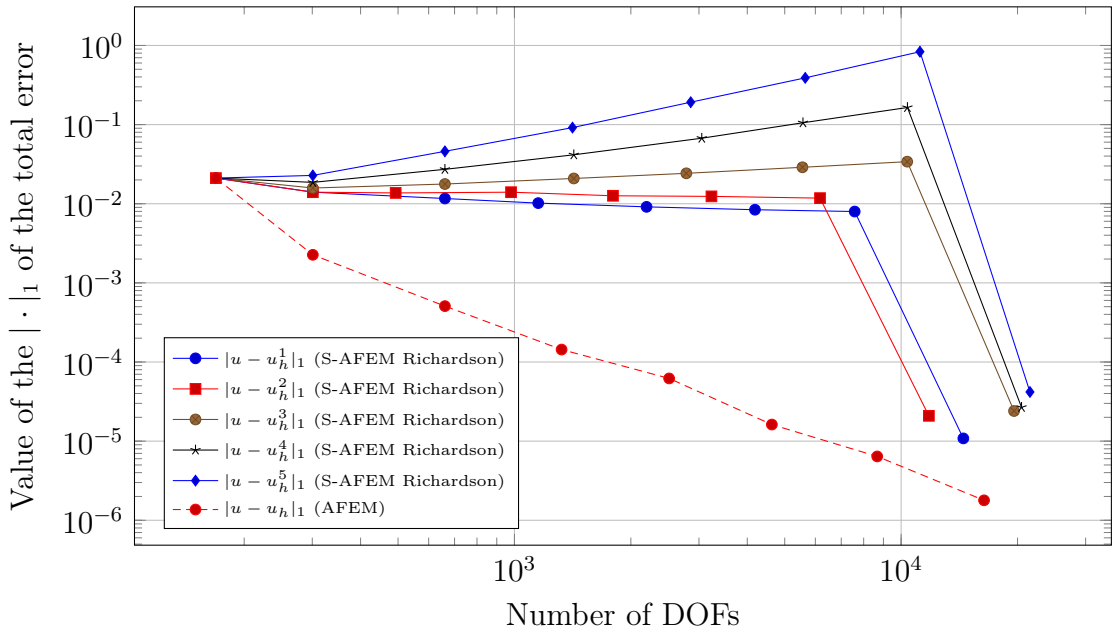


Figure 5.60: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

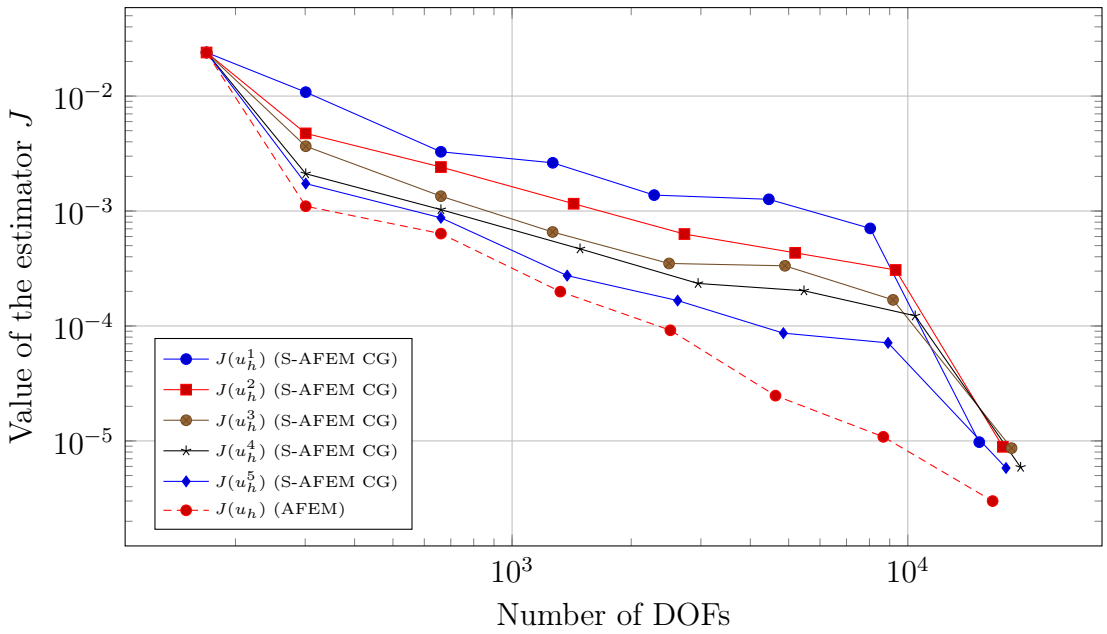


Figure 5.61: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

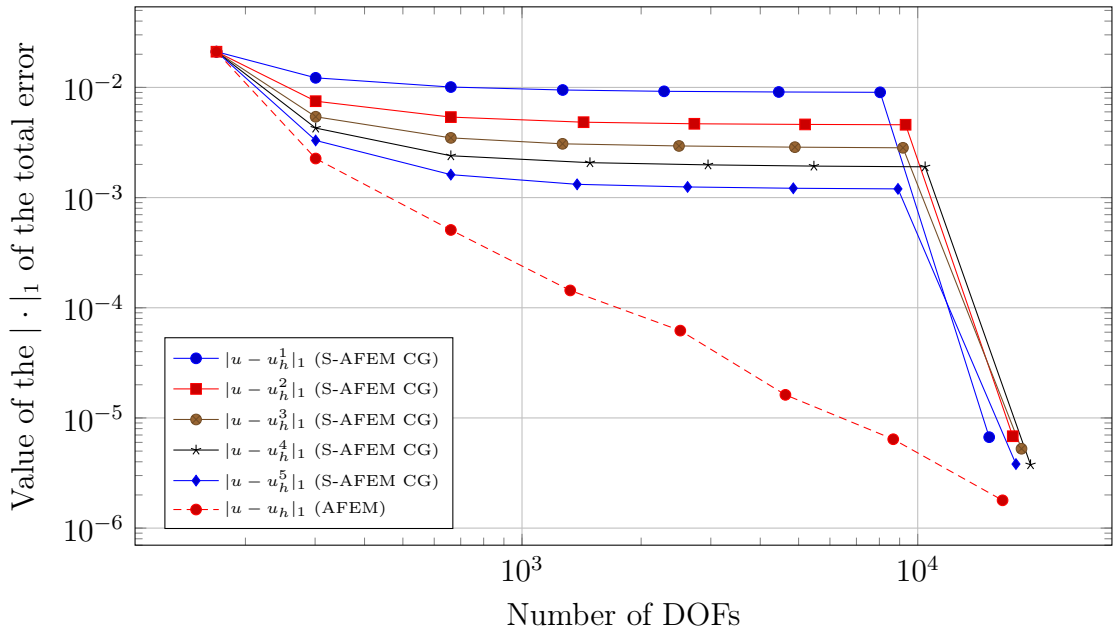


Figure 5.62: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

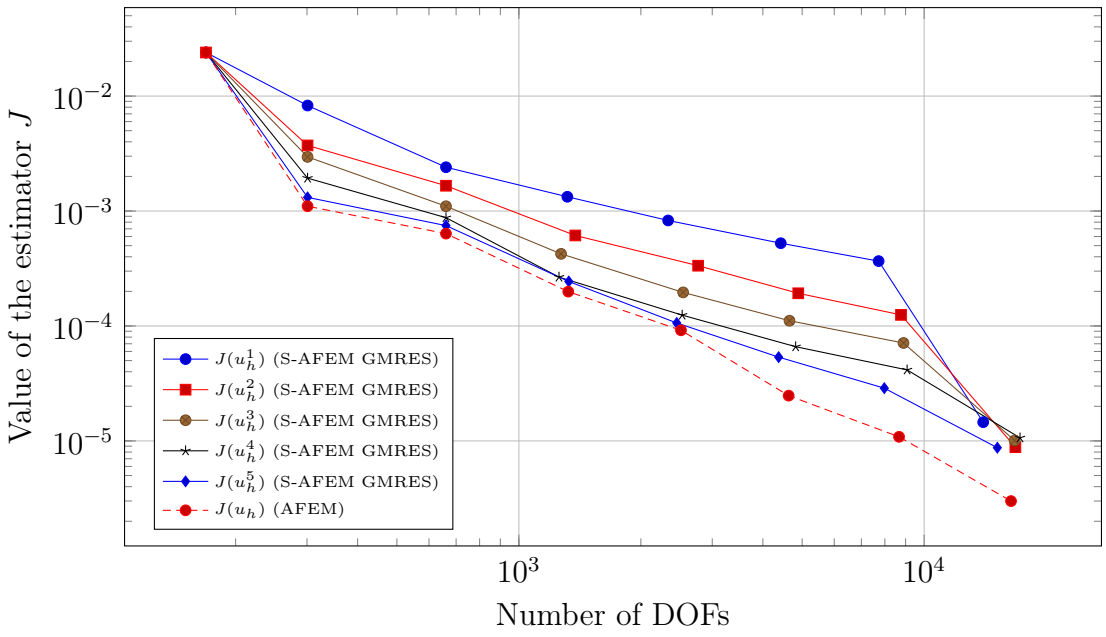


Figure 5.63: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

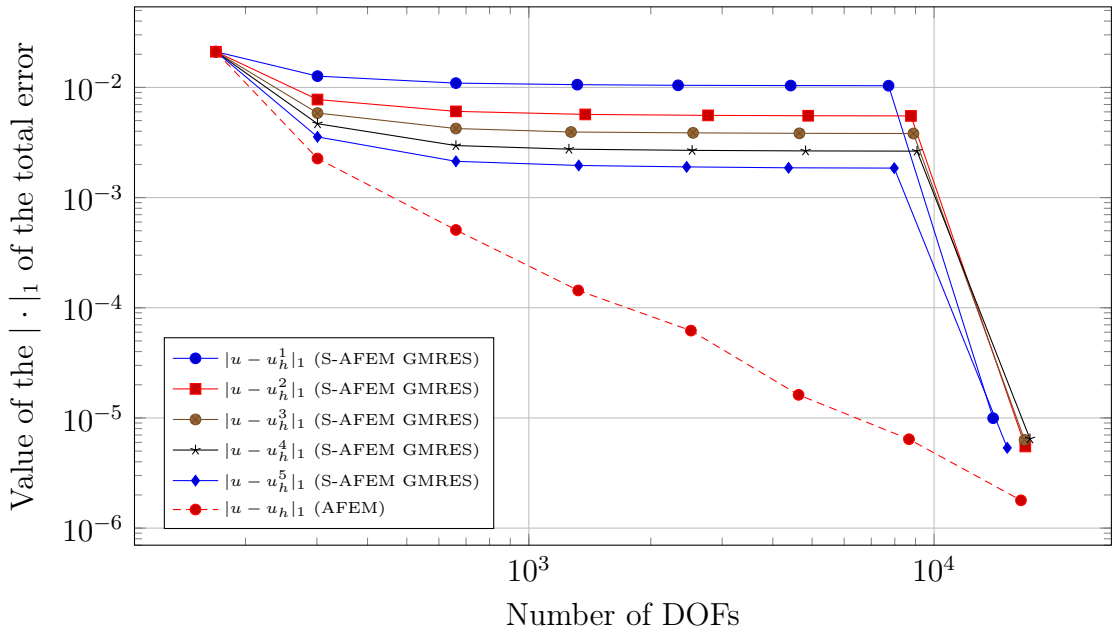


Figure 5.64: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

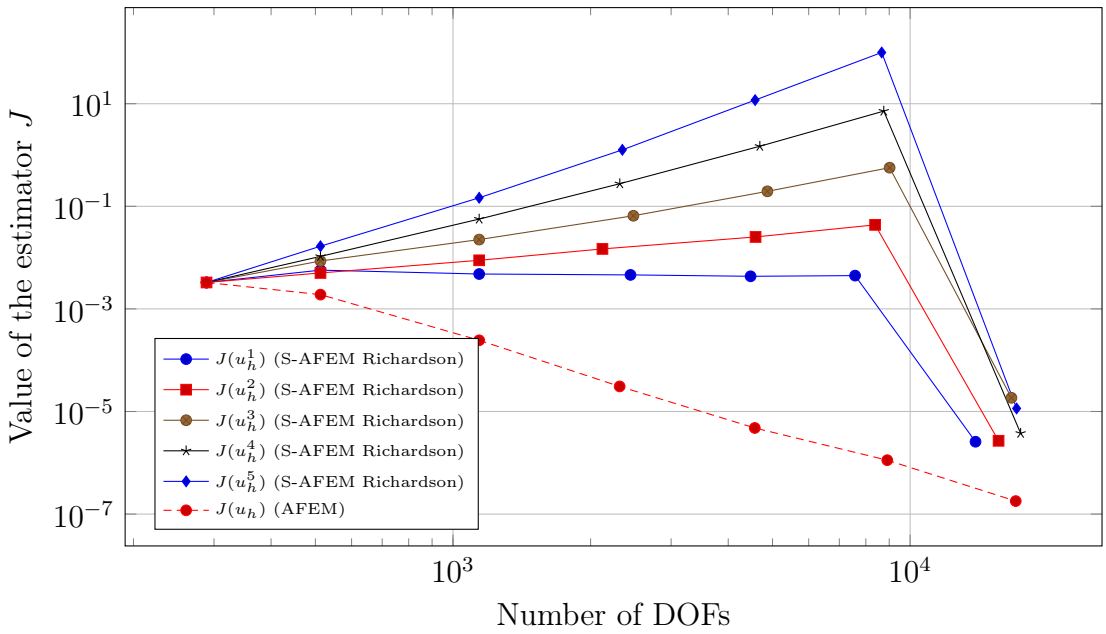


Figure 5.65: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

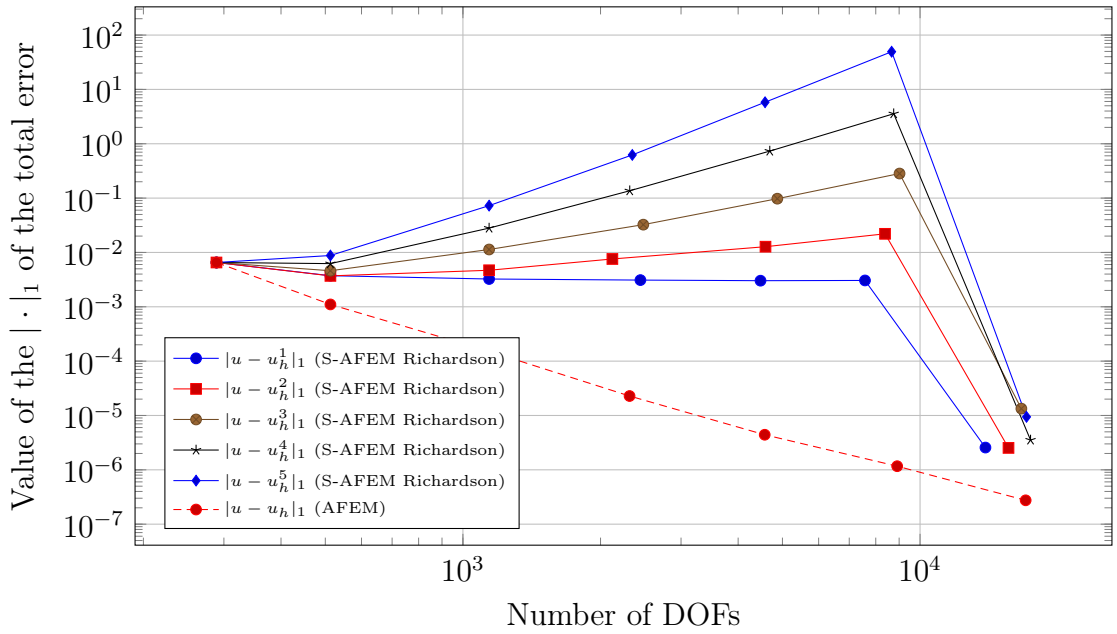


Figure 5.66: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

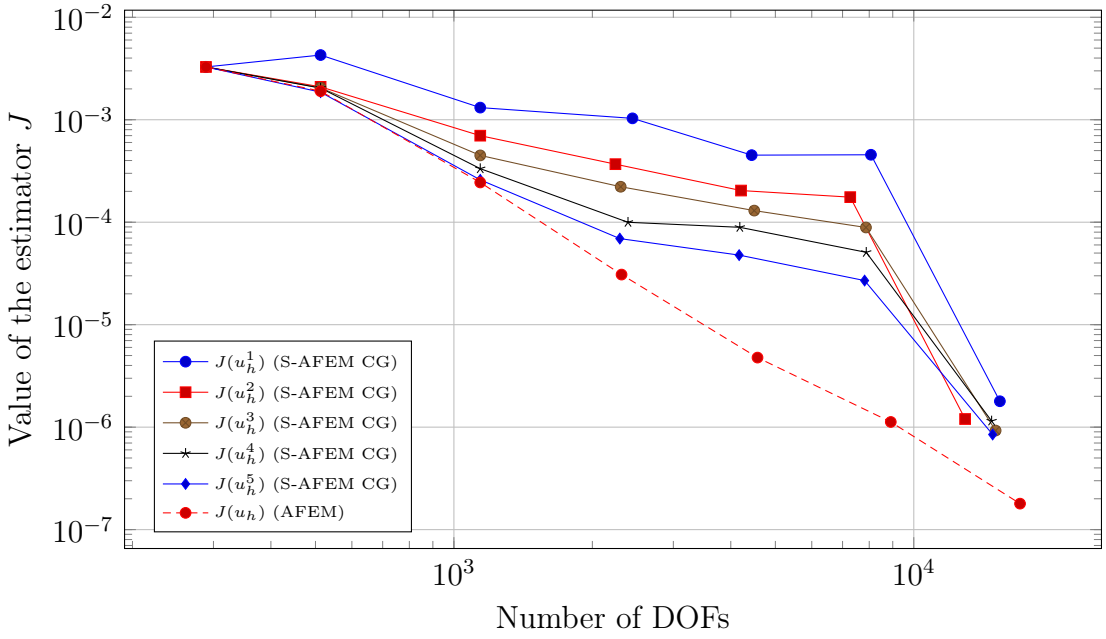


Figure 5.67: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

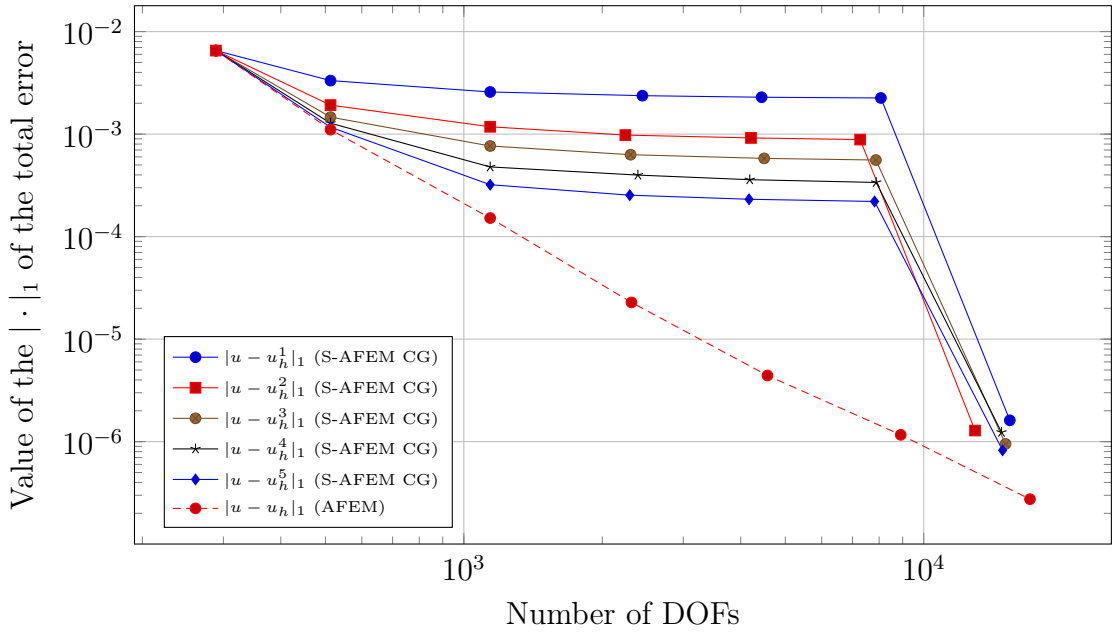


Figure 5.68: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

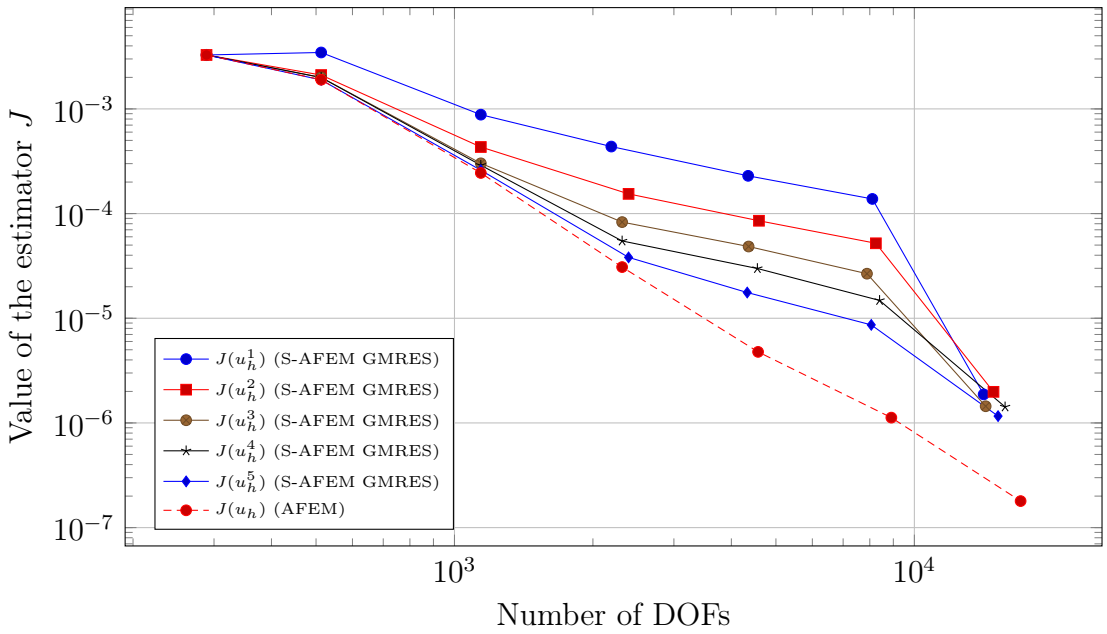


Figure 5.69: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

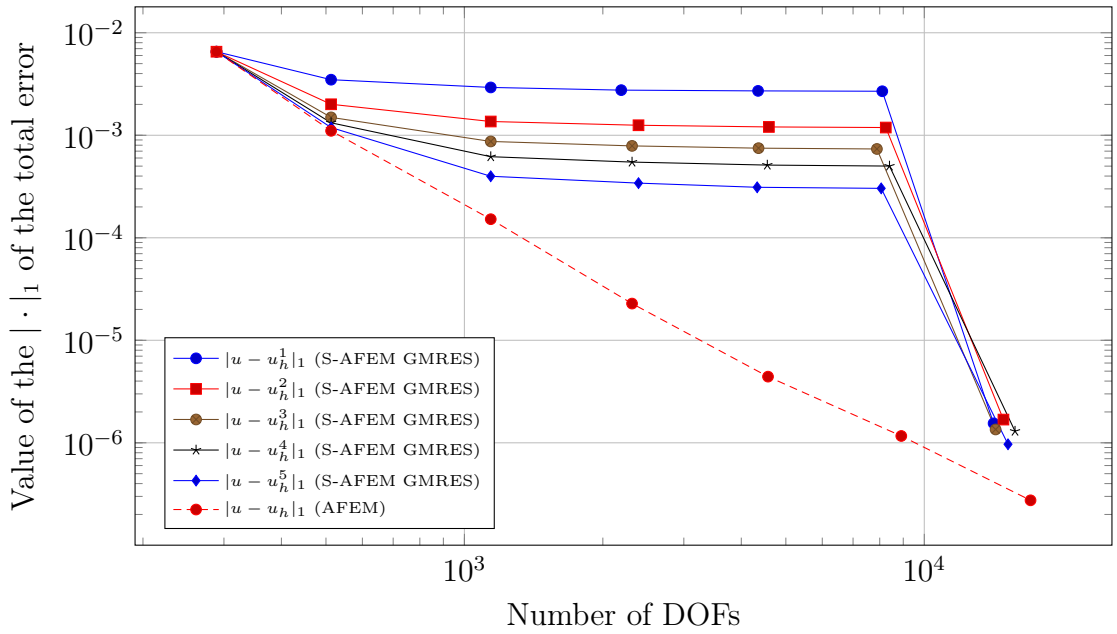


Figure 5.70: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with the GMRES iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

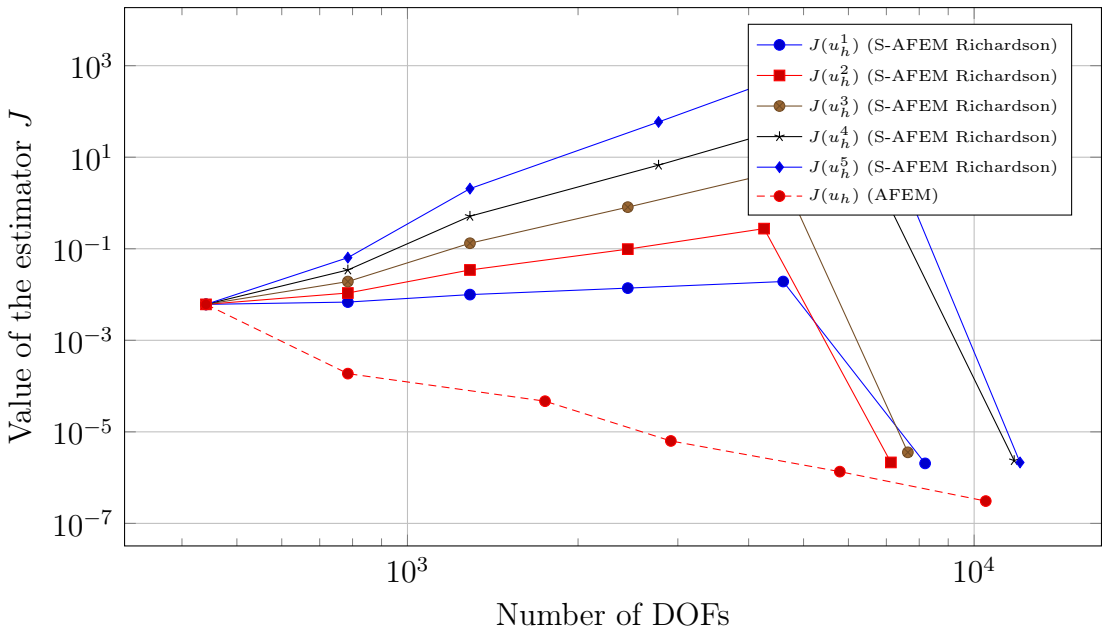


Figure 5.71: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

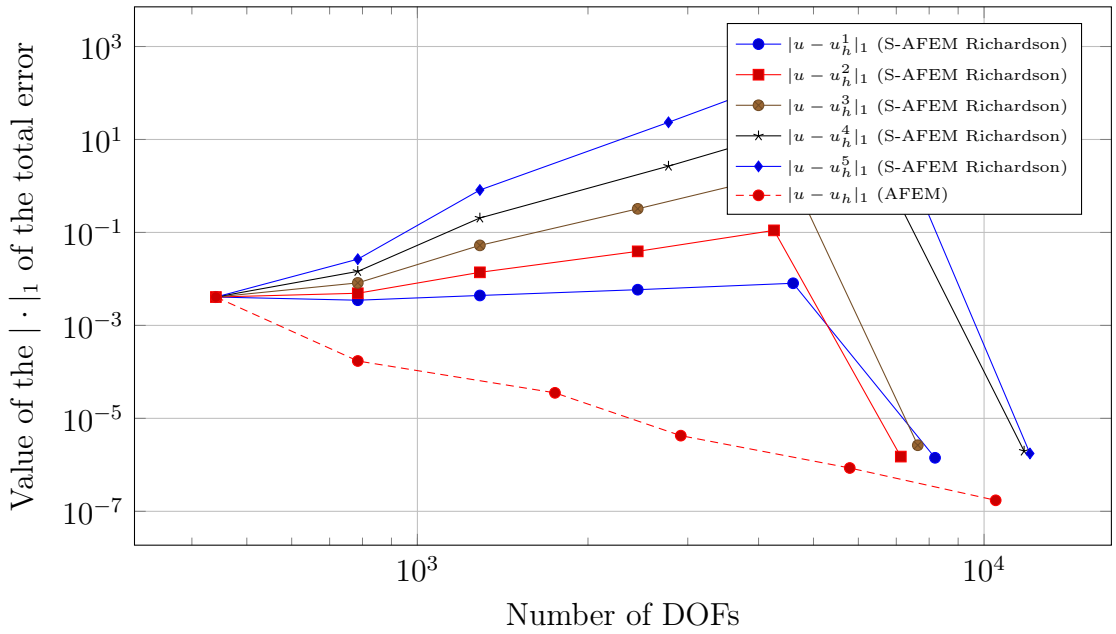


Figure 5.72: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson iteration as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

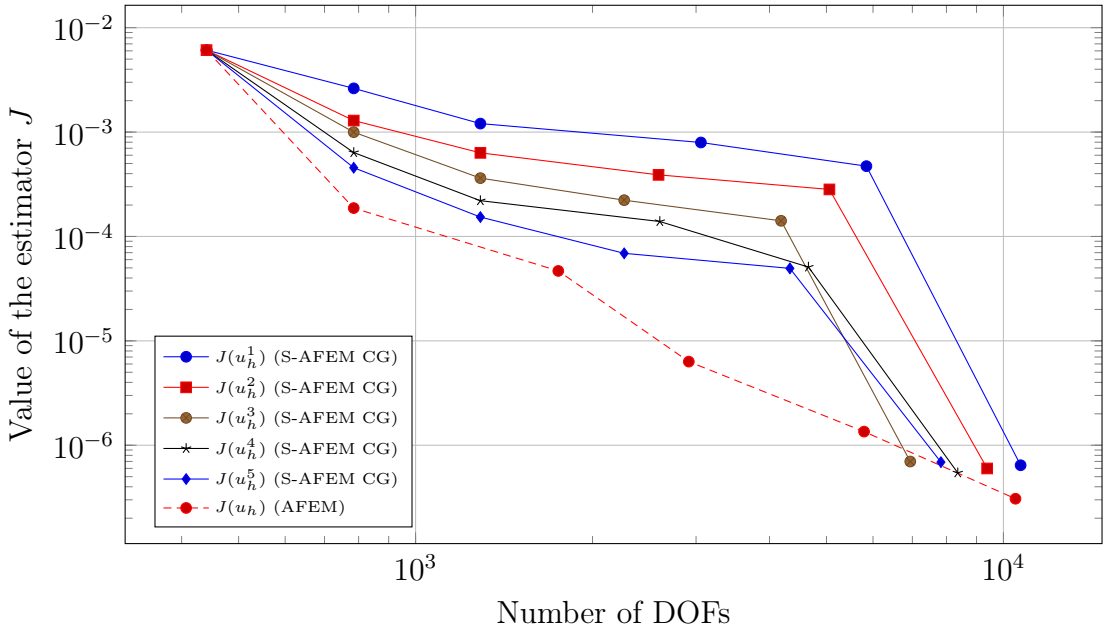


Figure 5.73: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

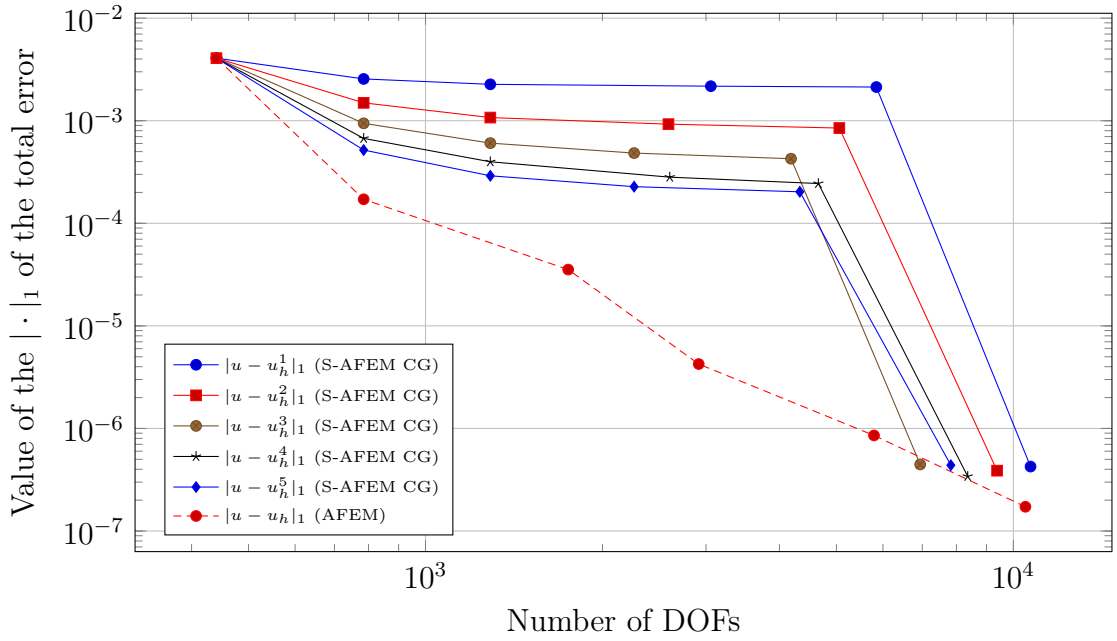


Figure 5.74: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the CG as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

