

Local multilevel preconditioners for elliptic equations with jump coefficients on bisection grids

Long Chen · Michael Holst ·
Jinchao Xu · Yunrong Zhu

Received: 27 January 2013 / Accepted: 15 August 2013 / Published online: 12 January 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract The goal of this paper is to design optimal multilevel solvers for the finite element approximation of second order linear elliptic problems with piecewise constant coefficients on bisection grids. Local multigrid and BPX preconditioners are constructed based on local smoothing only at the newest vertices and their immediate neighbors. The analysis of eigenvalue distributions for these local multilevel preconditioned systems shows that there are only a fixed number of eigenvalues which are deteriorated by the large jump. The remaining eigenvalues are bounded uniformly with respect to the coefficients and the meshsize. Therefore, the resulting preconditioned conjugate gradient algorithm will converge with an asymptotic rate independent of the coefficients and logarithmically with respect to the meshsize. As a result, the overall computational complexity is nearly optimal.

Keywords Local multilevel preconditioners · Multigrid · BPX · Discontinuous coefficients · Adaptive finite element methods · PCG · Effective condition number

L. Chen
Department of Mathematics, University of California at Irvine,
Irvine, CA 92697, USA
e-mail: chenlong@math.uci.edu

M. Holst
Department of Mathematics, University of California
at San Diego, La Jolla, CA 92093, USA
e-mail: mholst@math.ucsd.edu

J. Xu
Department of Mathematics, Pennsylvania State University,
University Park, PA 16802, USA
e-mail: xu@math.psu.edu

Y. Zhu (✉)
Department of Mathematics, Idaho State University,
Pocatello, ID 83209, USA
e-mail: zhuyunr@isu.edu

1 Introduction

In this article, we construct robust multilevel preconditioners for the finite element discretization of second order linear elliptic equations with strongly discontinuous coefficients. We extend corresponding results on uniform grids [58] to locally refined grids obtained by bisection methods.

Consider the following model problem :

$$\begin{cases} -\nabla \cdot (a \nabla u) = f & \text{in } \Omega, \\ u = g_D \text{ on } \Gamma_D, \quad a \frac{\partial u}{\partial n} = g_N & \text{on } \Gamma_N \end{cases} \quad (1.1)$$

where $\Omega \in \mathbb{R}^d$ is a polygon (for $d = 2$) or polyhedron (for $d = 3$) with Dirichlet boundary Γ_D and Neumann boundary Γ_N such that $\Gamma_D \cup \Gamma_N = \partial\Omega$. The diffusion coefficient $a = a(x)$ is piecewise positive constant. More precisely, the domain Ω is partitioned into M open disjoint polygonal or polyhedral regions Ω_i ($i = 1, \dots, M$) and

$$a|_{\Omega_i} = a_i, \quad i = 1, \dots, M$$

where each a_i is a positive constant. The regions Ω_i ($i = 1, \dots, M$) may possibly have complicated geometry but we assume that they are completely resolved by an initial triangulation \mathcal{T}_0 . Our analysis can be carried through to more general cases when $a(x)$ varies moderately in each subdomain and to other types of boundary conditions in a straightforward way.

The problem (1.1) belongs to the class of interface problems or transmission problems, which are relevant to many applications such as groundwater flow [31], electromagnetics [29], semiconductor modeling [24, 33], and fuelcells [50]. The coefficients in these applications may have large jumps across interfaces between regions with different material properties, i.e. $J(a) := \max_i a_i / \min_i a_i \gg 1$. Due to $J(a)$ and the mesh size, the finite element discretization of (1.1) is usually very ill-conditioned, which leads to deterioration

in the rate of convergence of multilevel and domain decomposition methods [3, 27, 47].

