Approximate inexpensive multigrid block-Jacobi solver using extension of basis functions and a levelwise optimal step-size

Zhaonan Dong, Clément Maradei, Martin Vohralík

Approximate inexpensive multigrid block-Jacobi solver using extension of basis functions and a levelwise optimal step-size

Zhaonan Dong[1,2], Clément Maradei[1,2], and Martin Vohralík[1,2]

[1]Inria, 48 rue Barrault, 75647 Paris, France.
[2]CERMICS, ENPC, Institut Polytechnique de Paris, CNRS, 6 & 8 avenue B. Pascal, 77455 Marne-la-Vallée, France.

June 30, 2026

Abstract

We develop an inexpensive multigrid solver guided by an a posteriori estimator of the algebraic error. We consider a second-order elliptic diffusion problem discretized with conforming finite elements of arbitrary polynomial degree p in $1 \leqslant d \leqslant 3$ space dimensions on possibly highly graded simplicial meshes. We build on [SIAM J. Sci. Comput. 43, 5 (2021), S117–S145], where one step of the patchwise block-Jacobi smoother followed by line search per mesh level is proven to lead to p-uniform contraction. Our goal is to significantly reduce its computational cost, while retaining similar contraction rate. Our central idea is to approximate the solution of the block-Jacobi problems, one per vertex patch, by solving block-Jacobi problems, one per simplex instead, using a sequential loop over simplices sharing the vertex in question. This can typically be realized via factorization of $(d+1)$ matrices on the reference simplex once and for all before the iterations start. In addition, we employ a line-search procedure to determine, on each mesh level, the optimal step sizes for the smoothing correction. We additionnaly introduce a simple and effective strategy that allows the solver to adaptively select the number of (post-)smoothing steps on each level, resulting in further contraction rate improvement. Numerical tests show that our method reduces the computational cost of the original p-uniform patchwise block-Jacobi approach by roughly two orders of magnitude, being comparable in cost to the standard Jacobi (diagonal) multigrid smoother, while converging much faster. These tests also highlight the benefits of adaptively selecting the number of post-smoothing steps.

1 Introduction

Multilevel methods are widely recognized for their effectiveness and versatility as solvers and preconditioners for large, sparse algebraic linear systems arising from the numerical discretization of partial differential equations. We refer the reader to the overviews of Hackbusch [10], Briggs *et al.* [4], Zhang [23], Oswald [16], and Griebel and Oswald [9]. A key property of these methods is that their convergence typically does not depend on the mesh size, i.e., they are h-robust. Nevertheless, their performance remains influenced by the polynomial degree of approximation p.

Early developments of p-robust solvers include the work of Quarteroni and Sacchi-Landriani [19] and Widlund and Pavarino [18], which addressed specific domain configurations decomposable

into rectangles without internal cross points, and the extension to more general quadrilateral and hexahedral meshes by Pavarino [17]. These results were further generalized to triangular and tetrahedral meshes by Schöberl *et al.* [20] through the introduction of an additive Schwarz preconditioner. For quadrilateral or hexahedral meshses, Huismann, Stiller, and Fröhlich [11] derived matrix-free explicit inverse static condensed operator enabling an additive Schwarz-type smoother that scales linearly in terms of DoFs. Brubeck and Farell showed how to construct optimal complexity solvers avoinding the patch problems for high order polynomial degrees [5, 6] in that setting. Within the framework of discontinuous Galerkin methods, Antonietti *et al.* [1, 2] established p-robust convergence results under the requirement of at least p^2 smoothing steps. In the finite element and simplicial mesh setting, Miraçi *et al.* [13, 14] demonstrated that it is possible to construct a p-robust solver driven by an *a posteriori* estimator of the algebraic error, which is localizable across mesh levels and element patches.

In this work, we present an inexpensive multilevel iterative solver built upon the solver introduced in [13]. We aim to improve its computational efficiency by reducing the need to solve one patch problem per each vertex (in the finest mesh or in each mesh level) to $d+1$ problems per reference simplex, where d is the space dimension. To this end, we construct a lifting that can be interpreted as an approximation of the algebraic error by continuous piecewise polynomials of degree p. This lifting is obtained through a V-cycle multigrid method without presmoothing and with a single postsmoothing step, complemented by an optimal step size correction by line search on each level, while the coarse correction is provided by a lowest polynomial degree (piecewise linear) function. The proposed solver belongs to the class of additive Schwarz methods applied to overlapping subdomains composed of element patches. Locally, the patch problems are approximated via an appropriate sequential procedure based on an extension operator and individual mesh element spaces.

We prove that the solver contracts the algebraic error under the assumption that a specific ratio remains uniformly bounded by a constant that may depend on p. The connection between this ratio, its possible p-dependence, and the contraction property is rigorously established both in the theoretical analysis and in the numerical experiments. Moreover, the solver is equipped with a built-in *a-posteriori* estimate of the algebraic error, associated with an error reduction formula quantifying the decrease of the algebraic error achieved at each iteration. Exploiting this feature, we design a procedure that adaptively selects the number of smoothing steps to be performed on each mesh level. Furthermore, for the case $\mathrm{Id} = \mathcal{K}$, we derive complexity estimates and propose efficient implementation strategies, demonstrating the effectiveness of the approach in terms of floating-point operations.

This contribution is organized as follows. In Section 2, we introduce the multilevel setting and the notation used throughout the paper. Section 3 presents the motivation and outlines the key ideas underlying the proposed solver. In Section 4, we introduce our main tool, namely the extension of basis functions, and define the resulting solver replacing the patch problems by sequential loops with cell problems, together with basic properties that follow from its construction. In Section 6, we propose a simple adaptive procedure aimed at reducing both the total number of smoothing steps and the computational cost required for convergence. Section 7 provides insight into how the proposed approach can be efficiently implemented in the case $\mathcal{K} = \mathrm{Id}$. Section 8 introduces alternative solvers used for comparisons in Section 10. Section 9 recalls and introduces complexity estimates in terms of floating-point operations, which form the basis for the comparisons presented in Section 10, where numerical experiments are reported. Section 11 is devoted to the proof of the main results. Finally, in Appendices A–C, we present a series of numerical experiments on a simple example designed to complement and support the theoretical analysis developed throughout the paper. These experiments provide numerical evidence for the link introduced in Section 3 and offer additional insight into the assumptions underlying our

main results.

2 Setting

2.1 Model problem

Let $\Omega \in \mathbb{R}^d, 1 \leqslant d \leqslant 3$, be an open bounded connected domain with Lipschitz boundary $\partial\Omega$. Let $f \in L^2(\Omega)$ be a source term and $\mathcal{K} \in [L^\infty(\Omega)]^{d\times d}$ a symmetric positive definite diffusion coefficient. We consider a second-order elliptic diffusion problem defined over Ω where we look for the weak solution $u \in H^1_0(\Omega)$ such that

$$(\mathcal{K}\nabla u, \nabla v) = (f, v) \quad \forall v \in H^1_0(\Omega). \tag{2.1}$$

Then, we use the standard notation $(\cdot,\cdot)$ for the $L^2(\Omega)$ or $[L^2(\Omega)]^d$ scalar product.

2.2 Finite element discretization and algebraic system

Let $\mathcal{T}_J, J \geqslant 1$ be a matching simplicial mesh of Ω, and let an integer $p \geqslant 1$ denote a polynomial degree. The finite element space is

$$V_J^p := \mathcal{P}_p(\mathcal{T}_J) \cap H^1_0(\Omega), \tag{2.2}$$

where $\mathcal{P}_p(\mathcal{T}_J) := \{v_J \in L^2(\Omega),\ v_J|_K \in \mathcal{P}_p(K)\ \forall K \in \mathcal{T}_J\}$. Let $N_J := \dim(V_J^p)$. The discrete counterpart of (2.1) consists in finding $u_J \in V_J^p$ such that

$$(\mathcal{K}\nabla u_J, \nabla v_J) = (f, v_J) \quad \forall v_J \in V_J^p. \tag{2.3}$$

If we introduce a basis $\phi_J^l, 1 \leqslant l \leqslant N_J$, of the space V_J^p, then problem (2.3) becomes equivalent to solving a system of linear algebraic equations: find $U_J \in \mathbb{R}^{N_J}$ such that

$$\mathbb{A}_J U_J = F_J, \tag{2.4}$$

where $(\mathbb{A}_J)_{lm} := (\mathcal{K}\nabla\phi_J^m, \nabla\phi_J^l)$ denotes the symmetric and positive definite stiffness matrix and $(F_J)_l := (f, \phi_J^l)$ the load vector. The solution can then be expressed as $u_J = \sum_{m=1}^{N_J}(U_J)_m\phi_J^m$. In this work we, however, prefer to work with (2.3), which is independent of the choice of the basis ϕ_J^l, in a crucial contrast to (2.4), which depends on it. Consequently, all our results are basis independent.

2.3 A hierarchy of meshes and spaces

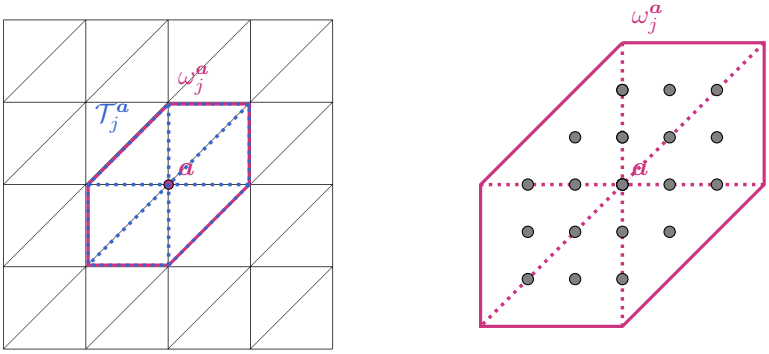
In order to design an efficient solver for the linear system (2.4), we suppose that there is a hierarchy of matching simplicial meshes $\{\mathcal{T}_j\}_{0\leqslant j\leqslant J}, J \geqslant 1$, where $\mathcal{T}_J$ has been introduced earlier, and where each mesh $\mathcal{T}_j$ is obtained as a refinement of $\mathcal{T}_{j-1}, 1 \leqslant j \leqslant J$. Alongside this mesh hierarchy, we define a corresponding hierarchy of the finite element spaces. To this end, for each mesh level $j \in \{0,\dots,J\}$, we fix p_j, the polynomial degree associated to mesh level j, such that $1 = p_0 \leqslant p_1 = \ldots = p_{J-1} \leqslant p_J = p$. In particular, let:

$$\text{for } j = 0: \qquad V_0^1 := \mathcal{P}_1(\mathcal{T}_0) \cap H^1_0(\Omega) \quad \text{(lowest-order space)}, \tag{2.5a}$$

$$\text{for } 1 \leqslant j \leqslant J-1: \qquad V_j^{p_j} := \mathcal{P}_{p_j}(\mathcal{T}_j) \cap H^1_0(\Omega) \quad (p_j\text{-th order spaces}), \tag{2.5b}$$

$$\text{for } j = J: \qquad V_J^p := \mathcal{P}_p(\mathcal{T}_J) \cap H^1_0(\Omega) \quad (p\text{-th order space}), \tag{2.5c}$$

where $\mathcal{P}_{p_j}(\mathcal{T}_j) := \left\{v_j \in L^2(\Omega),\ v_j|_K \in \mathcal{P}_{p_j}(K)\ \forall K \in \mathcal{T}_j\right\}$. Note that, crucially, we have nestedness of the FEM spaces, that is $V_0^1 \subseteq V_1^{p_1} \subseteq \ldots \subseteq V_{J-1}^{p_{J-1}} \subseteq V_J^p$.



Patch $\mathcal{T}_j^{\boldsymbol{a}}$ around a vertex $\boldsymbol{a}$ Degrees of freedom of the space V_j^a for $p = 3$ with the corresponding patch subdomain $\omega_j^{\boldsymbol{a}}$

Figure 1: Illustration of the patch subdomain $\omega_j^{\boldsymbol{a}}$ of the patch $\mathcal{T}_j^{\boldsymbol{a}}$ (left) and the degrees of freedom of the space $V_j^{\boldsymbol{a}}$ for the Lagrange finite elements where $p_j = 3$ (right).

Let $\mathcal{V}_j, 0 \leqslant j \leqslant J$, be the set of vertices of the mesh $\mathcal{T}_j$. In what follows, we need to define the notion of patches of elements, illustrated in Figure 1. Given a vertex $\boldsymbol{a} \in \mathcal{V}_j$, we denote by $\mathcal{T}_j^{\boldsymbol{a}}$ all the mesh elements of $\mathcal{T}_j$ that share the vertex $\boldsymbol{a}$, $\mathcal{T}_j^{\boldsymbol{a}} := \{K \in \mathcal{T}_j, \boldsymbol{a} \in \mathcal{V}_K\}$, where $\mathcal{V}_K$ is the set of vertices of an element K. The corresponding open patch subdomain is denoted by $\omega_j^{\boldsymbol{a}}$. We also denote by $\psi_{j,\boldsymbol{a}}$ the standard hat function associated to the vertex $\boldsymbol{a} \in \mathcal{V}_j$, i.e., the piecewise affine function with respect to the mesh $\mathcal{T}_j$ taking value 1 at vertex $\boldsymbol{a}$ and vanishing at all other vertices of $\mathcal{V}_j$. Note that $\omega_j^{\boldsymbol{a}}$ is the support of $\psi_{j,\boldsymbol{a}}$. Finally, the local spaces $V_j^{\boldsymbol{a}}$ are defined by

$$V_j^{\boldsymbol{a}} := \mathcal{P}_{p_j}\left(\mathcal{T}_j^{\boldsymbol{a}}\right) \cap H_0^1\left(\omega_j^{\boldsymbol{a}}\right). \tag{2.6}$$

We set $N_{j,\boldsymbol{a}} := \dim(V_j^{\boldsymbol{a}})$, cf. Figure 1 for the illustration of the Lagrange degrees of freedom when $p_j = 3$.

2.4 Mesh refinement and regularity assumptions

For any mesh level $j \in \{0, \ldots, J\}$ and $K \in \mathcal{T}_j$, we denote by $h_K := \operatorname{diam}(K)$ the diameter of the element K whereas ρ_K denotes the diameter of the largest ball contained in K, and $h_j := \max_{K\in\mathcal{T}_j} h_K$ stands for the mesh size on level j. We will work with uniformly shape-regular meshes:

Assumption 2.1. (Mesh shape regularity). There exists $\kappa_{\mathcal{T}} > 0$ such that

$$\max_{K\in\mathcal{T}_j} \frac{h_K}{\rho_K} \leqslant \kappa_{\mathcal{T}} \text{ for all } 0 \leqslant j \leqslant J. \tag{2.7}$$

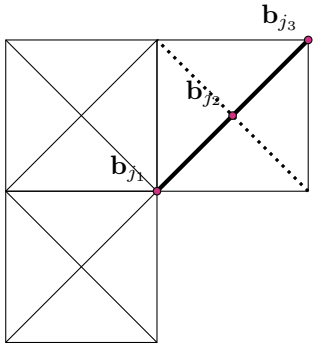
Below, we work under one of the two following additional assumptions. In the first setting, the hierarchy consists of quasi-uniform meshes with a bounded refinement factor between consecutive levels:

Assumption 2.2. (Refinement strength and mesh quasi-uniformity). There exists a fixed positive real number $0 < C_{\text{ref}} \leqslant 1$ such that for all $j \in \{1, \ldots, J\}$, for all $K \in \mathcal{T}_{j-1}$, and for any $K^* \in \mathcal{T}_j$ such that $K^* \subset K$, there holds

$$C_{\text{ref}}\, h_K \leqslant h_{K^*} \leqslant h_K, \tag{2.8}$$

There further exists a fixed positive real number $0 < C_{\text{qu}} \leqslant 1$ such that for all $j \in \{0, \ldots, J\}$ and for all $K \in \mathcal{T}_j$, there holds

$$C_{\text{qu}} h_j \leqslant h_K \leqslant h_j. \tag{2.9}$$



$\mathcal{T}_j$ obtained by a bisection of $\mathcal{T}_{j-1}$. Neighboring vertices on the refinement edge $\mathbf{b}_{j_1}, \mathbf{b}_{j_3}$. The new vertex after refinement is $\mathbf{b}_{j_2}$. $\mathcal{B}_j = \{\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \mathbf{b}_{j_3}\} \subset \mathcal{V}_j$

Figure 2: Illustration of the set $\mathcal{B}_j$; the refinement $\mathcal{T}_j$ (dotted lines) of the mesh $\mathcal{T}_{j-1}$ (full lines).