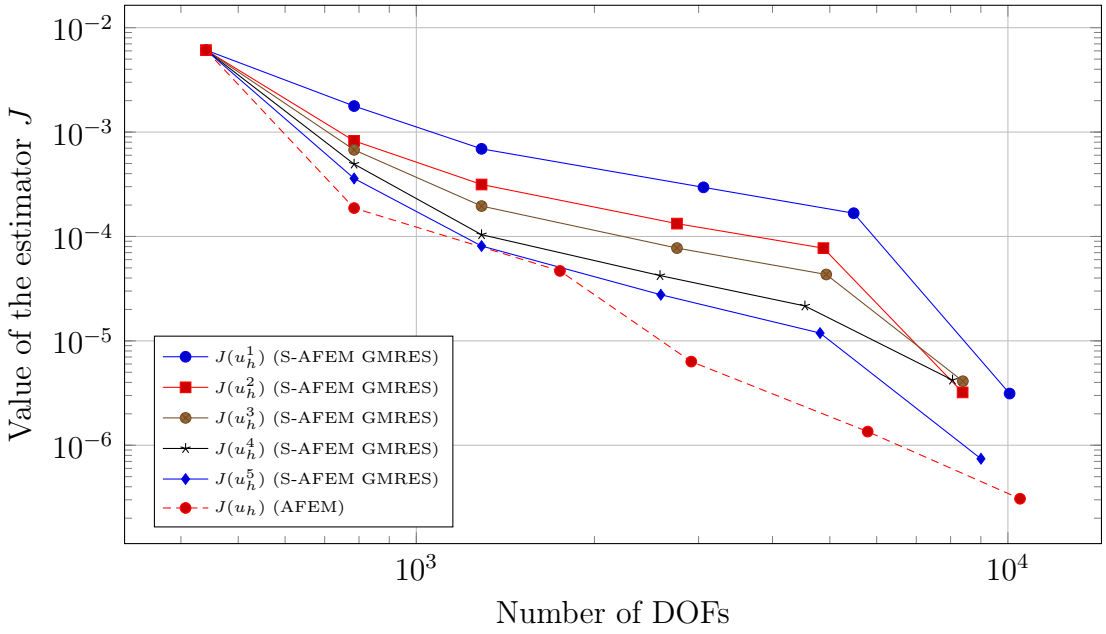


Figure 5.75: Value of the error estimator J for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

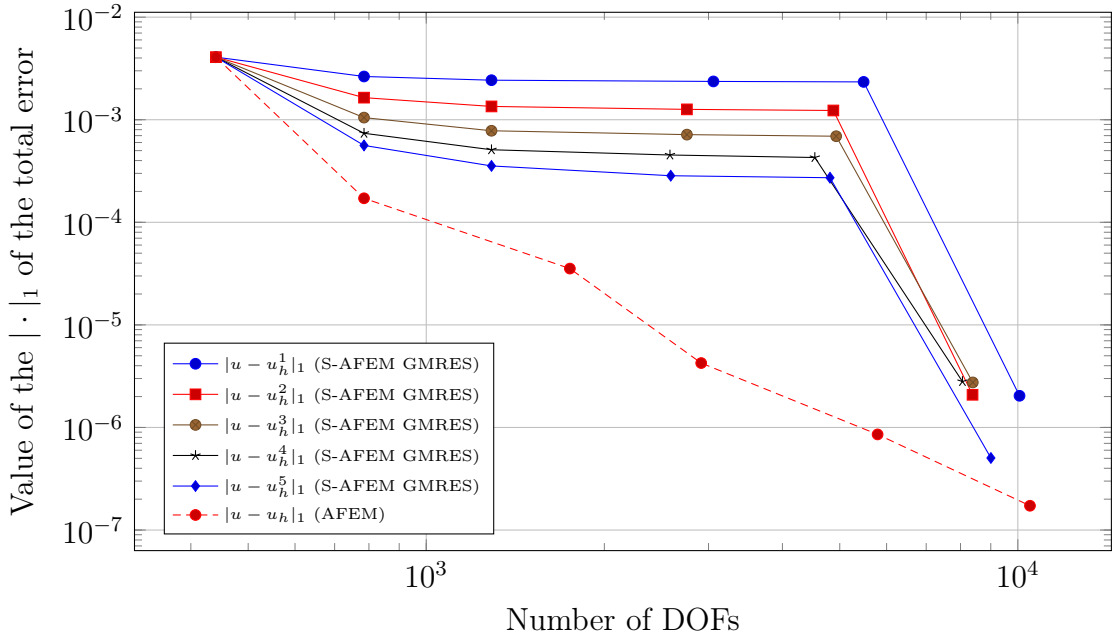


Figure 5.76: Value of the $|\cdot|_1$ of the total error for the peak problem in 2D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with the GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5. The initial global refinement is 2 and we select a fraction of 1/3 of cells for refinement at each cycle.

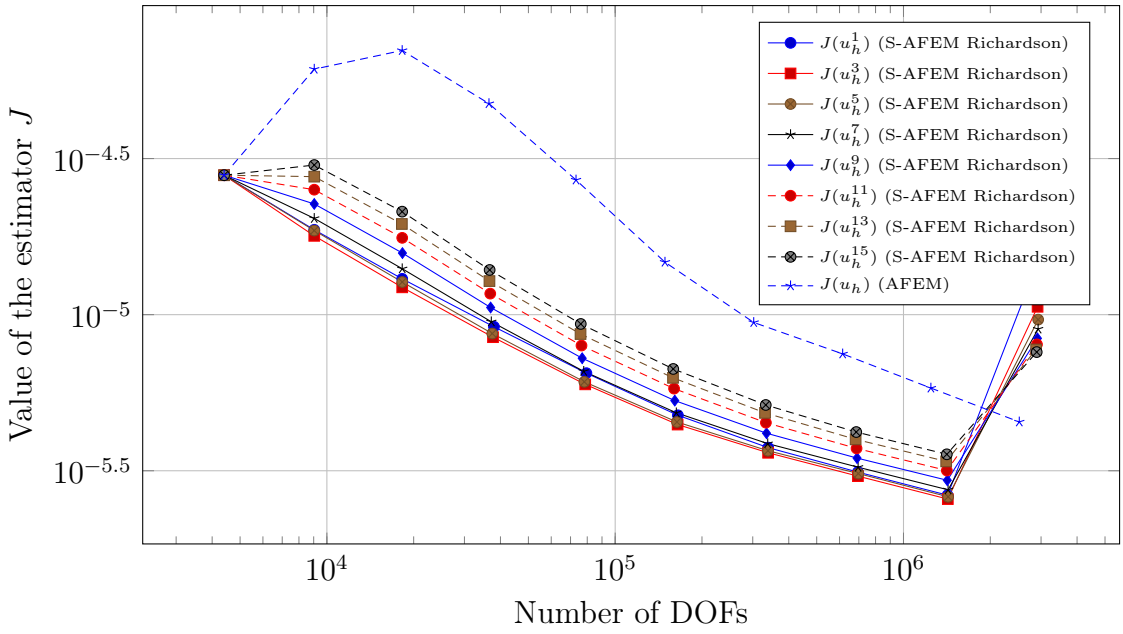


Figure 5.77: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

three or five smoothing steps. Also Richardson iterations shows nearly almost the same accuracy of the final approximation, as shown in Figure 5.78. In this case, immediately after five or seven smoothing steps, the error estimator for S-AFEM behaves as the one for the classical AFEM when the CG and the GMRES are used as smoothers (see 5.79 and 5.81), but it works less well in case of the Richardson iteration, as shown in Figure 5.77.

For $deg = 2$, Figures 5.86 and 5.88 show that the accuracy of the final approximations, for classic AFEM and S-AFEM, is almost the same immediately after three or five smoothing iterations when we use the CG and the GMRES. These smoothers work better than Richardson, which however shows a comparable final accuracy of both methods in Figure 5.84, after three or five iterations.

For $deg = 3, 4, 5$, Figures 5.89-5.106 show that the CG and the GMRES turn out to be better smoothers compared to Richardson iteration. We want to point out that the performance of the Richardson iteration might be improved due to further studies of the evaluation of the optimal relaxation parameter γ, which we are considering as fixed $\gamma = 0.25$. All tests show that the accuracy of the final approximations is nearly the same, and the estimator always underestimates the total error.

Next, we apply both AFEM and S-AFEM to the three dimensional Peak Problem and we show a comparison for different fixed FEM degrees, as deg $= 1, \ldots, 5$, and for different choices of smoothers for the intermediate cycles, respectively the Richardson iteration, the CG method and the GMRES method. For all cases,