In some special situations, one is able to show the (nearly) uniform convergence of the multilevel and (overlapping) domain decomposition methods (see [13, 25, 38, 48, 49] for examples). For general cases, one usually need some special techniques to obtain robust iterative methods, (cf. [2, 17, 28, 42]). Recently in [58, 61], we analyzed the eigenvalue distributions of the standard multilevel and overlapping domain decomposition preconditioned systems, and showed that there are only a small fixed number of eigenvalues that may deteriorate due to the discontinuous jump or mesh size, and that all the other eigenvalues are bounded below and above nearly uniformly with respect to the jump and mesh size. As a result, we proved that the convergence rate of the preconditioned conjugate gradient method is uniform with respect to the large jump, and depends logarithmically on mesh size. These results ensure that the standard multilevel and domain decomposition preconditioners are efficient and robust for finite element discretization of (1.1) on *quasi-uniform grids*. In this paper, we extend our results to *locally refined grids*.

The discontinuity of diffusion coefficients causes a lack of regularity of the solution to (1.1), which in turn, leads to deterioration in the rate of convergence for finite element approximations over quasi-uniform triangulations. Adaptive finite element methods through local mesh refinement can be applied to recover the optimal rate of convergence [16]. In order to achieve optimal computational complexity in adaptive finite element methods, it is imperative to design fast algorithms for solving the linear system of equations arising from the finite element discretization. The distinct feature of applying multigrid methods on locally refined meshes is that the number of nodes of nested meshes obtained by local refinements may not grow exponentially, violating one of the key properties of multilevel methods on uniform meshes that leads to optimal $\mathcal{O}(N)$ complexity. Indeed, let N be the number of unknowns in the finest space, the complexity of multilevel methods with global smoothers can be as bad as $\mathcal{O}(N^2)$ [35]. This prevents direct application of algorithms and theories developed in [58] for quasi-uniform grids to locally refined grids.

To achieve optimal $\mathcal{O}(N)$ complexity, the smoothing step in each level must be restricted to the newly added unknowns and their neighbors (see [7, 12, 35]). Such methods are referred to as *local multilevel methods* in [7]. As an extreme case, one can preform the smoothing only on newly added nodes turning a coarse grid to a fine grid. The resulting method is known as the hierarchical basis method [8, 60]. In two dimensions, hierarchical basis methods are proven to be robust for jump coefficient problems on locally refined meshes (cf. [8]). In three dimensions, however, classic multilevel and domain decomposition methods, including the

hierarchical basis multigrid methods, deteriorate rapidly due to the presence of discontinuity of coefficients. To obtain robust rates of convergence for multigrid methods, one has to use special coarse spaces [25, 41] or assume that the distribution of diffusion coefficients satisfies the so called quasi-monotone condition [25]. Therefore the three dimensional case is much more difficult. There are other works [1, 30] on optimal complexity of local multilevel methods in three dimensions, but the problems with discontinuous coefficients remain open.

In this article, we shall design and prove the efficiency and robustness of local multilevel preconditioners for the finite element discretization of problem (1.1) on bisection grids—one class of locally refined grids. In these preconditioners, we use a global smoothing in the finest mesh; and for each newly added node, we perform smoothing only for three vertices—the new vertex and its two parents vertices (the vertices sharing the same edge with the new vertex). We analyze the eigenvalue distribution of the multilevel preconditioned matrix, and prove that there are only a fixed number of small eigenvalues deteriorated by the coefficient and mesh-size; the other eigenvalues are bounded nearly uniformly. Thus, the resulting preconditioned conjugate gradient algorithm converges uniformly with respect to the jump and logarithmically with respect to the mesh size of the discretization. We establish our results of this type in both two and three dimensions.

To employ the geometric structure of bisection grids, we use the decomposition of bisection grids developed in the recent work [23, 52]. This approach enables us to introduce a natural decomposition of the finite element space into subspaces consisting only the newest vertices and their two parents vertices. In the analysis of these local multilevel preconditioners, one of the key ingredient is the *stable decomposition* (see Theorem 4.2). For the standard multilevel preconditioners on uniform mesh, in [58] we used the approximation and stability properties of the weighted L^2 projection (cf. [13]) to construct a stable decomposition. This weighted L^2 projection is no longer applicable for the local multilevel preconditioners, since it is a global projection. In order to preserve the local natural of the highly graded meshes, we introduce a local interpolation operator, which we manage to prove similar approximation and stability properties (see Theorems 3.1 and 3.2) as the weighted L^2 -projection. Our local quasi-interpolation operator and the corresponding analysis is more delicate than that in [23, 52] for the Poisson equation. We should remark that due to this space decomposition, we are able to remove the assumption, *nested local refinement*, which is used in most existing work on multilevel methods on local refinement grids [1, 30].