In the second setting, we work with a hierarchy generated from a quasi-uniform coarse mesh by a series of bisections, e.g., newest vertex bisection, cf. Mitchell [15] or Sewell [21]. In this case, one refinement of an edge of $\mathcal{T}_{j-1}$, for $j \in \{1, \ldots, J\}$, gives us a new finer mesh $\mathcal{T}_j$. We denote by $\mathcal{B}_j \subset \mathcal{V}_j$ the set consisting of the new vertex obtained after the bisection together with its two neighbors on the refinement edge; see Figure 2 for an illustration with $d = 2$. We also denote by $h_{\mathcal{B}_j}$ the maximal diameter of elements having a vertex in the set $\mathcal{B}_j$, for $j \in \{1, \ldots, J\}$. Here we assume:

Assumption 2.3. (Local refinement strength and the coarsest mesh quasi-uniformity of bisection-generated meshes). The coarsest mesh $\mathcal{T}_0$ is a conforming quasi-uniform mesh in the sense of (2.9), with parameter $0 < C^0_{\text{qu}} \leqslant 1$. The (possibly highly graded) conforming mesh $\mathcal{T}_J$ is generated from $\mathcal{T}_0$ by a series of bisections. There exists a fixed positive real number $0 < C_{\text{loc,qu}} \leqslant 1$ such that for all $j \in \{1, \ldots, J\}$, there holds $C_{\text{loc,qu}}\, h_{\mathcal{B}_j} \leqslant h_K \leqslant h_{\mathcal{B}_j}$ for all $K \in \mathcal{T}_j$ such that a vertex of K belongs to $\mathcal{B}_j$.

3 Motivation

We motivate here our approach.

3.1 p-robust solver with patch solves

As demonstrated in [13], a multilevel solver that ensures p-robust contraction with merely one post-smoothing step can be constructed for the finite element problem (2.3). Let u_J^i be the approximation of u_J on iteration i. Then the error on the next step $i+1$ is contracted according to

$$\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\| \leqslant \alpha\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|, \tag{3.1}$$

where the constant $0 \leqslant \alpha < 1$ is independent of p. In addition, the solver provides a built-in a posteriori estimator that gives a guaranteed and efficient (two-sided) bound on the algebraic error:

$$(\sqrt{1-\alpha^2})\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\| \leqslant \eta_{\mathrm{alg}}^i \leqslant \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|, \tag{3.2}$$

and more precisely

$$\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\|^2 = \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 - (\eta_{\mathrm{alg}}^i)^2, \tag{3.3}$$

thereby making the method *a posteriori steered*. The estimate of the algebraic error η_{alg}^i is defined

$$\eta_{\mathrm{alg}}^i := \left(\sum_{j=0}^{J}(\lambda_j^i\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|)^2\right)^{\frac{1}{2}}, \tag{3.4}$$

and requires the computation of the descent directions

$$\rho_j^i := \sum_{\boldsymbol{a}\in\mathcal{V}_j}\rho_{j,\boldsymbol{a}}^i \tag{3.5a}$$

for each mesh level $1 \leqslant j \leqslant J$, where $\rho_{j,\boldsymbol{a}}^i \in V_j^{\boldsymbol{a}}$ are the solutions of residual local problems of the form

$$\left(\mathcal{K}\nabla\rho_{j,\boldsymbol{a}}^i, \nabla v_{j,\boldsymbol{a}}\right)_{\omega_j^{\boldsymbol{a}}} = (f, v_{j,\boldsymbol{a}})_{\omega_j^{\boldsymbol{a}}} - \left(\mathcal{K}\nabla u_{J,j-1}^i, \nabla v_{j,\boldsymbol{a}}\right)_{\omega_j^{\boldsymbol{a}}} \quad \forall v_{j,\boldsymbol{a}} \in V_j^{\boldsymbol{a}}, \tag{3.5b}$$

while the coarse contribution $\rho_0^i \in V_0^1$ is given by the solution of the problem

$$\left(\mathcal{K}\nabla\rho_0^i, \nabla v_0\right) = (f, v_0) - \left(\mathcal{K}\nabla u_J^i, \nabla v_0\right) \quad \forall v_0 \in V_0^1. \tag{3.6}$$

Here $u_{J,j-1}^i$ is the approximate solution obtained on the level $j-1$, $u_{J,j-1}^i := u_J^i + \sum_{k=0}^{j-1}\lambda_k^i\rho_k^i$. The descent direction is paired with the corresponding optimal step size, defined by the following formula:

$$\lambda_j^i := \frac{\left(f, \rho_j^i\right) - \left(\mathcal{K}\nabla u_{J,j-1}^i, \nabla\rho_j^i\right)}{\|\mathcal{K}^{1/2}\nabla\rho_j^i\|^2}. \tag{3.7}$$

The new approximation is then $u_J^{i+1} := u_{J,J}^i$. Using (2.3), we note that (3.5b) can be equivalently formulated as local minimization problems over each vertex patch

$$\rho_{j,\boldsymbol{a}}^i := \underset{v_{j,\boldsymbol{a}} \in V_j^{\boldsymbol{a}}}{\arg\min} \quad \|\mathcal{K}^{\frac{1}{2}} \nabla((u_J - u_{J,j-1}) - v_{j,\boldsymbol{a}})\|_{\omega_j^{\boldsymbol{a}}}^2. \tag{3.8}$$

This approach, however, remains computationally demanding in terms of number of operations and memory. Indeed, one need to solve the vertex patch problem (3.5b) for each vertex on each mesh level. Whereas for $p_j = 1$, the block Jacobi matrix corresponding to (3.5b) is diagonal, it becomes larger and denser for increasing p_j. This paper looks for improvements of this limitation. Let us now introduce our key idea.

3.2 Sequential construction avoiding patch solves

Following the ideas developed in [8], it is possible to construct, element by element in a vertex patch, a piecewise polynomial which is, up to a constant only depending on the mesh shape regularity $\kappa_{\mathcal{T}}$ and space dimension d, as good as a solution of a problem over the piecewise polynomial space on the patch. More precisely, we can deduce the following result as a consequence of [8, Theorem 2.2]. Let $\tau_{j,\boldsymbol{a}}$ be a piecewise polynomial in $\mathcal{P}_{p_j}(\mathcal{T}_j^{\boldsymbol{a}})$ such that $\tau_{j,\boldsymbol{a}}|_F = 0$ for all faces F located on $\partial\omega_j^{\boldsymbol{a}}$. Then, there is a constant C only depending on $\kappa_{\mathcal{T}}$ and d, thus independent of the polynomial degree of approximation p_j such that

$$\min_{v_{j,\boldsymbol{a}} \in V_j^{\boldsymbol{a}}} \|\nabla_{\mathcal{T}}(\tau_{j,\boldsymbol{a}} - v_{j,\boldsymbol{a}})\|_{\omega_j^{\boldsymbol{a}}} \leqslant C \left(\sum_{i=1}^{|\mathcal{T}_j^{\boldsymbol{a}}|} \min_{v \in V_j^{\boldsymbol{a}}|_{K_i}, v|_{\partial K_i \cap \partial K_{i-1}} = \tilde{\rho}_{j,\boldsymbol{a}}^i|_{\partial K_{i-1}}} \|\nabla_{\mathcal{T}}(\tau_{j,\boldsymbol{a}} - v)\|_{K_i}^2 \right)^{\frac{1}{2}}, \tag{3.9}$$

where

$$\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_i} := \underset{v \in V_j^{\boldsymbol{a}}|_{K_i},\ v|_{\partial K_i \cap \partial K_{i-1}} = \tilde{\rho}_{j,\boldsymbol{a}}^i|_{\partial K_{i-1}}}{\arg\min} \|\nabla_{\mathcal{T}}(\tau_{j,\boldsymbol{a}} - v)\|_{K_i}, \text{ with } K_0 = \varnothing. \tag{3.10}$$

The procedure relies on suitable enumeration (shelling) of the cells K_i in $\mathcal{T}_j^{\boldsymbol{a}}$; in two space dimensions, this is simply a clockwise or counterclockwise loop. Thus, a patch solve on the left-hand side of (3.9) is as good as a sequential loop with elementwise problems on the right-hand of (3.9). Crucially, computing the elementwise minimizers $\tilde{\rho}_{j,\boldsymbol{a}}^i$ is significantly cheaper than computing the global minimizer $\rho_{j,\boldsymbol{a}}^i$ of the left-hand side of (3.9), while still providing a controlled p-robust approximation in the energy norm. This procedure is computationally illustrated in Appendix A. Unfortunately, we can not employ this idea directly, since in our case $\tau_{j,\boldsymbol{a}} = u_{J,j-1} - u_J$, see (3.8), where u_J is unknown. To gain access to the right hand side of (3.5b) element-wise, we need to employ an extension idea.

4 Multilevel Solver

We describe here our multilevel solver with the main idea being the replacement of the patch solve (3.5b) by the sequence of element solves, as motivated in Section 3.

4.1 Patch enumeration

Let us first consider the enumeration of the cells in a given patch $\mathcal{T}_j^{\boldsymbol{a}}$ in the form $K_1, ..., K_{|\mathcal{T}_j^{\boldsymbol{a}}|}$. To describe our idea simply, let us consider two space dimensions, a vertex $\boldsymbol{a} \in \mathcal{V}_j$ in the interior

of Ω, and the case where $|\mathcal{T}_j^{\boldsymbol{a}}|$ is even. Then the enumeration is simply in the counter-clockwise direction. The vertices on the boundary $\partial\omega_j^{\boldsymbol{a}}$ can be colored using two colors, with no two adjacent vertices sharing the same color. We use this to define $\mathbf{T}_{l,m} : K_l \to K_m$ as the unique bijective affine map between K_l and K_m that preserves the vertex $\boldsymbol{a}$ and the color of the other vertices. We show an illustration in Figure 3. For further details, we refer the reader to [8, Appendix B], where the case $d = 3$ as well as patches $\mathcal{T}_j^{\boldsymbol{a}}$ consisting of an odd number of elements are treated.

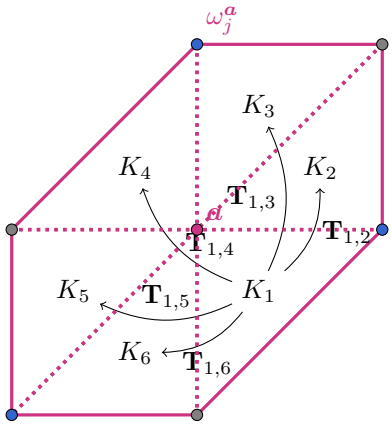


Figure 3: Illustration of the mappings $\mathbf{T}_{1,k}$ for a vertex patch $\mathcal{T}_j^{\boldsymbol{a}}$

4.2 Cell spaces

Let us now introduce the element infinite-dimensional spaces

$$\begin{aligned}
H_j^{\boldsymbol{a}}(K_1) &:= \left\{ v \in H^1(K_1),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0 \right\}, \\
H_j^{\boldsymbol{a}}(K_i) &:= \left\{ v \in H^1(K_i),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0, v|_{\partial K_i \cap \partial K_{i-1}} = 0 \right\}, \quad \forall 2 \leqslant i \leqslant |\mathcal{T}_j^{\boldsymbol{a}}| - 1, \\
H_j^{\boldsymbol{a}}(K_{|\mathcal{T}_j^{\boldsymbol{a}}|}) &:= \left\{ v \in H^1(K_{|\mathcal{T}_j^{\boldsymbol{a}}|}),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0, v|_{K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|-1}} = 0, v|_{\partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_1} = 0 \right\}
\end{aligned} \tag{4.1}$$

and their discrete counterparts

$$\begin{aligned}
V_j^{\boldsymbol{a}}(K_1) &:= \left\{ v \in \mathcal{P}_{p_j}(K_1),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0 \right\}, \\
V_j^{\boldsymbol{a}}(K_i) &:= \left\{ v \in \mathcal{P}_{p_j}(K_i),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0, v|_{\partial K_i \cap \partial K_{i-1}} = 0 \right\}, \quad \forall 2 \leqslant i \leqslant |\mathcal{T}_j^{\boldsymbol{a}}| - 1, \\
V_j^{\boldsymbol{a}}(K_{|\mathcal{T}_j^{\boldsymbol{a}}|}) &:= \left\{ v \in \mathcal{P}_{p_j}(K_{|\mathcal{T}_j^{\boldsymbol{a}}|}),\ v|_{\partial\omega_j^{\boldsymbol{a}}} = 0, v|_{\partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|-1}} = 0, v|_{\partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_1} = 0 \right\}.
\end{aligned} \tag{4.2}$$

Later on, we use the notation $N_j^{\boldsymbol{a}}(K_i)$ for the dimension of the space $V_j^{\boldsymbol{a}}(K_i)$.

4.3 Extension operators

We now introduce the extension operators. For all $1 \leqslant k \leqslant |\mathcal{T}_j^{\boldsymbol{a}}|$, we define the extension operators $E_k : H_j^{\boldsymbol{a}}(K_k) \to H_0^1(\omega_j^{\boldsymbol{a}})$ as follows:

$$\begin{aligned} E_1(v)|_{K_1} &:= v|_{K_1}, \quad E_1(v)|_{K_k} := v \circ \mathbf{T}_{1,k}^{-1} \quad \forall 2 \leqslant k \leqslant |\mathcal{T}_j^{\boldsymbol{a}}|, \\ E_k(v)|_{K_k} &:= v|_{K_k}, \quad E_k(v)|_{K_{k+1}} := v \circ \mathbf{T}_{k,k+1}^{-1} \quad \forall 2 \leqslant k \leqslant |\mathcal{T}_j^{\boldsymbol{a}}| - 1, \text{ and 0 otherwise}, \\ E_{|\mathcal{T}_j^{\boldsymbol{a}}|}(v)|_{K_{|\mathcal{T}_j^{\boldsymbol{a}}|}} &:= v|_{K_{|\mathcal{T}_j^{\boldsymbol{a}}|}} \text{ and 0 otherwise.} \end{aligned} \tag{4.3}$$

We use the notation $\mathrm{supp}(E_k)$ for the support of the extension operator E_k. In particular $|\mathrm{supp}(E_k)|$ denotes the number of elements K where E_k is non zero. We show an illustration in Figures 4–6. This operator coupled with the local cell spaces allows us to design a sequential procedure to produce an object $\tilde{\rho}_{j,\boldsymbol{a}}^i \in V_j^{\boldsymbol{a}}$, an approximation to $\rho_{j,\boldsymbol{a}}^i \in V_j^{\boldsymbol{a}}$ from (3.5b). This is the subject of Section 4.4.

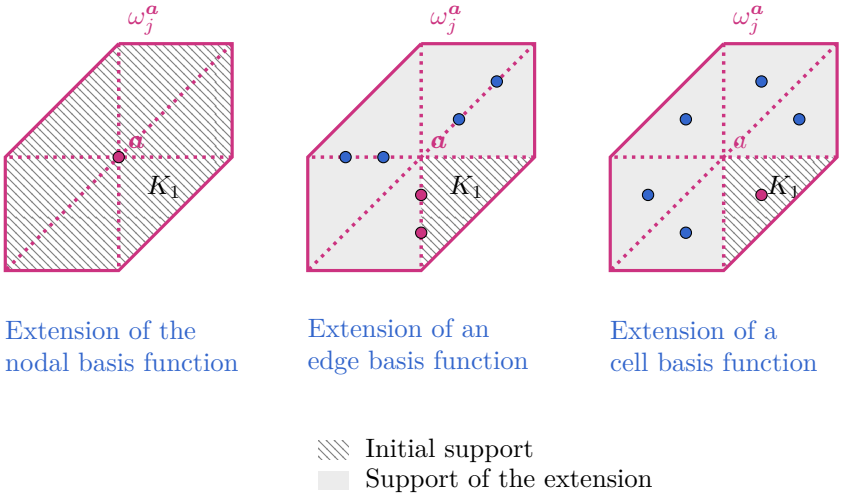


Figure 4: Illustration of the extension E_1 of basis functions for the Lagrange finite elements for different types of degrees of freedom when $p_J = 3$.

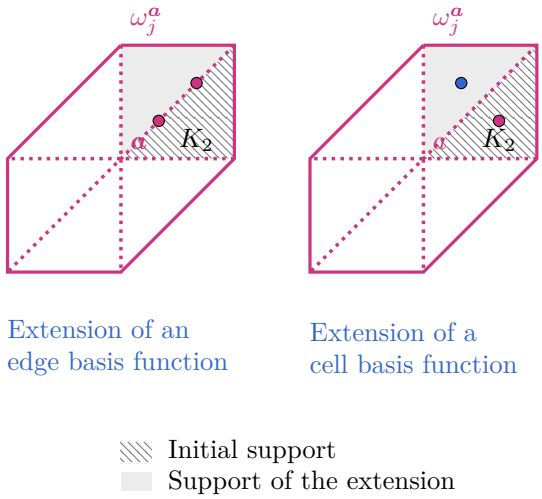


Figure 5: Illustration of the extension E_2 of basis functions for the Lagrange finite elements for different types of degrees of freedom when $p_J = 3$.