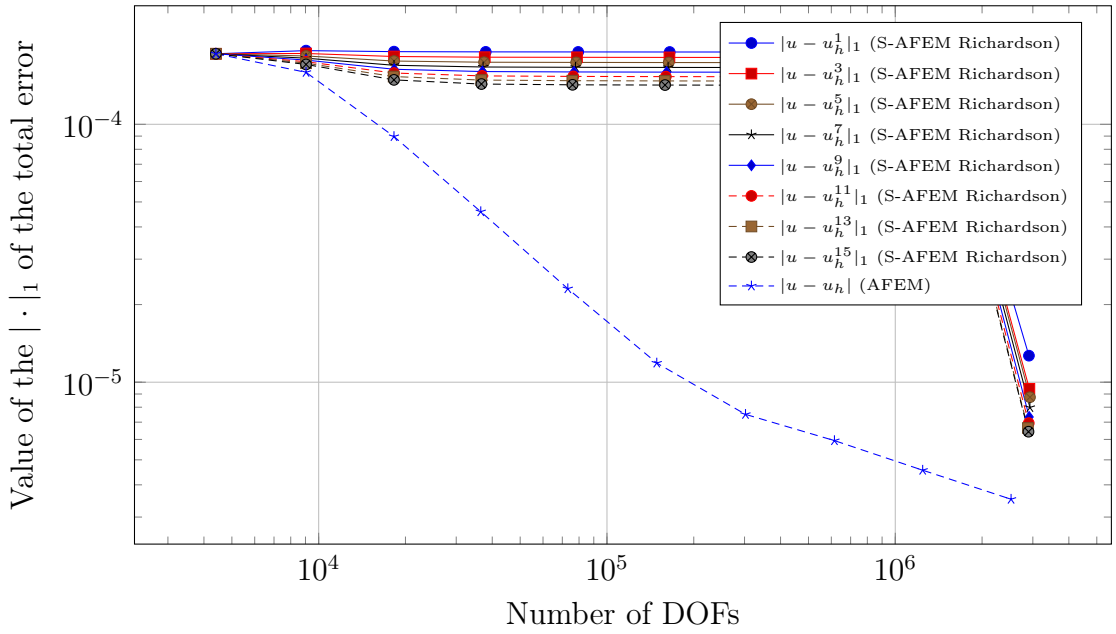


Figure 5.78: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

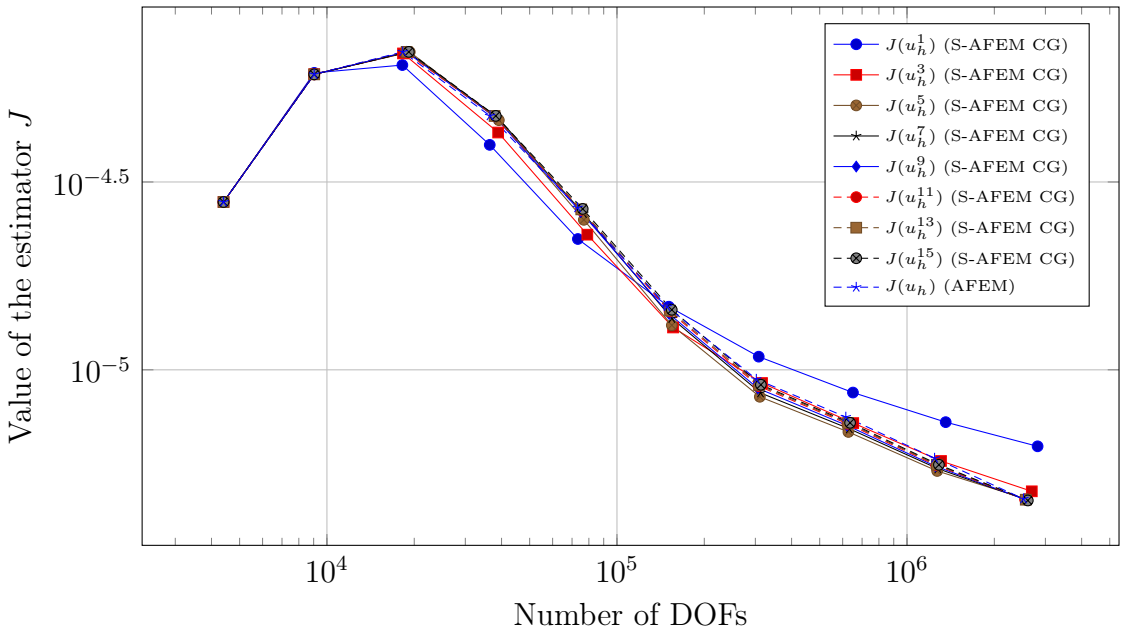


Figure 5.79: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

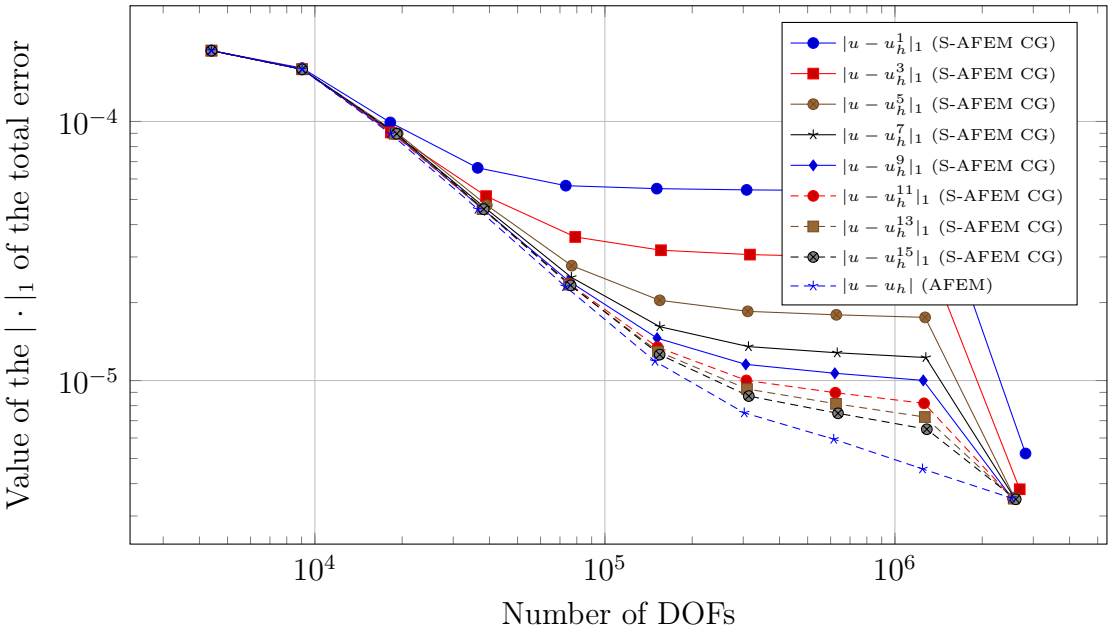


Figure 5.80: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

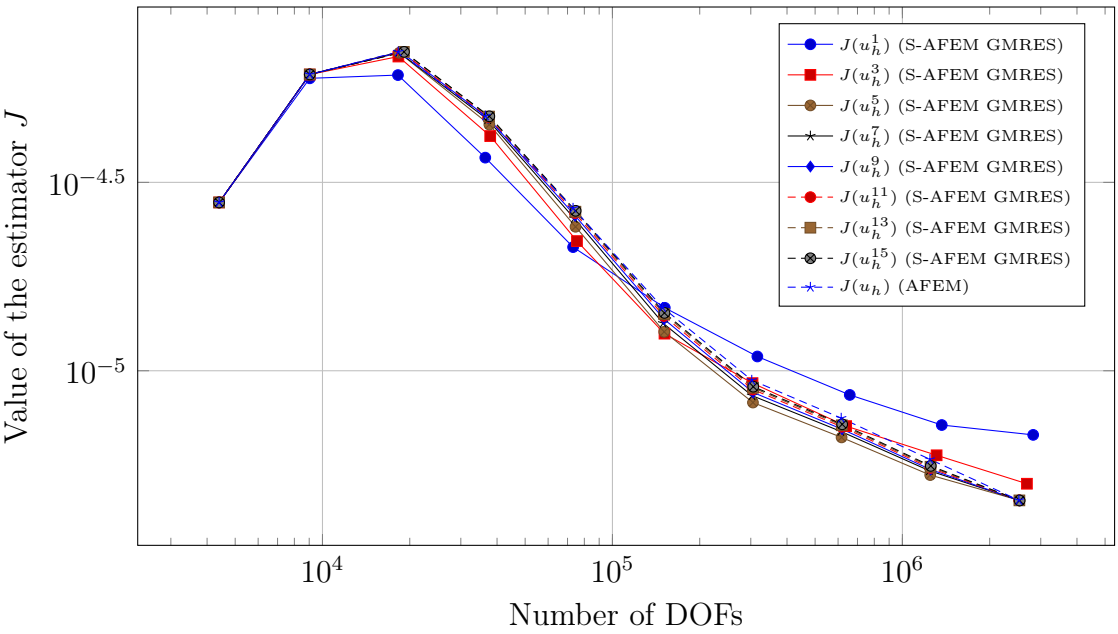


Figure 5.81: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

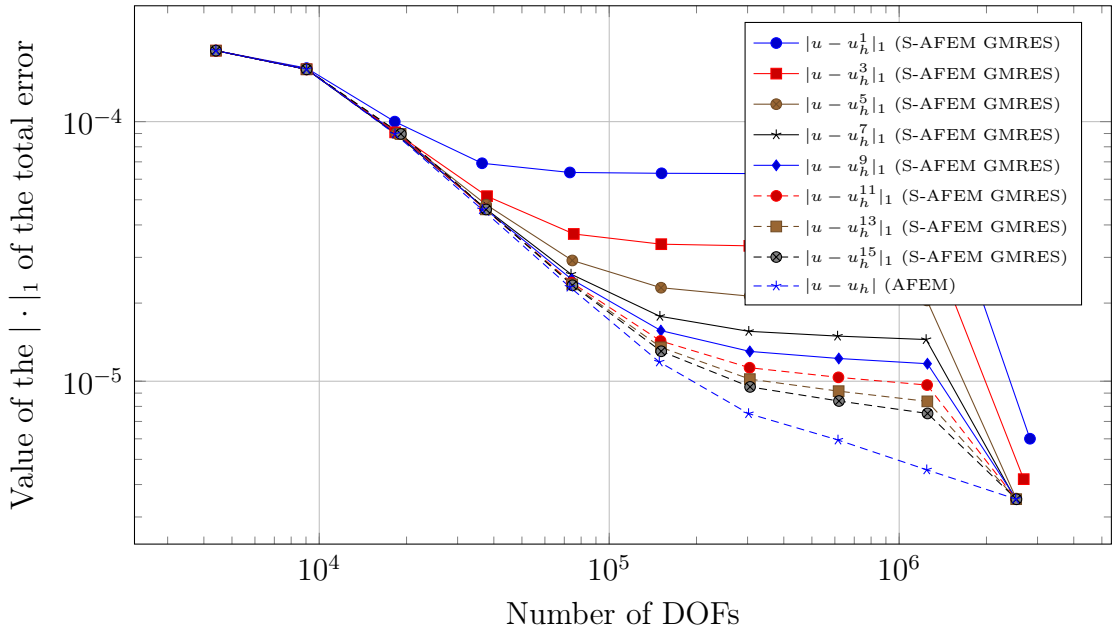


Figure 5.82: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

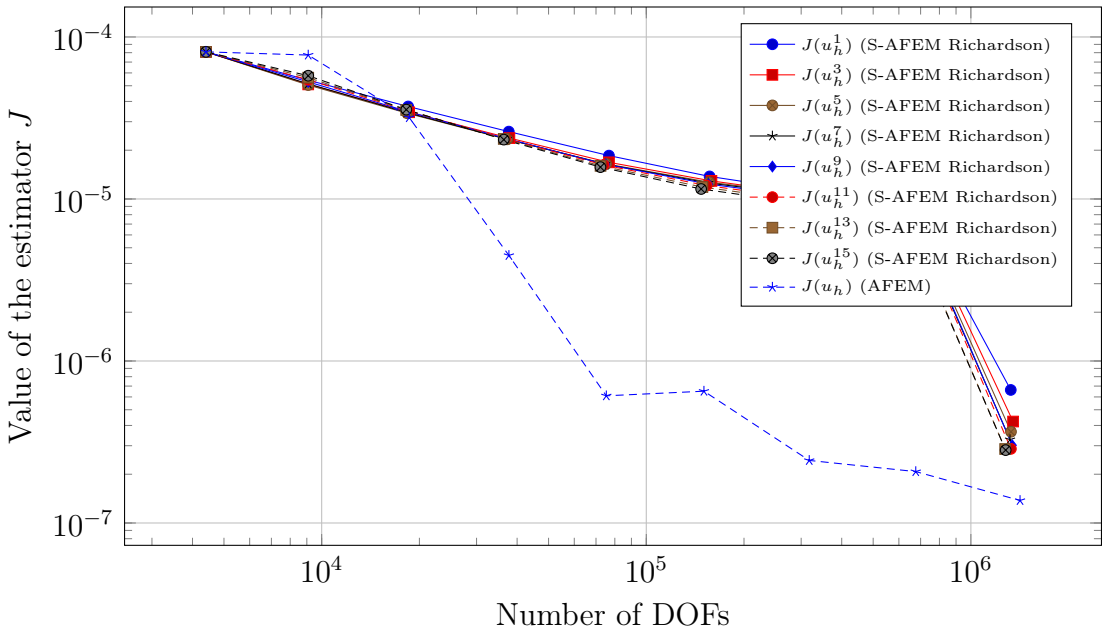


Figure 5.83: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

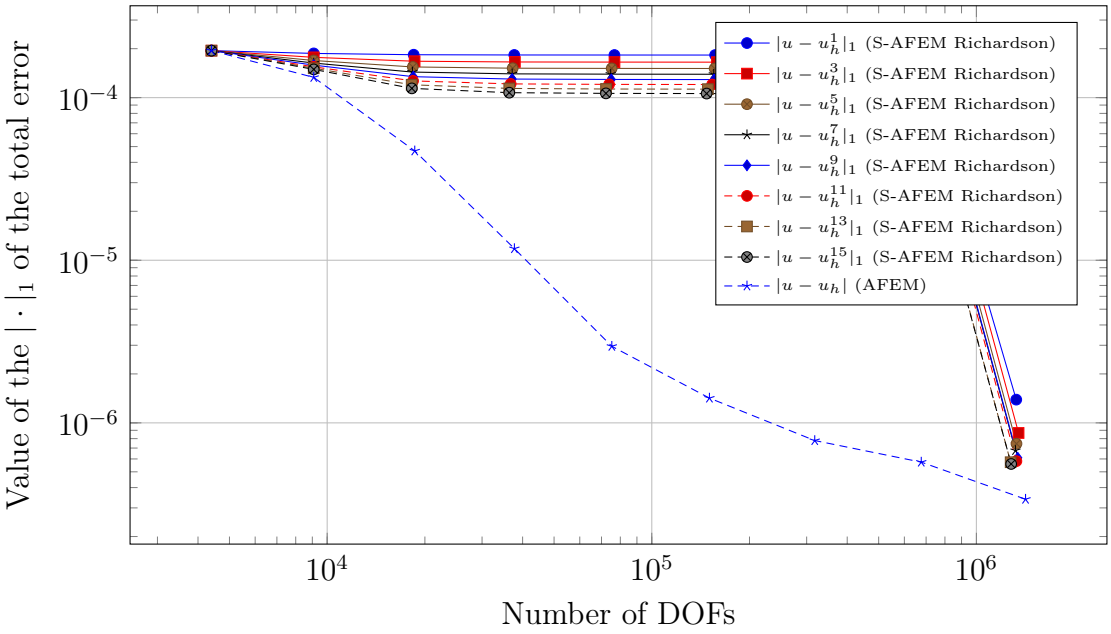


Figure 5.84: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

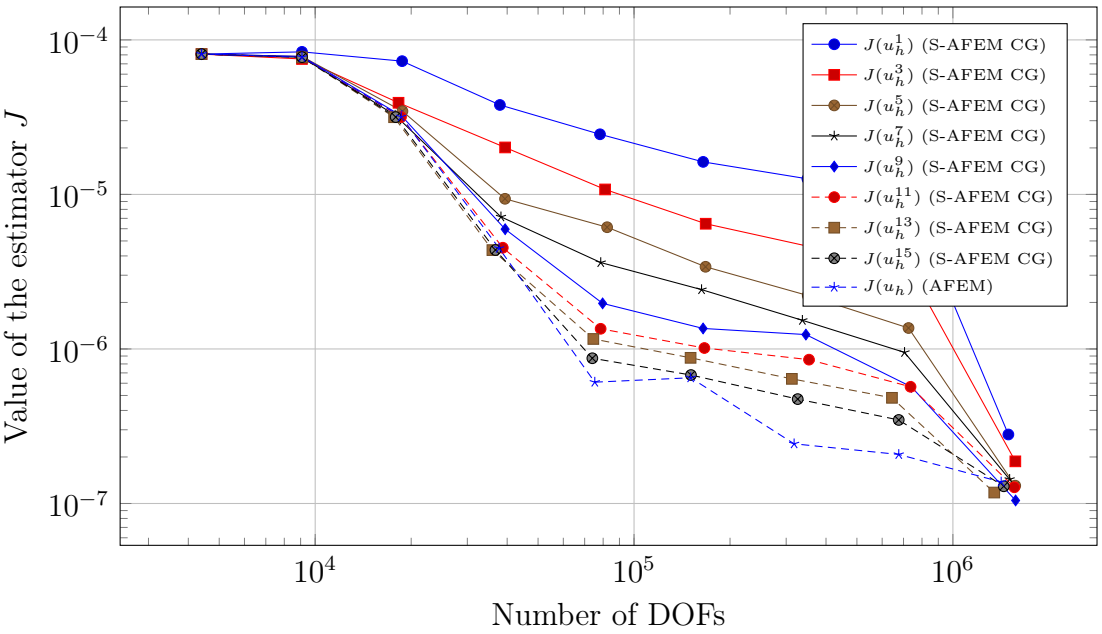


Figure 5.85: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

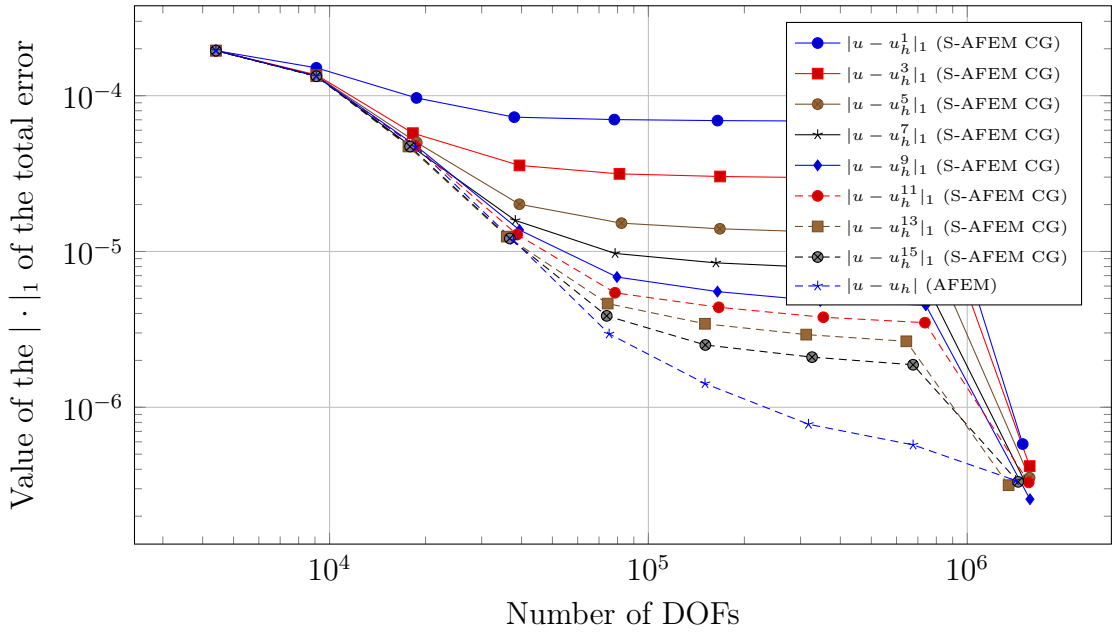


Figure 5.86: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

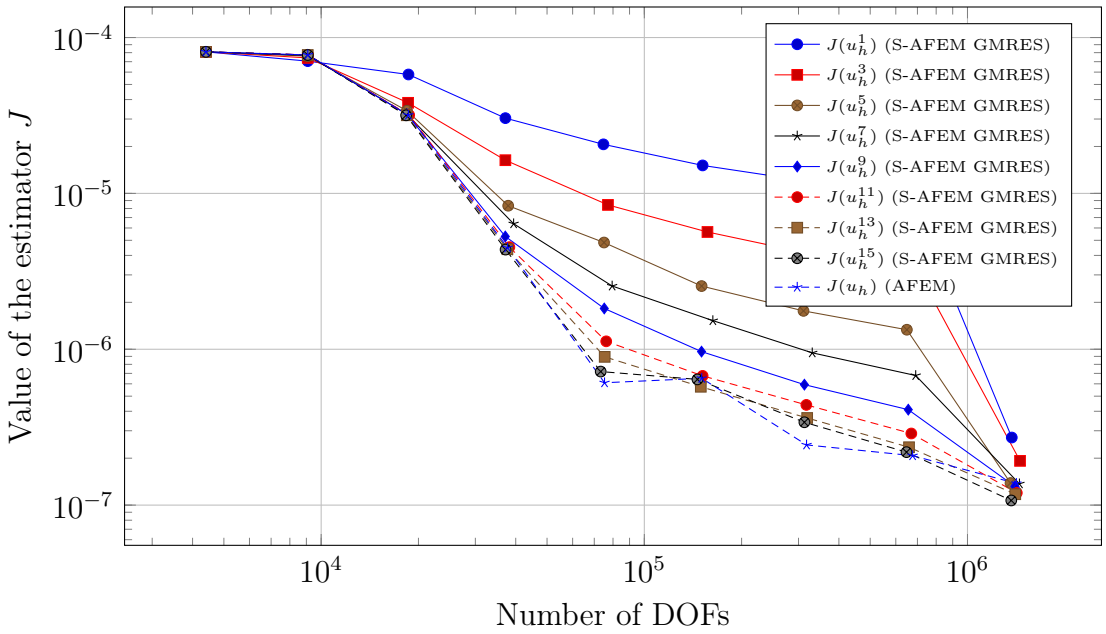


Figure 5.87: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

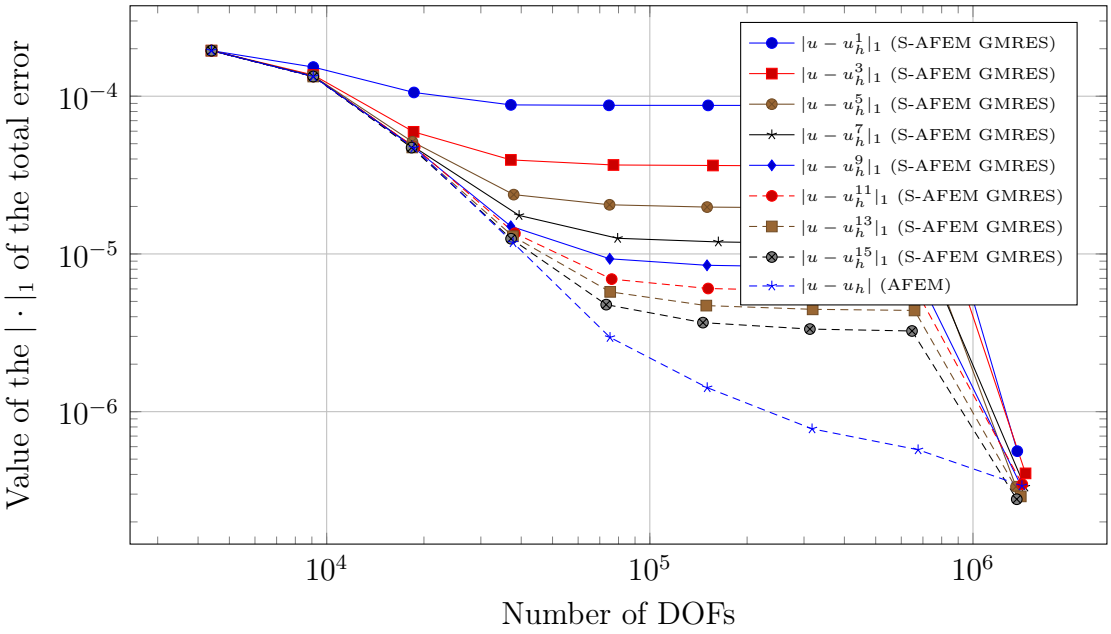


Figure 5.88: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

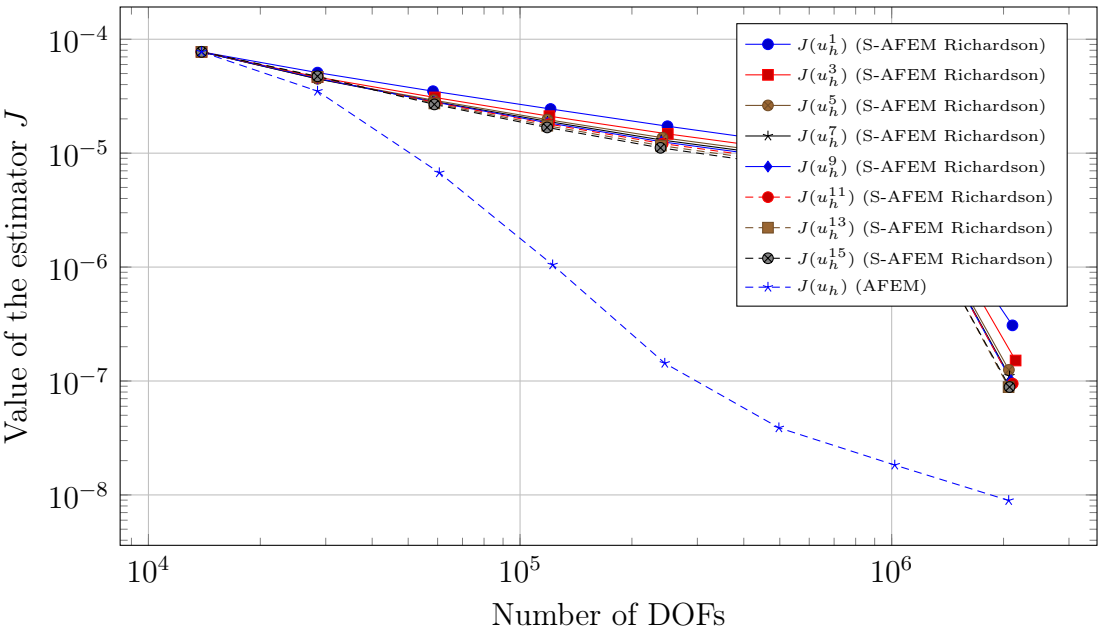


Figure 5.89: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

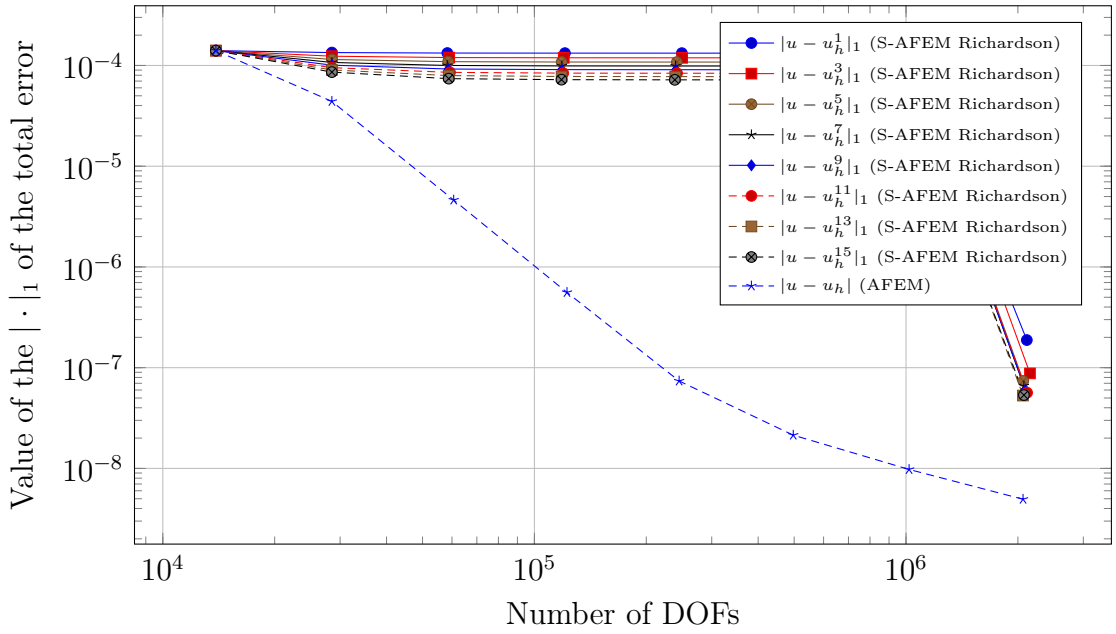


Figure 5.90: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

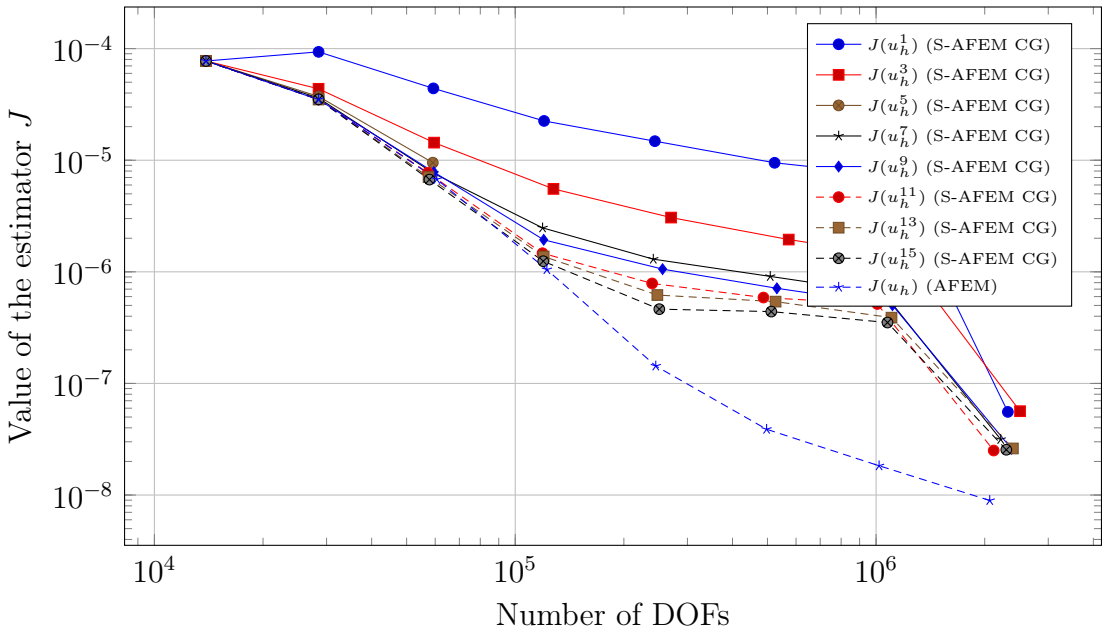


Figure 5.91: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

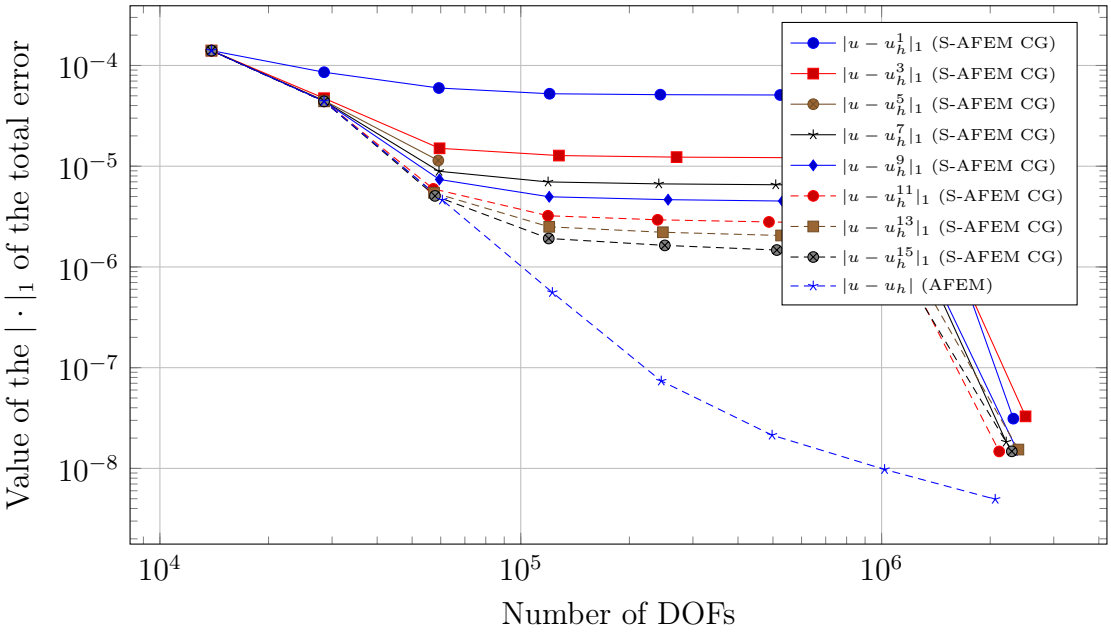


Figure 5.92: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

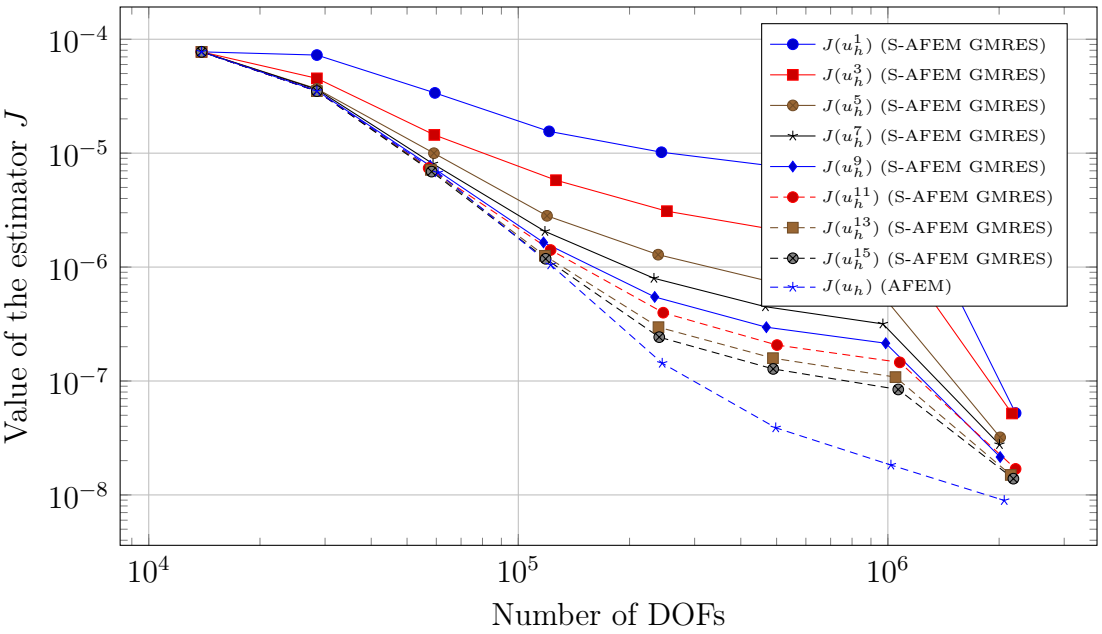


Figure 5.93: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

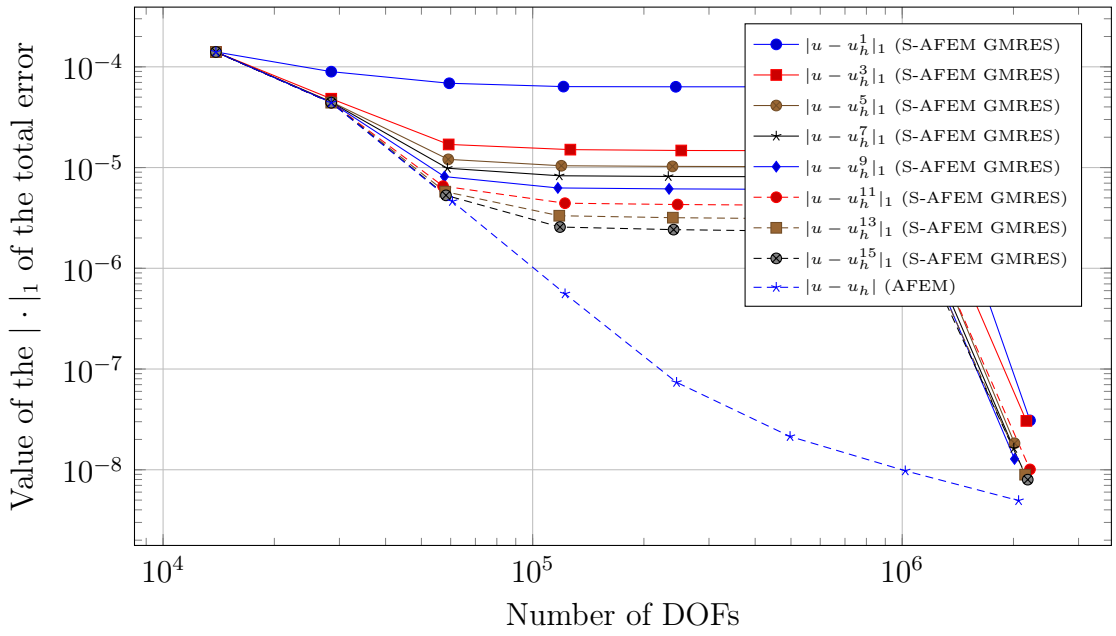


Figure 5.94: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

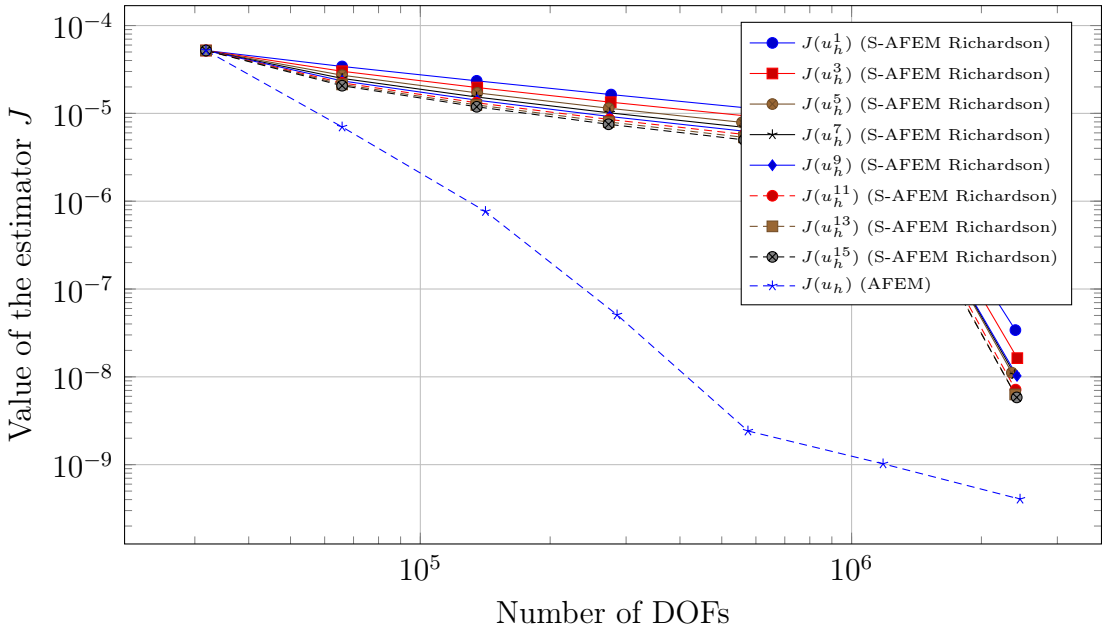


Figure 5.95: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

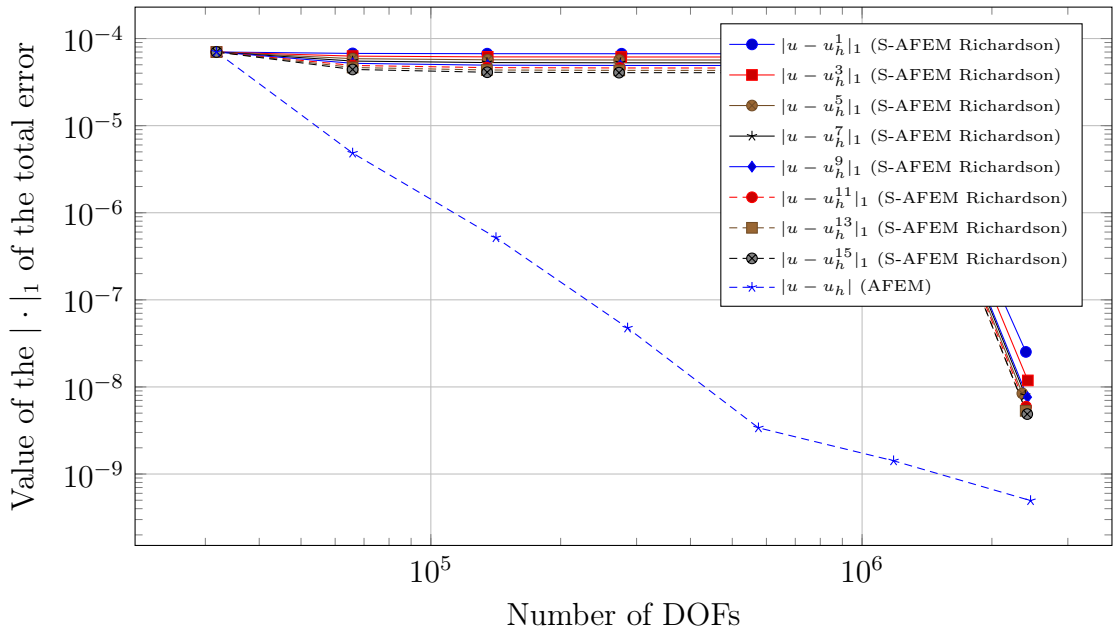


Figure 5.96: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

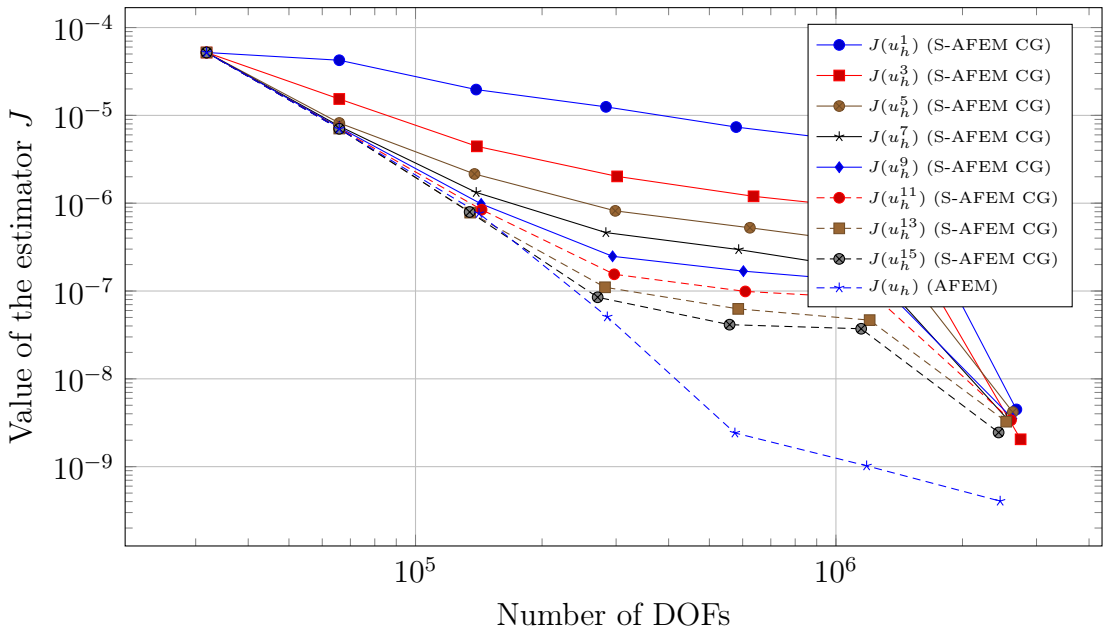


Figure 5.97: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

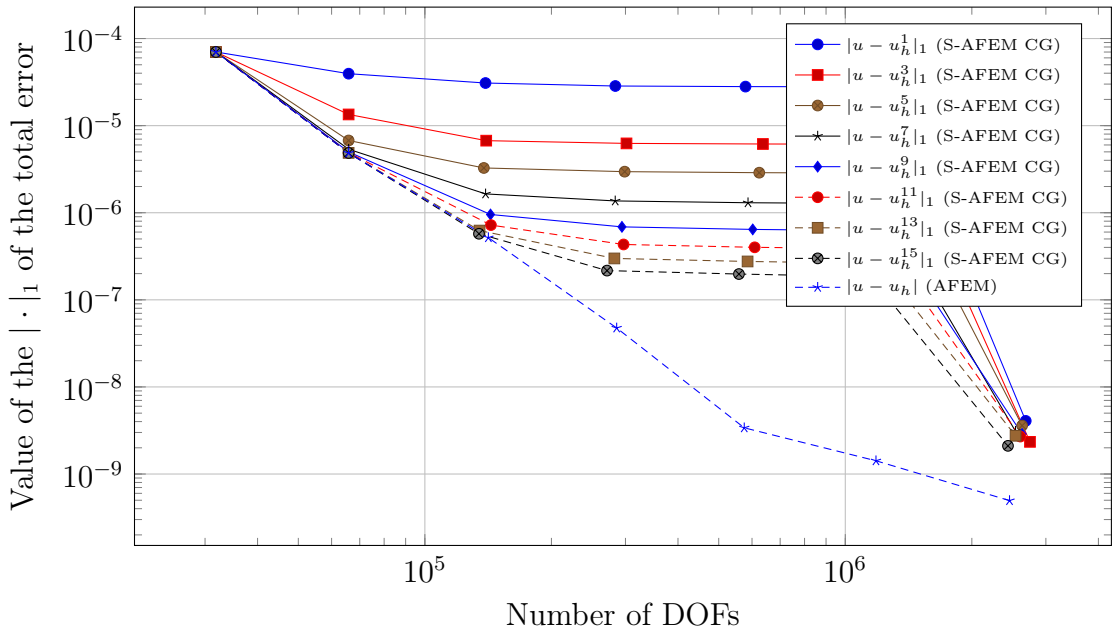


Figure 5.98: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

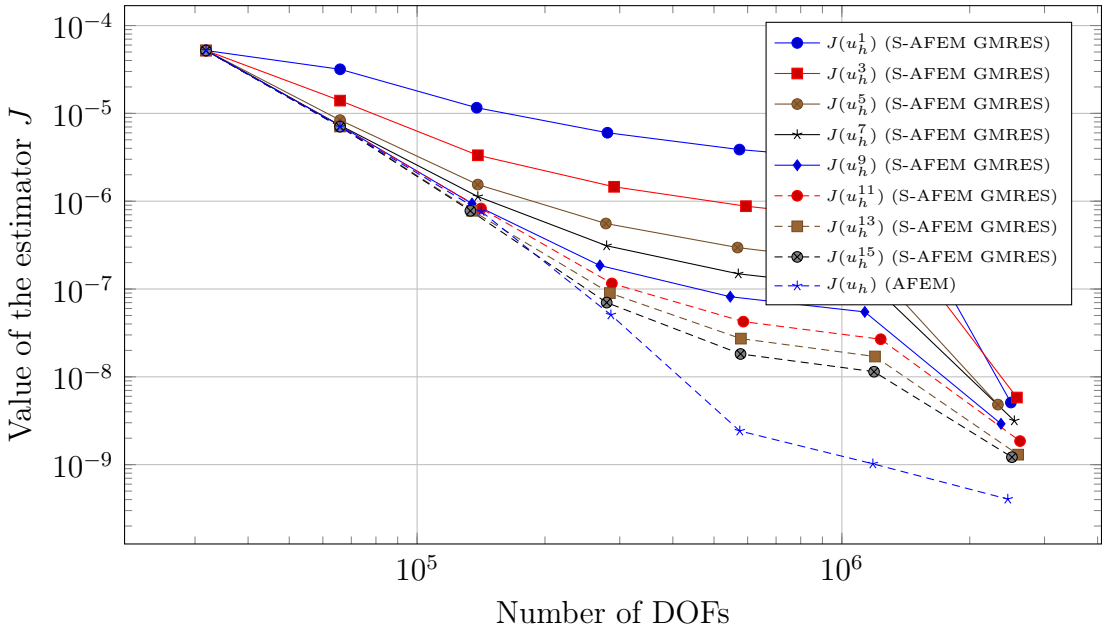


Figure 5.99: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

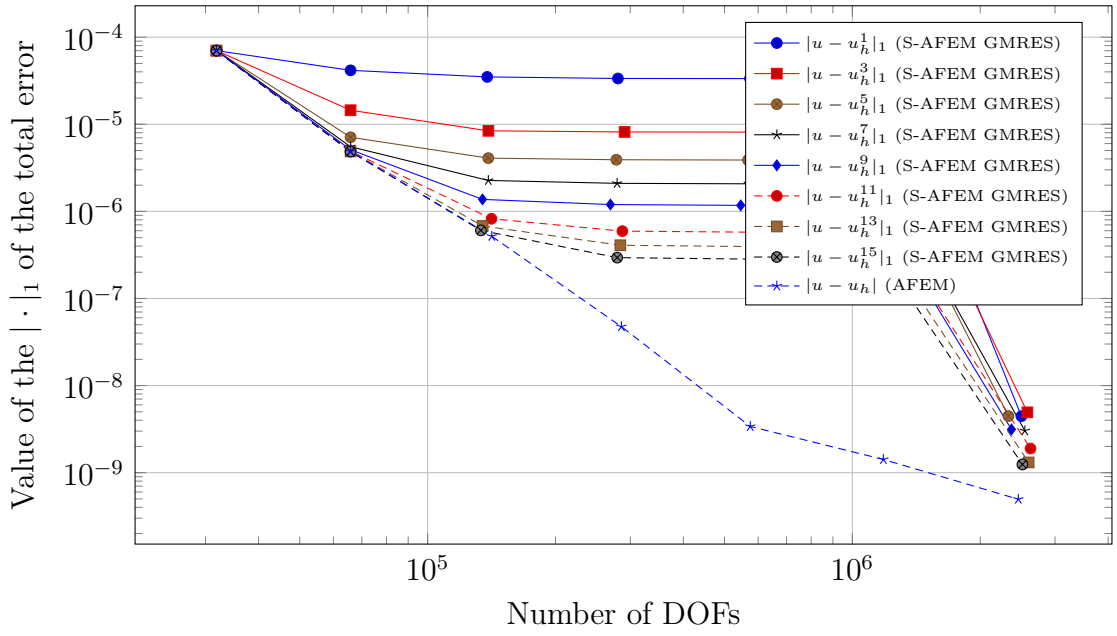


Figure 5.100: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

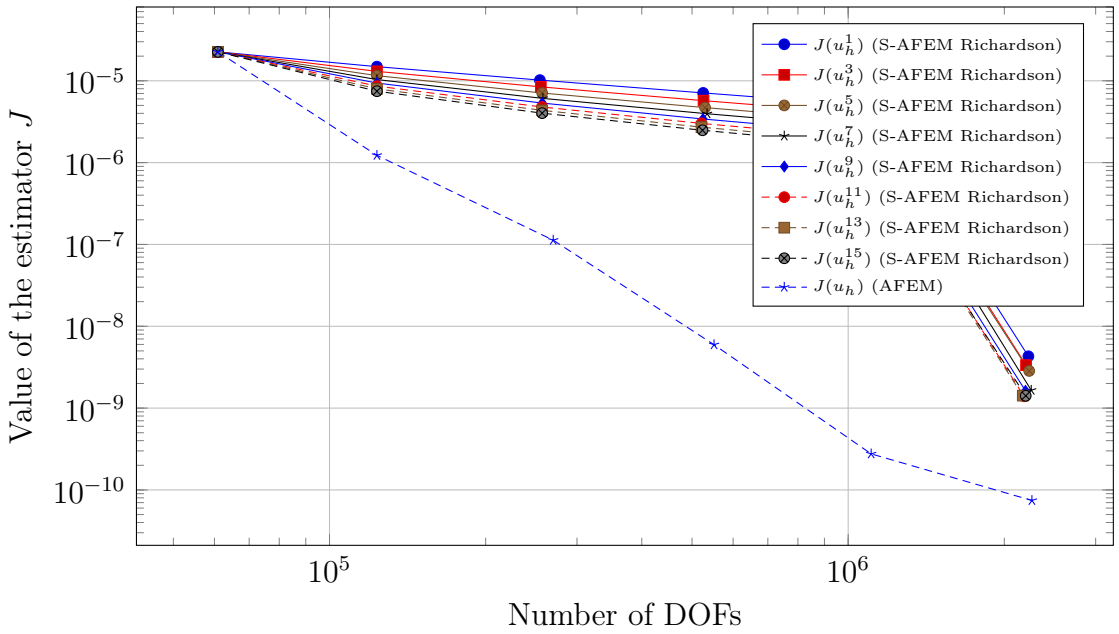


Figure 5.101: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

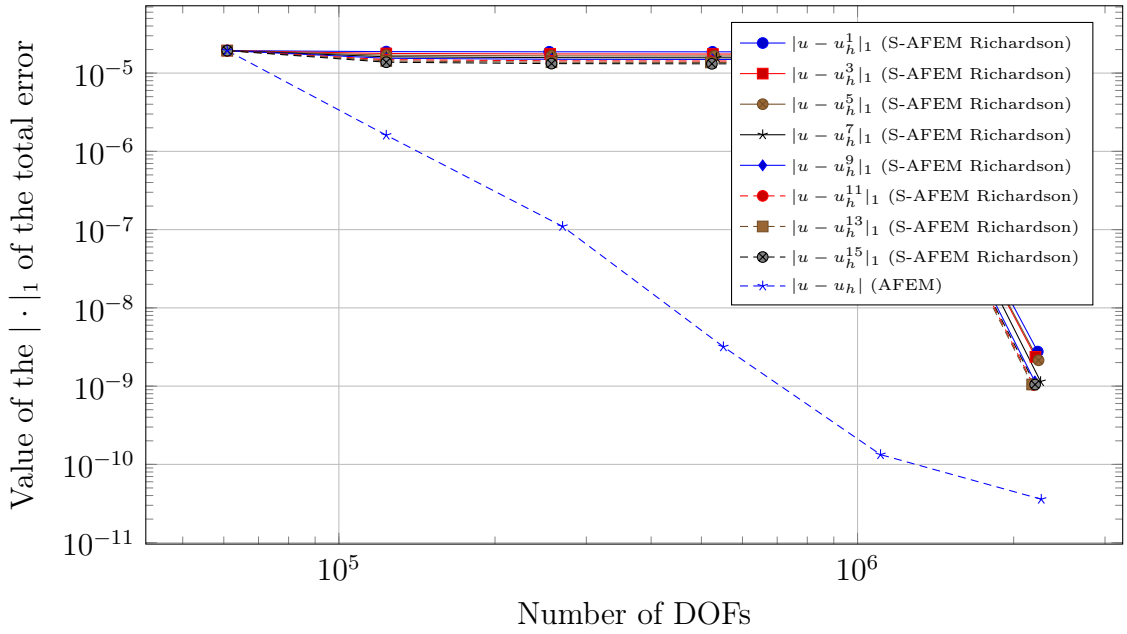


Figure 5.102: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

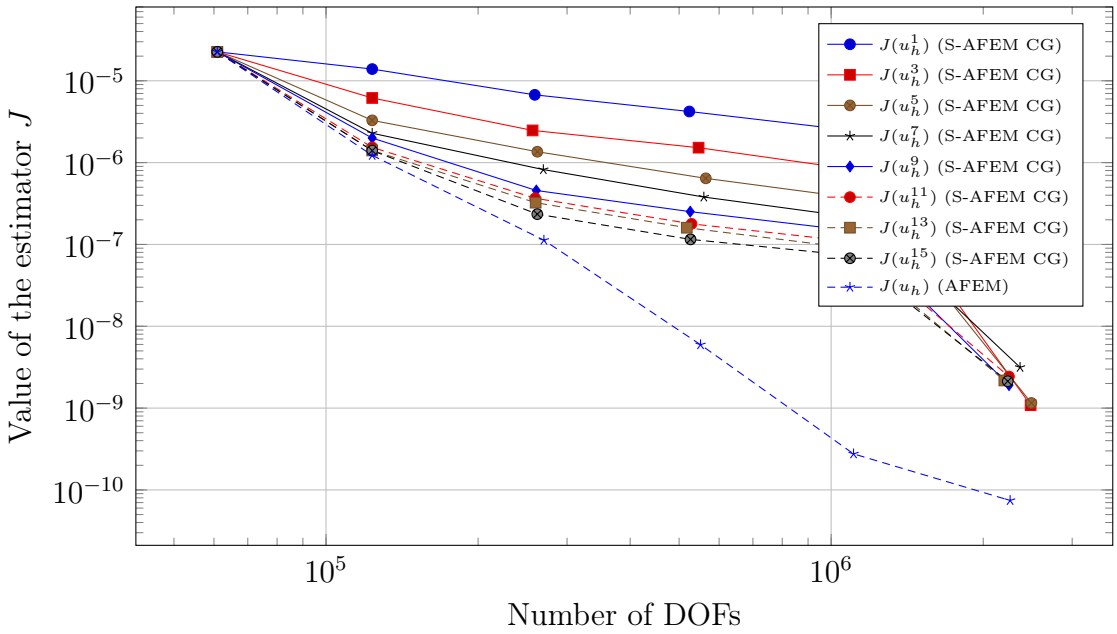


Figure 5.103: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

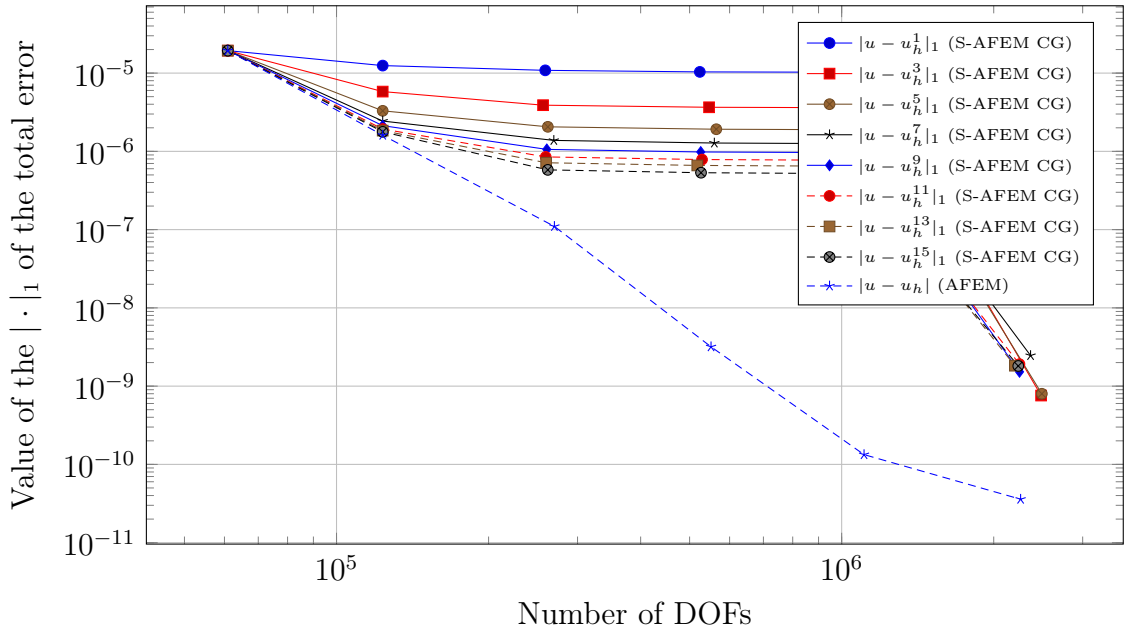


Figure 5.104: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

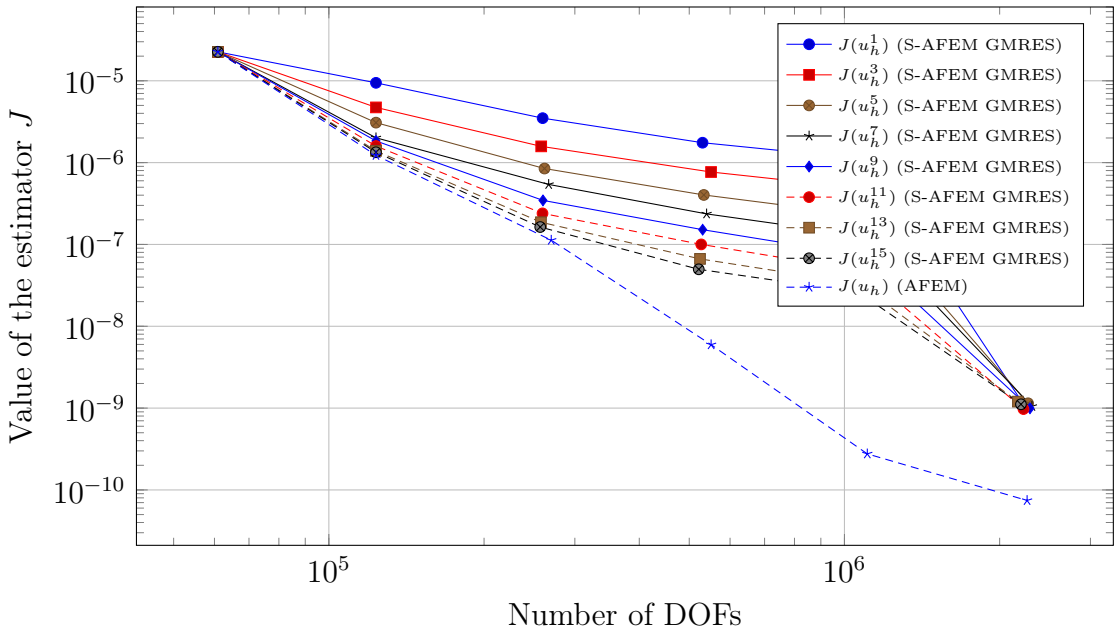


Figure 5.105: Value of the error estimator J for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

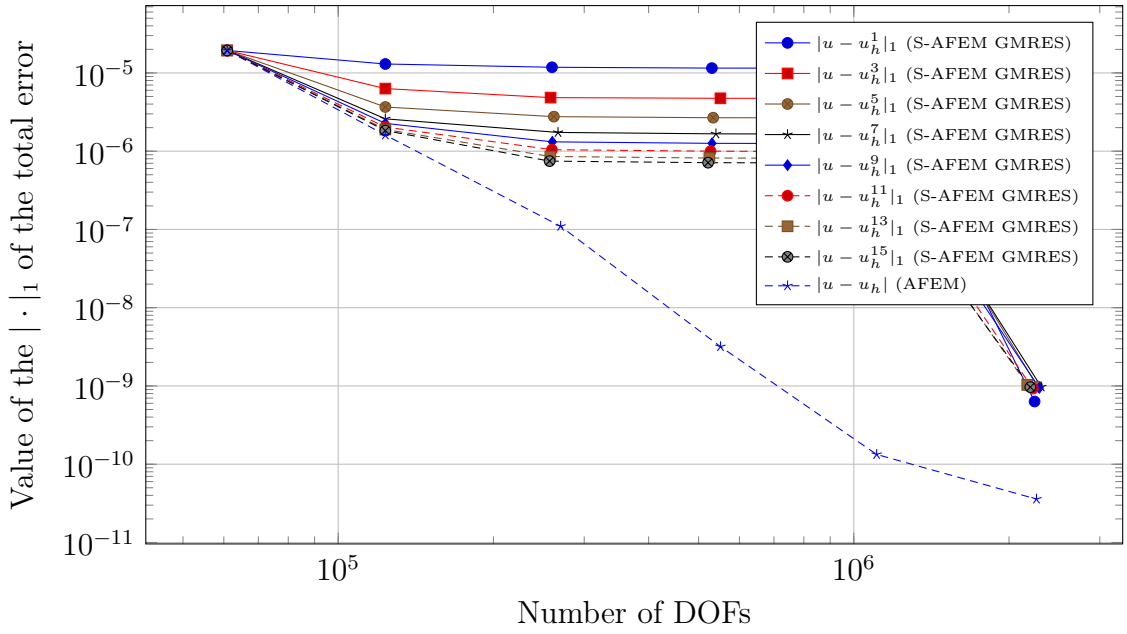


Figure 5.106: Value of the $|\cdot|_1$ error for the Fichera corner problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

we plot the value of the error estimator J and the value of the $|\cdot|_1$ seminorm of the total error, as the number of smoothing iterations $l = 2i - 1$ increases for $i = 1, \ldots, 8$. As a marking criterion we select a fraction of $1/7$ of cells with the largest error indicator to be marked for refinement.