The rest of the paper is organized as follows. In Sect. 2, we give some notation and recall some fundamental results as in [58]. In Sect. 4, we study bisection grids, and review

some technical tools from [23, 52]. Here we restrict ourself to a kind of special bisection scheme, namely the newest vertex bisection. Then in Sect. 4, we study some technical results of space decomposition, and present the optimal/stable decomposition and the strengthened Cauchy–Schwarz inequality on bisection grids. In Sect. 5, we analyze multilevel preconditioners, i.e., the BPX preconditioner and the multigrid V -cycle preconditioner, and prove convergence results for the preconditioned conjugate gradient algorithm. In Sect. 6, we present numerical experiments to support our theoretical results.

Throughout the article, we will use the following short notation, $x \lesssim y$ means $x \leq Cy$, $x \gtrsim y$ means $x \geq cy$ and $x \approx y$ means $cx \leq y \leq Cx$ where c and C are generic positive constants independent of the variables appearing in the inequalities and any other parameters related to mesh, space and coefficients.

2 Preliminaries

In this section, we introduce some notation, set up our problem, and review briefly some facts about the preconditioned conjugate gradient algorithm.

2.1 Notation and problem

Given a set of positive constants $\{a_i\}_{i=1}^M$, we define the following weighted inner products on the space $H^1(\Omega)$

$$(u, v)_{0,a} = \sum_{i=1}^M a_i (u, v)_{L^2(\Omega_i)},$$

and

$$(u, v)_{1,a} = \sum_{i=1}^M a_i (\nabla u, \nabla v)_{L^2(\Omega_i)}$$

with the induced weighted L^2 norm $\|\cdot\|_{0,a}$, and the weighted H^1 -seminorm $|\cdot|_{1,a}$, respectively. We denote by

$$\|u\|_{1,a} = \left(\|u\|_{0,a}^2 + |u|_{1,a}^2 \right)^{\frac{1}{2}},$$

and the related inner product and the induced energy norm by

$$(u, v)_A = A(u, v) := (u, v)_{1,a}, \quad \|u\|_A = \sqrt{A(u, u)}.$$

To impose the Dirichlet boundary condition in (1.1), we define

$$H_{g_D, \Gamma_D}^1 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = g_D \text{ in the trace sense}\},$$

and $H_D^1 := H_{0, \Gamma_D}^1$. Given a shape regular triangulation \mathcal{T}_h , which could be highly graded, we define \mathcal{V}_h as the standard piecewise linear and global continuous finite element space

on \mathcal{T}_h . Given $f \in H^{-1}(\Omega)$ and $g_N \in H^{1/2}(\Gamma_N)$, the linear finite element approximation of (1.1) is the function $u \in \mathcal{V}_h \cap H_{g_D, \Gamma_D}^1$, such that

$$A(u, v) = \langle f, v \rangle + \int_{\Gamma_N} g_N v, \quad \text{for all } v \in \mathcal{V}_h \cap H_D^1. \quad (2.1)$$

Given any $u_0 \in \mathcal{V}_h \cap H_{g_D, \Gamma_D}^1$, the problem (2.1) is equivalent to finding $u \in \mathcal{V}_h \cap H_D^1$ such that

$$A(u, v) = \langle f, v \rangle + \int_{\Gamma_N} g_N v - A(u_0, v), \quad \forall v \in \mathcal{V}_h \cap H_D^1. \quad (2.2)$$