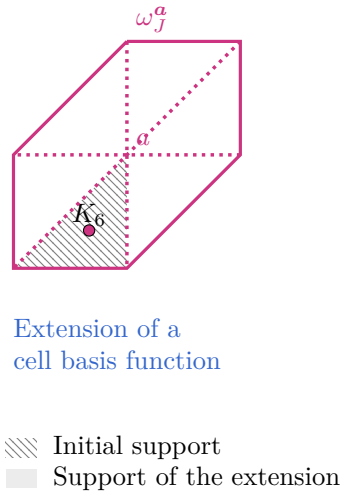


Figure 6: Illustration of the extension E_6 of basis functions for the Lagrange finite elements when $p_J = 3$. The initial support is in hatched, the support of the extension is in grey.

4.4 Solver with element-wise block-Jacobi smoother using extension of basis functions

In this section, we present our solver. It relies on the construction of approximate local residual liftings $\tilde{\rho}^i_{j,\boldsymbol{a}} \in V_j^{\boldsymbol{a}}$ via the extension operators of Section 4.3.

Algorithm 1: Multigrid solver using element solves and extension of basis functions

1. Initialize u_J^0 as the zero function of the space V_J^p and set $i = 0$.

2. Perform the following steps (a)–(d)

(a) Define $\tilde{\rho}_0^i$ as the solution of the coarse residual problem: no space

$$\left(\mathcal{K}\nabla\tilde{\rho}_0^i, \nabla v_0\right) = (f, v_0) - \left(\mathcal{K}\nabla u_J^i, \nabla v_0\right) \quad \forall v_0 \in V_0^1. \tag{4.4}$$

Define $\tilde{\lambda}_0^i := 1$ and set

$$u_{J,0}^i := u_J^i + \tilde{\lambda}_0^i \tilde{\rho}_0^i.$$

(b) **For** $j = 1 : J$ and for all vertices, define the local contributions $\tilde{\rho}_{j,\boldsymbol{a}}^i \in V_j^{\boldsymbol{a}}$ on the elements K_k of the patch $\mathcal{T}_j^{\boldsymbol{a}}$ as solutions of a sequence of cell problems :

For $k = 1 : |\mathcal{T}_j^{\boldsymbol{a}}|$, define the elementwise contributions $\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_k} \in \mathcal{P}_{p_j}(K_k)$
with $\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_k} = 0$ on $\partial\omega_j^{\boldsymbol{a}}$ such that
$(\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_k})|_{\partial K_k \cap \partial K_{k-1}} = (\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_{k-1}})|_{\partial K_k \cap \partial K_{k-1}}$ for $k = 2, \ldots, |\mathcal{T}_j^{\boldsymbol{a}}|$,
$(\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_{|\mathcal{T}_j^{\boldsymbol{a}}|}})|_{\partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_1} = (\tilde{\rho}_{j,\boldsymbol{a}}^i|_{K_1})|_{\partial K_{|\mathcal{T}_j^{\boldsymbol{a}}|} \cap \partial K_1}$,
as solutions of cell problems

$$\left(\mathcal{K}\nabla\tilde{\rho}_{j,\boldsymbol{a}}^i, \nabla v_{j,\boldsymbol{a}}\right)_{K_k} = \frac{(f, E_k v_{j,\boldsymbol{a}})_{\omega_j^{\boldsymbol{a}}} - \left(\mathcal{K}\nabla u_{J,j-1}^i, \nabla(E_k v_{j,\boldsymbol{a}})\right)_{\omega_j^{\boldsymbol{a}}}}{|\mathrm{supp}(E_k)|} \quad \forall v_{j,\boldsymbol{a}} \in V_j^{\boldsymbol{a}}(K_k). \tag{4.5a}$$

End For

- Define the descent direction $\tilde{\rho}_j^i \in V_j^{p_j}$ on the level j by

$$\tilde{\rho}_j^i := \sum_{\boldsymbol{a} \in V_j} \tilde{\rho}_{j,\boldsymbol{a}}^i. \tag{4.5b}$$

- **If** $\tilde{\rho}_j^i \neq 0$, define the optimal step-size $\tilde{\lambda}_j^i$ by

$$\tilde{\lambda}_j^i := \frac{\left(f, \tilde{\rho}_j^i\right) - \left(\mathcal{K}\nabla u_{J,j-1}^i, \nabla\tilde{\rho}_j^i\right)}{\|\mathcal{K}^{1/2}\nabla\tilde{\rho}_j^i\|^2}, \tag{4.6}$$

Else set $\tilde{\lambda}_j^i := 1$. Define the level update by

$$u_{J,j}^i = u_{J,j-1}^i + \tilde{\lambda}_j^i \tilde{\rho}_j^i. \tag{4.7}$$

End For

(c) Set $u_J^{i+1} := u_{J,J}^i \in V_J^p$.

(d) If the estimate of the algebraic error $\tilde{\eta}_{\mathrm{alg}}^i$ is small, then we stop the solver. Otherwise set $i := i + 1$ and return to step 2.

4.5 Basic properties

In this section, we collect properties that follows directly from the construction of our solver. We first present how the solver of Algorithm 1 induces an a posteriori estimator $\tilde{\eta}^i_{\text{alg}}$.

Definition 4.1. (Algebraic error estimator). Let $u^i_J \in V^p_J$ be arbitrary, and let $u^{i+1}_J \in V^p_J$ be the update at the end of one step of the solver introduced in Algorithm 1. We define the algebraic error estimator

$$\tilde{\eta}^i_{\text{alg}} := \left(\sum_{j=0}^{J} (\tilde{\lambda}^i_j \| \mathcal{K}^{\frac{1}{2}} \nabla \tilde{\rho}^i_j \|)^2 \right)^{\frac{1}{2}}. \tag{4.8}$$

We continue with several remarks.

Remark 4.2. (Localization on cells). The left-hand side of (4.5a) is posed solely on the element K_k in question. Below, in Section 7, we use it to deduce merely $d+1$ type of problems on the reference element $\hat{K}$ when $\mathcal{K} = \text{Id}$. In contrast, the right-hand side of (4.5a) evaluates the algebraic residual on the whole patch subdomain $\omega^{\boldsymbol{a}}_j$, using the extension operators of Section 4.3.

Remark 4.3. (Compact writing of the iteration update). Let $u^i_J \in V^p_J$. The update given in (4.7) writes equivalently as

$$u^i_{J,j} = u^i_J + \sum_{k=0}^{j} \tilde{\lambda}^i_k \tilde{\rho}^i_k \tag{4.9}$$

and in particular

$$u^{i+1}_J = u^i_J + \sum_{k=0}^{J} \tilde{\lambda}^i_k \tilde{\rho}^i_k. \tag{4.10}$$

The next lemma justifies the choice of the optimal step size given by (4.6).

Lemma 4.4. *(Levelwise optimal step size). Let $j \in \{1, \dots, J\}$, $u^i_{J,j-1} \in V^p_J$ be arbitrary, and let $\tilde{\rho}^i_j$ and $\tilde{\lambda}^i_j$ be given by (4.5) and (4.6), respectively. Then*

$$\tilde{\lambda}^i_j = \arg\min_{\lambda \in \mathbb{R}} \| \mathcal{K}^{\frac{1}{2}} \nabla \left(u_J - \left(u^i_{J,j-1} + \lambda \tilde{\rho}^i_j \right) \right) \|.$$

Proof. We write the algebraic error associated to $u^i_{J,j-1} + \lambda \tilde{\rho}^i_j$ as a function of λ

$$\begin{aligned} \| \mathcal{K}^{\frac{1}{2}} \nabla (u_J - (u^i_{J,j-1} + \lambda \tilde{\rho}^i_j)) \|^2 =& \| \mathcal{K}^{\frac{1}{2}} \nabla (u_J - u^i_{J,j-1}) \|^2 \\ & - 2\lambda \left(\mathcal{K} \nabla \left(u_J - u^i_{J,j-1} \right), \nabla \tilde{\rho}^i_j \right) + \lambda^2 \| \mathcal{K}^{\frac{1}{2}} \nabla \tilde{\rho}^i_j \|^2. \end{aligned}$$

The minimum of this quadratic function is given by the following formula (using (2.3)):

$$\tilde{\lambda}^i_j = \frac{\left(f, \tilde{\rho}^i_j \right) - \left(\mathcal{K} \nabla u^i_{J,j-1}, \nabla \tilde{\rho}^i_j \right)}{\| \mathcal{K}^{1/2} \nabla \tilde{\rho}^i_j \|^2}.$$

□

5 Main results

In this section, we collect our main results. Those results are obtained using a polynomial hierarchy such that $p_j = 1, 0 \leqslant j \leqslant J-1$, and $p_J = p$. We first begin by establishing an energy error decay formula that quantifies the progress achieved at each iteration of the solver of Algorithm 1. The identity below shows that the energy norm of the error diminishes with each update, with the total reduction of the form of orthogonal decomposition of the residual lifting contributions.

Theorem 5.1. (Error decrease formula). For $u_J^i \in V_J^p$, let $u_J^{i+1} \in V_J^p$ be given by one iteration of the solver of Algorithm 1. The following holds true

$$\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\|^2 = \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 - \sum_{j=0}^{J}(\tilde{\lambda}_j^i\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_j^i\|)^2. \tag{5.1}$$

Proof. This is the consequence of the line search (4.6), as in [13]. We go through the levels from finest to coarse while exploiting the relation between the level update and the optimal step-size on each level.

$$\begin{aligned}
\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\|^2 &\overset{(4.7)}{=} \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - (u_{J,J-1}^i + \tilde{\lambda}_J^i\tilde{\rho}_J^i))\|^2 \\
&\overset{(4.6)}{=} \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_{J,J-1}^i)\|^2 - (\tilde{\lambda}_J^i\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_J^i\|)^2 \\
&= \cdots = \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - (u_J^i + \tilde{\lambda}_0^i\tilde{\rho}_0^i))\|^2 - \sum_{j=1}^{J}(\tilde{\lambda}_j^i\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_j^i\|)^2 \\
&\overset{(4.4)}{=} \|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 - \sum_{j=0}^{J}(\tilde{\lambda}_j^i\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_j^i\|)^2.
\end{aligned}$$

□

We now state the main assumption that we use to prove Theorem 5.3.

Assumption 5.2. (Connection between levelwise residual lifting and its approximate counterpart). Let ρ_J^i and $\tilde{\rho}_J^i$ defined respectively by (3.5) and (4.5). Let λ_J^i and $\tilde{\lambda}_J^i$ be their corresponding optimal step size defined by (3.7) and (4.6). We introduce the ratio ξ_J^i:

$$\xi_J^i := \frac{\lambda_J^i\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_J^i\|^2}{(\tilde{\lambda}_J^i)^2\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_J^i\|^2}. \tag{5.2}$$

We assume that there exists a constant $0 \leqslant C(p) < +\infty$ such that $\xi_J^i \leqslant C(p)$. Thus we can write the following inequality

$$\lambda_J^i\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_J^i\|^2 \leqslant C(p)(\tilde{\lambda}_J^i)^2\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_J^i\|^2. \tag{5.3}$$

Following the framework of [13], we now state our main theorem, which holds under Assumption 5.2.

Theorem 5.3. (Contraction of the algebraic error) Let $u_J \in V_J^p$ be the unknown finite element solution of (2.3) and let $i \geqslant 0$ and $u_J^i \in V_J^p$ be arbitrary. Take u_J^{i+1} constructed from the

multilevel solver of Algorithm 1 with the polynomial hierarchy $p_j = 1, 1 \leqslant j \leqslant J-1$. Let Assumption 2.1 and either Assumption 2.2 or 2.3 hold. Suppose also Assumption 5.2. Then there holds

$$\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\| \leqslant \tilde{\alpha}\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|, \tag{5.4}$$

where $0 \leqslant \tilde{\alpha} < 1$ depends only on the space dimension d, the mesh shape regularity parameter $\kappa_{\mathcal{T}}$, the ratio of the largest and the smallest eigenvalue of the diffusion coefficient $\mathcal{K}$, the number of mesh levels J, and the constant $C(p)$.

Following Theorem 5.1, the estimator $\tilde{\eta}^i_{\text{alg}}$ is immediately a guaranteed lower bound on the algebraic error.

Theorem 5.4. (Guaranteed lower bound on the algebraic error). There holds

$$\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\| \geqslant \tilde{\eta}^i_{\text{alg}}.$$

Theorem 5.5. (Reliable and efficient bound on the algebraic error). Let the assumptions of Theorem 5.3 be satisfied. Then in addition to Lemma 5.4, there holds

$$\tilde{\eta}^i_{\text{alg}} \geqslant \tilde{\beta}\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|, \tag{5.5}$$

where $0 \leqslant \tilde{\beta} < 1$ is given by $\sqrt{1-\tilde{\alpha}^2}$ from (5.4).

As stated in Theorem 5.3, we assume that the constant $C(p)$ is bounded. If $C(p)$ were p-robust, one would consequently observe p-robust contraction. However, this behavior is not confirmed by our numerical experiments. A more detailed investigation of this relationship is provided in Appendix B and C. Theorem 5.5 allows to write $\tilde{\eta}^i_{\text{alg}}$ as a two-sided bound of the algebraic error thereby ensuring that the estimate is reliable and efficient. The two Theorems 5.3 and 5.5 are in fact equivalent. We use the error decrease formula of (5.1) to show this, following [13, Corollary 3].

Remark 5.6. (Equivalence between contraction and estimator efficiency). Let the assumptions of Theorem 5.3 be satisfied. Then (5.4) holds if and only if (5.5) holds.

Proof.

$$\begin{aligned}
&\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^{i+1})\|^2 \leqslant \tilde{\alpha}^2\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 \\
\overset{\substack{(5.1)\\(4.8)}}{\Leftrightarrow}&\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 - (\tilde{\eta}^i_{\text{alg}})^2 \leqslant \tilde{\alpha}^2\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 \\
\Leftrightarrow&(1-\tilde{\alpha})^2\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u_J^i)\|^2 \leqslant (\tilde{\eta}^i_{\text{alg}})^2.
\end{aligned}$$

□

Therefore, in Section 11, we will only prove Theorem 5.5.

6 Adaptive solver

Pursuing the idea of [13, Section 7], we propose a simple and practical approach that allows the solver described in Algorithm 1 to autonomously and adaptively choose the number of smoothing steps on each mesh level. The central idea of the adaptive variant is to perform additional post-smoothing steps, if necessary, on those levels that contribute most to the algebraic error. This

decision is based on the available a posteriori estimate of the algebraic error and follows a Dörfler-type marking criterion [7].

Algorithm 2: Adaptive multigrid solver using element solves and basis function extension

1. Initialize $u_J^0 \in V_J^p$ as the zero function and set $i := 0$.
2. Perform the following steps (a)–(d):
   (a) Construct $\tilde{\rho}_0^i$ using (4.4) and set
   $$\tilde{\rho}_{0,1}^i := \tilde{\rho}_0^i, \quad \tilde{\lambda}_{0,1}^i := 1, \quad \nu_0^i := 1, \quad u_{J,0}^i := u_J^i + \tilde{\lambda}_{0,1}^i \tilde{\rho}_{0,1}^i.$$
   (b) **For** $j = 1, \dots, J$:
      i. Set $\nu := 1$.
      ii. From $u_{J,j-1}^i$, construct $\tilde{\rho}_j^i$ and $\tilde{\lambda}_j^i$ using (4.5)–(4.6). Set
      $$\tilde{\rho}_{j,\nu}^i := \tilde{\rho}_j^i, \quad \tilde{\lambda}_{j,\nu}^i := \tilde{\lambda}_j^i, \quad u_{J,j,\nu}^i := u_{J,j-1}^i + \tilde{\lambda}_{j,\nu}^i \tilde{\rho}_{j,\nu}^i.$$
      While
      $$\nu < \nu_{\max} \quad \text{and}$$
      $$\left(\tilde{\lambda}_{j,\nu}^i \|\mathcal{K}^{1/2} \nabla \tilde{\rho}_{j,\nu}^i\|\right)^2 \geqslant \theta^2 \left(\sum_{k=0}^{j-1} \sum_{\ell=1}^{\nu_k^i} \left(\tilde{\lambda}_{k,\ell}^i \|\mathcal{K}^{1/2} \nabla \tilde{\rho}_{k,\ell}^i\|\right)^2 + \sum_{\ell=1}^{\nu-1} \left(\tilde{\lambda}_{j,\ell}^i \|\mathcal{K}^{1/2} \nabla \tilde{\rho}_{j,\ell}^i\|\right)^2 \right) \tag{6.1}$$
      Do:
      - Set $\nu := \nu + 1$.
      - From $u_{J,j,\nu-1}^i$, construct $\bar{\rho}_j^i$ and $\bar{\lambda}_j^i$ by (4.5)–(4.6)
      - Update
      $$\tilde{\rho}_{j,\nu}^i := \bar{\rho}_j^i, \quad \tilde{\lambda}_{j,\nu}^i := \bar{\lambda}_j^i, \quad u_{J,j,\nu}^i := u_{J,j,\nu-1}^i + \tilde{\lambda}_{j,\nu}^i \tilde{\rho}_{j,\nu}^i.$$
      Endwhile
      iii. Set $\nu_j^i := \nu$ and $u_{J,j}^i := u_{J,j,\nu}^i$.
   (c) Define the final update at iteration i as
   $$u_J^{i+1} := u_{J,J}^i \in V_J^p.$$
   (d) If $\tilde{\eta}_{\text{alg}}^i$ is below the user specified tolerance, then stop the solver. Otherwise, set $i := i + 1$ and return to step 2.