For piecewise bi-linear finite elements, when the Richardson iteration is used as a smoother, the estimator for S-AFEM exhibits the same order of convergence as the estimator for classic AFEM starting from five to seven smoothing iterations, as shown in Figure 5.107, and the final errors have almost the same accuracy, as shown in Figure 5.108 Surprisingly, things work perfectly when the CG method is used as a smoother: after three CG smoothing iterations both the estimator for S-AFEM and for classic AFEM show the same order of convergence, as shown in Figure 5.109, and the accuracy of the final approximation is the same, as shown in Figure 5.108. The same behaviour is shown when the GMRES method is used as a smoother, as shown in Figures 5.111 and 5.112.

For higher degree finite element approximations, we can see how the presence of the algebraic error influences the error estimator, which turns out to be quasi optimal generally after five or seven iterations when the CG and GMRES are used as smoothers, while it shows less optimal behaviour compared to classic AFEM, when the Richardson iteration is used as a smoother. We again point out that the performance of the Richardson iteration might be improved due to further studies of the evaluation of the optimal relaxation parameter γ, which we are considering as fixed $\gamma = 0.25$. Finally, as evidenced in

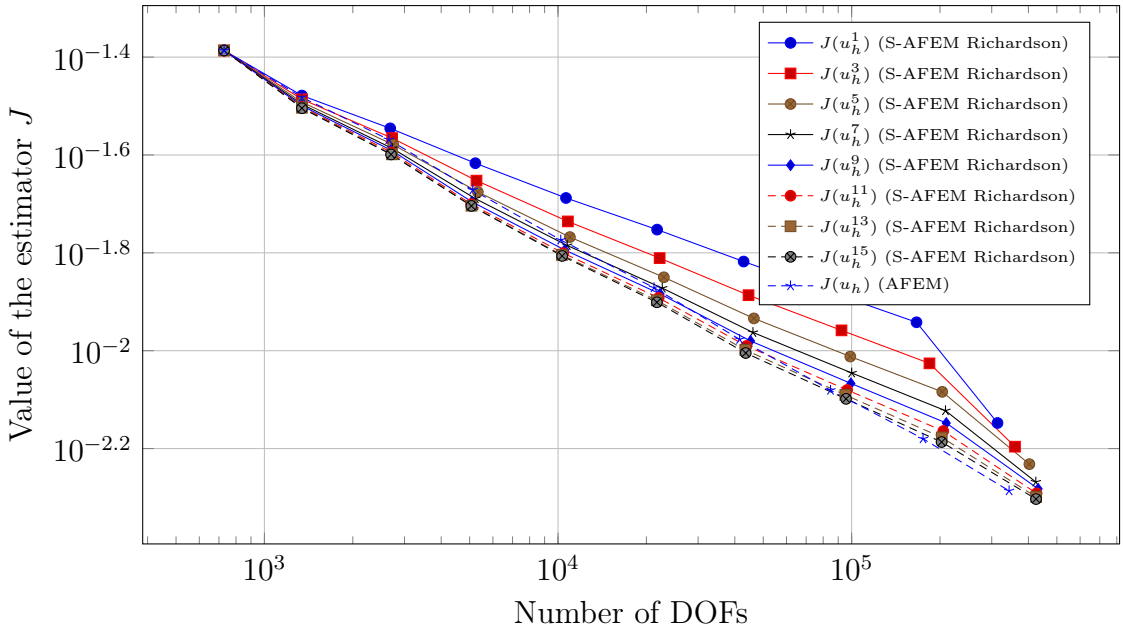


Figure 5.107: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

Figures 5.114, 5.116, 5.118, 5.120, 5.122, 5.124, 5.126, 5.128, 5.130, 5.132, 5.134 and 5.136 the accuracy of the final solutions produced by S-AFEM and by classic AFEM is almost the same and S-AFEM turns out to be a good method also for higher-order FEM discretisations.

5.5 S-AFEM applied to diffusion-transport problems: an example

In this section, we test the accuracy of S-AFEM for the class of diffusion-transport problems of the following form

$$-\Delta u + \boldsymbol{\beta} \cdot \nabla u = f \text{ in } \Omega \text{ and } u = 0 \text{ on } \partial\Omega, \tag{5.5}$$

where we add to our classical model problem the transport term $\boldsymbol{\beta} \cdot \nabla u$. For our numerical examples, we will consider a two-dimensional problem where the transport term $\boldsymbol{\beta} \cdot \nabla u$ is determined by $\boldsymbol{\beta} = (\beta, \beta)^T$, where the parameter β takes the values $1, 10$ and 50. It is well known that in this case Problem (5.5) has a unique solution (cf. eg. [77]). In particular, we impose a forcing term equals to $f = 2\beta$ to the system such that the exact solution is

$$x + y + \frac{-e^{\beta x} + 1}{e^{\beta} - 1} - \frac{e^{\beta y} - 1}{e^{\beta} - 1}. \tag{5.6}$$

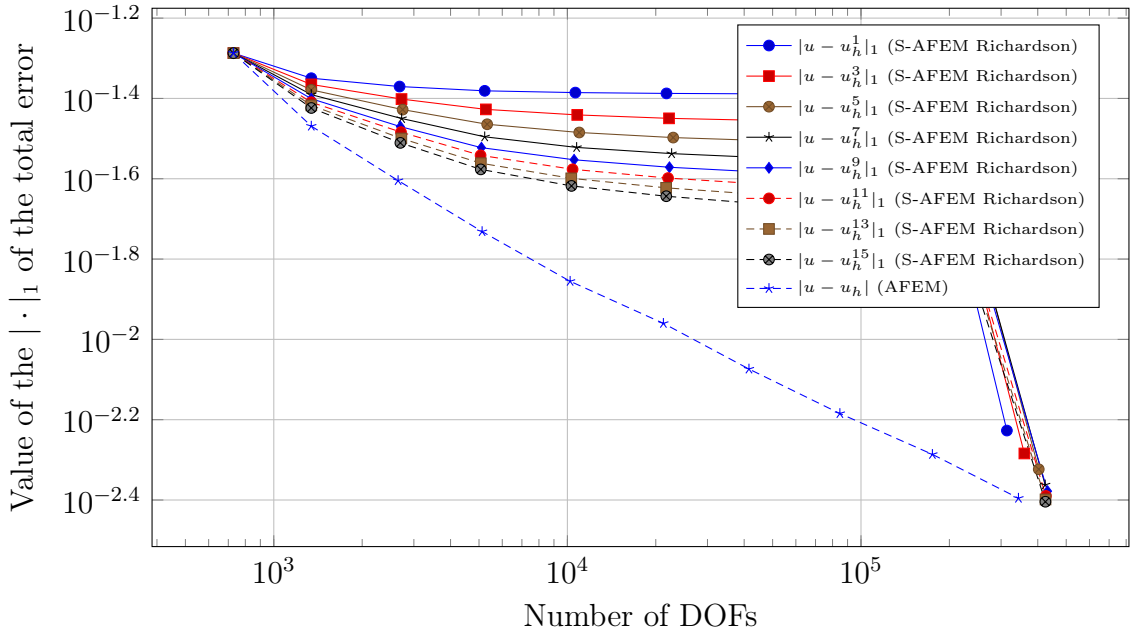


Figure 5.108: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

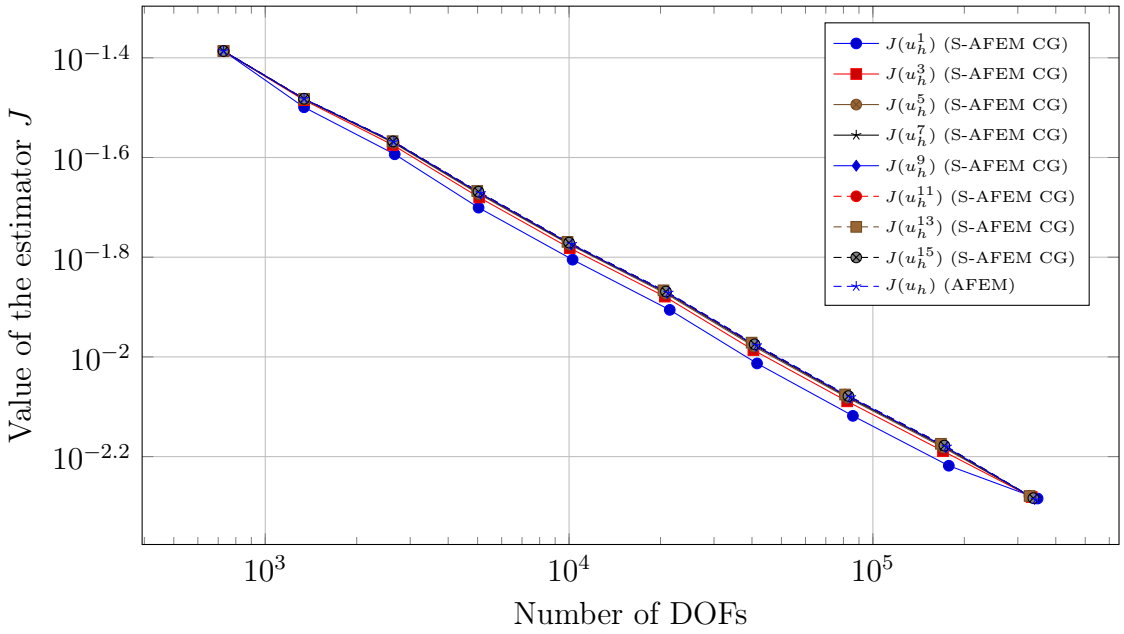


Figure 5.109: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i-1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

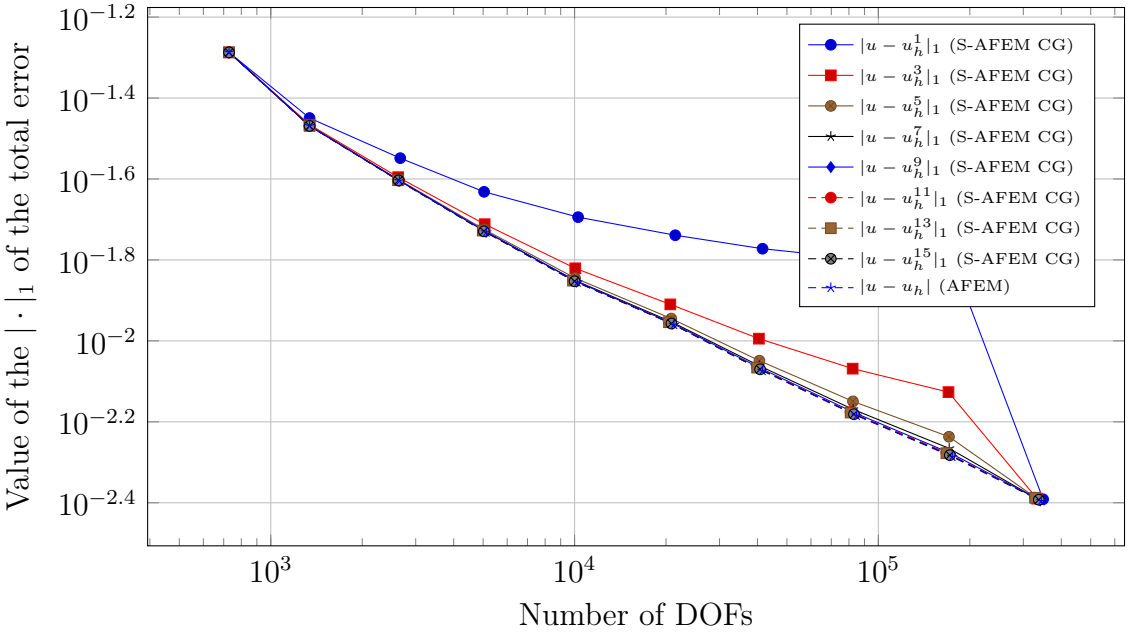


Figure 5.110: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

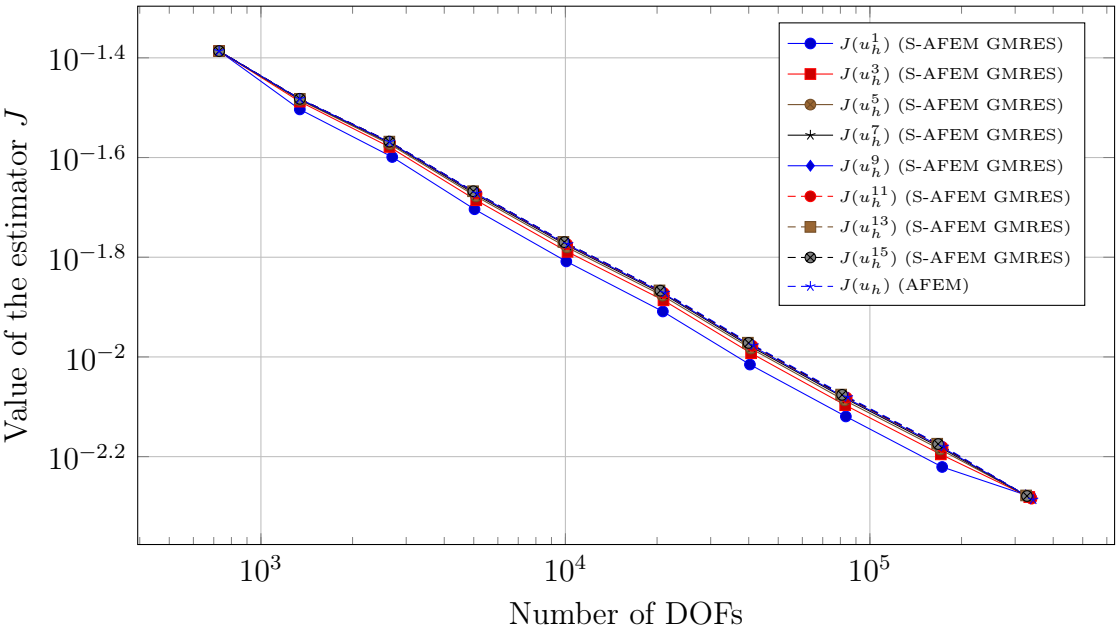


Figure 5.111: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

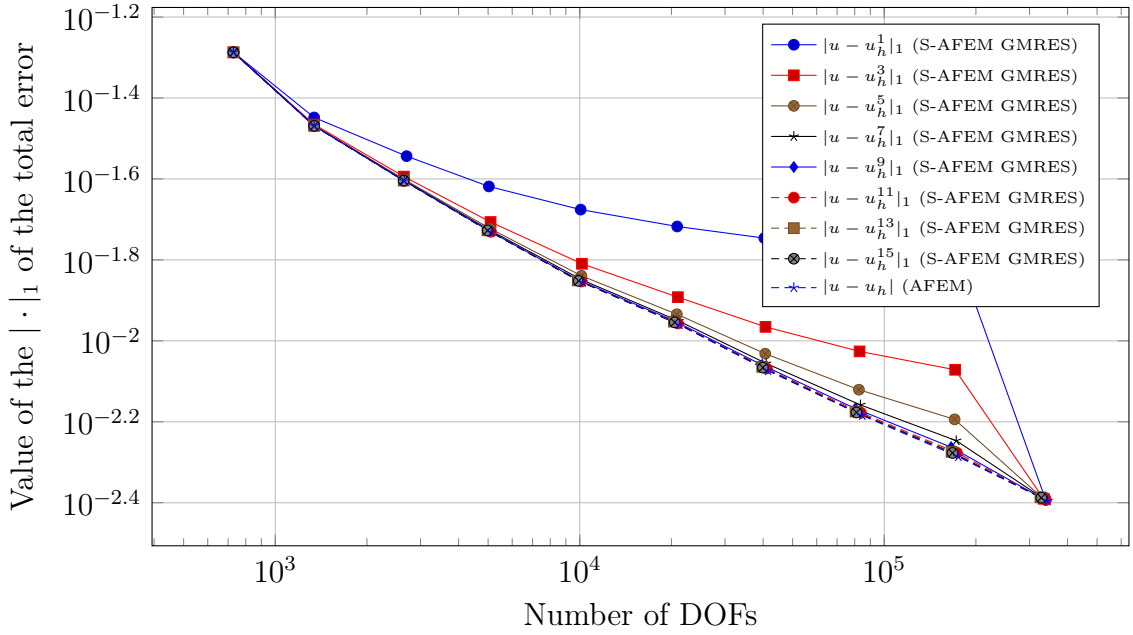


Figure 5.112: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 3 and we select a fraction of 0.15 of all cells for refinement at each cycle