We thus consider the space $\mathcal{V}_{h,D} := \mathcal{V}_h \cap H_D^1$. The bilinear form $A(\cdot, \cdot)$ will then introduce a symmetric positive definite (with respect to standard L^2 -inner product) operator, still denoted by A , from $\mathcal{V}_{h,D}$ to $\mathcal{V}_{h,D}$ as

$$(Au, v) = A(u, v).$$

Define $b \in \mathcal{V}_{h,D}$ as

$$(b, v) = \langle f, v \rangle + \int_{\Gamma_N} g_N v - A(u_0, v) \quad \forall v \in \mathcal{V}_{h,D}.$$

We then get the following operator equation on $\mathcal{V}_{h,D}$

$$Au = b. \quad (2.3)$$

For simplicity, in the remainder of the paper, we should omit the subscript D in $\mathcal{V}_{h,D}$ without ambiguity.

We are interested in solving Eq. (2.3) by the preconditioned conjugate gradient methods with BPX and multigrid preconditioners. Let us now review briefly some basic results concerning the preconditioned conjugate gradient method.

2.2 Preconditioned conjugate gradient method

Let B be a symmetric positive definite (SPD) operator. Applying it to both sides of (2.3), we get an equivalent equation

$$BAu = Bb. \quad (2.4)$$

We apply the conjugate gradient method to solve (2.4) and the resulting method is known as the *preconditioned conjugate gradient (PCG)* method, where B is called a *preconditioner*.

Let $\kappa(BA) = \lambda_{\max}(BA)/\lambda_{\min}(BA)$ be the (generalized) condition number of the preconditioned system BA . Starting from an arbitrary initial guess u_0 , we have the following well known convergence rate estimate for the k th iteration u_k ($k \geq 1$) in PCG (see e.g. [40])

$$\frac{\|u - u_k\|_A}{\|u - u_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa(BA)} - 1}{\sqrt{\kappa(BA)} + 1} \right)^k.$$

So if the condition number $\kappa(BA)$ is uniformly bounded, then PCG algorithm converges uniformly. Here the uniformity means the independence of the size of the matrix A . Later on, when A is related to Eq. (1.1), we shall also discuss the uniformity of convergence with respect to the jump of diffusion coefficients.

If there are some isolated small or large eigenvalues, we can sharpen the above convergence rate estimate as stated in the following theorem.

Theorem 2.1 [5] *Suppose that $\sigma(BA) = \sigma_0(BA) \cup \sigma_1(BA)$ such that there are m elements in $\sigma_0(BA)$ and $\alpha \leq \lambda \leq \beta$ for each $\lambda \in \sigma_1(BA)$. Then*

$$\frac{\|u - u_k\|_A}{\|u - u_0\|_A} \leq 2K \left(\frac{\sqrt{\beta/\alpha} - 1}{\sqrt{\beta/\alpha} + 1} \right)^{k-m}, \tag{2.5}$$

where

$$K = \max_{\lambda \in \sigma_1(BA)} \prod_{\mu \in \sigma_0(BA)} \left| 1 - \frac{\lambda}{\mu} \right|.$$

If there are only m small eigenvalues in $\sigma_0(BA)$, say

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m \ll \lambda_{m+1} \leq \dots \leq \lambda_n,$$

then

$$K = \prod_{i=1}^m \left| 1 - \frac{\lambda_n}{\lambda_i} \right| \leq \left(\frac{\lambda_n}{\lambda_1} - 1 \right)^m = (\kappa(BA) - 1)^m. \tag{2.6}$$

Therefore the convergence rate of PCG algorithm will be dominated by the factor $(\sqrt{\beta/\alpha} - 1)/(\sqrt{\beta/\alpha} + 1)$, i.e. by β/α where $\beta = \lambda_n(BA)$ and $\alpha = \lambda_{m+1}(BA)$. We define the “effective condition number” as follows.

Definition 1 Let \mathcal{V} be an n -dimensional Hilbert space and $T : \mathcal{V} \rightarrow \mathcal{V}$ be a symmetric and positive definite operator. For any integer $m \in [1, n - 1]$, the m th effective condition number of T is defined by

$$\kappa_m(T) = \frac{\lambda_{\max}(T)}{\lambda_{m+1}(T)}$$

where $\lambda_{m+1}(T)$ is the $(m + 1)$ -th minimal eigenvalue of T .