Remark 6.1. (Adaptive substep). If the adaptive substep in 2(b) of Algorithm 2 is skipped by setting $\nu_{\max} = 1$, we recover the non adaptive solver described in Algorithm 1. Otherwise, the algorithm continues with additional smoothing iterations while the algebraic error on mesh level j and solver iteration i as estimated by the left-hand side of (6.1) continues to be more important than the cumulative decrease already achieved during iteration i.

Remark 6.2. (Bulk-chasing criterion). The bulk-chasing (Dörfler's) marking criterion used above

in (6.1) is not essential. Other marking strategies, such as the maximal criterion, can also be employed.

7 Implementation with $d+1$ reference problems for $\mathcal{K} = \mathrm{Id}$

Let $\mathcal{K} = \mathrm{Id}$. In this section, we aim at giving further insight on how the cell procedure leads to a significant cost reduction in terms of memory and overall performance of Algorithm 1 with respect to that of Section 3.1. First, let us consider $\tilde{\rho}^i_{j,\boldsymbol{a}}|_{K_k}$ from (4.5a) and the decomposition in the basis of $V_j^{\boldsymbol{a}}(K_k) := \mathrm{span}(\phi^1_{j,\boldsymbol{a}}, \dots, \phi^{N_{j,\boldsymbol{a}}(K_k)}_{j,\boldsymbol{a}})$, in addition to the part fixed by the inhomogeneous Dirichlet condition on the boundary with already visited neighbors that we denote by g_k. This requests solution of the algebraic linear system that arises from (4.5a):

$$\sum_{l=1}^{N_{j,\boldsymbol{a}}(K_k)} (\tilde{\rho}^i_{j,\boldsymbol{a}})^l \left(\nabla\phi^l_{j,\boldsymbol{a}}, \nabla\phi^m_{j,\boldsymbol{a}}\right)_{K_k} = \left(f, E_k\phi^m_{j,\boldsymbol{a}}\right)_{\omega_j^{\boldsymbol{a}}} - \left(\nabla u^i_{J,j-1}, \nabla(E_k\phi^m_{j,\boldsymbol{a}})\right)_{\omega_j^{\boldsymbol{a}}} - (\nabla g_k, \nabla\phi^m_{j,\boldsymbol{a}})_{K_k} \quad \forall m = 1, \dots, N_{j,\boldsymbol{a}}(K_k). \tag{7.1}$$

By employing a reference to physical geometric mapping $\mathbf{T}_{s,K_k} : \hat{K}_s \to K_k$ and defining for all $m = 1, \dots, N_{j,\boldsymbol{a}}(K_k)$ the reference basis functions $\hat{\phi}^m_s \in \mathcal{P}_{p_j}(\hat{K}_s)$ such that $\hat{\phi}^m_s := \phi^m_{j,\boldsymbol{a}} \circ \mathbf{T}_{s,K_k}(\hat{x})$, one can write the following and equivalent algebraic linear system where the left hand side matrix is constructed on the domain of a reference element $\hat{K}_s$:

$$\sum_{l=1}^{N_{j,\boldsymbol{a}}(K_k)} (\tilde{\rho}^i_{j,\boldsymbol{a}})^l |\mathrm{det} J_{s,K_k}| (\mathbb{J}^{-1}_{s,K_k}\mathbb{J}^{-\mathrm{t}}_{s,K_k}\nabla\hat{\phi}^l_s, \nabla\hat{\phi}^m_s)_{\hat{K}_s} = \left(f, E_k\phi^m_{j,\boldsymbol{a}}\right)_{\omega_j^{\boldsymbol{a}}} - \left(\nabla u^i_{J,j-1}, \nabla(E_k\phi^m_{j,\boldsymbol{a}})\right)_{\omega_j^{\boldsymbol{a}}} - (\nabla g_k, \nabla\phi^m_{j,\boldsymbol{a}})_{K_k} \quad \forall m = 1, \dots, N_{j,\boldsymbol{a}}(K_k). \tag{7.2}$$

Here J_{s,K_k} is the Jacobian of the affine transformation $\mathbf{T}_{s,K_k}$. Crucially, we want to stress that in the case where the physical element K_k is merely a rotation, translation or dilation of the reference element K_s, i.e,

$$\mathbf{T}_{s,K_k}(\hat{X}) = \alpha_{s,K_k}\mathbb{Q}_{s,K_k}\hat{X} + \beta_{s,K_k}, \tag{7.3}$$

with $\mathbb{Q}_{s,K_k} \in \mathbb{R}^{d\times d}$ the rotation matrix of determinant 1, $\alpha_{s,K_k} \in \mathbb{R}$ the coefficient of dilatation and $\beta_{s,K_k} \in \mathbb{R}^d$ the translation vector, the matrix product $\mathbb{J}^{-1}_{s,K_k}\mathbb{J}^{-\mathrm{t}}_{s,K_k}$ simplifies into $|\mathrm{det}\mathbb{J}_{\mathrm{s,K_k}}|^{-\frac{2}{d}}$. Indeed, in that case, the Jacobian is

$$J_{s,K_k} = \alpha_{s,K_k}\mathbb{Q}_{s,K_k}. \tag{7.4}$$

Then, formula (7.2) reads as

$$|\mathrm{det}\mathbb{J}_{\mathrm{s,K_k}}|^{-\frac{2}{d}} 1 \sum_{l=1}^{N_{j,\boldsymbol{a}}(K_k)} (\tilde{\rho}^i_{j,\boldsymbol{a}})^l (\nabla\hat{\phi}^l_s, \nabla\hat{\phi}^m_s)_{\hat{K}_s} = \left(f, E_k\phi^m_{j,\boldsymbol{a}}\right)_{\omega_j^{\boldsymbol{a}}} - \left(\nabla u^i_{J,j-1}, \nabla(E_k\phi^m_{j,\boldsymbol{a}})\right)_{\omega_j^{\boldsymbol{a}}} - (\nabla g_k, \nabla\phi^m_{j,\boldsymbol{a}})_{K_k} \quad \forall m = 1, \dots, N_{j,\boldsymbol{a}}(K_k). \tag{7.5}$$

Therefore, one can store the Cholesky decomposition of the left hand side of $(d+1)$ matrices once and for all and employ it with the physical data information of the right hand side. Note that we use an index $s \in \{1, \dots, d+1\}$ in the above formulas, reflecting the different boundary conditions in (4.5a). While the reference simple $\hat{K}_s$ can remain the same, the dimension of the

problem ($N_{j,\boldsymbol{a}}(K_k)$) depends on the ordering of the element K_k within the procedure and the corresponding boundary conditions (see formula (4.5a)). As an example, in two space dimensions, there is a total of three distinct cases to take care of, this is illustrated in Figure 7. Consequently, we only need to compute three Cholesky decompositions in this case. In Section 9, we take advantage of this cost reduction to derive complexity formulas.

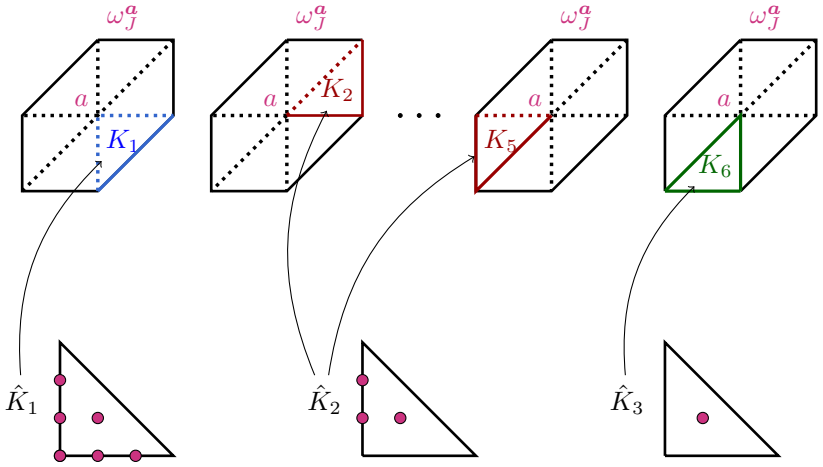


Figure 7: Visualization of the mapping to the reference triangles $\hat{K}_s$ under different boundary conditions. The representation shows the Lagrange degrees of freedom for $p_J = 3$.

8 Additionnal solvers for numerical comparisons

In this section, we introduce some additionnal multigrid solvers for numerical comparisons in Section 10 below.

Remark 8.1. (Block-Jacobi smoother: $\mathrm{MG}_{\mathrm{patch}}$). Using ρ_j^i of (3.5a) and λ_j^i of (3.7), as defined in Section 3.1, in the solver of Algorithm 1, is exactly the approach explored in [13]. We will refer this approach as being the $\mathrm{MG}_{\mathrm{patch}}$ solver later in Section 9.

Remark 8.2. (Diagonal Jacobi smoother with optimal step-size: $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}$). In order to make further comparisons later, we would like to introduce the following smoother in our setting. We introduce $\rho_j^{i,*} \in V_j^{p_j}$ defined by

$$\rho_j^{i,*} = \sum_{l=1}^{N_j} \rho_{j,l}^{i,*} \phi_j^l, \tag{8.1a}$$

where the coefficients $\rho_{j,l}^{i,*}, 1 \leqslant l \leqslant N_j$ are the solutions of

$$\rho_{j,l}^{i,*}(\mathcal{K}\nabla\phi_j^l, \nabla\phi_j^l) = (f, \phi_j^l) - (\mathcal{K}\nabla u_{J,j-1}^i, \nabla\phi_j^l). \tag{8.1b}$$

We recall that $\phi_j^l, 1 \leqslant l \leqslant N_j$ form a basis of $V_j^{p_j}$. Crucially, (8.1b) is one equation for one unknown $\rho_{j,l}^{i,*}$: this is a functional writing of the Jacobi (diagonal) smoother for the space $V_j^{p_j}$ on the level j. Employing the $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}$ solver means replacing $\tilde{\rho}_j^i$ of formula (4.5b) by $\rho_j^{i,*}$ of

formula (8.1a). Similarly to what has been done in Algorithm 1, an optimal step size can be defined by

$$\lambda_j^{i,*} := \frac{(f, \rho_j^{i,*}) - (\mathcal{K}\nabla u_{J,j-1}^i, \nabla \rho_j^{i,*})}{\|\mathcal{K}^{1/2}\nabla \rho_j^{i,*}\|^2}. \tag{8.2}$$

Remark 8.3. (Diagonal Jacobi Smoother: $\mathrm{MG_J}$). This is the classical Jacobi smoother where we set $p_j := p \quad \forall j \geqslant 1$, $p_0 = 1$ remains the same. The correction is given by (8.1a) without the use of optimal step-sizes.

9 Complexity of the solvers

We give in this section some more insight into the algorithmic complexity of the solver of Algorithm 1, those of Section 8, and their variants with additonnal number of smoothing steps. Recall the matrix writing of one iteration of the solver of Algorithm 1. Performing one iteration of the solver of Algorithm 1 would mean first solving the coarse problem

$$\tilde{R}_0^i := \mathbb{A}_0^{-1}(F_0 - \mathbb{A}_0 U_{J,0}^i), \tag{9.1a}$$

and then solving on the level j

$$U_{J,j}^i := U_{J,j-1} + \tilde{\lambda}_j^i \tilde{R}_j^i, \tag{9.1b}$$

where $\tilde{R}_j^i := \mathbb{R}_j(F_j - \mathbb{A}_j U_{J,j-1}^i)$, with the computation of $\tilde{\lambda}_j^i$ done by

$$\tilde{\lambda}_j^i := \frac{(\tilde{R}_j^i)^{\mathrm{t}} F_j - (\mathbb{A}_j U_{J,j-1}^i)^{\mathrm{t}} \tilde{R}_j^i}{(\mathbb{A}_j \tilde{R}_j^i)^{\mathrm{t}} \tilde{R}_j^i}, \tag{9.1c}$$

making use of intergrid operators between levels. In (9.1), we use the representation $\tilde{\rho}_j^i = \sum_{m=1}^{N_j} (\tilde{R}_j^i)_m \phi_j^m$. Here $\mathbb{R}_j$ is the approximate of the right inverse operator $\mathbb{A}_j^{-1}$ stemming from the element wise procedure of (4.5b).

We now turn to the computational complexity of the method. In particular, following the work of [13], we introduce a complexity formula that accounts for the number of floating-point operations performed after i_s iterations of the solver described in Algorithm 2. To enable a fair comparison, we also derive analogous complexity formulas for the solver of Algorithm 2, as well as for each solver variant numerically studied in Section 10. In particular, we obtain for the cellwise solver of Algorithm 2:

$$\text{nflops}_{\ \mathrm{MG}_{\mathrm{cell}}(0,\nu_j^i)} := \frac{|\mathcal{V}_0|^3}{3} + \sum_{j=1}^{J-1} \text{ndof}\left(V_j^1\right) + \sum_{k=1}^{3} \frac{\text{ndof}\left(V_J^{\boldsymbol{a}}(\hat{K}_k)\right)^3}{3} \tag{9.2a}$$

$$+ \sum_{i=1}^{i_s} \left[2\,|\mathcal{V}_0|^2 + \sum_{j=1}^{J-1} \nu_j^i \text{ndof}\left(V_j^1\right) + \nu_j^i \sum_{\boldsymbol{a}\in\mathcal{V}_J} \sum_{K_k \in \mathcal{T}_J^{\boldsymbol{a}}} 2\text{ndof}\left(V_J^{\boldsymbol{a}}(K_k)\right)^2 \right] \tag{9.2b}$$

$$+ \sum_{i=1}^{i_s} \sum_{j=1}^{J} \left[2\text{nnz}\left(\mathcal{I}_{j-1}^j\right) + 2\text{nnz}\left(\mathcal{I}_j^{j-1}\right) + 2\nu_j^i \text{nnz}\left(\mathbb{A}_j\right) + 3\nu_j^i \left(2\,\text{size}\left(\mathbb{A}_j\right)\right) \right]. \tag{9.2c}$$

For the patchwise solver of Remark 8.1, we have

$$\text{nflops}_{\,\mathrm{MG}_{\mathrm{patch}}(0,\nu_j^i)} := \frac{|\mathcal{V}_0|^3}{3} + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\frac{\text{ndof}\left(V_j^{\boldsymbol{a}}\right)^3}{3} \tag{9.3a}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\left[2\,|\mathcal{V}_0|^2 + \sum_{j=1}^{J}\nu_j^i\sum_{\boldsymbol{a}\in\mathcal{V}_j}2\text{ndof}\left(V_j^{\boldsymbol{a}}\right)^2\right] \tag{9.3b}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\sum_{j=1}^{J}\left[2\text{nnz}\left(\mathcal{I}_{j-1}^{j}\right) + 2\text{nnz}\left(\mathcal{I}_{j}^{j-1}\right) + 2\nu_j^i\text{nnz}\left(\mathbb{A}_j\right) + 3\nu_j^i\left(2\,\text{size}\left(\mathbb{A}_j\right)\right)\right]. \tag{9.3c}$$

The computational cost of the multigrid Jacobi method with ν number of post-smoothing steps described in Remark 8.3 $\mathrm{MG_J}(0,\nu)$:

$$\text{nflops}_{\,\mathrm{MG_J}(0,\nu)} := \frac{|\mathcal{V}_0|^3}{3} + \sum_{j=1}^{J}\text{ndof}\left(V_j^{p_j}\right) \tag{9.4a}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\left[2\,|\mathcal{V}_0|^2 + \sum_{j=1}^{J}\nu\text{ndof}\left(V_j^{p_j}\right)\right] \tag{9.4b}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\sum_{j=1}^{J}\left[2\text{nnz}\left(\mathcal{I}_{j-1}^{j}\right) + 2\text{nnz}\left(\mathcal{I}_{j}^{j-1}\right)\right]. \tag{9.4c}$$

The computational cost of the Jacobi smoother of Remark 8.2 with ν additionnal post-smoothing steps and an optimal step-size $\mathrm{MG_J^{opt}}(0,\nu)$:

$$\text{nflops}_{\,\mathrm{MG_J^{opt}}(0,\nu)} := \frac{|\mathcal{V}_0|^3}{3} + \sum_{j=1}^{J}\text{ndof}\left(V_j^{p_j}\right) \tag{9.5a}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\left[2\,|\mathcal{V}_0|^2 + \sum_{j=1}^{J}\nu\text{ndof}\left(V_j^{p_j}\right)\right] \tag{9.5b}$$

$$+\sum_{i=1}^{i_{\mathrm{s}}}\sum_{j=1}^{J}\left[2\text{nnz}\left(\mathcal{I}_{j-1}^{j}\right) + 2\text{nnz}\left(\mathcal{I}_{j}^{j-1}\right) + 2\nu\text{nnz}\left(\mathbb{A}_j\right) + 3\nu\left(2\,\text{size}\left(\mathbb{A}_j\right)\right)\right]. \tag{9.5c}$$

The formulas (9.2), (9.3), (9.4), and (9.5) are derived under the assumption that the global coarse matrix is factorized (for a matrix of size n, this cost is estimated as $1/3n^3$). For mesh levels $1 \leqslant j \leqslant J-1$, the cost of inverting all local systems (4.5a) associated with a single degree of freedom is given by the total number of degrees of freedom in V_j^1, this is in fact equivalent to the computation of the diagonal Jacobi smoother globally, see formulas (9.4) and (9.5). For the last level J we only need the factorization of three local cell matrices (9.2a) instead of all the patch matrices (9.3a), see Figure 7. For (9.2) and (9.3), the cost of solves by forward and backward substitutions is estimated as $2n^2$, while it is linear in terms of dofs for the Jacobi-type variant (9.4) (9.5). The construction of intergrid operators $\mathcal{I}_{j-1}^{j} : V_{j-1}^{p_{j-1}} \to V_j^{p_j}$ is estimated at twice the number of nonzeros in the associated interpolation matrix and remain the same for all the solvers (first two terms of equations labeled "c"). The evaluation of the optimal step-sizes $\tilde{\lambda}_j^i$, as defined in (4.6), requires a cost equal to twice the number of nonzeros of the stiffness matrix $\mathbb{A}_j$ at the corresponding level, together with three inner products (last two terms of (9.2c),(9.3c), (9.5c)).