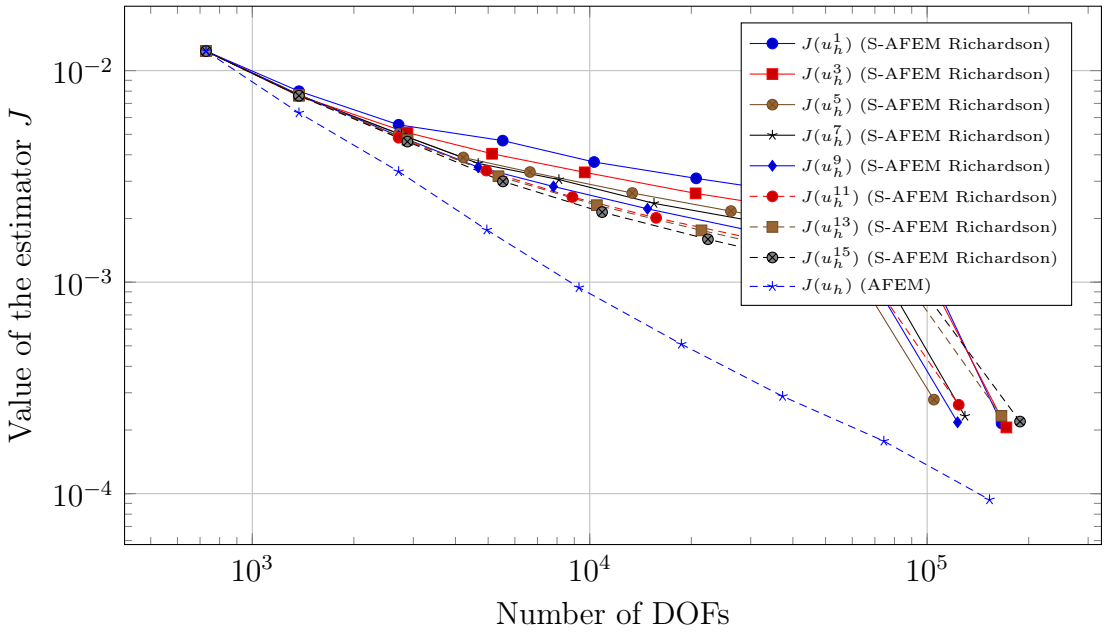


Figure 5.113: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

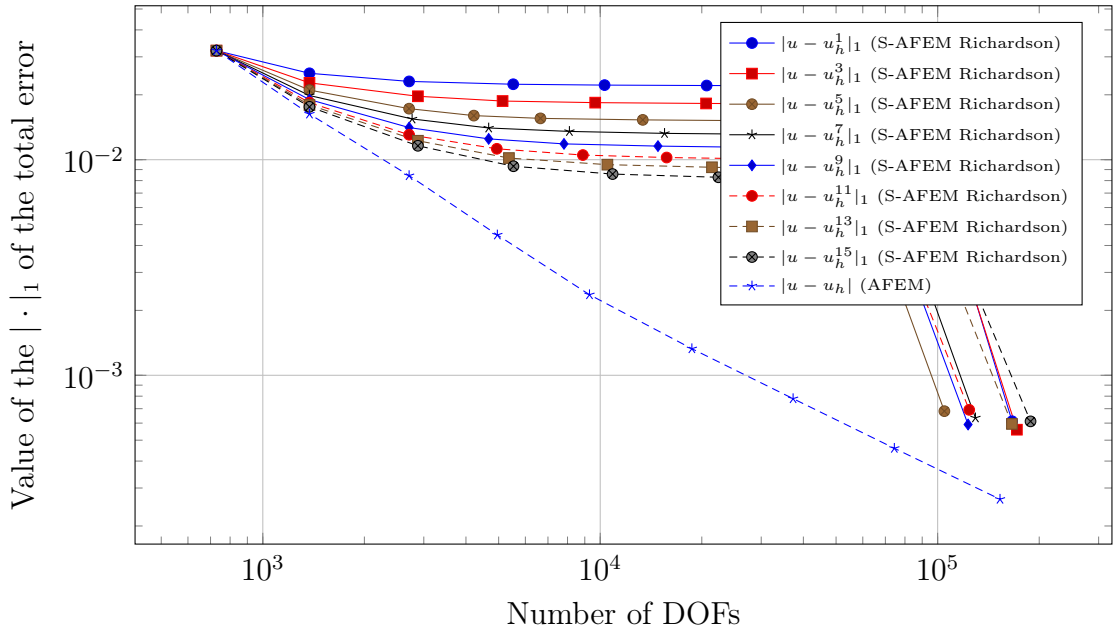


Figure 5.114: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

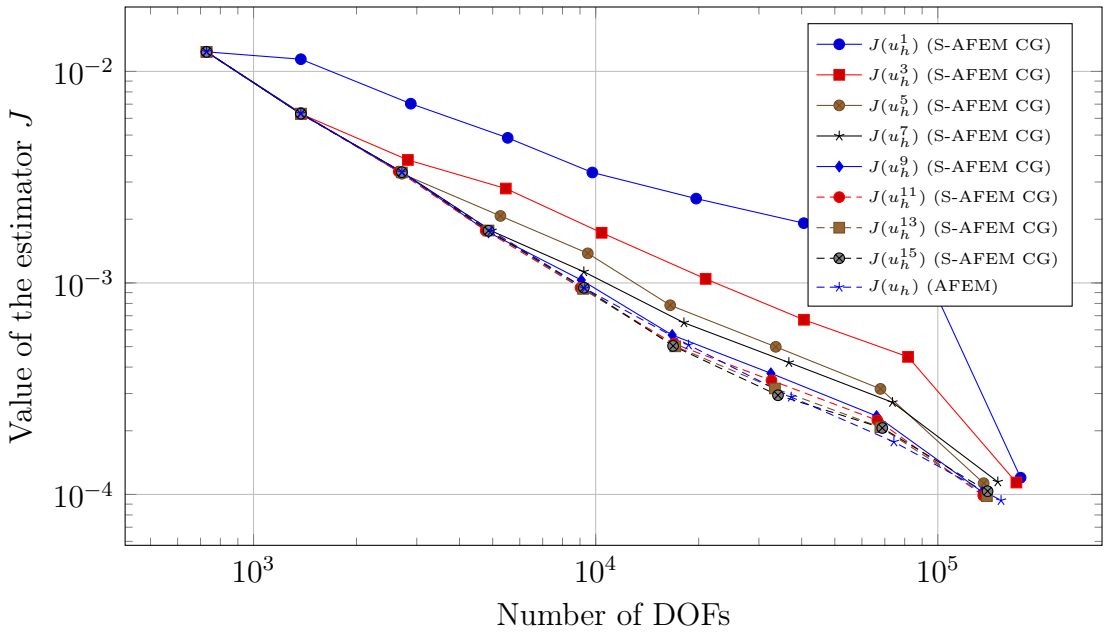


Figure 5.115: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

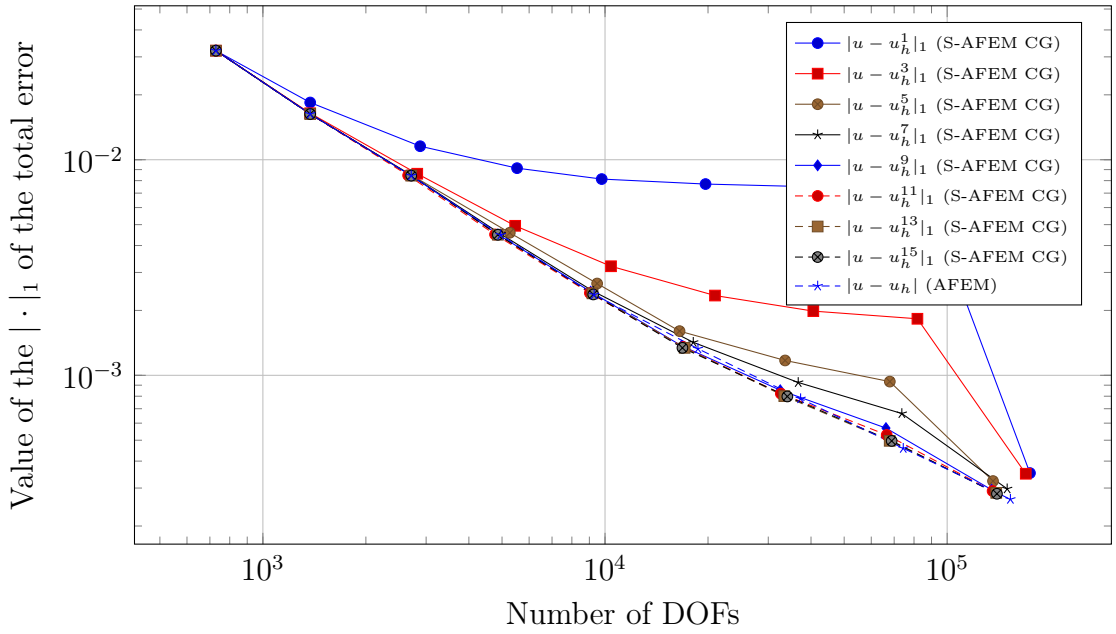


Figure 5.116: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

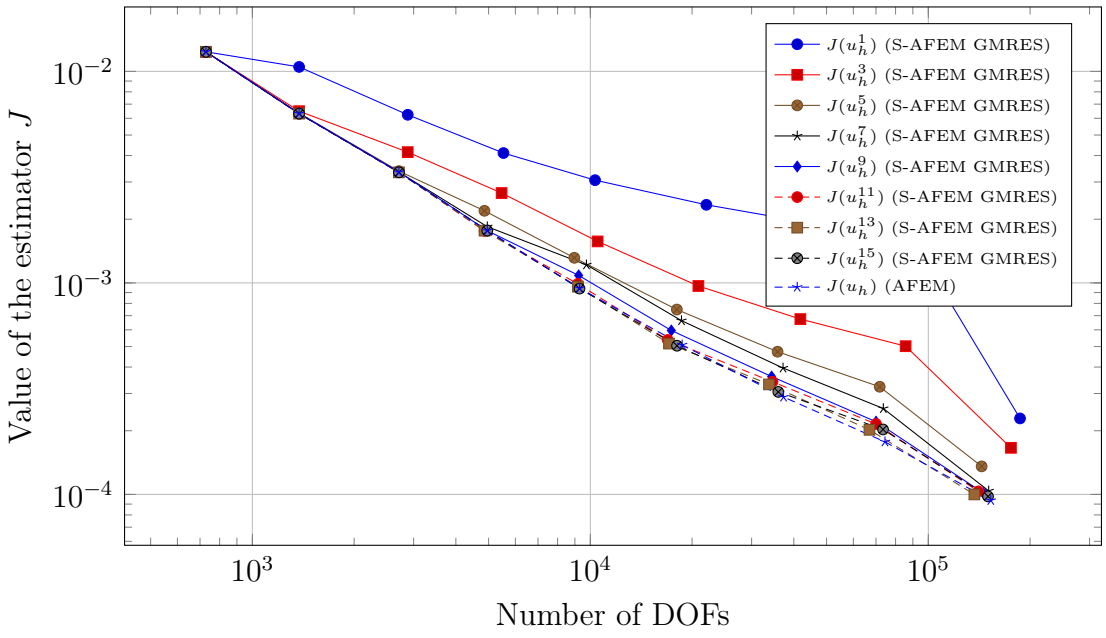


Figure 5.117: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

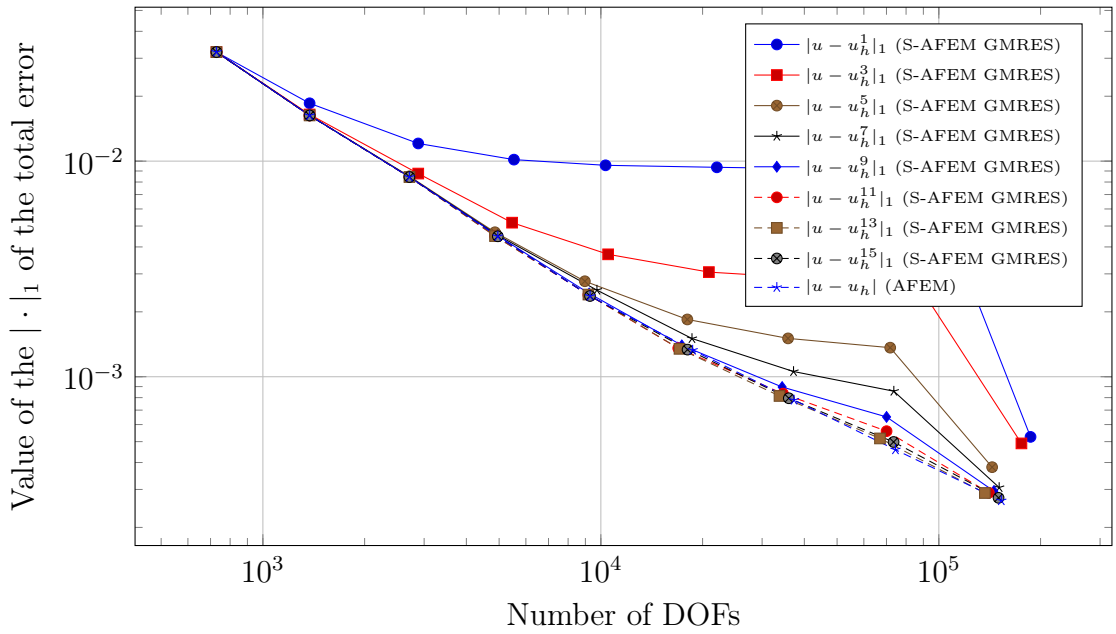


Figure 5.118: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 2$, when we apply 9 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

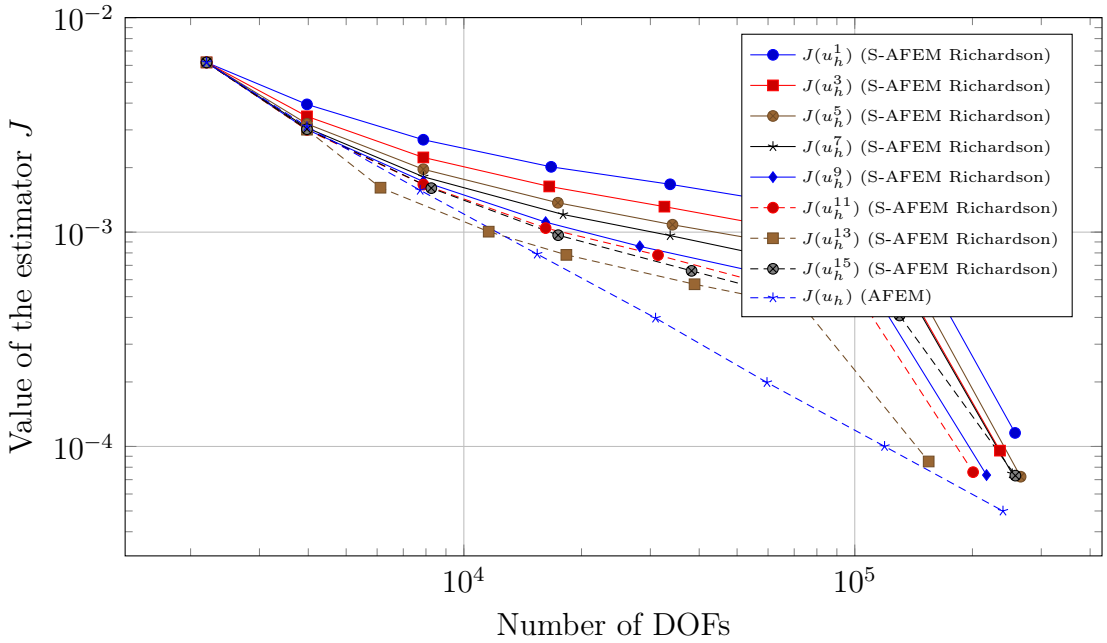


Figure 5.119: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

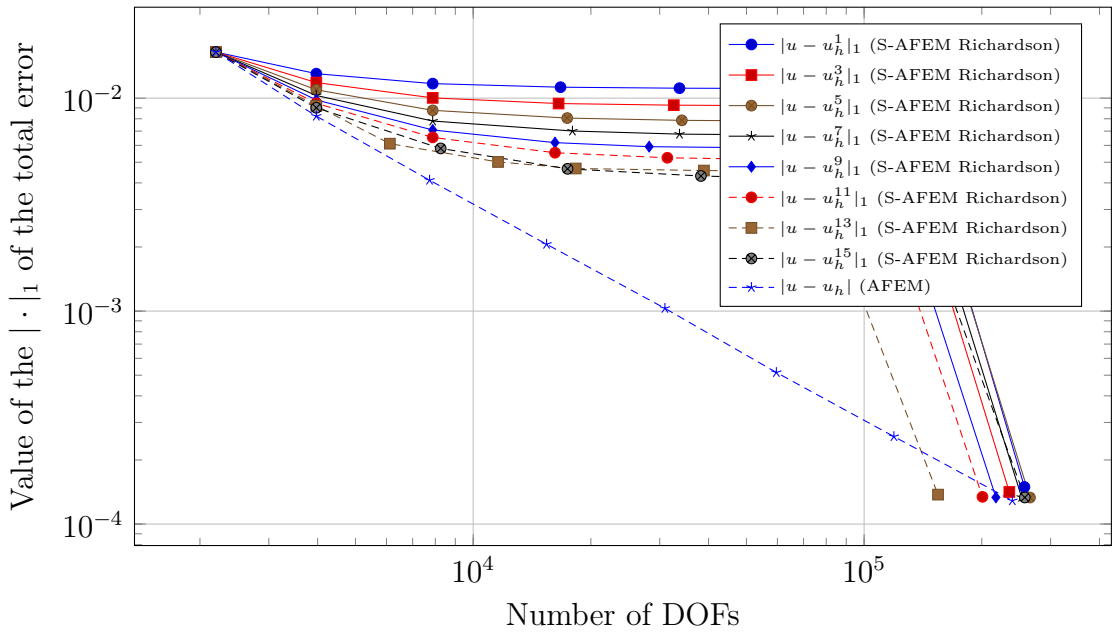


Figure 5.120: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

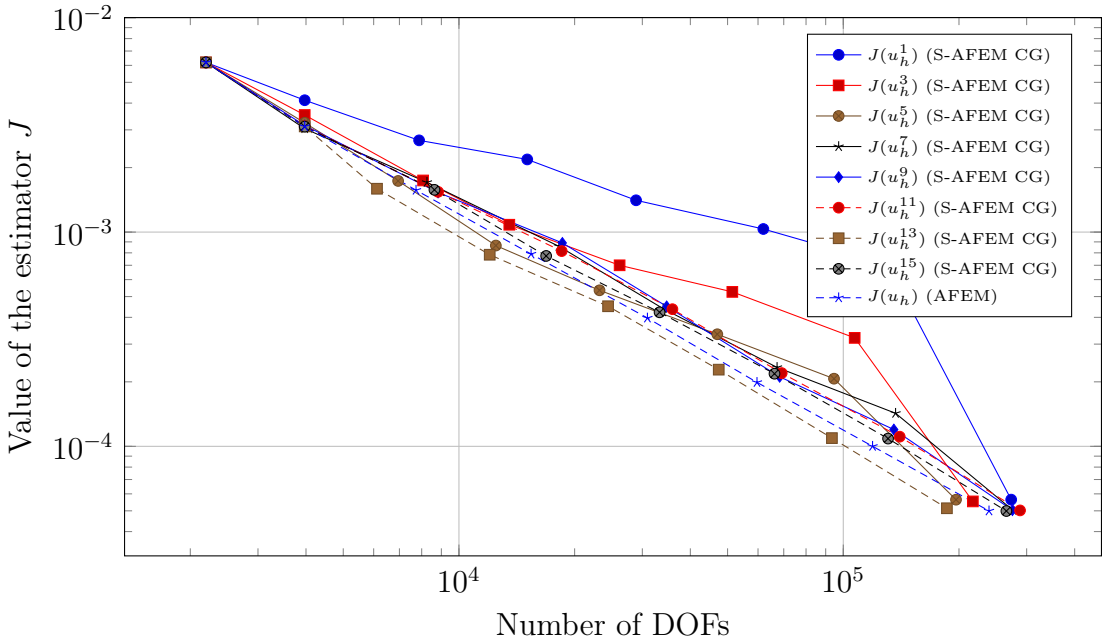


Figure 5.121: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

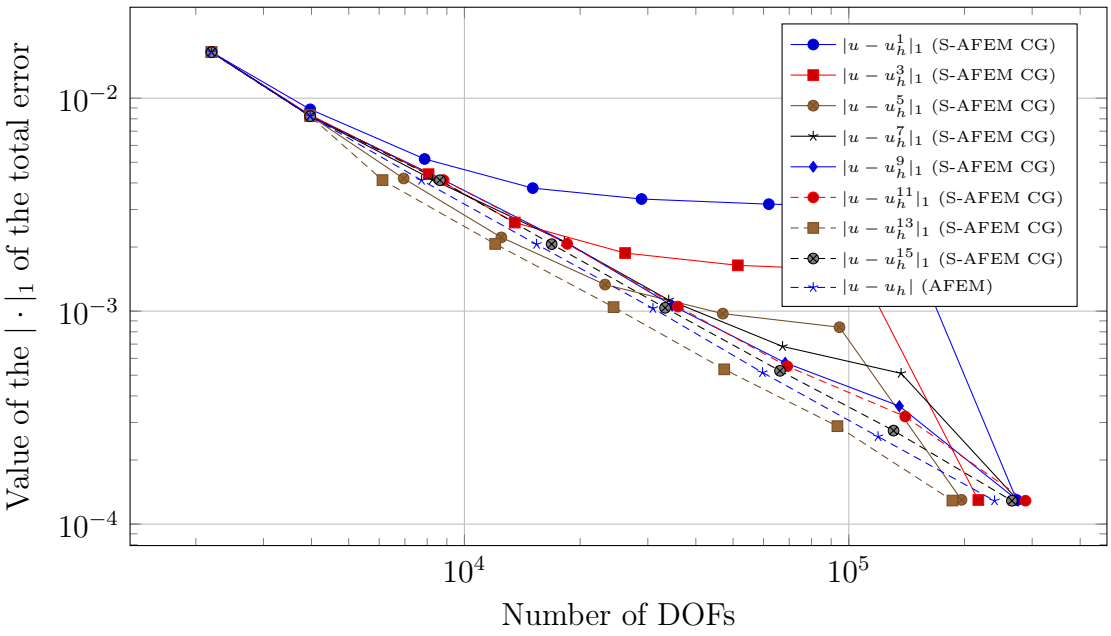


Figure 5.122: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

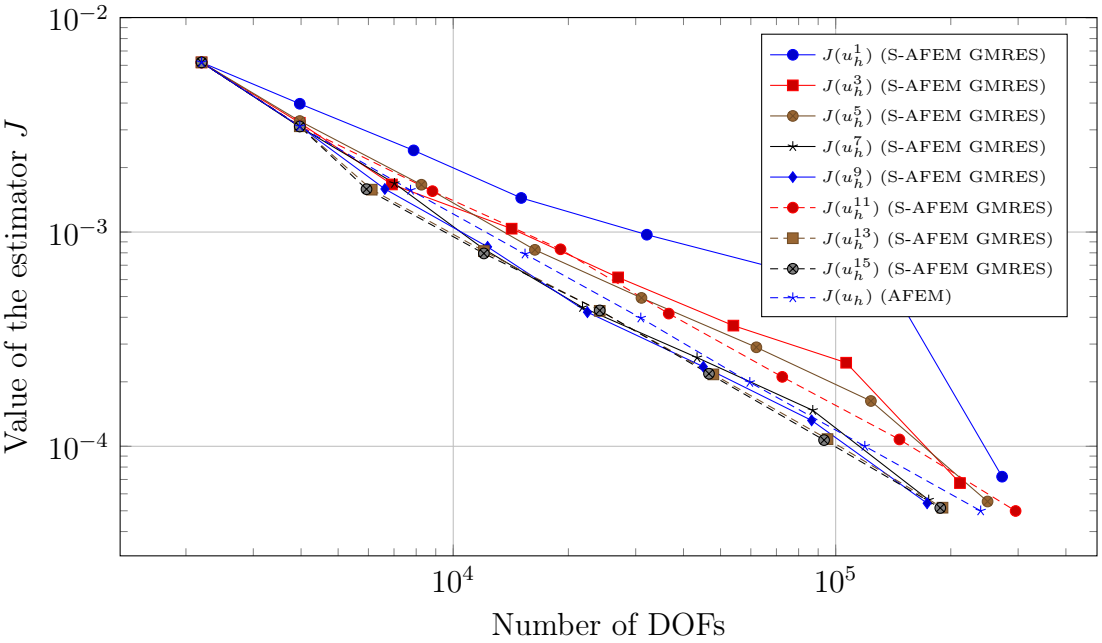


Figure 5.123: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

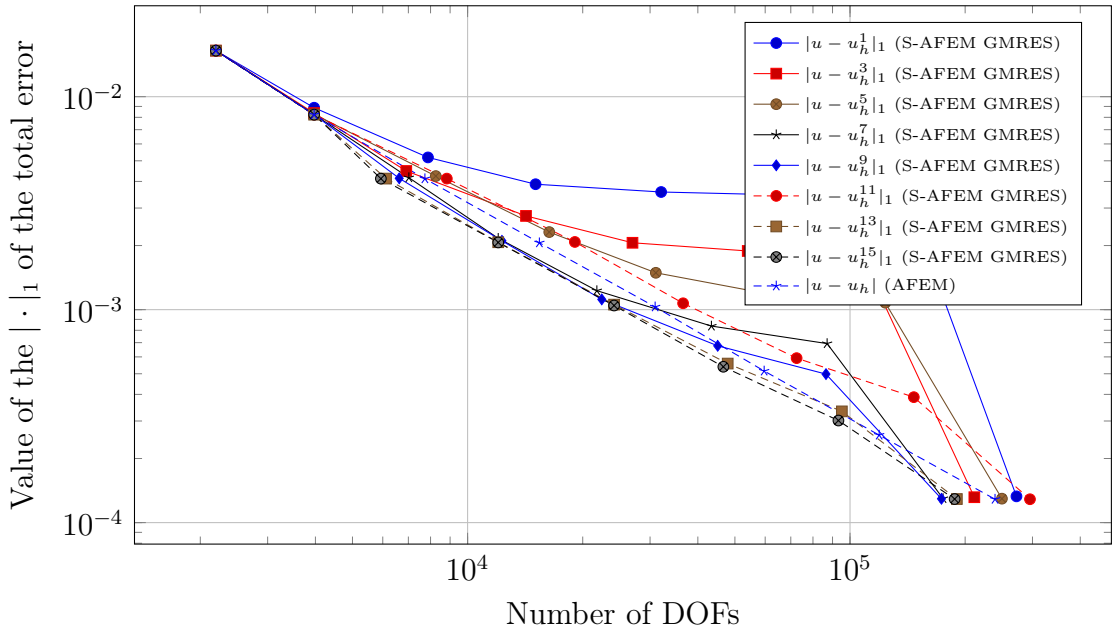


Figure 5.124: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 3$, when we apply 8 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

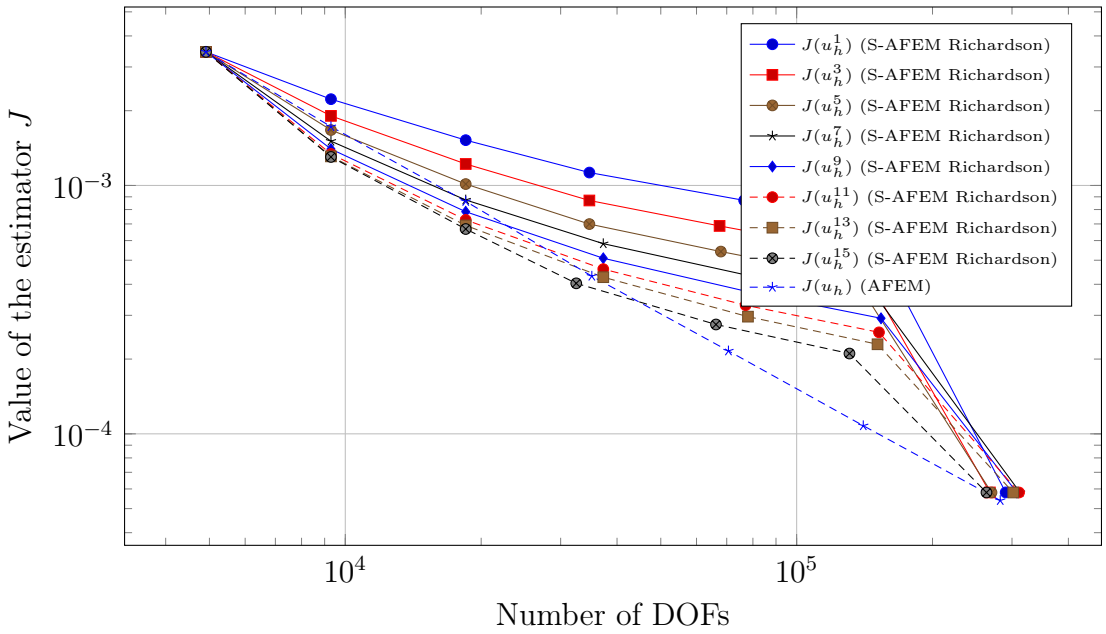


Figure 5.125: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

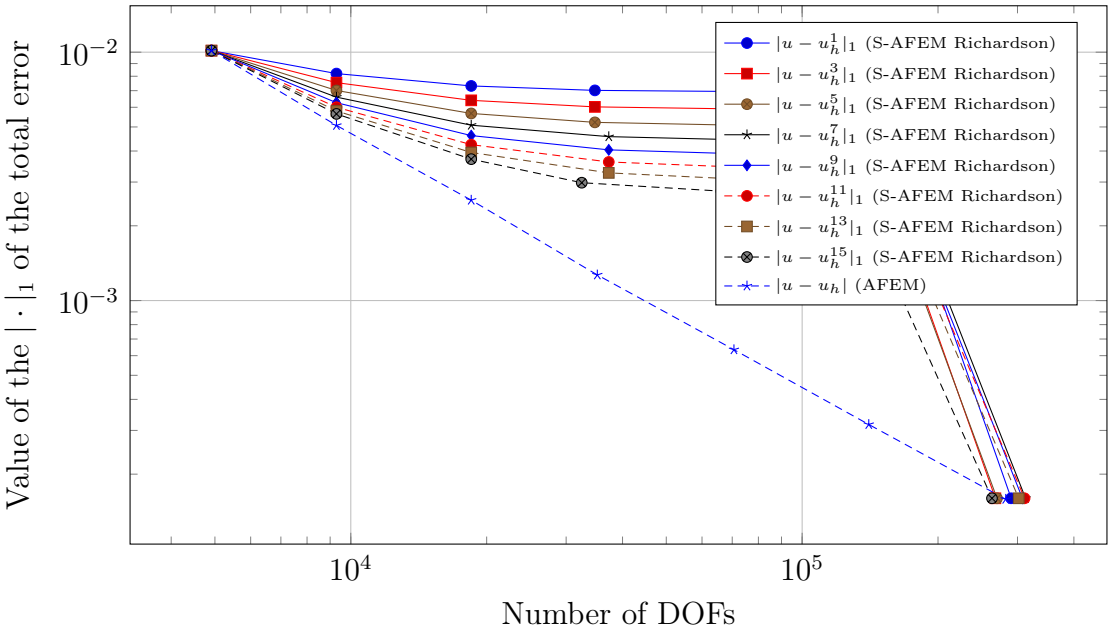


Figure 5.126: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

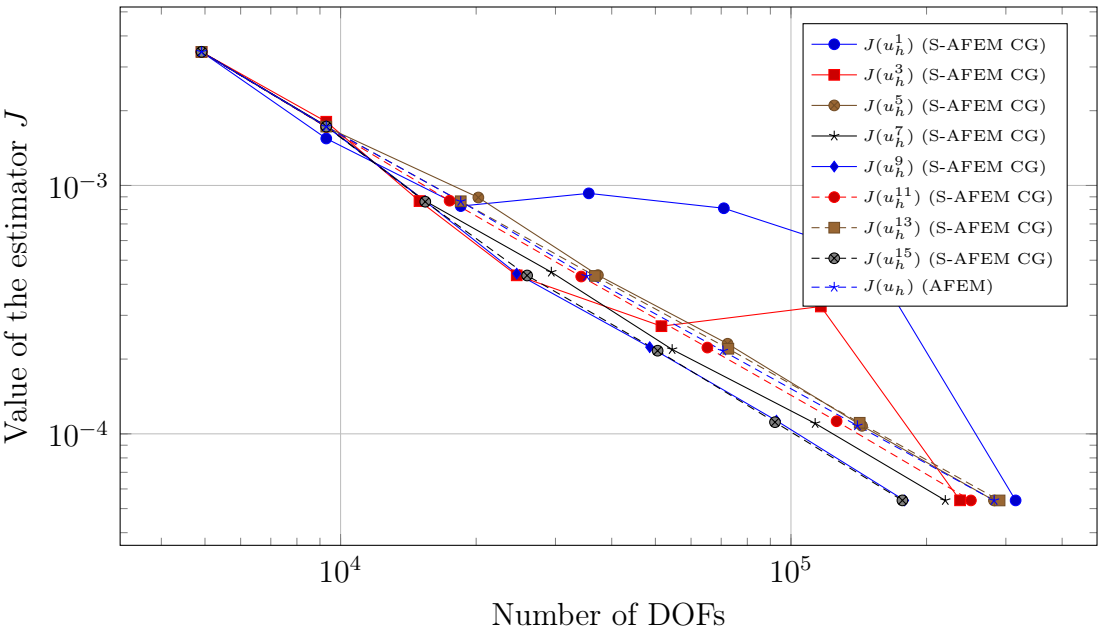


Figure 5.127: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

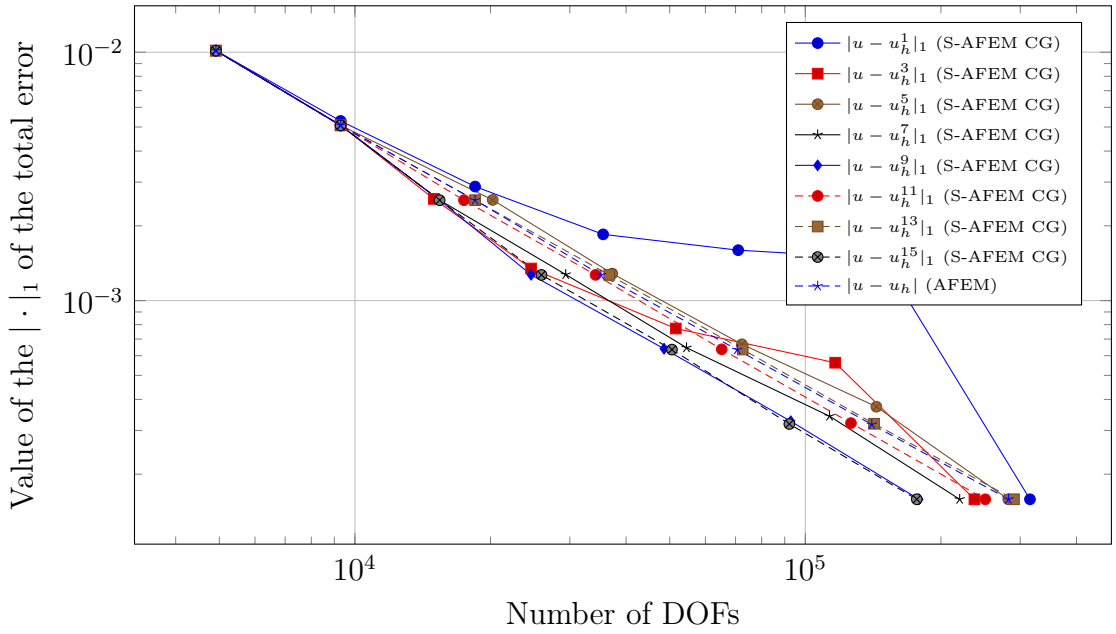


Figure 5.128: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

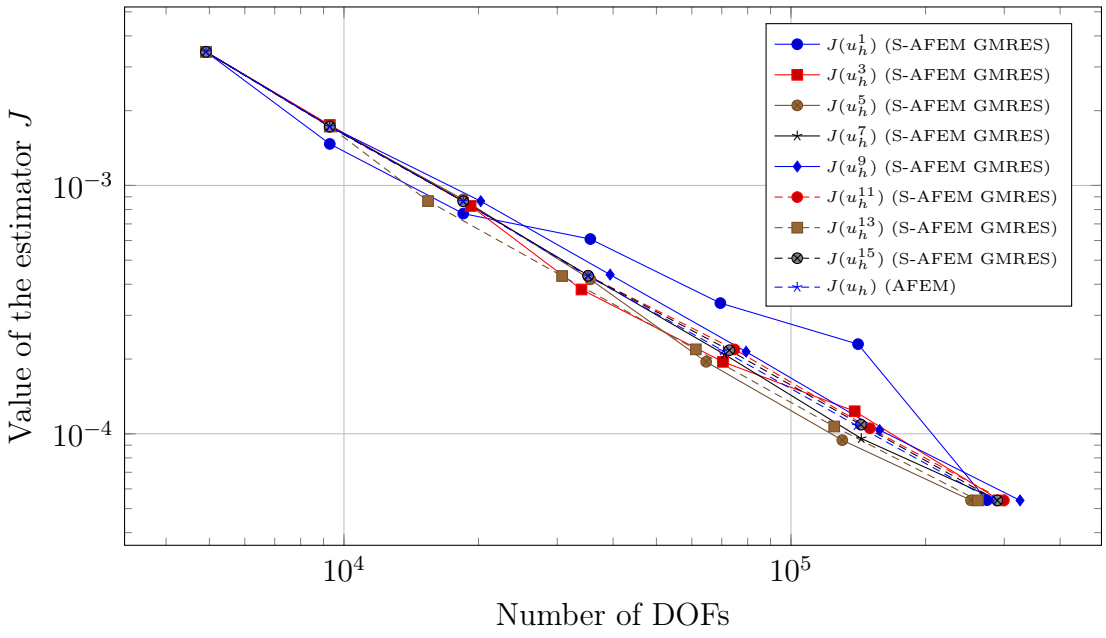


Figure 5.129: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

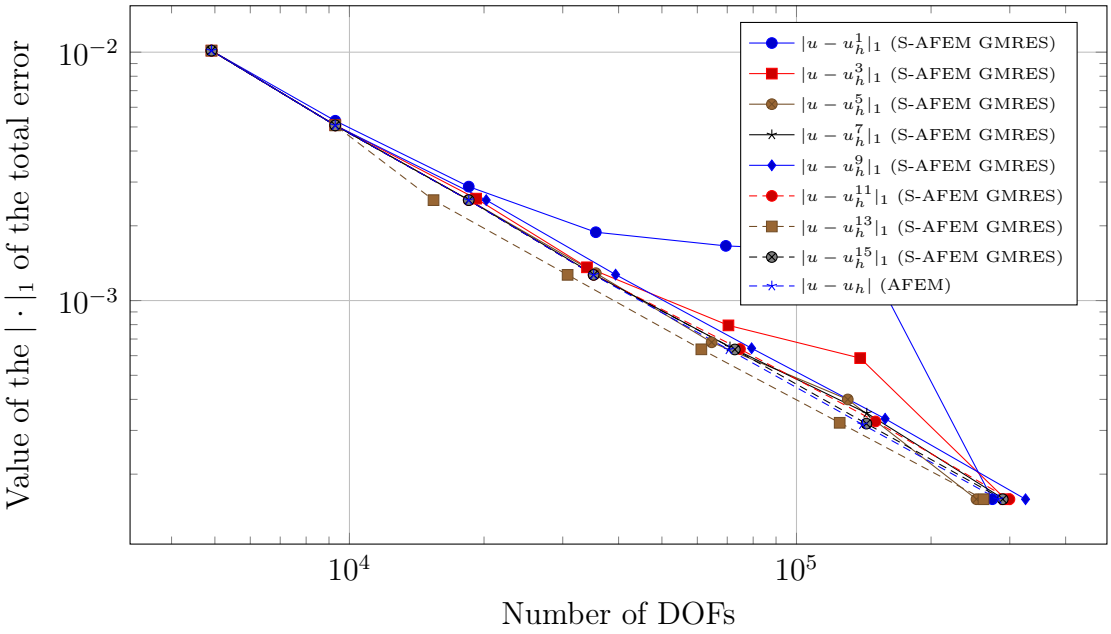


Figure 5.130: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 4$, when we apply 7 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