We want to stress that formulas (9.2)–(9.5) do not account for the operations required to evaluate the right-hand sides of the problems. While such evaluations may contribute to the overall flop count especially for $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$ and $\mathrm{MG}_{\mathrm{J}}(0,1)$, their impact depends heavily on the particular implementation and are dominated by the other components. We aim to emphasize that the computational cost of the solver of Algorithm 1 is very close to the one of the classical Jacobi multigrid method of (9.4) and the step-size variant (9.5). On the level J, the additional cost arising from the sequential patch procedure at level J is given by the last term in equations (9.2a) and (9.2b). While the cost of the last term (9.2a) is negligeable, the cost of the last term in (9.2b) is indeed superior to the one required for the diagonal Jacobi smoother. However, we want to stress that the formula (9.2) allows for a parallel implementation. In that regard the question of efficiency in terms of number of floating-points operations is less relevant. In terms of memory requirements, the solver presented in Algorithm 1 represents a significant improvement over the work in [13] (Section 3.1), as it no longer needs to operate with local matrices constructed on patches.

10 Numerical experiments

In this section, we analyze the performance in terms of iterations and cost of the proposed solver. We will consider problem (2.3) with $\mathcal{K} = 1$ and in (10.3), an inhomogenous boundary conditions given by the exact solution. The test cases where the exact solutions is given are the following:

$$\text{sine}: \quad \Omega := (-1,1)^2, u(x,y) := \sin(2\pi x)\sin(2\pi y), \tag{10.1}$$

$$\text{peak}: \quad \Omega := (0,1)^2, u(x,y) := (x-1)y(y-1)\mathrm{e}^{-100\left((x-0.5)^2-(y-0.117)^2\right)}, \tag{10.2}$$

$$\text{L-shape}: \quad \Omega := (-1,1)^2 \backslash ([0,1]\times[-1,0]), u(r,\varphi) := r^{2/3}\sin(2\varphi/3). \tag{10.3}$$

We consider mesh hierarchies obtained by successive uniform refinement using the newest vertex bisection method on a coarse Delaunay triangulation mesh. We study the convergence of the solver of Algorithm 1 and its adaptive variant of Algorithm 2 in terms of number of iterations and cost. We stop the solver when the algebraic error $\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J^i - u_J)\|$ drops below 10^{-5}.

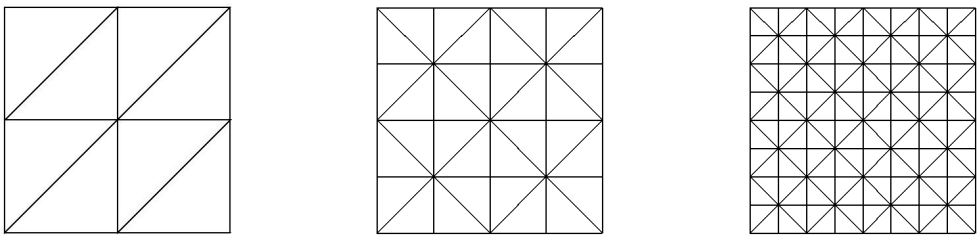


Figure 8: Illustration of an initial grid $\mathcal{T}_0$ (left) with its refinement $\mathcal{T}_1$ (middle) and $\mathcal{T}_2$ (right) obtained by two uniform iterations of the newest vertex bisection method.

10.1 Convergence in terms of number of iterations

We first present numerical results focusing exclusively on the number of iterations required by the proposed solver (Algorithm 1) to achieve convergence. The behavior of the method is compared

with the other solvers detailed in Section 8.

Sine Problem

J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$	J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$
4	1	4225	19	19	19	11	5	1	4225	20	20	20	20
	2	16641	31	19	32	19		2	16641	35	20	34	div
	3	37249	47	21	59	37		3	37249	52	20	70	36
	4	66049	66	20	101	160		4	66049	82	20	77	div
	5	103041	70	20	127	div		5	103041	84	20	240	div
	6	148225	74	20	893	div		6	148225	86	20	856	div
	7	201601	92	20	+3000	div		7	201601	135	20	+3000	div

Peak Problem

J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$	J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$
4	1	4225	13	10	10	8	5	1	4225	10	10	10	9
	2	16641	35	13	21	24		2	16641	33	13	20	26
	3	37249	39	14	36	div		3	37249	40	14	47	div
	4	66049	47	14	47	div		4	66049	47	14	66	div
	5	103041	51	14	94	div		5	103041	51	14	101	div
	6	148225	60	14	393	div		6	148225	62	14	403	div
	7	201601	52	14	1581	div		7	201601	54	14	1635	div

L-shape Problem

J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$	J	p	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$	$\mathrm{MG}_{\mathrm{J}}(p,p)$
4	1	3201	33	33	33	20	5	1	3201	33	33	33	25
	2	12545	61	34	63	102		2	12545	59	33	64	104
	3	28033	66	36	116	div		3	28033	66	33	58	div
	4	49665	66	36	145	div		4	49665	103	33	56	div
	5	77441	130	36	320	div		5	77441	77	33	116	div
	6	111361	120	36	2484	div		6	111361	98	33	2485	div
	7	151425	246	36	+3000	div		7	151425	190	33	+3000	div

Table 1: Number of iterations i_s needed for convergence. The solver of Algorithm 1 is compared with the others solvers detailed in Section 8. Different test cases with different polynomial degrees p_J and number of mesh levels J. For $J = 4$, the initial grid corresponds to the first level of refinement when $J = 5$.

Table 1 reports the number of iterations required to reach convergence for each solver across the different test cases. As shown in [13], the solver $\mathrm{MG}_{\mathrm{patch}}(0,1)$ exhibits p-robust behavior; consequently, the number of iterations remains unchanged as the polynomial degree p increases. In contrast, the standard multigrid Jacobi solver $\mathrm{MG}_{\mathrm{J}}(p,p)$ is observed to not converge for $p \geqslant 3$, a behavior that is well documented in the literature (see the references in Section 1). For the optimized multigrid Jacobi solver $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$, this divergence is no longer observed. However, the number of iterations required for convergence increases significantly with p, already exceeding 2000 iterations for $p = 6$ in the third test case. By contrast, the solver $\mathrm{MG}_{\mathrm{cell}}(0,1)$ presented in Algorithm 1, which has a comparable computational cost, does not exhibit this behavior. Overall, while an increase in the number of iterations with respect to p is observed, this growth remains moderate. Moreover, for certain test cases, such as the Peak problem, almost p-robust behavior can be observed.

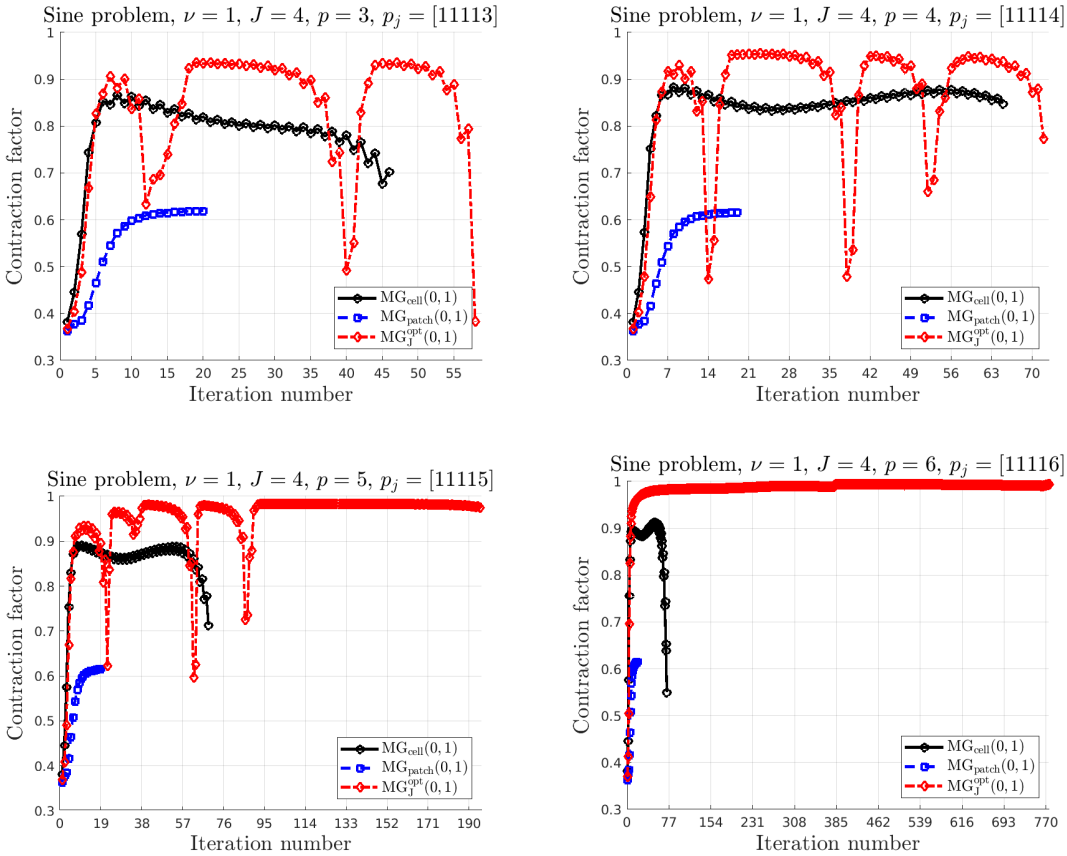


Figure 9: Contraction factors over the iterations for different polynomial degrees and mesh levels with $J = 4$. The solver of Algorithm 1 is compared with the other solvers introduced in Section 9 for the sine problem (10.1) with different polynomial degrees.

Figure 9 reports the contraction factors of the solvers $\mathrm{MG}_{\mathrm{cell}}(0,1)$, $\mathrm{MG}_{\mathrm{patch}}(0,1)$, as well as $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$. Consistently with the observations reported in Table 1, we observe that the contraction factor of $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,1)$ approaches one rapidly as the polynomial degree p increases, whereas the contraction factor of $\mathrm{MG}_{\mathrm{cell}}(0,1)$ is significantly less affected by p. Furthermore, for all solvers, the contraction factor is significantly smaller during the initial iterations. This behavior can be attributed to the exact coarse solve, which efficiently eliminates the lowest-order components of the algebraic error.

10.2 Convergence in terms of cost

In this section, we investigate the number of floating point operations required for convergence as per Section 9 and the influence of the number of smoothing steps on the overall count.

Table 2: Estimated number of floating point operations to match the stopping criterion given for the solvers presented in Section 8. Different test cases, different polynomial degrees p and different number of mesh levels J.

Sine Problem, $p_J = 6$

J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$	J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$
4	1	37249	$9.95e^8$	$9.16e^8$	$7.89e^9$	div	5	1	148225	$4.03e^9$	$3.51e^9$	$3.01e^{10}$	div
	5	37249	$7.78e^8$	$9.92e^8$	$6.20e^9$	div		5	148225	$2.69e^9$	$4.04e^9$	$2.28e^{10}$	div
	10	37249	$6.49e^8$	$1.41e^9$	$4.93e^9$	div		10	148225	$2.60e^9$	$5.74e^9$	$1.78e^{10}$	div
	15	37249	$7.56e^8$	$1.94e^9$	$5.30e^9$	div		15	148225	$2.65e^9$	$7.88e^9$	$1.87e^{10}$	div
	20	37249	$7.44e^8$	$2.04e^9$	$4.87e^9$	div		20	148225	$2.43e^9$	$8.29e^9$	$1.76e^{10}$	div

L-shape Problem, $p_J = 5$

J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$	J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$
4	1	19521	$4.60e^8$	$4.08e^8$	$1.23e^9$	div	5	1	77441	$1.43e^9$	$1.73e^9$	$1.39e^{10}$	div
	5	19521	$3.25e^8$	$4.58e^8$	$1.12e^9$	div		5	77441	$1.36e^9$	$1.23e^9$	$9.97e^9$	div
	10	19521	$3.46e^8$	$5.96e^8$	$5.99e^8$	div		10	77441	$1.39e^9$	$1.13e^9$	$6.08e^9$	div
	15	19521	$3.23e^8$	$7.38e^8$	$9.58e^8$	div		15	77441	$1.44e^9$	$1.53e^9$	$7.17e^9$	div
	20	19521	$3.76e^8$	$9.56e^8$	$6.35e^8$	div		20	77441	$1.55e^9$	$1.74e^9$	$9.17e^9$	div

L-shape Problem, $p_J = 6$

J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{J}(0,\nu)$	J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,\nu)$	$\mathrm{MG}_{\mathrm{patch}}(0,\nu)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{J}(0,\nu)$
4	1	28033	$1.03e^9$	$9.40e^8$	$1.59e^{10}$	div	5	1	111361	$4.15e^9$	$3.92e^9$	$6.58e^{10}$	div
	5	28033	$7.68e^8$	$1.07e^9$	$9.21e^9$	div		5	111361	$3.08e^9$	$4.36e^9$	$3.79e^{10}$	div
	10	28033	$7.29e^8$	$1.37e^9$	$7.04e^9$	div		10	111361	$2.93e^9$	$5.59e^9$	$2.88e^{10}$	div
	15	28033	$7.78e^8$	$1.68e^9$	$7.96e^9$	div		15	111361	$3.12e^9$	$6.85e^9$	$3.27e^{10}$	div
	20	28033	$7.43e^8$	$2.15e^9$	$7.16e^9$	div		20	111361	$3.36e^9$	$8.78e^9$	$2.94e^{10}$	div

L-shape Problem, $p_J = 7$

J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$	J	ν	DoFs	$\mathrm{MG}_{\mathrm{cell}}(0,1)$	$\mathrm{MG}_{\mathrm{patch}}(0,1)$	$\mathrm{MG}^*_J(0,\nu)$	$\mathrm{MG}_{\mathrm{J}}(0,\nu)$
4	1	38081	$1.96e^9$	$4.15e^9$	div	div	5	1	151425	$8.94e^9$	$2.13e^{10}$	div	div
	5	38081	$2.22e^9$	$1.98e^9$	div	div		5	151425	$8.33e^9$	$1.19e^{10}$	div	div
	10	38081	$2.80e^9$	$1.92e^9$	$8.98e^{10}$	div		10	151425	$8.39e^9$	$1.15e^{10}$	$9.58e^{11}$	div
	15	38081	$3.39e^9$	$1.78e^9$	$1.13e^{11}$	div		15	151425	$7.64e^9$	$1.38e^{10}$	$3.12e^{12}$	div
	20	38081	$4.29e^9$	$1.83e^9$	$9.29e^{10}$	div		20	151425	$8.82e^9$	$1.51e^{10}$	$9.36e^{11}$	div