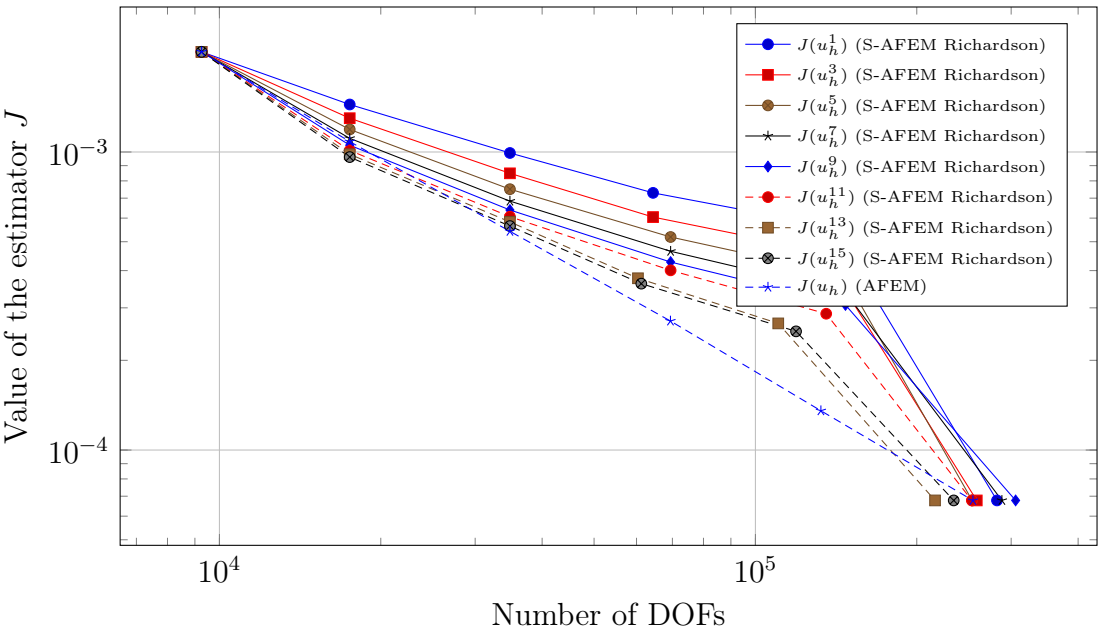


Figure 5.131: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

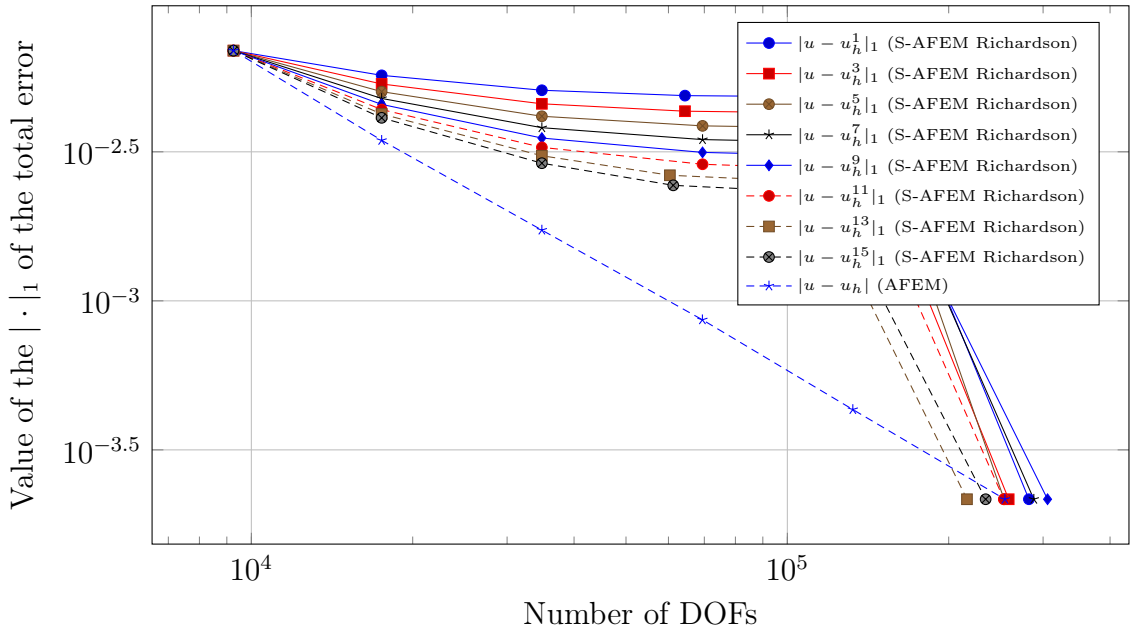


Figure 5.132: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with Richardson as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

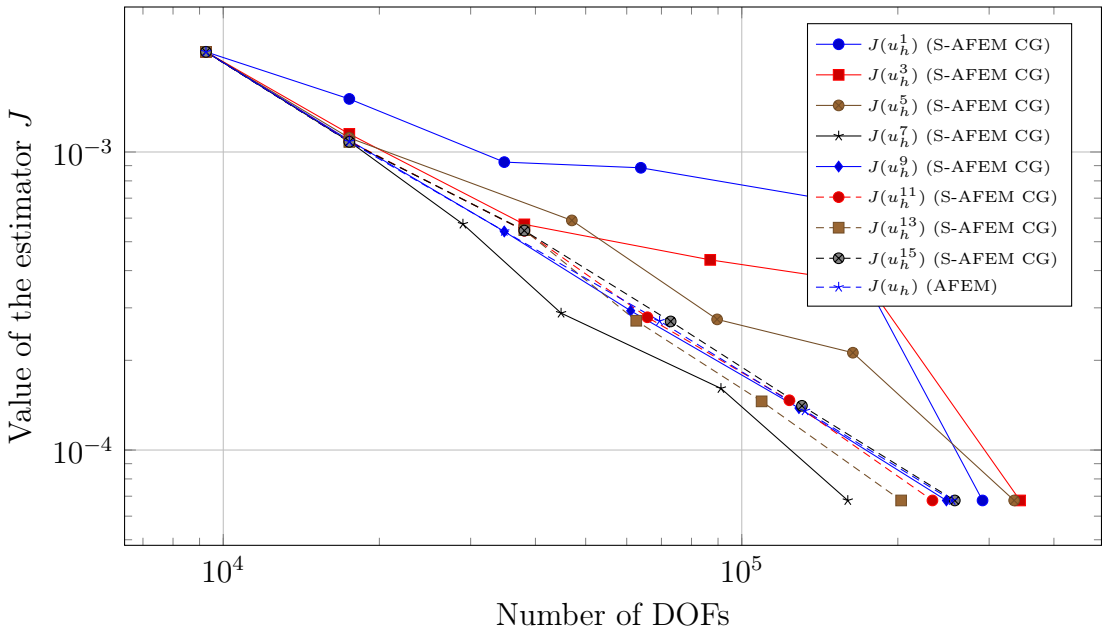


Figure 5.133: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

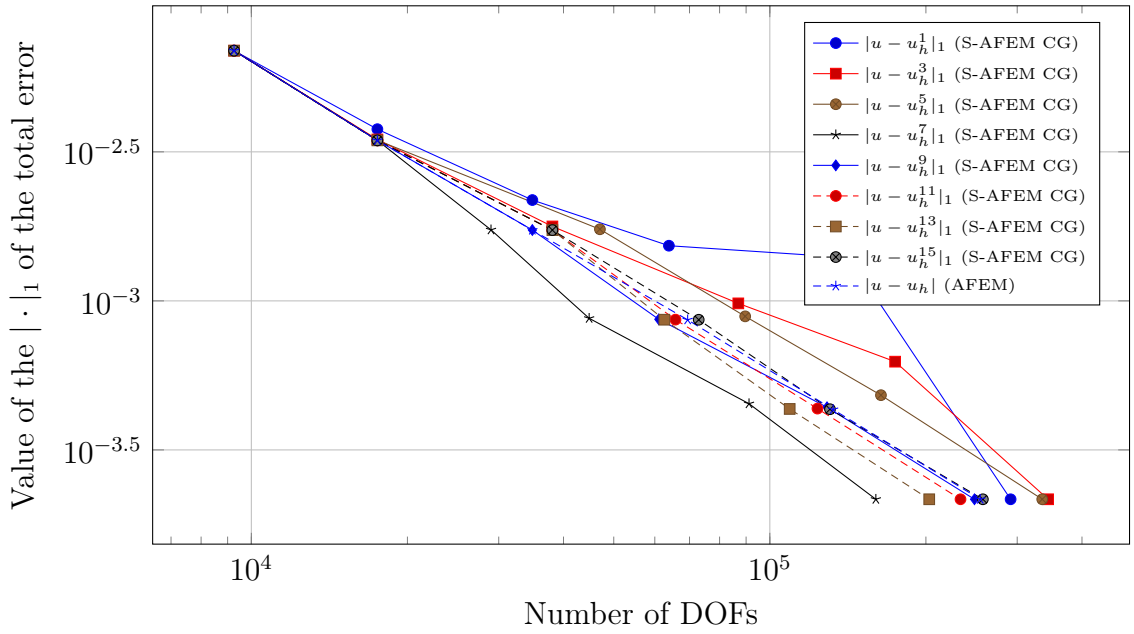


Figure 5.134: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with CG as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

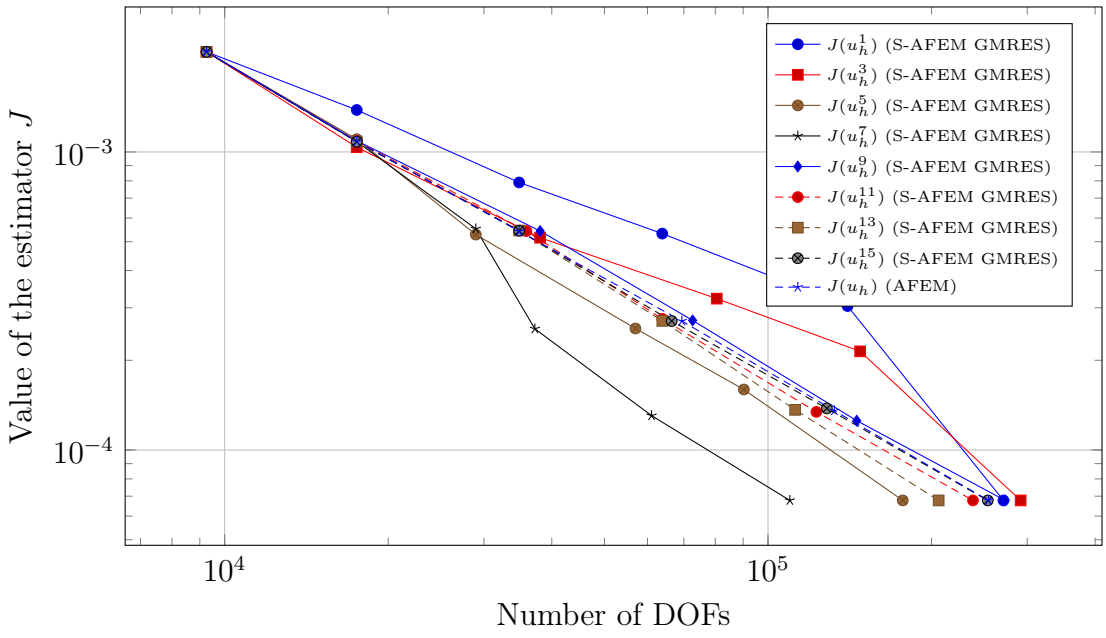


Figure 5.135: Value of the error estimator J for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

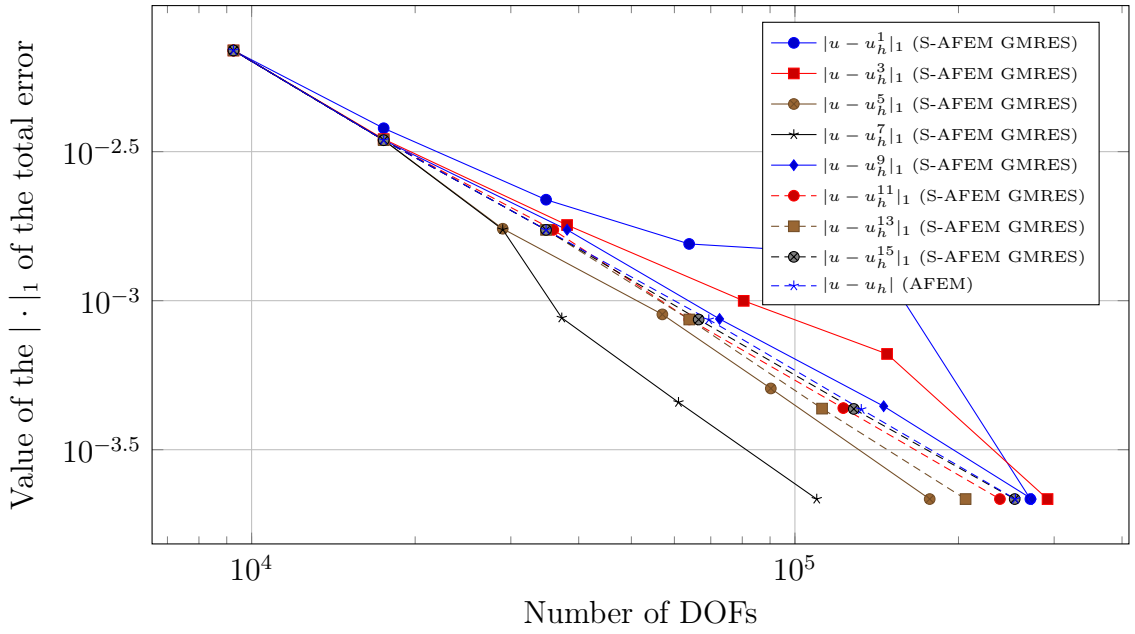


Figure 5.136: Value of the $|\cdot|_1$ error for the peak problem in 3D, for FEM discretisation degree $deg = 5$, when we apply 6 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count $l = 2i - 1$ goes from $i = 1$ to 8. The initial global refinement is 2 and we select a fraction of 0.15 of all cells for refinement at each cycle

Since the system associated to Problem (5.6) is not symmetric, the only possibility for both a robust solver and for the smoother is given by the GMRES method (cf. eg. [67]). We apply classic AFEM and S-AFEM for both bi-linear piecewise and higher order finite element discretisations for $deg = 1, \ldots, 5$. We plot the value of the estimator J and the value of the $|\cdot|_1$ of the total error, as the GMRES smoothing iteration count l goes from 1 to 5. For all the cases where the transport term $\boldsymbol{\beta} = (1, 1)\hat{}T$ (corresponding to small transport), the behaviour of the estimator for S-AFEM ($\forall l = 1, \ldots, 5$) is exactly the same as the one given by the classic AFEM, for the case $deg = 1$, as shown in Figure 5.137, while it approaches it as the GMRES smoothing iteration count increases from 1 to 5 for higher order FEM discretisations (i.e. for $deg \geq 2$), as shown in Figures 5.139, 5.139, 5.141, 5.143 and 5.145. However, in all cases, the accuracy of the final approximations is almost the same, as evidenced in Figures 5.138, 5.140, 5.142, 5.144 and 5.146. 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 behaviour of the estimator for S-AFEM ($\forall l = 1, \ldots, 5$) is exactly the same as the one given by the classic AFEM, for the case $deg = 1$, as shown in Figures 5.147 and 5.157, while it approaches it as the GMRES smoothing iteration count increases from 1 to 5 for higher order FEM discretisations, as evidenced in Figures 5.149, 5.149, 5.151, 5.153, 5.165, 5.159, 5.159, 5.161, 5.163 and 5.165. In all cases, however, the accuracy of the final approximations is almost the same, as evidenced in Figures 5.148,

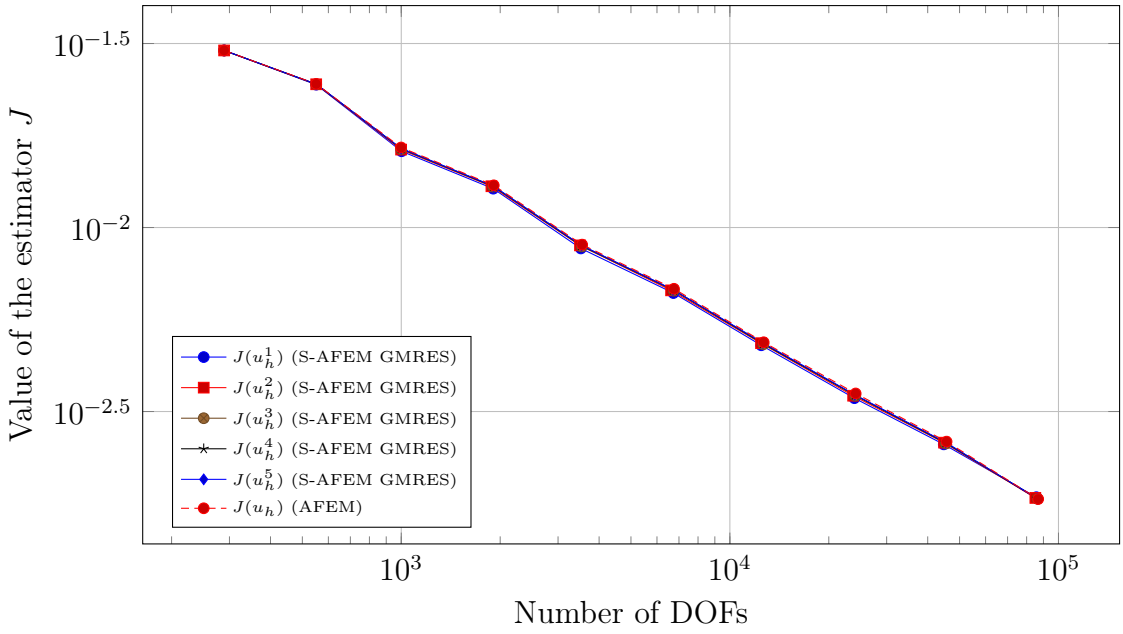


Figure 5.137: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

5.150, 5.152, 5.154, 5.156, 5.158, 5.160, 5.162, 5.164 and 5.166,showing that S-AFEM turns out to be a good method also for more complicated and non-symmetric problems.

5.6 Conclusions

In this work we propose and analyse 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. One of the key discoveries of this research program, that was our motivational starting point, has been the fact 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 solution, 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 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 at applying the $\mathcal{E}stimate$-$\mathcal{M}ark$-$\mathcal{R}efine$ steps at the rough approximations obtained at the intermediate levels. More importantly, we show that the accuracy of the final approximation obtained through S-AFEM is almost the same to the one that would be obtained by classical AFEM, at a

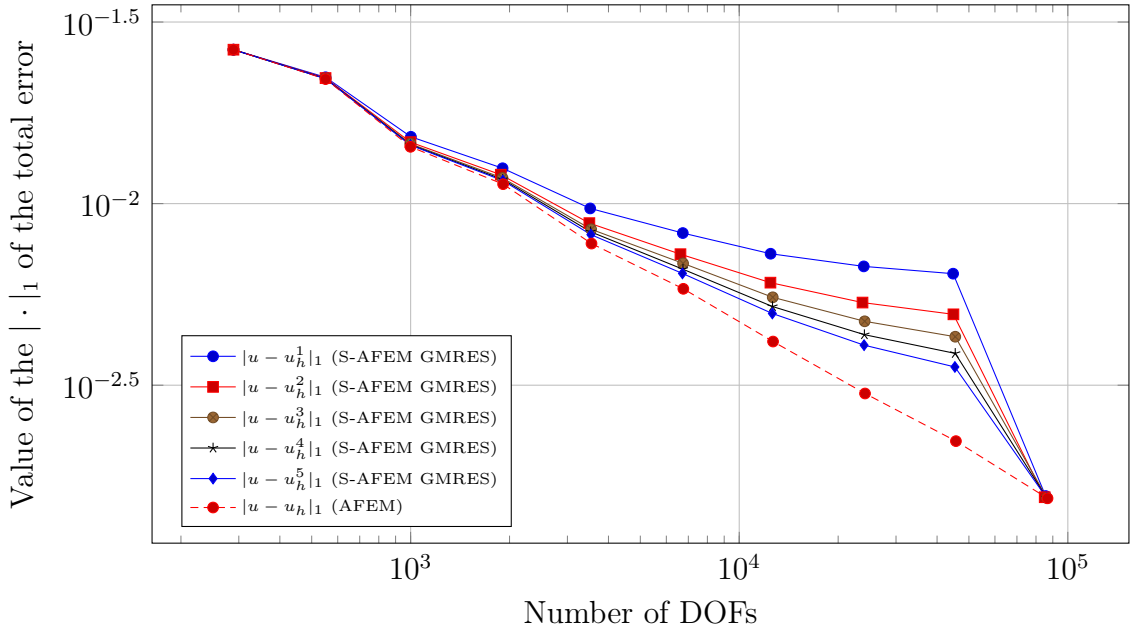


Figure 5.138: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

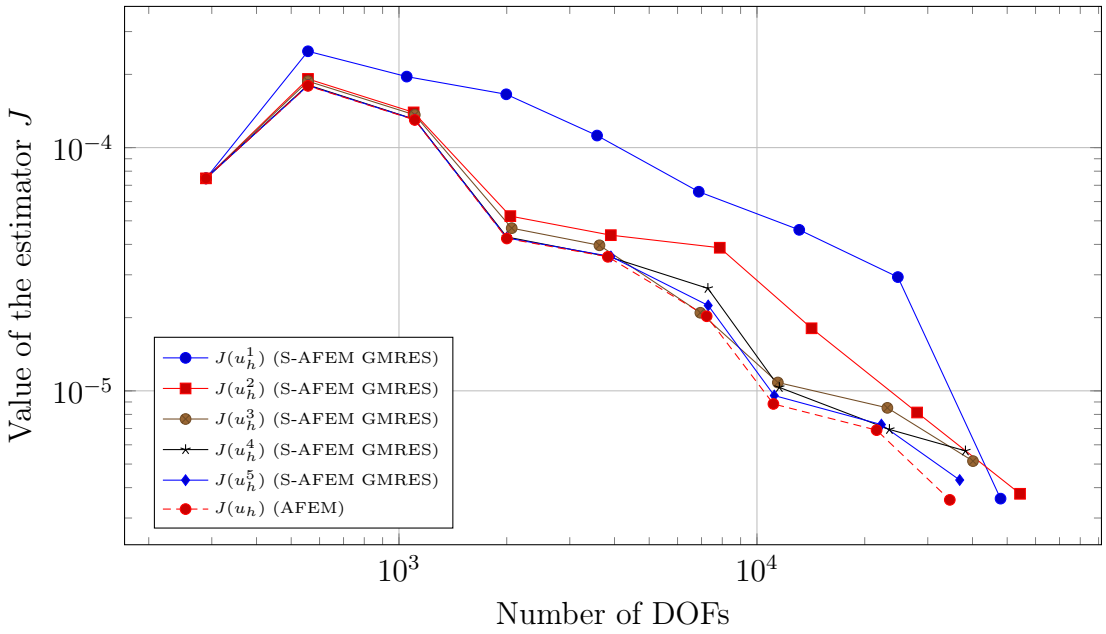


Figure 5.139: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

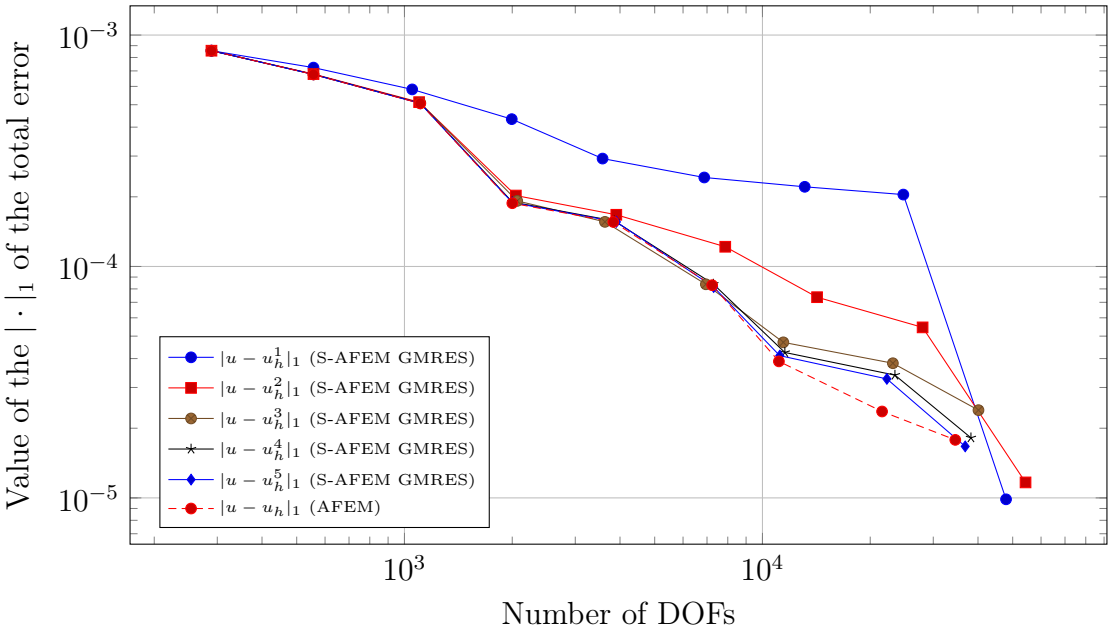


Figure 5.140: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

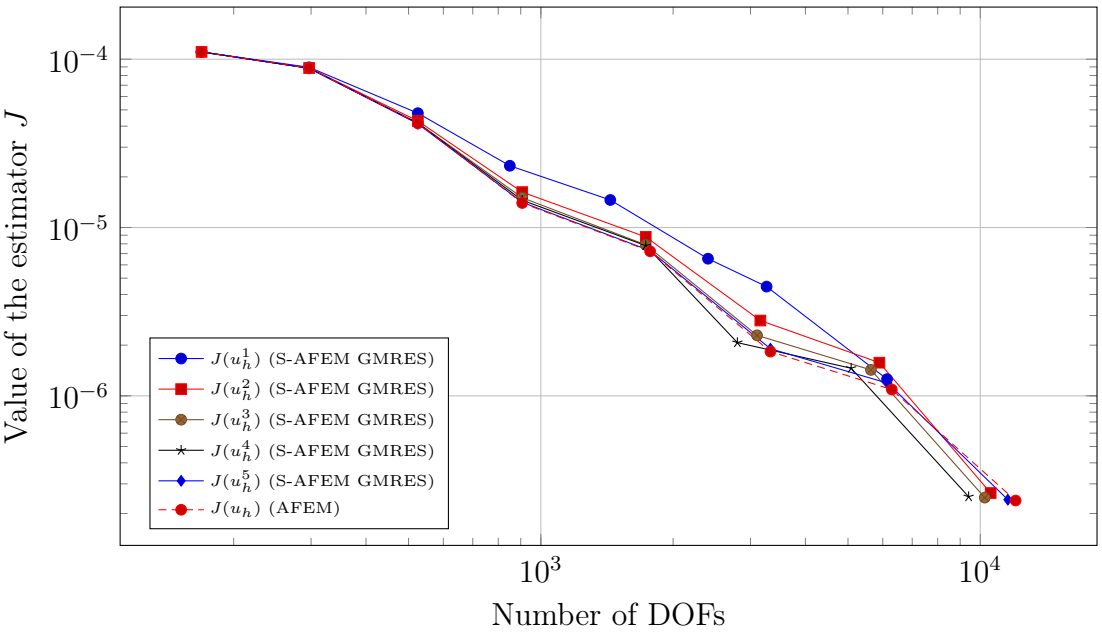


Figure 5.141: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

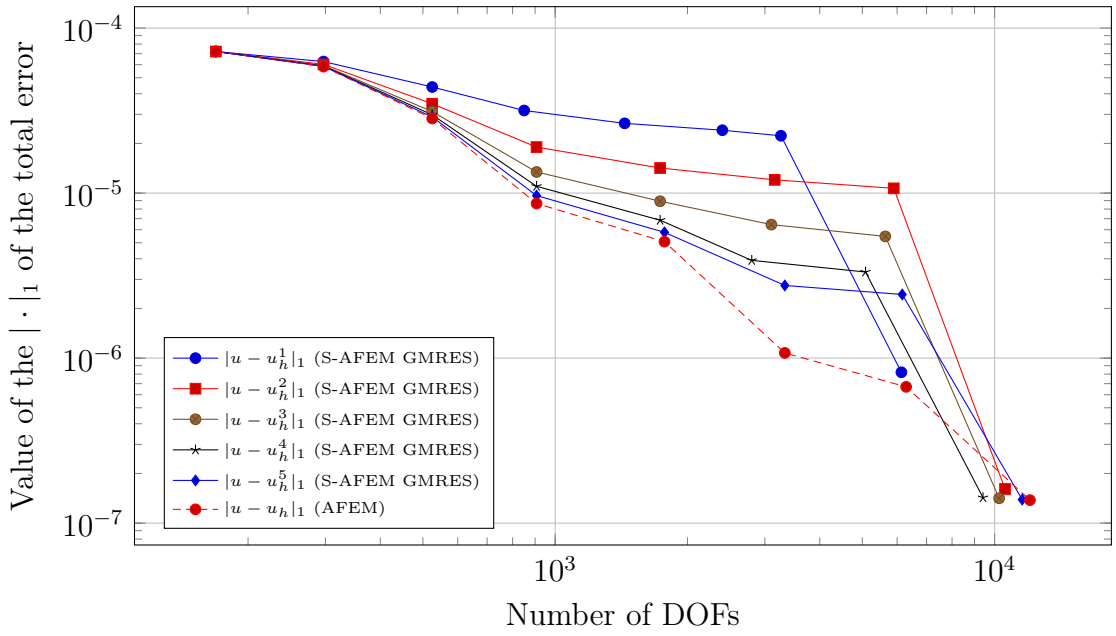


Figure 5.142: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

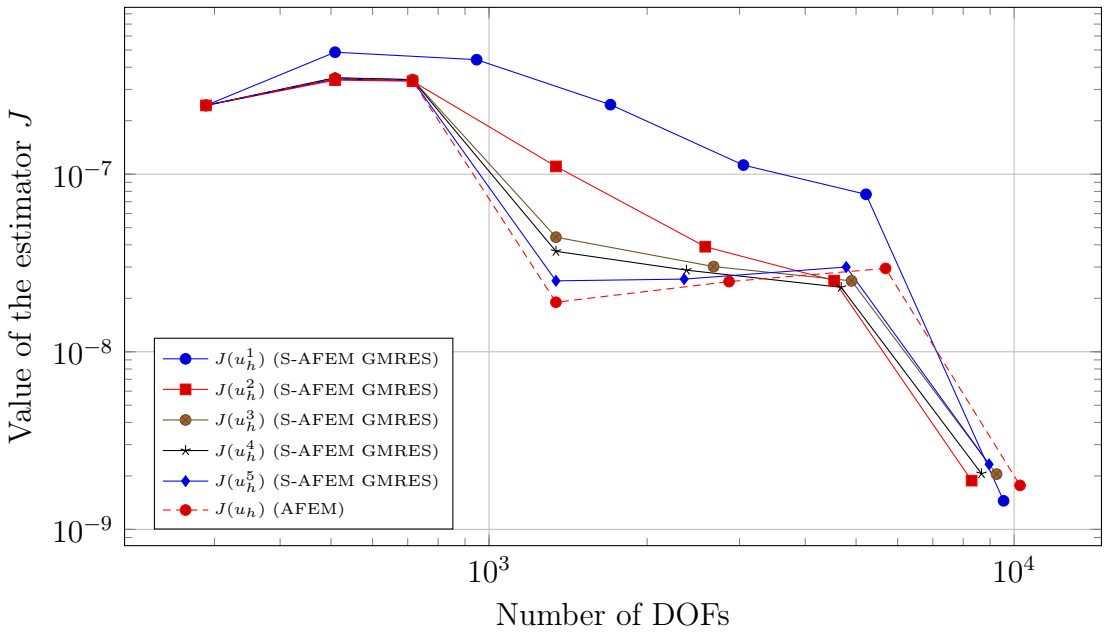


Figure 5.143: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

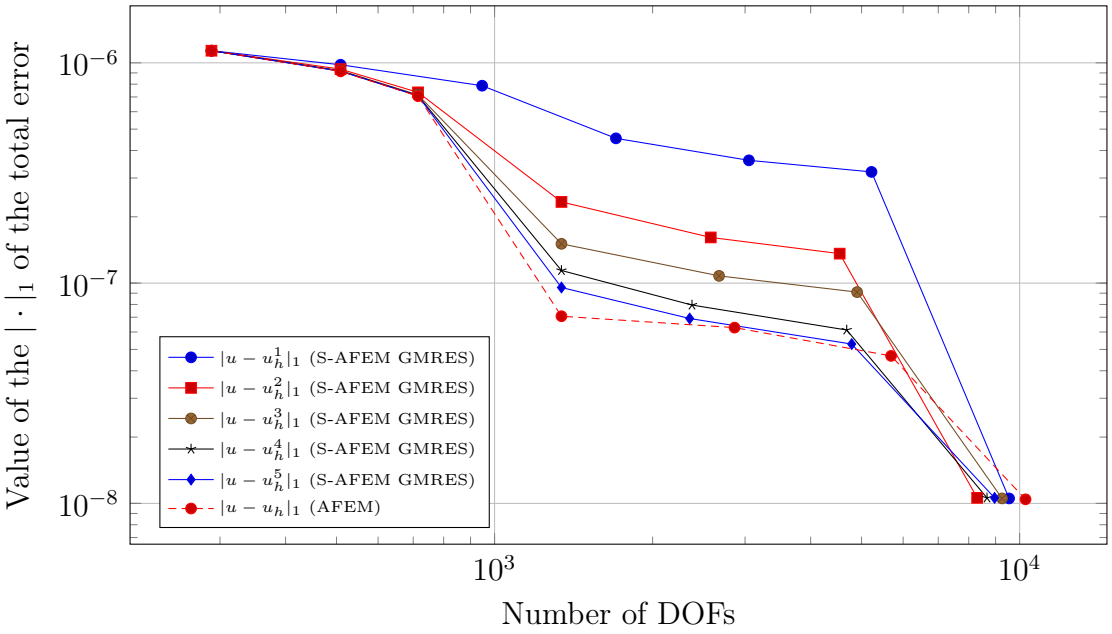


Figure 5.144: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

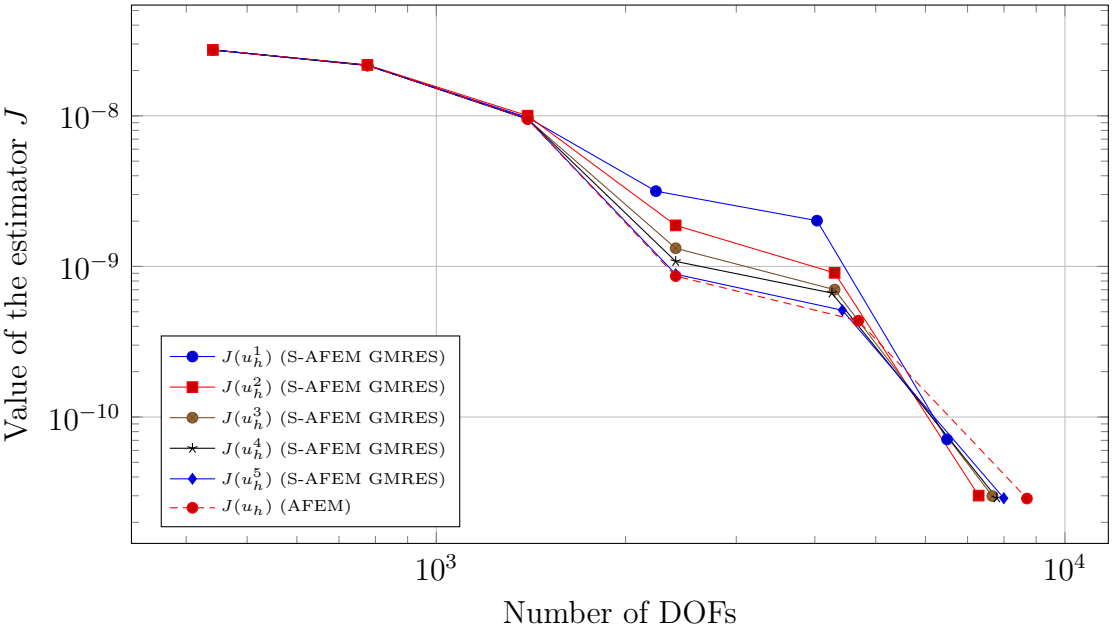


Figure 5.145: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

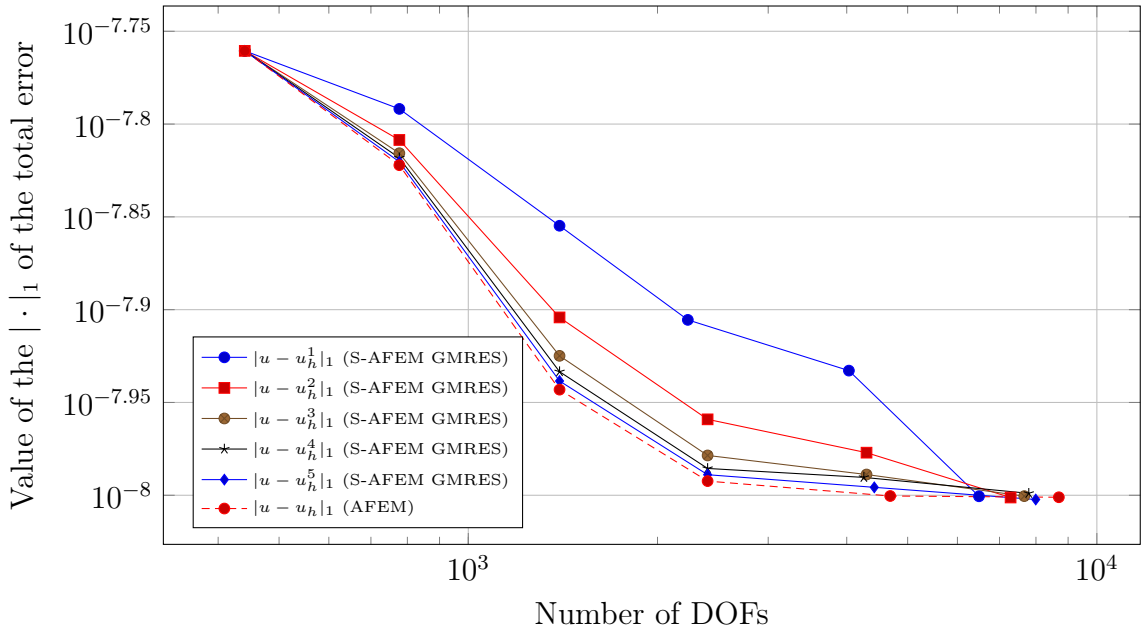


Figure 5.146: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (1, 1)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

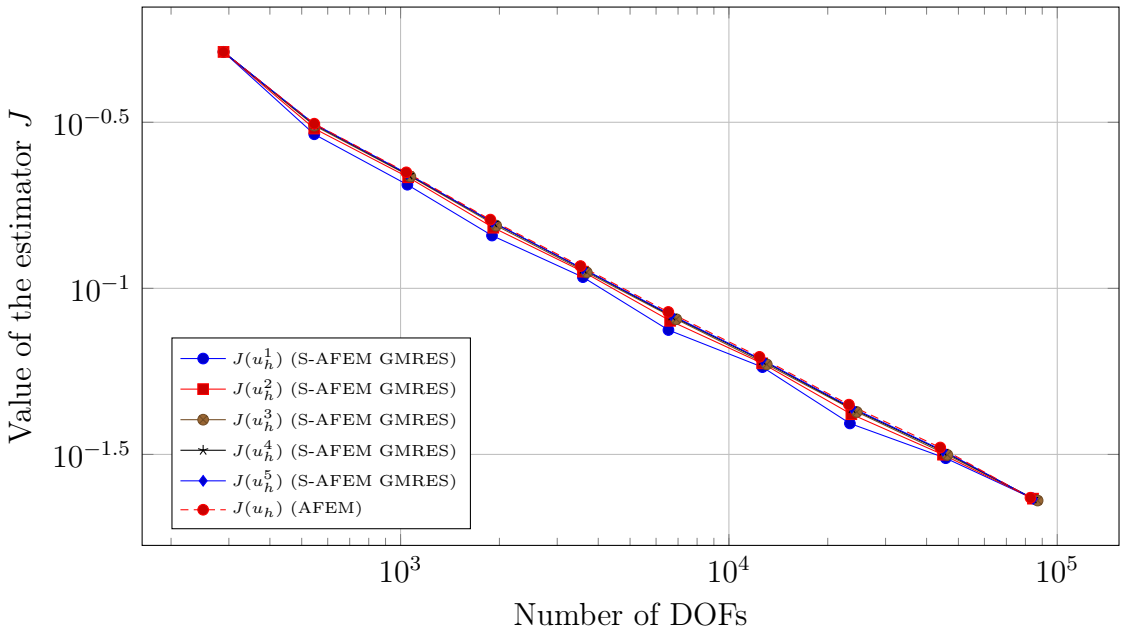


Figure 5.147: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

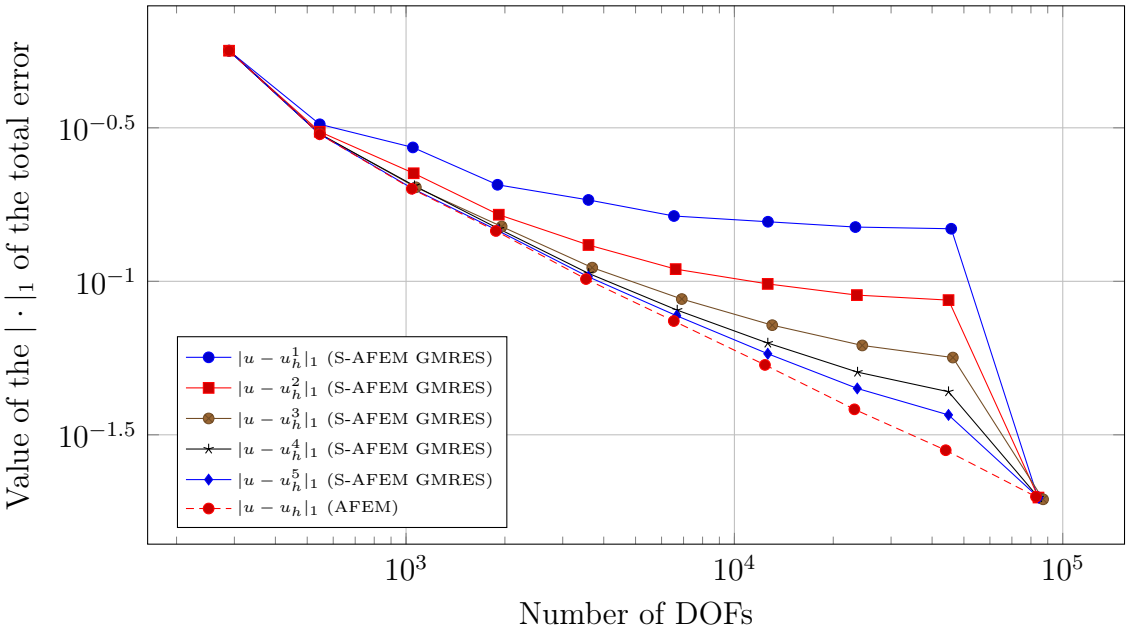


Figure 5.148: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

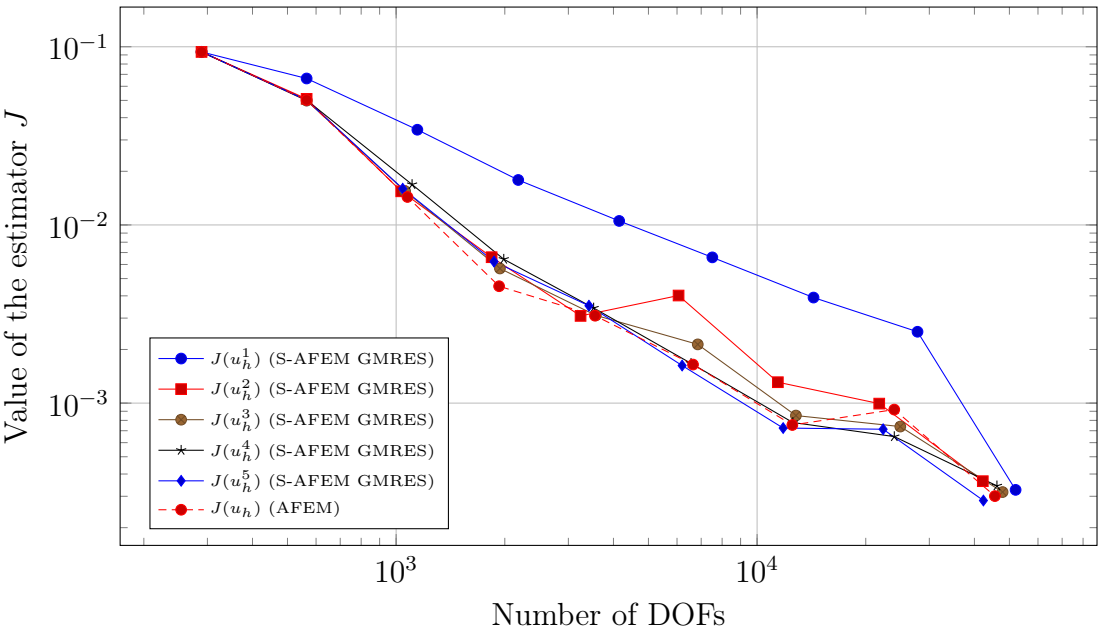


Figure 5.149: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

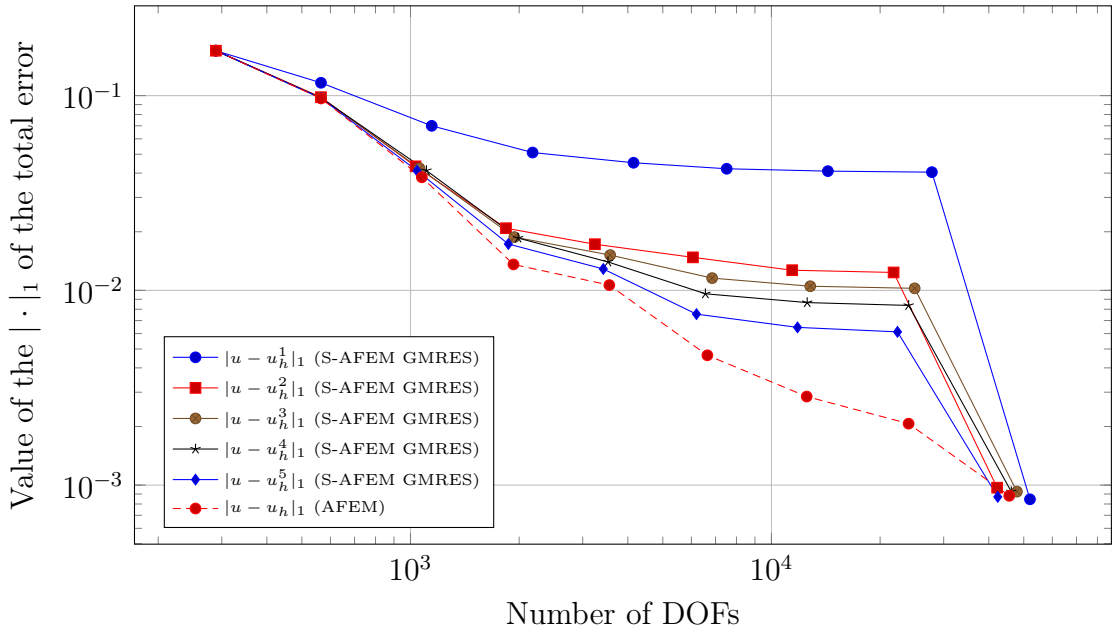


Figure 5.150: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

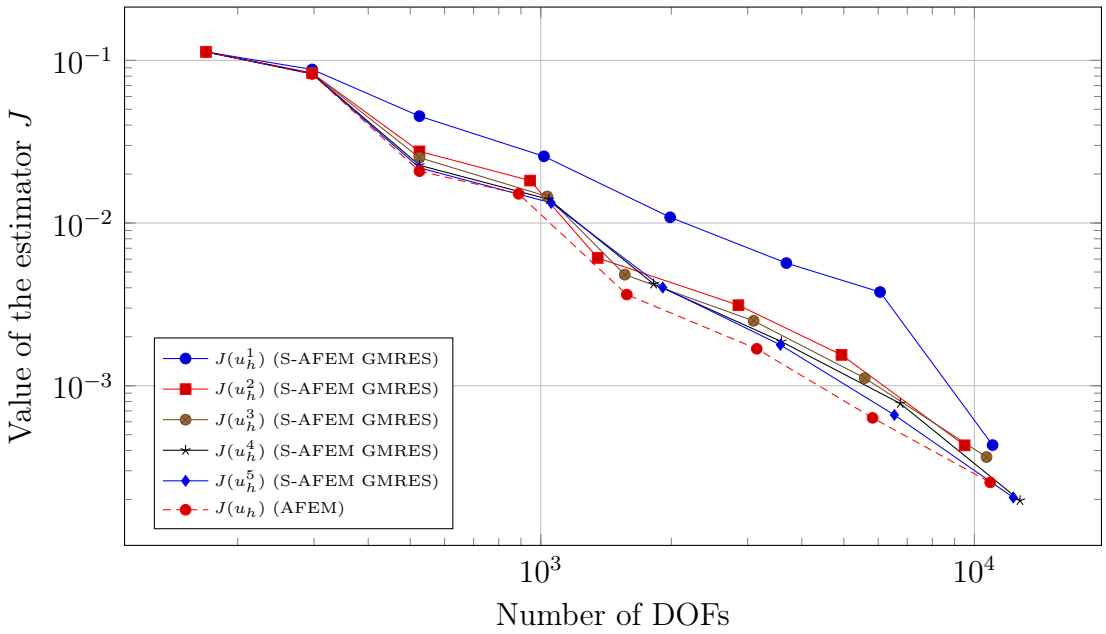


Figure 5.151: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

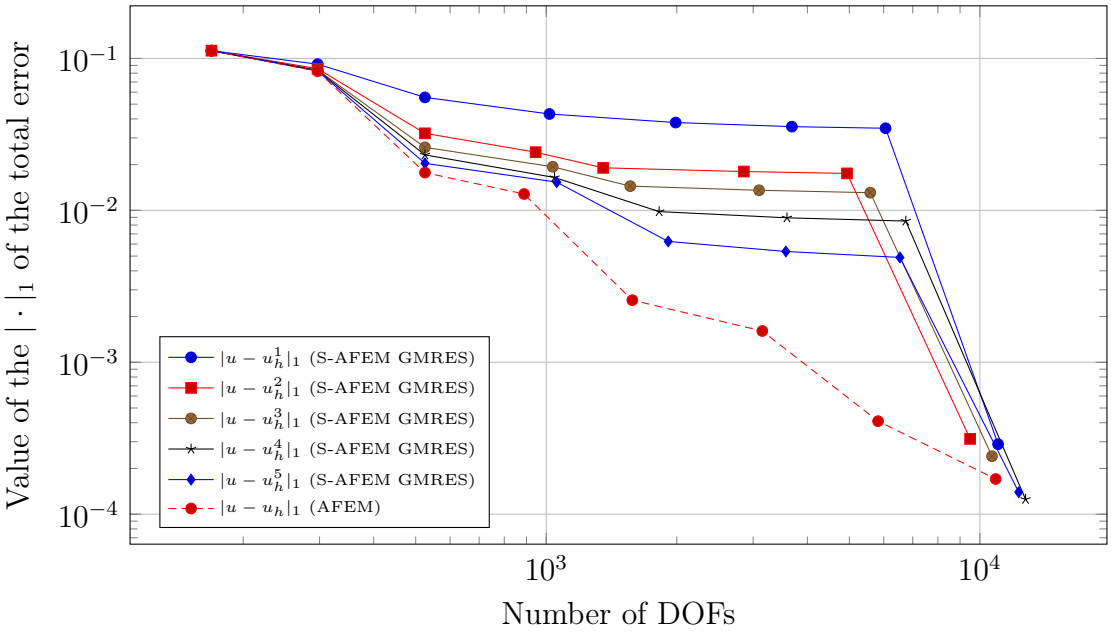


Figure 5.152: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

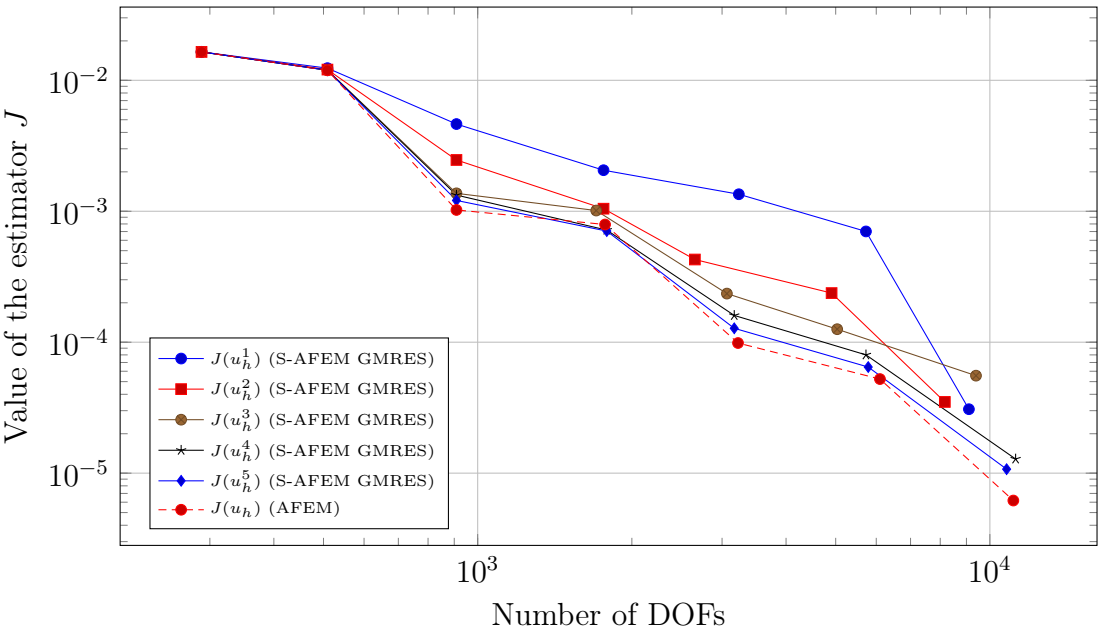


Figure 5.153: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

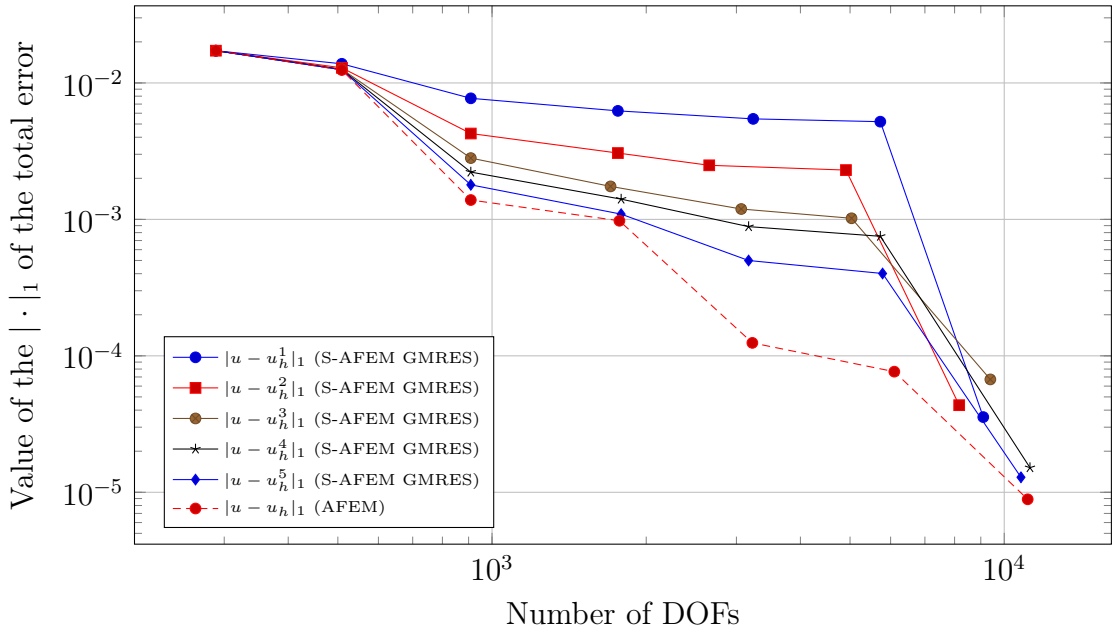


Figure 5.154: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

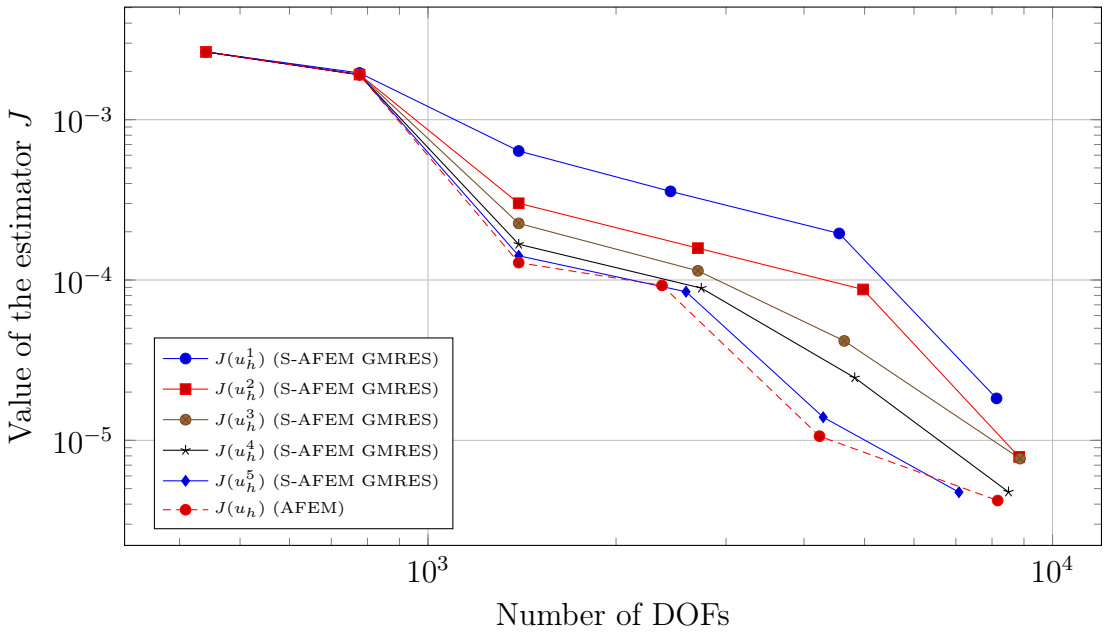


Figure 5.155: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

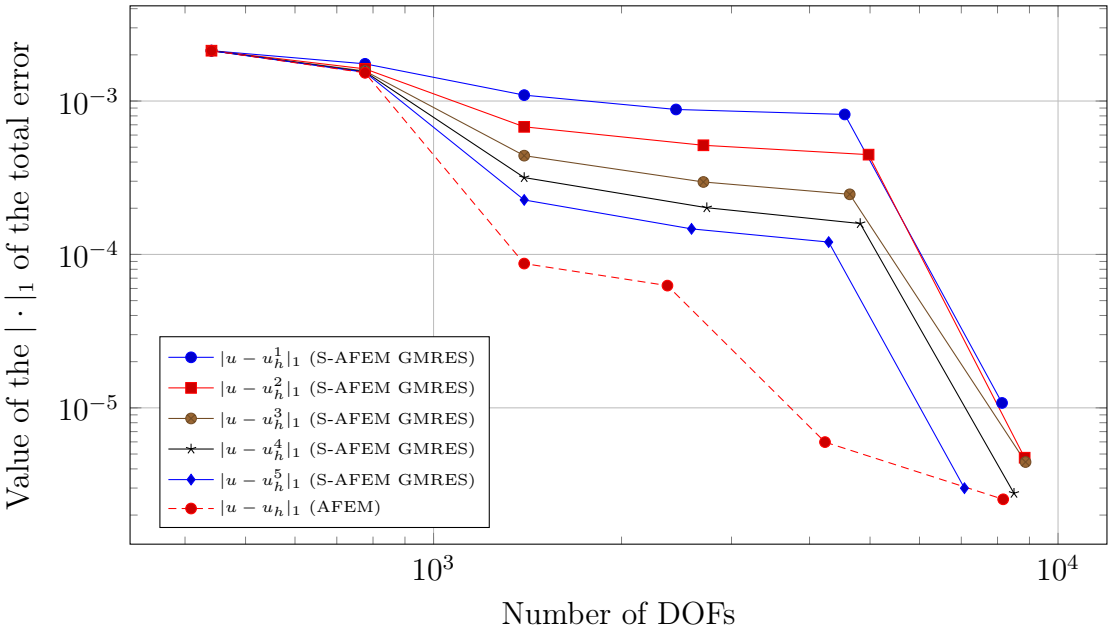


Figure 5.156: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (10, 10)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

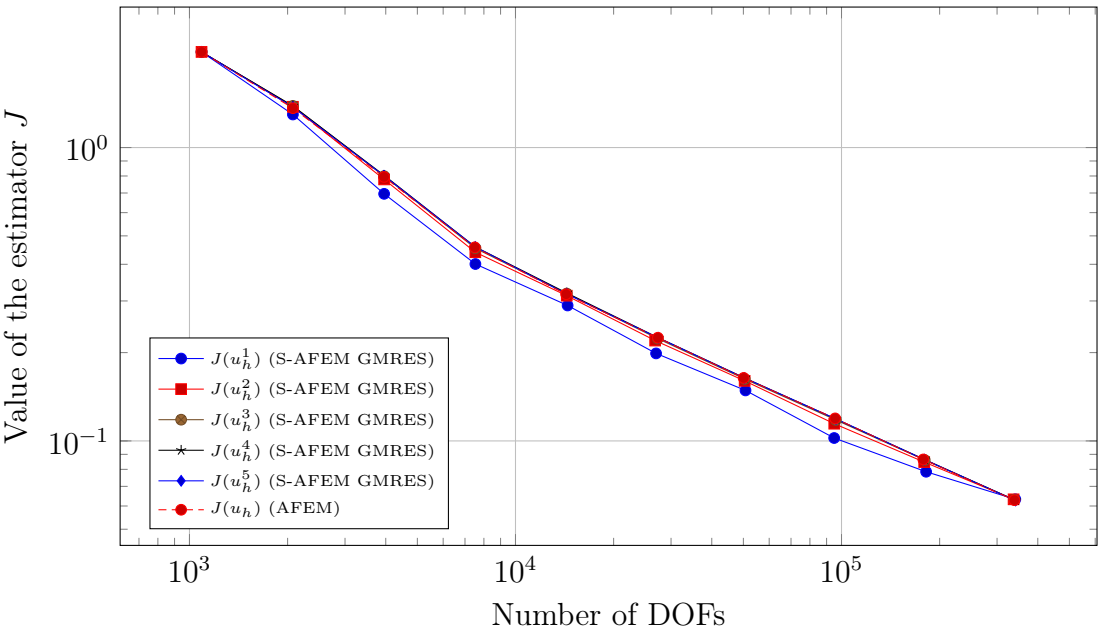


Figure 5.157: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

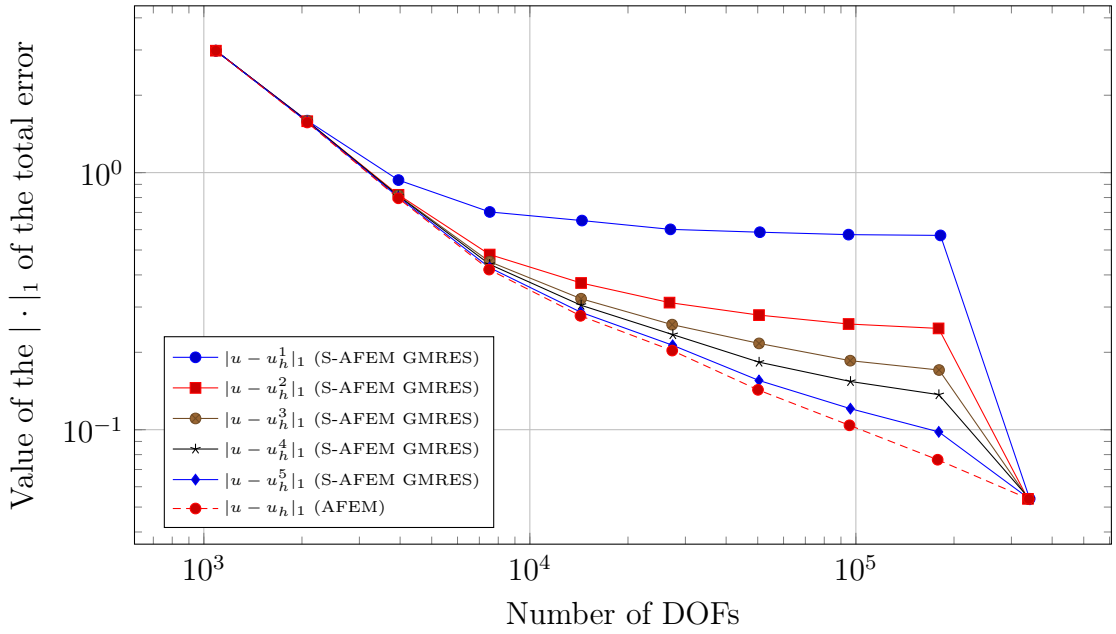


Figure 5.158: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 1$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