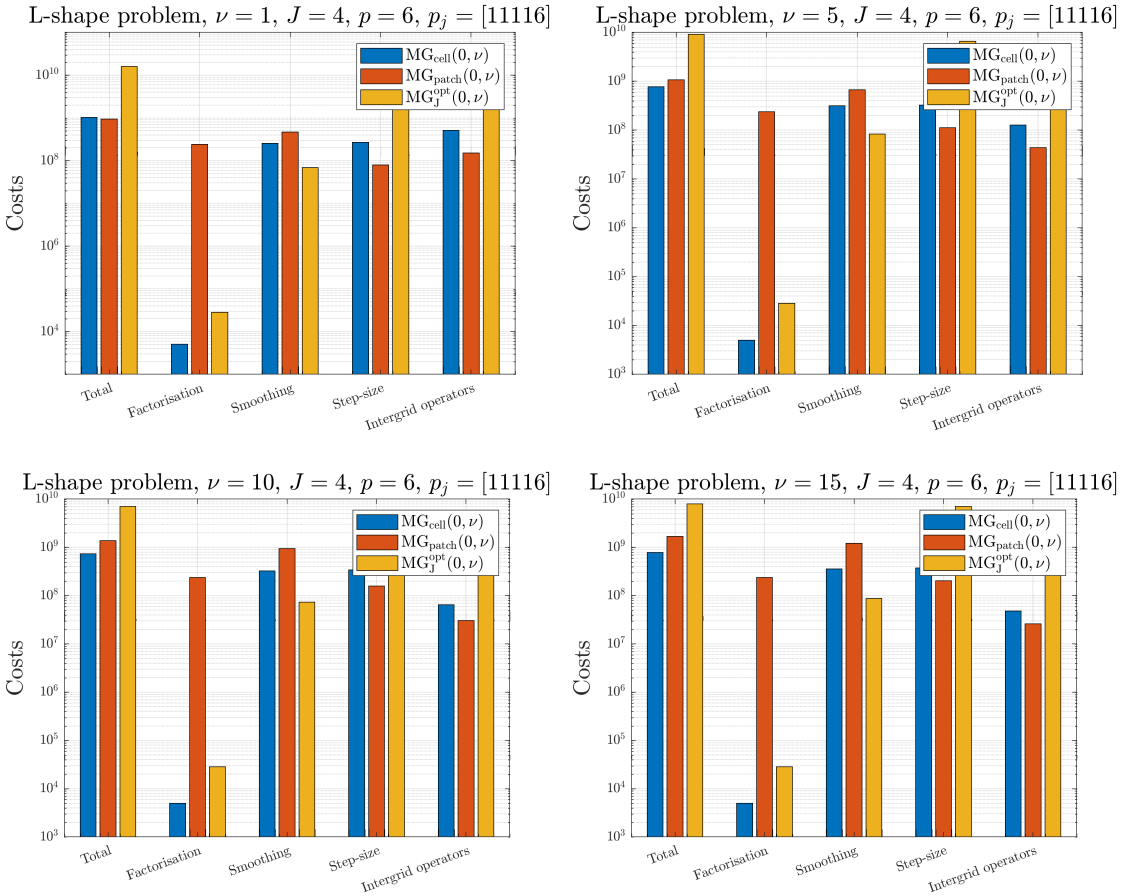


Figure 10: Cost comparison between the solvers $\mathrm{MG}_{\mathrm{cell}}(0,\nu)$, $\mathrm{MG}_{\mathrm{patch}}(0,\nu)$, and $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,\nu)$, where the total cost (shown in the leftmost column) is decomposed into different contributions appearing in formulas of Section 9. Ilustrative example with the L-shape problem (10.3) with $J = 4, p = 6$, and different numbers of smoothing steps.

In Figure 10, the total computational cost is decomposed into its different components, indicated on the abscissa and corresponding to the terms appearing in the cost formulas. The label "Factorisation" denotes the cost associated with the Cholesky decomposition, "Smoothing" refers to the cost of applying the approximate inverse of the operator to the residual, while "Step-size" counts (9.5) and "Intergrid Operators" are for restriction and prolongation. One can observe that $\mathrm{MG}_{\mathrm{cell}}(0,\nu)$ and $\mathrm{MG}_{\mathrm{patch}}(0,\nu)$ exhibit total costs of the same order of magnitude, whereas the total cost of $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,\nu)$ is significantly higher. A crucial observation is that $\mathrm{MG}_{\mathrm{cell}}(0,\nu)$ does not incur a high factorisation cost. The reason is, as discussed in Section 7, that the procedure of Algorithm 1 no longer requires the factorisation of all the patch matrices (second term of (9.3a)).

Although this is not directly reflected in the total cost, it represents a major improvement, as factorisation is the only component of the solver cost that cannot be expressed solely in terms of matrix–vector products that are particularly well suited for efficient and highly parallel implementations on modern architectures. Moreover, factorisation is also the most dominant

contribution to memory requirements, which are known to constitute a major bottleneck for solvers and preconditionners on modern architectures. One can also observe the influence of adding more smoothing steps and its influence on the different costs. Adding more smoothing steps reduces the use of intergrid operators and therefore reduces the associated cost, this is particulary visible for $\mathrm{MG}_{\mathrm{J}}^{\mathrm{opt}}(0,\nu)$, while it plays a less significant role for the other solvers.

10.3 Adaptive variant of the solver

We now investigate the behavior of the solver introduced in Algorithm 2. To compare the solver's performance across different configurations, we make use of the estimated number of floating-point operations given in (9.2), and additionally introduce the number of global synchronizations linked to the step-size optimisation (9.1c):

$$\text{sync} := i_s + \sum_{i=1}^{i_s} \sum_{j=1}^{J} \nu_j^i. \tag{10.4}$$

10.4 Influence of the parameter θ

In this section, we report the cumulated number of smoothing steps at each level for different choices of θ. We use the solver of Algorithm 1 for comparison. In Figure 11 we look at different setting of the sine problem where the polynomial degree p varies and investigate the influence of the choice of the polynomial hierarchy p_j for $j = 1, ..., J-1$.

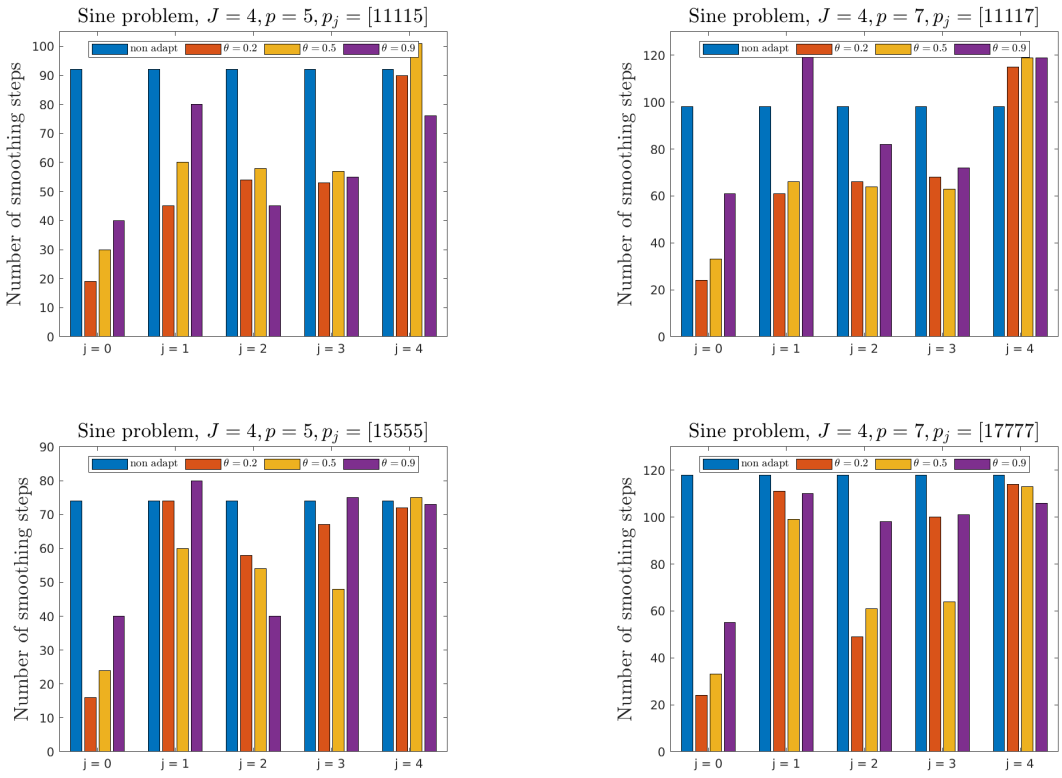


Figure 11: Number of smoothing steps per level for the sine problem, number of mesh levels $J = 4$, polynomial degree $p = 5$ on the left and $p = 7$ on the right. On the top, $p_j = 1$ for $j = 1, ..., J - 1$, and on the bottom $p_j = p$ for $j = 1, ..., J$. For the adaptive variant, $\nu_{\max}$ is set to 5, for the non adaptive variant, it is $\nu = 1$.

In Figure 11, we observe that the adaptive procedure, for all choices of θ, yields an overall reduction in the number of smoothing steps per level when compared to the non-adaptive variant. For mesh hierarchies satisfying $p_j = 1, j \leqslant J - 1$, the computational effort is predominantly concentrated on level J. This behavior can be attributed to the fact that the cell solve procedure is more effective at damping the high-order components of the algebraic error than the standard diagonal Jacobi smoother. Indeed, as shown in the bottom panel of Figure 11, the distribution of smoothing steps becomes significantly more homogeneous across levels, while still achieving a reduction in the total number of smoothing steps relative to the non-adaptive approach.

10.5 Performance of the adaptive solver

We now present a set of results intended to provide further insight into the efficiency of the adaptive procedure described in Algorithm 2. In particular, we simultaneously examine the influence of the choice of the parameters θ and $\nu_{\max}$ on the number of iterations, the number of synchronizations defined in (10.4), and the number of floating-point operations estimated using the formulas introduced in Section 9.

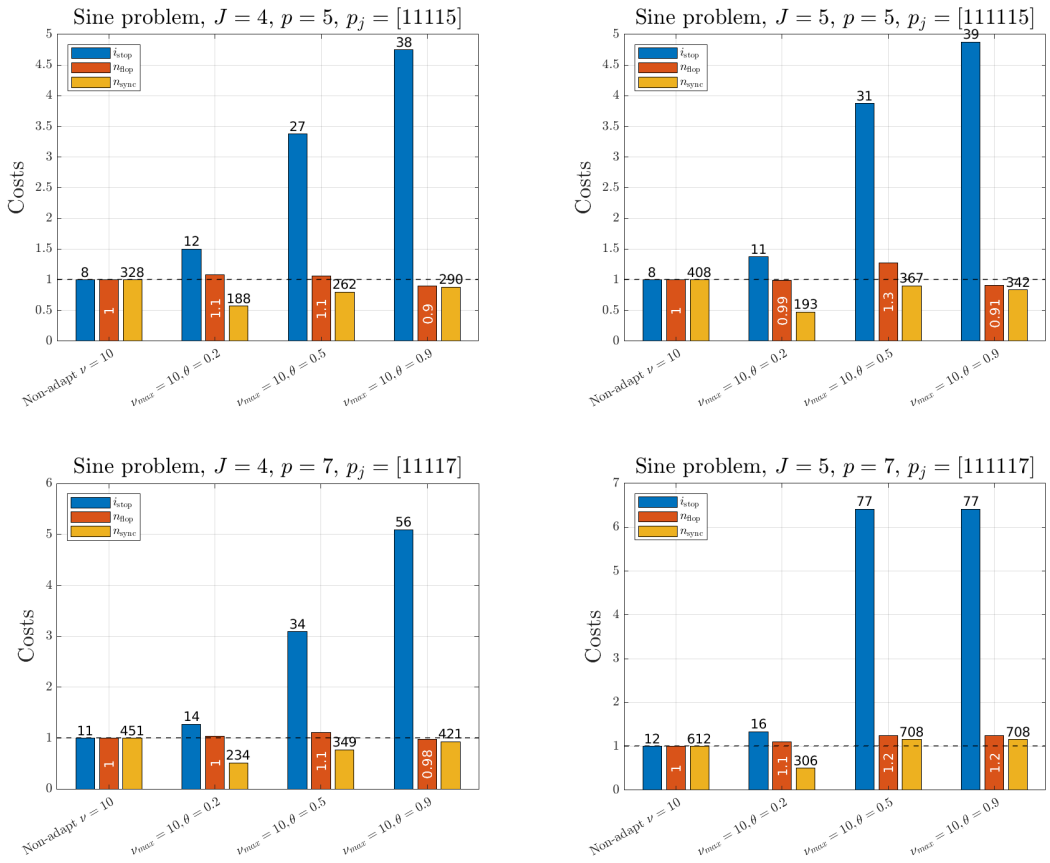


Figure 12: Comparison between a fixed number of smoothing steps ν on all levels (Algorithm 1) and the adaptive variant with different values of θ (Algorithm 2). Number of i_s, floating point operations given by (9.2), and number of synchronisations (10.4).

Figure 12 illustrates the influence of the parameter θ on the number of iterations i_s required for convergence, the total number of synchronisations (10.4), and the overall cost given by (9.2). As already observed in Figure 11, these results show that the adaptive procedure is effective in reducing the total number of smoothing steps: the values of sync are smaller for $\theta > 0$, at the expense of a slight increase in i_s.

In terms of floating-point operations, one can observe that the adaptive procedure can lead to performance gains, with cost reductions of up to 10%, although the outcome depends on the choice of θ. This sensitivity comes from the fact that the `While` condition in the adaptive procedure of Algorithm 2 bases its decision solely on the current estimate of the algebraic error, without accounting for the cost of a smoothing step. As a consequence, the solver tends to produce ν_j^i such that $\nu_j^i \geqslant \nu_j^i$ for all $1 \leqslant j \leqslant J-1$. While the total number of smoothing steps is reduced, this behavior may still result in a higher overall computational cost. This observation suggests that the `While` condition of (6.1) should be redesigned to explicitly incorporate the cost of smoothing. This is possible because the cost of a smoothing is known *a priori*.

11 Proofs

Similarly to [13], our strategy to prove contraction stated in Theorem 5.3 is to analyze the levelwise contributions of the uncomputable exact hierarchically constructed algebraic error $\bar{\rho}^i_{J,\mathrm{alg}} := u_J - u^i_J \in V^p_J$. To this end, we employ a polynomial-degree-robust stable decomposition that is obtained by combining the one of Schöberl *et al.* [20] that yields a one level p-stable decomposition with the work of Xu, Chen, and Nochetto [22] on h-stable multilevel decomposition for piecewise linear functions to first establish a connection between $\bar{\rho}^i_j$ and ρ^i_j, and then use Assumption 5.2 to link ρ^i_j to $\bar{\rho}^i_j$.