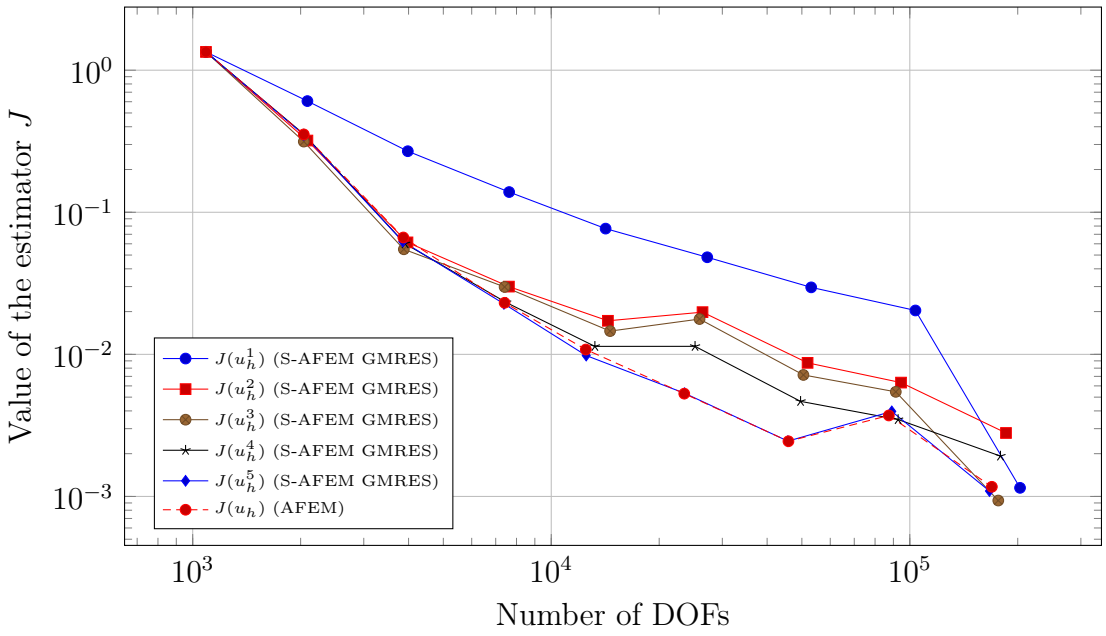


Figure 5.159: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

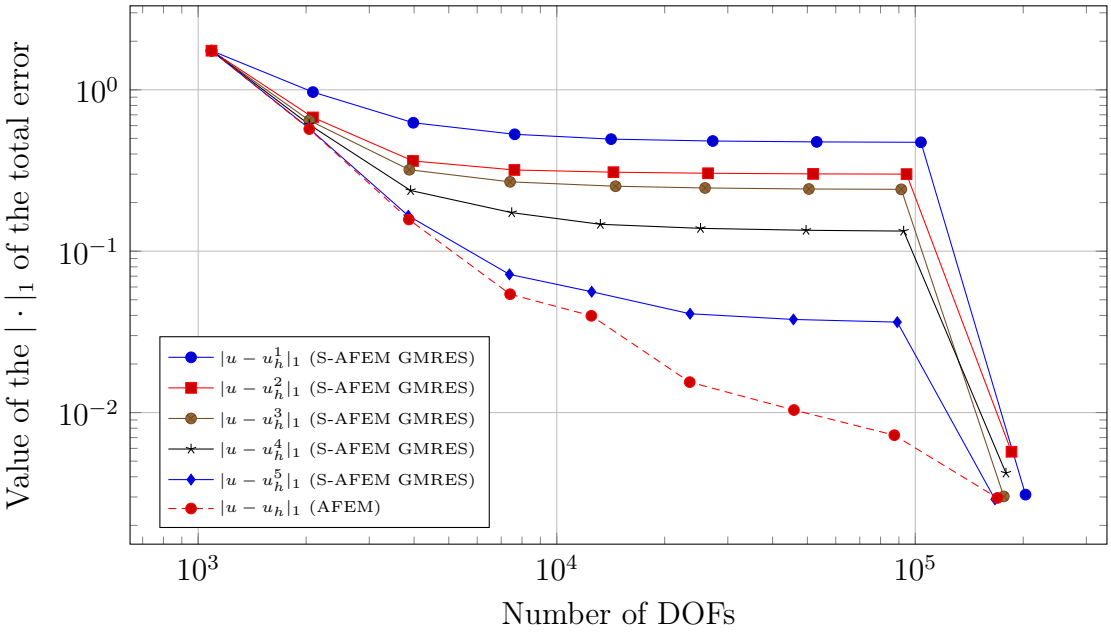


Figure 5.160: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 2$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

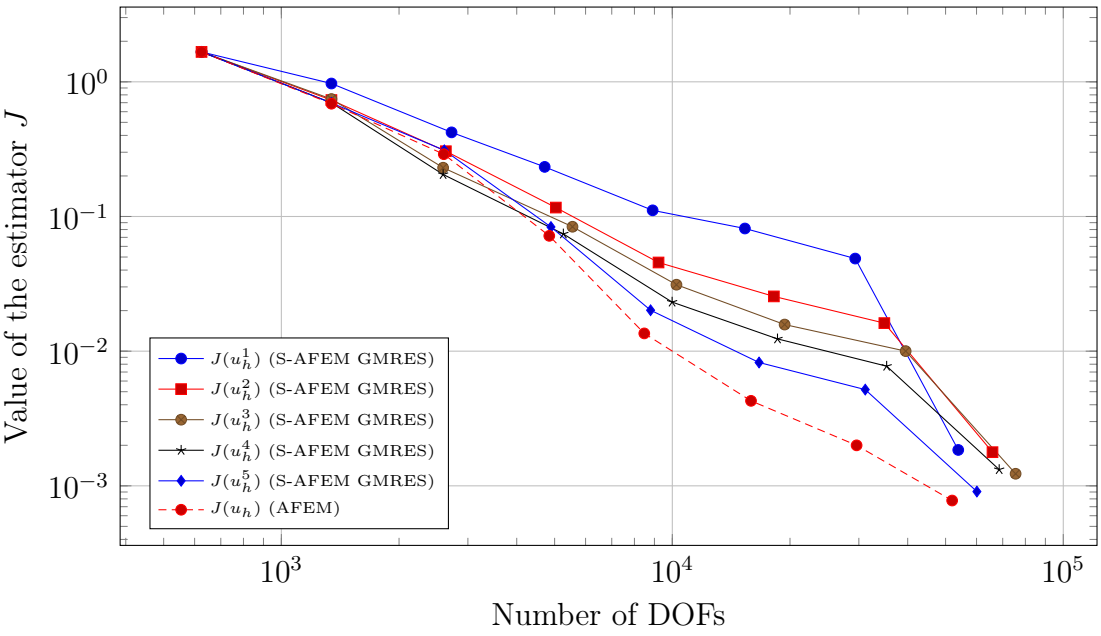


Figure 5.161: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

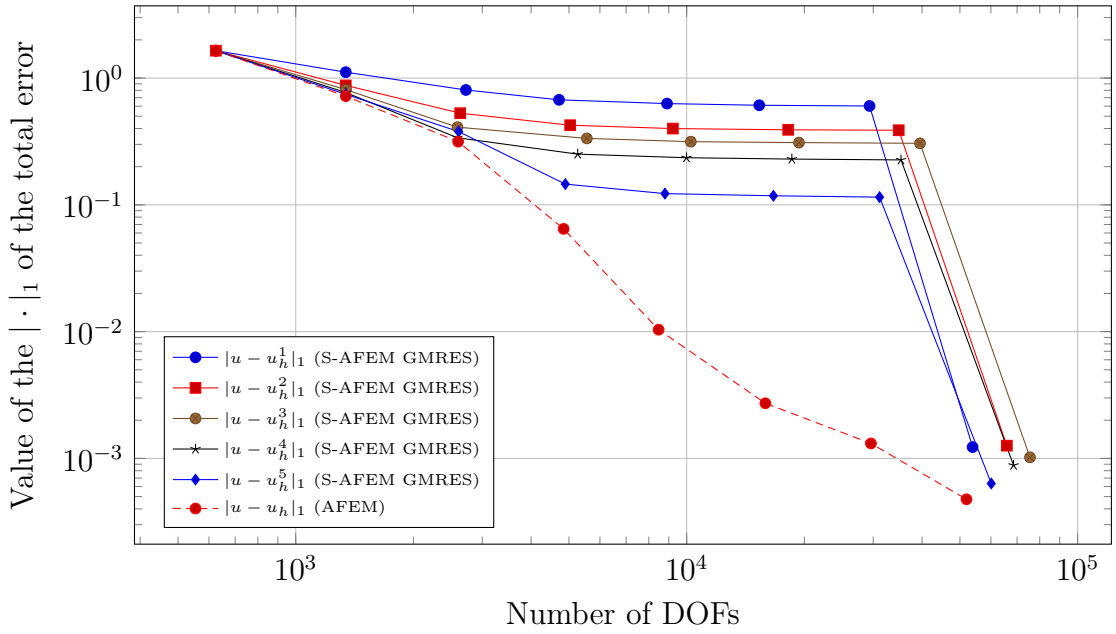


Figure 5.162: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 3$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

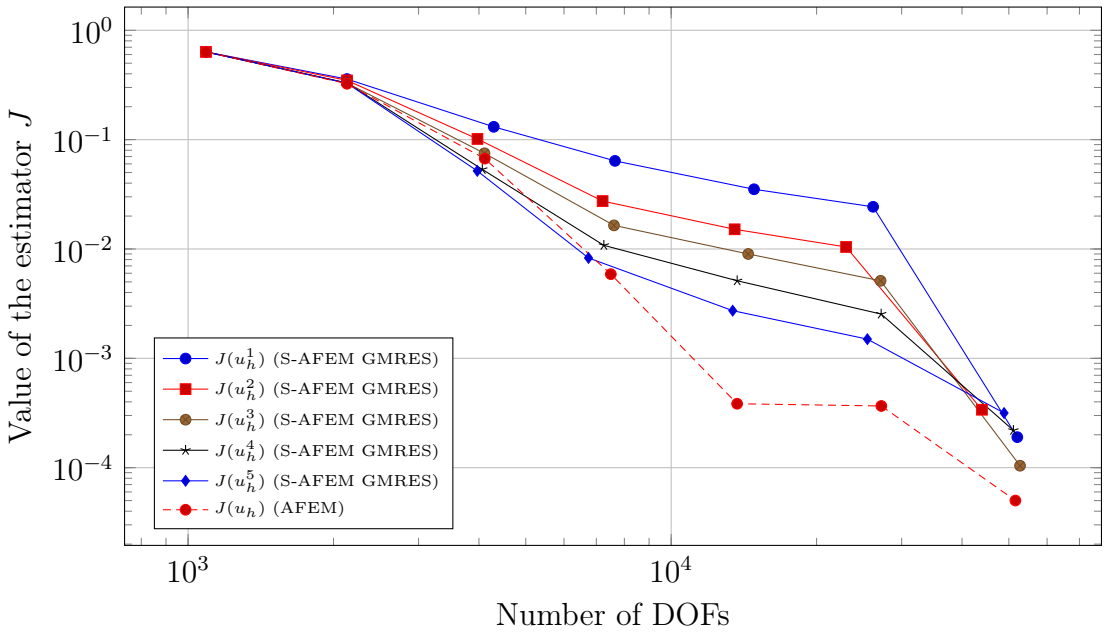


Figure 5.163: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

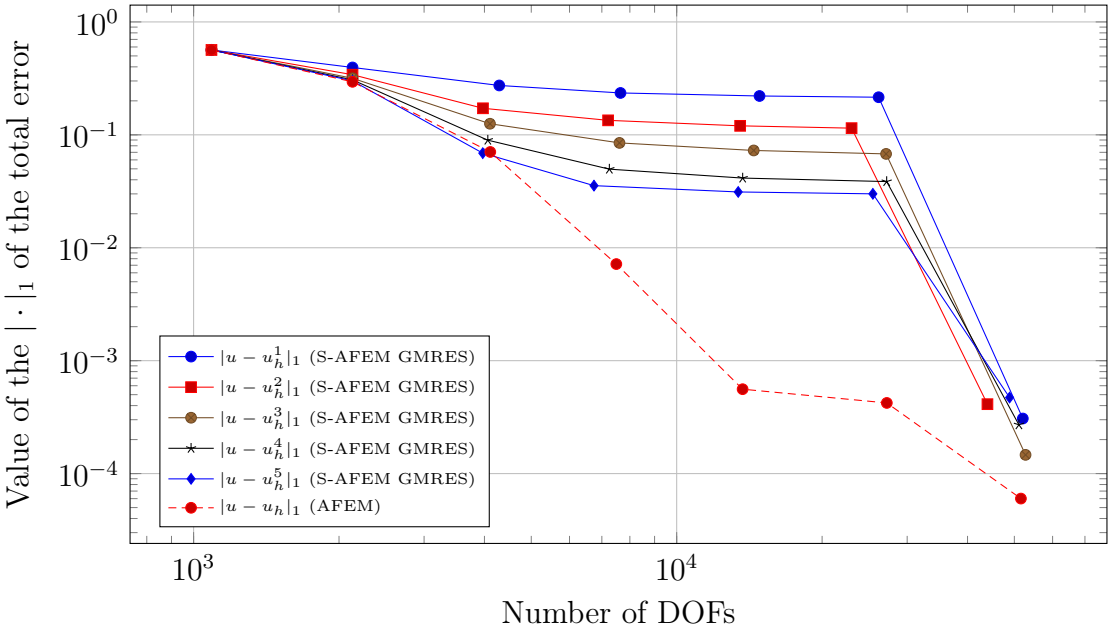


Figure 5.164: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 4$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

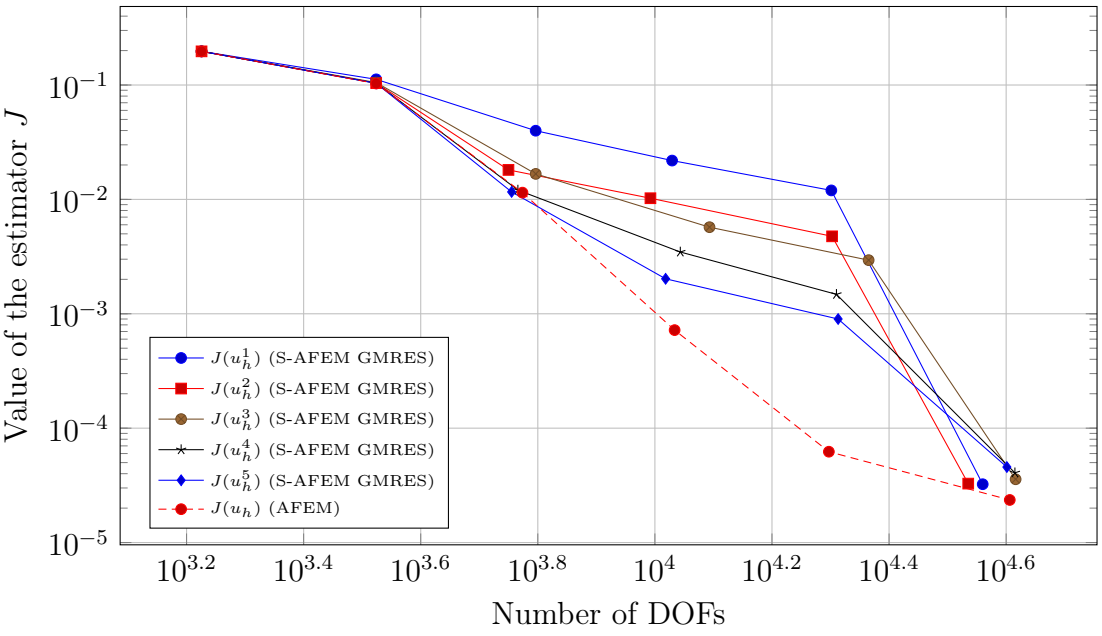


Figure 5.165: Value of the error estimator J for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

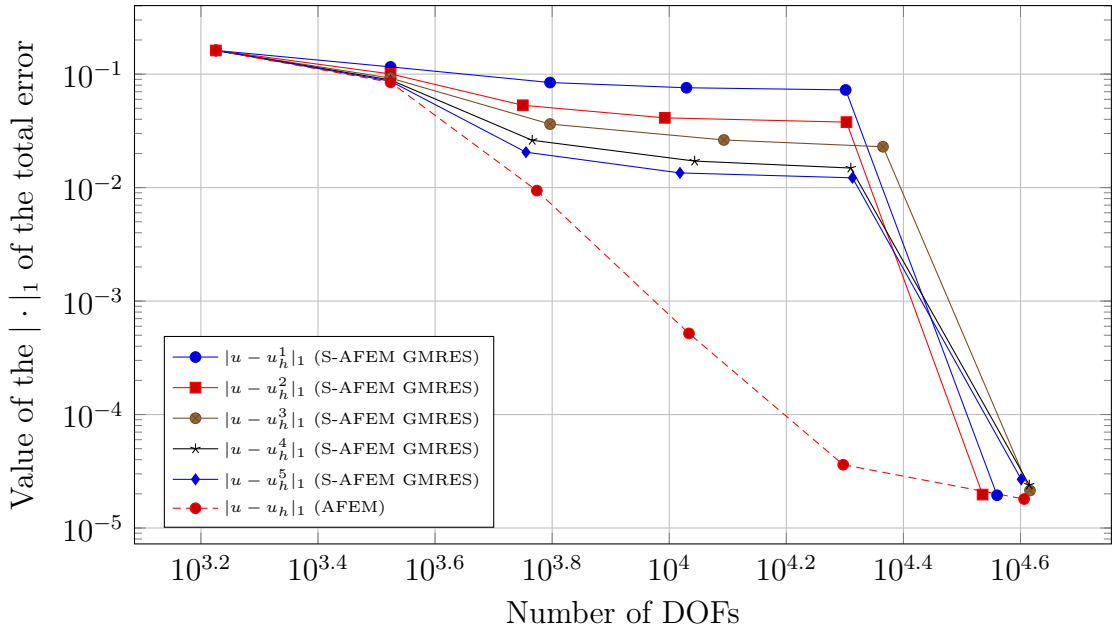


Figure 5.166: Value of the $|\cdot|_1$ error for the diffusion transport problem in 2D, with transport term $\boldsymbol{\beta} = (50, 50)$, and FEM discretisation degree $deg = 5$, when we apply 10 cycles of AFEM and S-AFEM with GMRES as a smoother, as the smoothing iteration count l goes from 1 to 5.

fraction of the computational cost.

This thesis work contains the detailed and rigorous analysis of the error propagation properties of the S-AFEM. In particular, we prove a series of theorems and results that rigorously estimate the algebraic error propagation between different nested levels, which shows that the algebraic error is made up by small contributions given by the accumulation of low frequency terms, which have in general a smaller influence on the estimator. The analysis of the discretisation error and convergence estimates for S-AFEM are not covered in this work but they are part of our current and future investigations.

We provide numerical evidences that the S-AFEM strategy is competitive in cost and accuracy by considering different variants of our algorithm, where different smoothers are considered for the intermediate cycles (respectively Richardson iteration, the CG method, and the GMRES method). We conclude that in general, all smoothers are good smoother candidates for the intermediate cycles. While in our numerical experiments the CG and the GMRES turn out to be better smoothers compared to Richardson iterations, we'd like to point out that its performance strictly depends on the relaxation parameter γ, and can be improved with a detailed study of the optimal choice for γ.

Moreover, we investigate the accuracy and of S-AFEM for high order finite element discretisations and we find out that S-AFEM turns out to be a good strategy also for higher order finite element discretisations, and one could use directly the CG method (or, alternatively, the GMRES method) as smoothers for the intermediate iterations. Our numerical evidences show that two smoothing

iterations are enough for the two-dimensional case, while from five up to seven smoothing iteration are enough for the three-dimensional case independently on the polynomial degree of the finite element approximation.

Finally, we show how the S-AFEM strategy works well not only for classical two and three dimensional problems that are used to benchmark adaptive mesh-refining techniques, but also for more complex non symmetric problems by providing a two-dimensional example of a diffusion-transport problem. Even if at the moment we don't have theoretical results for these more complex cases, there are good insights that S-AFEM algorithm leads to the shown speed-up even in realistic scenarios, making it a highly valuable and promising algorithm for many practical applications.

Motivated by the theoretical results on the discretisation error and by our numerical evidences, 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 at each cycle, through a smoothing method. We show that various types of smoothing methods work well for S-AFEM and only a few smoothing iterations are necessary, for the estimator to manifest a quasi optimal convergence order and for it to produce the same set of marked cells for refinement at each cycle.

Bibliography

[1] R. A. Adams and J. J. Fournier. *Sobolev spaces*, volume 140. Academic press, 2003.

[2] M. Ainsworth and J. T. Oden. *A posteriori error estimation in finite element analysis*, volume 37. John Wiley & Sons, 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. *Journal of Numerical Mathematics*, 26(4):173–183, 2018.

[4] J. Argyris and S. Kelsey. Energy theorems and structural analysis 1960, 1954.

[5] M. Arioli, E. H. Georgoulis, and D. Loghin. Stopping criteria for adaptive finite element solvers. *SIAM Journal on Scientific Computing*, 35(3):A1537–A1559, 2013.

[6] M. Arioli, J. Liesen, A. Miçdlar, and Z. Strakoš. Interplay between discretization and algebraic computation in adaptive numerical solutionof elliptic pde problems. *GAMM-Mitteilungen*, 36(1):102–129, 2013.

[7] M. Aurada, M. Feischl, T. Führer, M. Karkulik, J. M. Melenk, and D. Praetorius. Classical fem-bem coupling methods: nonlinearities, well-posedness, and adaptivity. *Computational Mechanics*, 51(4):399–419, 2013.

[8] I. Babuska and A. K. Aziz. Lectures on mathematical foundations of the finite element method. Technical report, MARYLAND UNIV., COLLEGE PARK. INST. FOR FLUID DYNAMICS AND APPLIED MATHEMATICS., 1972.

[9] I. Babuška and W. C. Rheinboldt. A-posteriori error estimates for the finite element method. *International Journal for Numerical Methods in Engineering*, 12(10):1597–1615, 1978.

[10] I. Babuška and T. Strouboulis. *The finite element method and its reliability*. Oxford university press, 2001.

[11] I. Babuska, T. Strouboulis, I. Babuška, et al. *The finite element method and its reliability*. Oxford university press, 2001.

[12] N. S. Bakhvalov. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Computational Mathematics and Mathematical Physics*, 6(5):101–135, 1966.

[13] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[14] W. Bangerth and R. Rannacher. *Adaptive finite element methods for differential equations*. Birkhäuser, 2013.

[15] R. E. Bank and T. Dupont. An optimal order process for solving finite element equations. *Mathematics of Computation*, 36(153):35–51, 1981.

[16] R. Becker and S. Mao. Convergence and quasi-optimal complexity of a simple adaptive finite element method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(6):1203–1219, 2009.

[17] P. Binev, W. Dahmen, and R. DeVore. Adaptive finite element methods with convergence rates. *Numerische Mathematik*, 97(2):219–268, 2004.

[18] D. Braess and W. Hackbusch. A new convergence proof for the multigrid method including the v-cycle. *SIAM Journal on Numerical Analysis*, 20(5):967–975, 1983.

[19] H. Brakhage. Über die numerische behandlung von integralgleichungen nach der quadraturformelmethode. *Numerische Mathematik*, 2(1):183–196, 1960.

[20] J. H. Bramble. Multigrid methods, vol. 294 of pitman research notes in mathematical sciences, 1993.

[21] J. H. Bramble. *Multigrid methods*. Routledge, 2018.

[22] J. H. Bramble and J. E. Pasciak. New convergence estimates for multigrid algorithms. *Mathematics of computation*, 49(180):311–329, 1987.

[23] J. H. Bramble and J. E. Pasciak. The analysis of smoothers for multigrid algorithms. *mathematics of computation*, 58(198):467–488, 1992.

[24] J. H. Bramble and J. E. Pasciak. New estimates for multilevel algorithms including the v-cycle. *Mathematics of computation*, 60(202):447–471, 1993.

[25] J. H. Bramble, J. E. Pasciak, J. P. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Mathematics of Computation*, 57(195):1–21, 1991.

[26] J. H. Bramble and X. Zhang. The analysis of multigrid methods. *Handbook of numerical analysis*, 7:173–415, 2000.

[27] A. Brandt. Multi-level adaptive technique (mlat) for fast numerical solution to boundary value problems. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 82–89. Springer, 1973.

[28] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[29] A. Brandt. Guide to multigrid development. In *Multigrid methods*, pages 220–312. Springer, 1982.

[30] A. Brandt. Algebraic multigrid theory: The symmetric case. *Applied mathematics and computation*, 19(1-4):23–56, 1986.

[31] A. Brandt, S. McCoruick, and J. Huge. Algebraic multigrid (amg) f0r sparse matrix equati0ns. *Sparsity and its Applications*, 257, 1985.

[32] S. Brenner. Convergence of the multigrid v-cycle algorithm for second-order boundary value problems without full elliptic regularity. *Mathematics of Computation*, 71(238):507–525, 2002.

[33] S. Brenner and R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007.

[34] S. C. Brenner and C. Carstensen. Finite element methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–47, 2017.

[35] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[36] W. L. Briggs, S. F. McCormick, et al. *A multigrid tutorial*, volume 72. Siam, 2000.

[37] I. Bubuška and M. Vogelius. Feedback and adaptive finite element solution of one-dimensional boundary value problems. *Numerische Mathematik*, 44(1):75–102, 1984.

[38] C. Carstensen. Quasi-interpolation and a posteriori error analysis in finite element methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(6):1187–1202, 1999.

[39] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. Axioms of adaptivity. *Computers & Mathematics with Applications*, 67(6):1195–1253, 2014.

[40] C. Carstensen and R. Verfürth. Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM journal on numerical analysis*, 36(5):1571–1587, 1999.

[41] J. M. Cascon, C. Kreuzer, R. H. Nochetto, and K. G. Siebert. Quasi-optimal convergence rate for an adaptive finite element method. *SIAM Journal on Numerical Analysis*, 46(5):2524–2550, 2008.

[42] Z. Chen. *Finite element methods and their applications*. Springer Science & Business Media, 2005.

[43] P. G. Ciarlet. The finite element method for elliptic problems. *Classics in applied mathematics*, 40:1–511, 2002.

[44] P. Clément. Approximation by finite element functions using local regularization. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):77–84, 1975.

[45] R. W. Clough. The finite element method in plane stress analysis. In *Proceedings of 2nd ASCE Conference on Electronic Computation, Pittsburgh Pa., Sept. 8 and 9, 1960*, 1960.

[46] R. W. Clough. Original formulation of the finite element method. *Finite Elements in Analysis and Design*, 7(2):89–101, 1990.

[47] R. W. Clough and E. L. Wilson. *Stress analysis of a gravity dam by the finite element method*. Laboratório Nacional de Engenharia Civil, 1962.

[48] R. Courant et al. *Variational methods for the solution of problems of equilibrium and vibrations*. Verlag nicht ermittelbar, 1943.

[49] M. Dauge. *Elliptic boundary value problems on corner domains: smoothness and asymptotics of solutions*, volume 1341. Springer, 2006.

[50] W. Dörfler. A convergent adaptive algorithm for poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.

[51] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. *Acta numerica*, 4:105–158, 1995.

[52] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*, volume 159. Springer Science & Business Media, 2013.

[53] R. P. Fedorenko. A relaxation method for solving elliptic difference equations. *USSR Computational Mathematics and Mathematical Physics*, 1(4):1092–1096, 1962.

[54] R. P. Fedorenko. The speed of convergence of one iterative process. *USSR Computational Mathematics and Mathematical Physics*, 4(3):227–235, 1964.

[55] M. Feischl, T. Führer, N. Heuer, M. Karkulik, and D. Praetorius. Adaptive boundary element methods. *Archives of Computational Methods in Engineering*, 22(3):309–389, 2015.

[56] M. Feischl, M. Karkulik, J. M. Melenk, and D. Praetorius. Quasi-optimal convergence rate for an adaptive boundary element method. *SIAM Journal on Numerical Analysis*, 51(2):1327–1348, 2013.

[57] T. Gantumur. Adaptive boundary element methods with convergence rates. *Numerische Mathematik*, 124(3):471–516, 2013.

[58] T. Grätsch and K.-J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & structures*, 83(4-5):235–265, 2005.

[59] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative schwarz algorithms. *Numerische Mathematik*, 70(2):163–180, 1995.

[60] P. Grisvard. *Elliptic problems in nonsmooth domains*. SIAM, 2011.

[61] W. Hackbusch. *Iterative solution of large sparse systems of equations*, volume 95. Springer, 1994.

[62] W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.

[63] W. Hackbusch. *Elliptic differential equations: theory and numerical treatment*, volume 18. Springer, 2017.

[64] P. Jiránek, Z. Strakoš, and M. Vohralík. A posteriori error estimates including algebraic error and stopping criteria for iterative solvers. *SIAM Journal on Scientific Computing*, 32(3):1567–1590, 2010.

[65] L. Kronsjö and G. Dahlquist. On the design of nested iterations for elliptic difference equations. *BIT Numerical Mathematics*, 12(1):63–71, 1972.

[66] S. Lang. *Introduction to linear algebra*. Springer Science & Business Media, 2012.

[67] J. Liesen and Z. Strakos. *Krylov subspace methods: principles and analysis*. Oxford University Press, 2013.

[68] S. F. McCormick. *Multigrid methods*. SIAM, 1987.

[69] P. Morin, R. H. Nochetto, and K. G. Siebert. Data oscillation and convergence of adaptive fem. *SIAM Journal on Numerical Analysis*, 38(2):466–488, 2000.

[70] P. Morin, R. H. Nochetto, and K. G. Siebert. Convergence of adaptive finite element methods. *SIAM review*, 44(4):631–658, 2002.

[71] O. Mulita. Dominating terms in the edge contribution of classical residual-based a posteriori error estimators. 2019.

[72] O. Mulita, S. Giani, and L. Heltai. Quasi-optimal mesh sequence construction through smoothed adaptive finite element method. *arXiv:1905.06924*, 2019.

[73] R. H. Nochetto, K. G. Siebert, and A. Veeser. Theory of adaptive finite element methods: an introduction. In *Multiscale, nonlinear and adaptive approximation*, pages 409–542. Springer, 2009.

[74] J. Papež, J. Liesen, and Z. Strakoš. Distribution of the discretization and algebraic error in numerical solution of partial differential equations. *Linear Algebra and its Applications*, 449:89–114, 2014.

[75] J. Papež and Z. Strakoš. On a residual-based a posteriori error estimator for the total error. *IMA Journal of Numerical Analysis*, 2017.

[76] J. Papež, Z. Strakoš, and M. Vohralík. Estimating and localizing the algebraic and total numerical errors using flux reconstructions. *Numerische Mathematik*, 138(3):681–721, 2018.

[77] A. Quarteroni. *Modellistica numerica per problemi differenziali*, volume 97. Springer, 2016.

[78] W. Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

[79] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*, pages 73–130. SIAM, 1987.

[80] Y. Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[81] A. Sartori, N. Giuliani, M. Bardelloni, and L. Heltai. deal2lkit: A toolkit library for high performance programming in deal.ii. *SoftwareX*, 7:318–327, 2018.

[82] K. Schellbach. Probleme der variationsrechnung. *Journal für die reine und angewandte Mathematik*, 41:293–363, 1851.

[83] L. R. Scott and S. Zhang. Higher-dimensional nonnested multigrid methods. *Mathematics of Computation*, 58(198):457–466, 1992.

[84] R. Stevenson. Optimality of a standard adaptive finite element method. *Foundations of Computational Mathematics*, 7(2):245–269, 2007.

[85] Z. Strakoš and J. Liesen. On numerical stability in large scale linear algebraic computations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, 85(5):307–325, 2005.

[86] G. Strang and G. J. Fix. *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[87] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid.* Elsevier, 2000.

[88] M. Turner, R. Clough, T. L. Martin, and L. Topp. Hc stiffness and deflection analysis of complex structures. *Journal of areonautical Sciences*, 23:805–823, 1956.

[89] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50(1-3):67–83, 1994.

[90] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques.* John Wiley & Sons Inc, 1996.

[91] R. Verfürth. Adaptive finite element methods lecture notes winter term 2007/08. 2008.

[92] P. Wesseling. An introduction to multigrid methods. rt edwards. *Inc.,*, 2:2, 2004.

[93] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4):581–613, 1992.

[94] X. Zhang. Multilevel schwarz methods. *Numerische Mathematik*, 63(1):521–539, 1992.

[95] O. C. Zienkiewicz and Y. K. Cheung. The finite element method in structural and continuum mechanics. Technical report, McGraw-Hill, 1967.

[96] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals.* Elsevier, 2005.