First, following the work done in [13], we recall the construction of $\bar{\rho}^i_{J,\,\mathrm{alg}} = u^i_J - u_J \in V^p_J$ for a given $u^i_J \in V^p_J$, i.e.,

$$\bar{\rho}^i_{J,\mathrm{alg}} := \bar{\rho}^i_0 + \sum_{j=1}^{J} \bar{\rho}^i_j,$$

where $\tilde{\rho}^i_0 = \bar{\rho}^i_0 \in V^1_0$ is the solution to the global lowest-order residual problem on the coarsest mesh

$$\left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla v_0\right) = (f, v_0) - \left(\mathcal{K}\nabla u^i_J, \nabla v_0\right) \quad \forall v_0 \in V^1_0,$$

and, for $j = 1 \leqslant J$, $\bar{\rho}^i_j \in V^{p_j}_j$ are the solutions of

$$\left(\mathcal{K}\nabla\bar{\rho}^i_j, \nabla v_j\right) = (f, v_j) - \left(\mathcal{K}\nabla u^i_J, \nabla v_j\right) \quad - \sum_{k=0}^{j-1} \left(\mathcal{K}\nabla\bar{\rho}^i_k, \nabla v_j\right) \quad \forall v_j \in V^{p_j}_j.$$

This means that the construction $\bar{\rho}^i_{J,\,\mathrm{alg}}$ satisfies

$$\left(\mathcal{K}\nabla\bar{\rho}^i_{J,\mathrm{alg}}, \nabla v_J\right) = (f, v_J) - \left(\mathcal{K}\nabla u^i_J, \nabla v_J\right) \quad \forall v_J \in V^p_J,$$

in accordance with the fact that $\bar{\rho}^i_{J,\mathrm{alg}} = u_J - u^i_J$. Moreover, there holds

$$\left(\mathcal{K}\nabla\bar{\rho}^i_j, \nabla\bar{\rho}^i_k\right) = 0, \quad \text{for } 0 \leqslant k, j \leqslant J, j \neq k.$$

This leads to the orthogonal decomposition of the error between u_J and u^i_J as

$$\|\mathcal{K}^{\frac{1}{2}}\nabla\left(u_J - u^i_J\right)\|^2 = \|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{J,\,\mathrm{alg}}\|^2 = \sum_{j=0}^{J} \|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_j\|^2. \tag{11.1}$$

Lemma 11.1. *(Connection between local contributions and levelwise correction). Let ρ^i_j given by (3.5a), $j \in \{1, \ldots, J\}$. We have*

$$\begin{aligned}\sum_{\boldsymbol{a}\in\mathcal{V}_j} \|\mathcal{K}^{\frac{1}{2}}\nabla\rho^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j} &\overset{(3.5\mathrm{b})}{=} \sum_{\boldsymbol{a}\in\mathcal{V}_j} \{(f, \rho^i_{j,\boldsymbol{a}})_{\omega^{\boldsymbol{a}}_j} - (\mathcal{K}\nabla u^i_{J,j-1}, \nabla\rho^i_{j,\boldsymbol{a}})_{\omega^{\boldsymbol{a}}_j}\} \\ &\overset{(3.5\mathrm{a})}{=} (f, \rho^i_j) - (\mathcal{K}\nabla u^i_{J,j-1}, \nabla\rho^i_j) \overset{(3.7)}{=} \lambda^i_j \|\mathcal{K}^{\frac{1}{2}}\nabla\rho^i_j\|^2.\end{aligned} \tag{11.2}$$

Proof. The first equality follows from using $\rho^i_{j,\boldsymbol{a}}$ as a test function in (3.5b). The second from summing over all patches, using (3.5a). The third follows directly from using the definition of λ^i_j in (3.7). □

Lemma 11.2. *(Estimation of $\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|$ by local contributions). Let $\rho_{j,\boldsymbol{a}}^i$ and ρ_j^i for $j\in\{1,\dots,J\}$, $\boldsymbol{a}\in\mathcal{V}_j$, be respectively given by (3.5b) and (3.5a). Then there holds*

$$\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|^2 \leqslant (d+1)\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_{j,\boldsymbol{a}}^i\|_{\omega_j^{\boldsymbol{a}}}^2 \tag{11.3}$$

Proof. Since $\rho_j^i=\sum_{\boldsymbol{a}\in\mathcal{V}_j}\rho_{j,\boldsymbol{a}}^i$, the inequality $\left|\sum_{k=1}^{d+1}a_k\right|^2\leqslant(d+1)\sum_{k=1}^{d+1}|a_k|^2$ leads to

$$\begin{aligned}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|^2 &= \sum_{K\in\mathcal{T}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|_K^2 = \sum_{K\in\mathcal{T}_j}\Big\|\sum_{\boldsymbol{a}\in\mathcal{V}_K}\mathcal{K}^{\frac{1}{2}}\nabla\rho_{j,\boldsymbol{a}}^i\Big\|_K^2\\ &\leqslant (d+1)\sum_{K\in\mathcal{T}_j}\sum_{\boldsymbol{a}\in\mathcal{V}_K}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_{j,\boldsymbol{a}}^i\|_K^2\\ &= (d+1)\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_{j,\boldsymbol{a}}^i\|_{\omega_j^{\boldsymbol{a}}}^2.\end{aligned}$$

□

Remark 11.3. (Lower bound on the optimal step-sizes). Note that (11.2) together with (11.3) and the definition $\lambda_j^i=1$ when $\rho_j^i=0$ or $j=0$ lead to

$$\lambda_j^i\geqslant\frac{1}{d+1},\quad 0\leqslant j\leqslant J. \tag{11.4}$$

Lemma 11.4. *(Consequence of Assumption 5.2). Let ρ_J^i and λ_J^i be given by their definition in Section 3.1. Let $\tilde{\rho}_J^i$ and $\tilde{\lambda}_J^i$ be given by the definition of Algorithm 1 with the polynomial hierarchy $p_j=1, 1\leqslant j\leqslant J-1$. Then there holds*

$$\sum_{j=1}^{J}\lambda_j^i\|\mathcal{K}^{\frac{1}{2}}\nabla\rho_j^i\|^2\leqslant C(p)\sum_{j=1}^{J}(\tilde{\lambda}_j^i)^2\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}_j^i\|^2. \tag{11.5}$$

Proof. We sum over all mesh levels and use Assumption 5.2 for the level J. □

We recall the following Lemma, introduced in [12, Proposition 7.6] that will be used in the proof of Theorem 5.5. This result is obtained by combining the work of Schöberl *et al.* [20] that yields a one level p-robust decomposition and the work of Xu, Chen, and Nochetto [22] on multilevel stable decomposition for piecewise linear functions.

Lemma 11.5. *(p-robust multilevel stable decomposition). Let $v_J\in V_J^p$. Under assumption 2.1, there exists a decomposition such that*

$$v_J = v_0+\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}v_{j,\boldsymbol{a}},\quad v_0\in V_0^1, v_{j,\boldsymbol{a}}\in V_j^{\boldsymbol{a}}, \tag{11.6a}$$

stable as

$$\|\nabla v_0\|^2+\sum_{j=1}^{J}\|\nabla v_{j,\boldsymbol{a}}\|_{\omega_j^{\boldsymbol{a}}}^2\leqslant C_S\|\nabla v_J\|^2, \tag{11.6b}$$

where C_S depends only on the space dimension d and the mesh shape regularity parameter κ_τ.

Proof of Theorem 5.5 (Estimator efficiency).
Let us consider the exact hierarchically construsted algebraic residual lifting $\bar{\rho}^i_{J,\mathrm{alg}}$ that yields $\|\mathcal{K}^{\frac{1}{2}}\nabla(u_J - u^i_J)\| = \|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{J,\mathrm{alg}}\|$, see (11.1). We begin by using Lemma 11.5 which allows the decomposition

$$\bar{\rho}^i_{J,\mathrm{alg}} = \bar{c}^i_0 + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\bar{\rho}^i_{j,\boldsymbol{a}}, \quad \bar{c}^i_0 \in V^1_0, \bar{\rho}^i_{j,\boldsymbol{a}} \in V^{\boldsymbol{a}}_j, \tag{11.7a}$$

$$\left\|\nabla\bar{c}^i_0\right\|^2 + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left\|\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right\|^2_{\omega^{\boldsymbol{a}}_j} \leqslant C^2_{\mathrm{S}}\left\|\nabla\bar{\rho}^i_{J,\mathrm{alg}}\right\|^2. \tag{11.7b}$$

Taking into account the variations of the diffusion coefficient $\mathcal{K}$, we have

$$\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{c}^i_0\|^2 + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j} \leqslant C^2_{\mathrm{S},\mathcal{K}}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{J,\mathrm{alg}}\|^2, \tag{11.8}$$

where the constant $C^2_{\mathrm{S},\mathcal{K}}$ additionally depends on the ratio of the largest and the smallest eigenvalue of the diffusion coefficient $\mathcal{K}$. We can assume $C_{\mathrm{S},\mathcal{K}} \geqslant 1$. We use this decomposition to develop

$$\begin{aligned}
\left\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{J,\mathrm{alg}}\right\|^2 &\overset{(11.7a)}{=} \left(\mathcal{K}\nabla\bar{\rho}^i_{J,\mathrm{alg}}, \nabla\bar{c}^i_0 + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)\\
&\overset{(4.4)}{=} \left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla\bar{c}^i_0\right) + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\mathcal{K}\nabla\bar{\rho}^i_{J,\mathrm{alg}}, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j}\\
&\overset{(2.3)}{=} \left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla\bar{c}^i_0\right) + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\left(f, \bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j} - \left(\mathcal{K}\nabla u^i_J, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j}\right)\\
&\overset{(4.9)}{=} \left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla\bar{c}^i_0\right) + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\left(f, \bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j} - \left(\mathcal{K}\nabla u^i_{J,j-1}, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j} + \sum_{k=0}^{j-1}\left(\tilde{\lambda}^i_k\mathcal{K}\nabla\tilde{\rho}^i_k, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j}\right)\\
&\overset{(3.5b)}{=} \left(\mathcal{K}\nabla\rho^i_0, \nabla\bar{c}^i_0\right) + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\left(\mathcal{K}\nabla\rho^i_{j,\boldsymbol{a}}, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j} + \sum_{k=0}^{j-1}\left(\tilde{\lambda}^i_k\mathcal{K}\nabla\tilde{\rho}^i_k, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j}\right)\\
&\overset{(3.5a)}{=} \left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla\bar{c}^i_0\right) + \sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\mathcal{K}\nabla\rho^i_{j,\boldsymbol{a}}, \nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j} + \sum_{j=1}^{J}\sum_{k=0}^{j-1}\left(\tilde{\lambda}^i_k\mathcal{K}\nabla\tilde{\rho}^i_k, \sum_{\boldsymbol{a}\in\mathcal{V}_j}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right).
\end{aligned}$$

We then estimate separately the above three terms using Young's inequality. For the first term, using Young's inequality and employing that $\tilde{\lambda}^i_0 = 1$,

$$\left(\mathcal{K}\nabla\tilde{\rho}^i_0, \nabla\bar{c}^i_0\right) \leqslant \frac{C^2_{\mathrm{S},\mathcal{K}}}{2}\left(\tilde{\lambda}^i_0\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}^i_0\|\right)^2 + \frac{1}{2C^2_{\mathrm{S},\mathcal{K}}}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{c}^i_0\|^2.$$

For the second term, we have

$$\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\left(\mathcal{K}\nabla\rho^i_{j,\boldsymbol{a}},\bar{\rho}^i_{j,\boldsymbol{a}}\right)_{\omega^{\boldsymbol{a}}_j}\leqslant C^2_{\mathrm{S},\mathcal{K}}\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\rho^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}+\frac{1}{4C^2_{\mathrm{S},\mathcal{K}}}\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}$$
$$\overset{(11.2)}{=} C^2_{\mathrm{S},\mathcal{K}}\sum_{j=1}^{J}\lambda^i_j\|\mathcal{K}^{\frac{1}{2}}\nabla\rho^i_j\|^2+\frac{1}{4C^2_{\mathrm{S},\mathcal{K}}}\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}$$
$$\overset{(11.5)}{\leqslant} C^2_{\mathrm{S},\mathcal{K}}C(p)\sum_{j=1}^{J}(\tilde{\lambda}^i_j\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}^i_j\|)^2+\frac{1}{4C^2_{\mathrm{S},\mathcal{K}}}\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}$$

Finally, for the third term as in Lemma 11.2,

$$\sum_{j=1}^{J}\sum_{k=0}^{j-1}\left(\tilde{\lambda}^i_k\mathcal{K}\nabla\tilde{\rho}^i_k,\sum_{\boldsymbol{a}\in\mathcal{V}_j}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\right)$$
$$\leqslant\frac{2(d+1)C^2_{\mathrm{S},\mathcal{K}}J}{2}\sum_{j=1}^{J}\sum_{k=0}^{j-1}\left(\tilde{\lambda}^i_k\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}^i_k\|\right)^2+\frac{\sum_{j=1}^{J}\sum_{k=0}^{j-1}\|\mathcal{K}^{\frac{1}{2}}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2}{2(2(d+1))C^2_{\mathrm{S},\mathcal{K}}J)}$$
$$\leqslant(d+1)C^2_{\mathrm{S},\mathcal{K}}J^2\sum_{k=0}^{J}\left(\tilde{\lambda}^i_k\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}^i_k\|^2\right)+\frac{1}{4C^2_{\mathrm{S},\mathcal{K}}}\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}$$

We now sum these three components together and obtain

$$\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}_{J,\mathrm{alg}}\|^2\leqslant 2(d+1)C(p)C^2_{\mathrm{S},\mathcal{K}}J^2\sum_{j=0}^{J}(\tilde{\lambda}^i_j\|\mathcal{K}^{\frac{1}{2}}\nabla\tilde{\rho}^i_j\|)^2+\frac{\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{c}^i_0\|^2+\sum_{j=1}^{J}\sum_{\boldsymbol{a}\in\mathcal{V}_j}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}^i_{j,\boldsymbol{a}}\|^2_{\omega^{\boldsymbol{a}}_j}}{2C^2_{\mathrm{S},\mathcal{K}}}$$
$$\overset{\substack{(4.8)\\(11.8)}}{\leqslant} 2(d+1)C(p)C^2_{\mathrm{S},\mathcal{K}}J^2(\tilde{\eta}^i_{\mathrm{alg}})^2+\frac{1}{2}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}_{J,\mathrm{alg}}\|^2.$$

We substract $\frac{1}{2}\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}_{J,\mathrm{alg}}\|^2$ on both sides and obtain the final result

$$\|\mathcal{K}^{\frac{1}{2}}\nabla\bar{\rho}_{J,\mathrm{alg}}\|^2\leqslant 4(d+1)C^2_{\mathrm{S},\mathcal{K}}C(p)J^2(\tilde{\eta}^i_{\mathrm{alg}})^2. \tag{11.9}$$

□

A Evaluation of local constants of approximation

In this appendix, we present numerical experiments to support the theoretical framework developed in our work, namely to evaluate the link introduced in Section 3.2. These results are included to give further insights towards a more complete understanding. We consider a simple setting of the sine problem (10.1), where the mesh hierarchy from the discretization (2.3) is composed of an initial grid $\mathcal{T}_0$ that contains eight triangles obtained by Delaunay triangulation and a single refinement $\mathcal{T}_J$ obtained by using two uniform iterations of the newest vertex bisection method ($J=1$). In what follows, the polynomial degree p_J will vary.

We define the following ratio

$$\alpha^i_{J,\boldsymbol{a}} := \frac{\left(\sum_{K_i \in \mathcal{T}_J^{\boldsymbol{a}}} \min_{v \in V_J^{\boldsymbol{a}}(K_i)} \left\|\nabla_{\mathcal{T}} \left(\tau^i_{J,\boldsymbol{a}} - v\right)\right\|^2_{K_i}\right)^{\frac{1}{2}}}{\min_{v_{J,\boldsymbol{a}} \in V_J^{\boldsymbol{a}}} \left\|\nabla_{\mathcal{T}} \left(\tau^i_{J,\boldsymbol{a}} - v_{J,\boldsymbol{a}}\right)\right\|_{\omega_J^{\boldsymbol{a}}}}, \tag{A.1}$$

where we set $\tau^i_{J,\boldsymbol{a}} := (u_J - u^i_{J,J-1})|_{\omega_J^{\boldsymbol{a}}}$ with $u^i_{J,J-1}$ constructed via Algorithm 1. The reference finite element solution u_J is obtained via a direct solver. By [8, Corollary 3.7], this ratio has to be uniformly bounded for all polynomial degrees, only depending on mesh shape regularity.

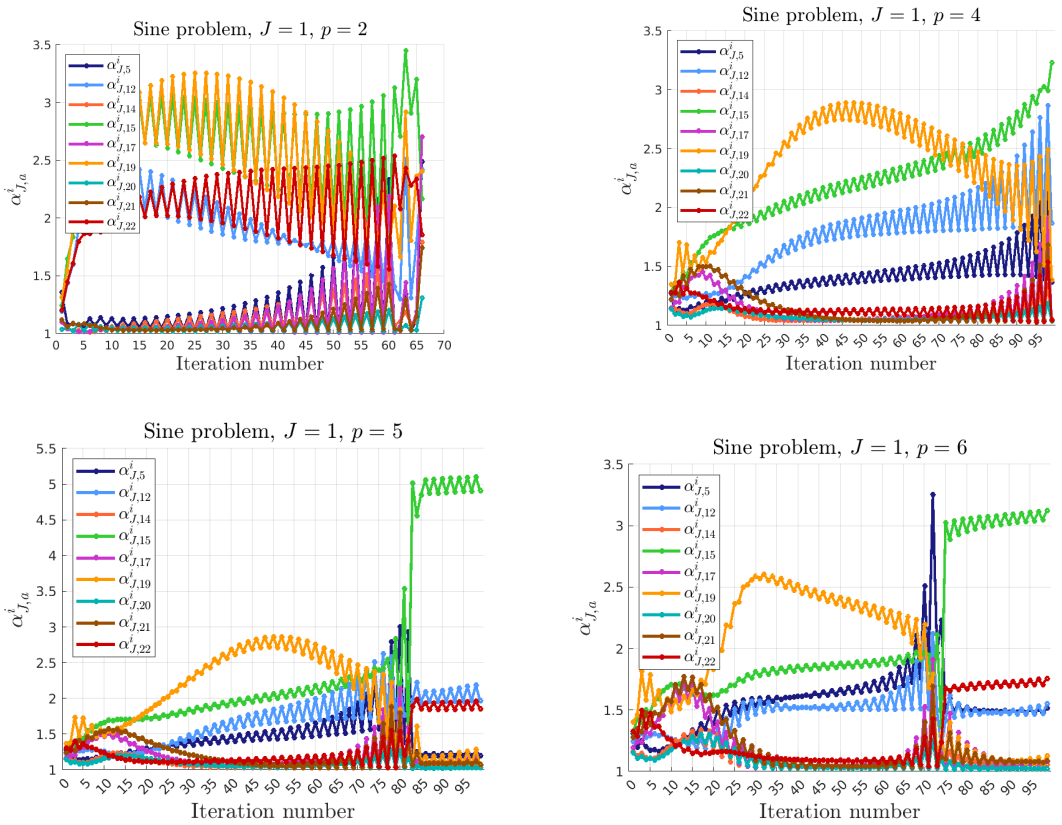


Figure 13: Values of the coefficient $\alpha^i_{J,\boldsymbol{a}}$ for interior patches over the iteration using the solver defined in Algorithm 1. Sine problem (10.1) with $J = 1$ for various polynomial degrees p_J.

As shown in Figure 13, for all interior vertices $\boldsymbol{a} \in \mathcal{V}_J$ the constant $\alpha^i_{J,\boldsymbol{a}}$ remains of comparable magnitude across the iterations and exhibits a p-robust behavior. One can observe that not all local constants behave identically: some remain close to one throughout the iterations, while others take slightly larger values. This indicates that certain patches contribute less to the global error reduction of the solver, whereas others play a more prominent role in the overall contraction.

B Evaluation of the proximity between ρ_j^i and $\tilde{\rho}_j^i$

We continue here the illustrations from Appendix A. We propose to evaluate the proximity between the two global objects $\tilde{\rho}_j^i$ and ρ_j^i defined respectively by (4.5b) and (3.5a). In that regard, we chose $\tau_J^i := u_J - u_{J,J-1}^i$ with u_J obtained using a direct solver and introduce

$$\beta_J^i := \frac{\|\nabla(\tilde{\rho}_J^i - \tau_J^i)\|}{\|\nabla(\rho_J^i - \tau_J^i)\|}.$$

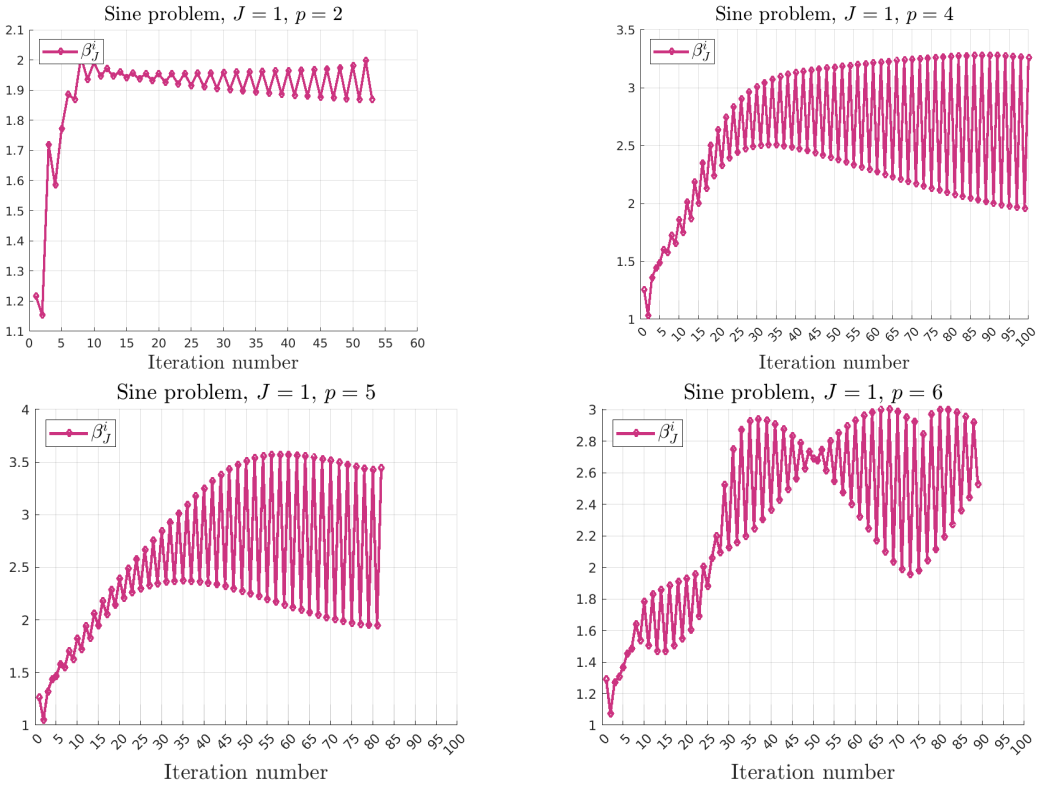


Figure 14: Values of the coefficient β_J^i for interior patches over the iteration using the solver defined in Algorithm 1 for various polynomial degrees p_J. Sine problem (10.1) with $J = 1$.

Similarly to the observations made for Figure 13, we observe that β_J^i is constant in terms of magnitude and therefore is robust with respect to the polynomial degree p. Once again, this corroborates the theoritical results of [8].

C Numerical evalutations of Assumption (5.2) and link between the ratio ξ_J^i and the solver's contraction

In this section, we complete Appendices A and B. We observe numerically the link between the objects $\tilde{\rho}_J^i$ and ρ_J^i that is crucial for the convergence of the solver of Algorithm 1. In that regard, we look at the progression of the ratio $\frac{\|\nabla \rho_J^i\|}{\|\nabla \tilde{\rho}_J^i\|}$, as well as the one of the ratio ξ_J^i of Assumption 5.2, with increasing p.

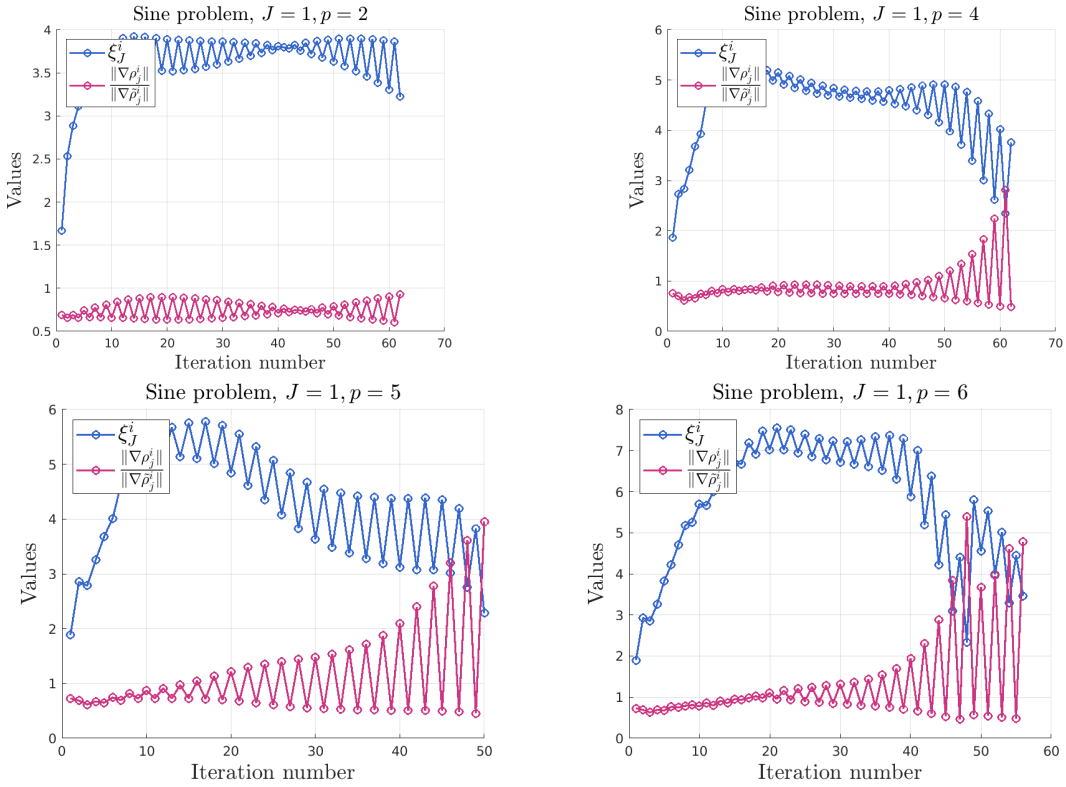


Figure 15: Values of the coefficient ratio $\frac{\|\nabla \rho_J^i\|}{\|\nabla \tilde{\rho}_J^i\|}$ and the ratio ξ_J^i from the Assumption 5.2 over the iterations using the solver of Algorithm 1 for various polynomial degrees. Sine problem (10.1) with $J = 1$.

One can observe that the ratio between ρ_J^i and $\tilde{\rho}_J^i$ and the factor ξ_J^i grow when p_J increases. This is tied to the non p-robustness of the solver observed in Section 10. We now look at the link between the value of the ratio ξ_J^i and the contraction of the solver.

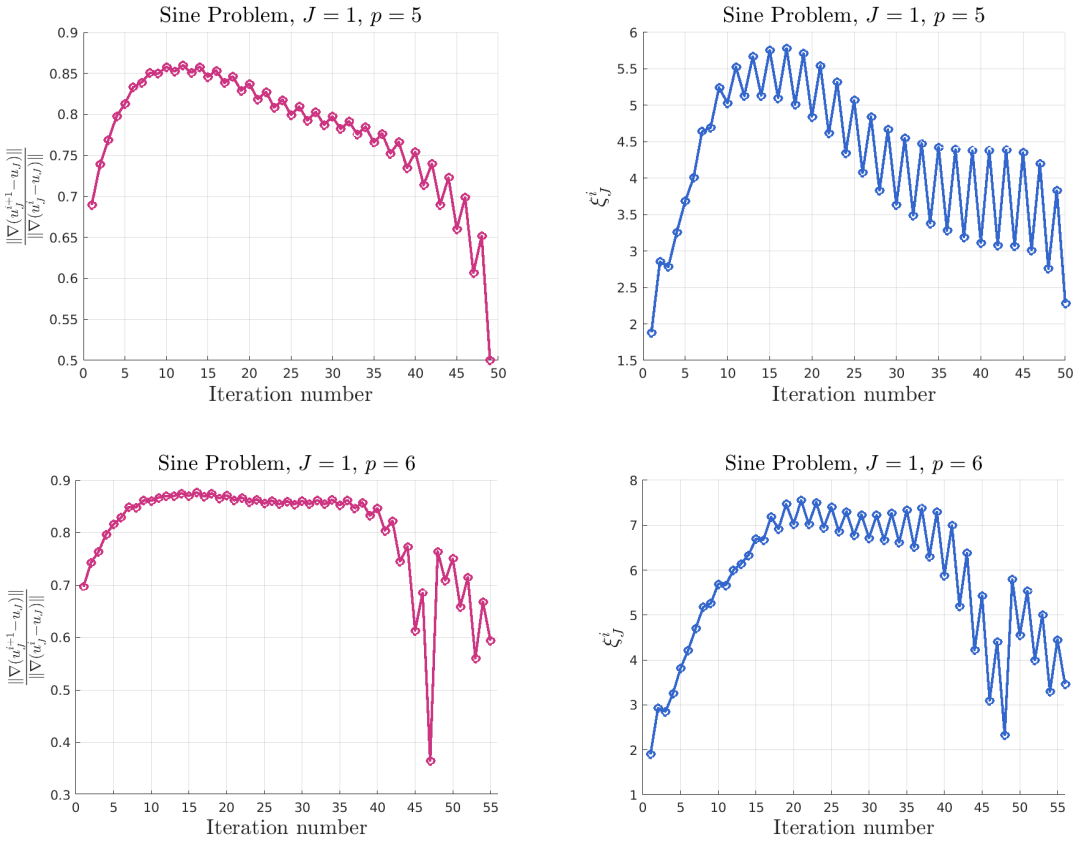


Figure 16: Contraction factor $\dfrac{\|\nabla(u_J^{i+1} - u_J)\|}{\|\nabla(u_J^i - u_J)\|}$ from Theorem 5.3 (left), and ratio ξ_J^i from (5.3) (right). Both quantities are shown for the same polynomial degree p_J of the Sine problem (10.1).

According to the more detailed analysis done in the proof of section 11, one observes a direct relation between the two quantities $\|\nabla\rho_J^i\|$ and $\|\nabla\tilde{\rho}_J^i\|$, defining the ratio ξ_J^i from Assumption 5.2, and the contraction factor of the solver of Algorithm 1. In particular, the value of ξ_J^i serves as an indicator of the efficiency of the iterative procedure: the closer ξ_J^i is to one, the stronger the contraction of the solver becomes. This observation highlights the key role of ξ_J^i in the analysis of the contraction of the solver.

References

[1] P. F. Antonietti, P. Houston, X. Hu, M. Sarti, and M. Verani. Multigrid algorithms for *hp*-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes. *Calcolo*, 54(4):1169–1198, 2017.

[2] P. F. Antonietti and G. Pennesi. *V*-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes. *J. Sci. Comput.*, 78(1):625–652, 2019.

[3] D. Braess, V. Pillwein, and J. Schöberl. Equilibrated residual error estimates are p-robust. *Comput. Methods Appl. Mech. Engrg.*, 198(13-14):1189–1197, 2009.

[4] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2000.

[5] P. D. Brubeck and P. E. Farrell. A scalable and robust vertex-star relaxation for high-order FEM. *SIAM J. Sci. Comput.*, 44(5):A2991–A3017, 2022.

[6] P. D. Brubeck and P. E. Farrell. Multigrid solvers for the de Rham complex with optimal complexity in polynomial degree. *SIAM J. Sci. Comput.*, 46(3):A1549–A1573, 2024.

[7] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[8] A. Ern and M. Vohralík. Stable broken H^1 and $\boldsymbol{H}(\mathrm{div})$ polynomial extensions for polynomial-degree-robust potential and flux reconstruction in three space dimensions. *Math. Comp.*, 89(322):551–594, 2020.

[9] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numer. Math.*, 70(2):163–180, 1995.

[10] W. Hackbusch. *Multigrid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.

[11] I. Huismann, J. Stiller, and J. Fröhlich. Scaling to the stars—a linearly scaling elliptic solver for p-multigrid. *J. Comput. Phys.*, 398:108868, 21, 2019.

[12] A. Miraçi, J. Papež, and M. Vohralík. A multilevel algebraic error estimator and the corresponding iterative solver with p-robust behavior. *SIAM J. Numer. Anal.*, 58(5):2856–2884, 2020.

[13] A. Miraçi, J. Papež, and M. Vohralík. A-posteriori-steered p-robust multigrid with optimal step-sizes and adaptive number of smoothing steps. *SIAM J. Sci. Comput.*, 43(5):S117–S145, 2021.

[14] A. Miraçi, J. Papež, and M. Vohralík. Contractive local adaptive smoothing based on Dörfler's marking in a-posteriori-steered p-robust multigrid solvers. *Comput. Methods Appl. Math.*, 21(2):445–468, 2021.

[15] W. F. Mitchell. Adaptive refinement for arbitrary finite-element spaces with hierarchical bases. *Journal of Computational and Applied Mathematics*, 36(1):65–78, 1991. Special Issue on Adaptive Methods.

[16] P. Oswald. *Multilevel finite element approximation*. Teubner Skripten zur Numerik. [Teubner Scripts on Numerical Mathematics]. B. G. Teubner, Stuttgart, 1994. Theory and applications.

[17] L. F. Pavarino. Additive Schwarz methods for the p-version finite element method. *Numer. Math.*, 66(4):493–515, 1994.

[18] L. F. Pavarino and O. B. Widlund. A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions. *SIAM J. Numer. Anal.*, 33(4):1303–1335, 1996.

[19] A. Quarteroni and G. Sacchi Landriani. Domain decomposition preconditioners for the spectral collocation method. *J. Sci. Comput.*, 3(1):45–76, 1988.

[20] J. Schöberl, J. M. Melenk, C. Pechstein, and S. Zaglmayr. Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements. *IMA J. Numer. Anal.*, 28(1):1–24, Mar. 2008.

[21] E. G. Sewell. *Automatic generation of triangulations for piecewise polynomial approximation.* ProQuest LLC, Ann Arbor, MI, 1972. Thesis (Ph.D.)–Purdue University.

[22] J. Xu, L. Chen, and R. H. Nochetto. Optimal multilevel methods for $H(\mathrm{grad})$, $H(\mathrm{curl})$, and $H(\mathrm{div})$ systems on graded and unstructured grids. In *Multiscale, nonlinear and adaptive approximation*, pages 599–659. Springer, Berlin, 2009.

[23] X. Zhang. Multilevel Schwarz methods. *Numer. Math.*, 63(4):521–539, 1992.