As a corollary of Theorem 2.1, we have

$$\frac{\|u - u_k\|_A}{\|u - u_0\|_A} \leq 2(\kappa(BA) - 1)^m \left(\frac{\sqrt{\kappa_m(BA)} - 1}{\sqrt{\kappa_m(BA)} + 1} \right)^{k-m}. \tag{2.7}$$

From (2.7), given a tolerance ε , the number of iterations of the PCG method to reduce the relative error below the tolerance ε is (cf. [5,6])

$$m + \left\lceil \left(\log \left(\frac{2}{\varepsilon} \right) + m |\log(\kappa(BA) - 1)| \right) / c_0 \right\rceil,$$

where $c_0 = \log((\sqrt{\kappa_m(BA)} + 1)/(\sqrt{\kappa_m(BA)} - 1))$. Therefore if there exists an $m \geq 1$ such that the m th effective condition number is bounded uniformly, then the PCG algorithm will still converge almost uniformly, even though the standard condition number $\kappa(BA)$ might be large.

To estimate the effective condition number, in particular $\lambda_{m+1}(A)$, we use a fundamental tool known as the Courant “minimax” principle (see e.g. [26]).

Theorem 2.2 *Let \mathcal{V} be an n -dimensional Hilbert space with inner product $(\cdot, \cdot)_{\mathcal{V}}$ and $T : \mathcal{V} \rightarrow \mathcal{V}$ a symmetric positive operator on \mathcal{V} . Suppose $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ are the eigenvalues of T , then*

$$\lambda_{m+1}(T) = \max_{\dim(S)=m} \min_{0 \neq v \in S^\perp} \frac{(Tv, v)_{\mathcal{V}}}{(v, v)_{\mathcal{V}}}$$

for $i = 1, 2, \dots, n - 1$. Especially, for any subspace $\mathcal{V}_0 \subset \mathcal{V}$ with $\dim(\mathcal{V}_0) = n - m$

$$\lambda_{m+1}(T) \geq \min_{0 \neq v \in \mathcal{V}_0} \frac{(Tv, v)_{\mathcal{V}}}{(v, v)_{\mathcal{V}}}. \tag{2.8}$$

If both A and B are SPD operators, then BA is SPD in the inner product induced by B^{-1} and A . Below, we shall apply Theorem 2.2 to $T = BA$ and $(u, v)_{\mathcal{V}} := (B^{-1}u, v)_{L^2}$. Therefore if we have an inequality of the type $(Av, v) \geq c(B^{-1}v, v)$ for all v in a suitable subspace \mathcal{V}_0 with $\dim(\mathcal{V}_0) = n - m$, we can get a lower bound of $\lambda_{m+1}(BA)$.

3 Local quasi-interpolation

The theoretical justification of the robustness of multilevel preconditioners relies on establishing approximation and stability properties of certain interpolation operators. There are two difficulties: one is the locality and stability and another is the robustness with respect to the coefficient.

The weighted L^2 -projection $Q_h^a : L^2(\Omega) \rightarrow \mathcal{V}_h$ defined by $(Q_h^a u, v_h)_{0,a} = (u, v_h)_{0,a} \quad \forall v_h \in \mathcal{V}_h$ was used in [58,61] for the case of uniform refinement. For the analysis of local multilevel preconditioners, the interpolation operator should preserve certain local structure. Therefore, the weighted L^2 -projection, which is a global operator, is not appropriate. On the other hand, the standard nodal interpolation operator is local but not stable in the energy norm. Local quasi-interpolation, such as Scott–Zhang operators [43], are developed to achieve both locality and stability.

However, the stability constant will in general depend on the jump of diffusion coefficients if we apply the standard quasi-interpolation globally on the whole domain. The value at a vertex is usually defined using a simplex in the patch of this vertex and thus depends on the diffusion coefficient in this simplex. For a vertex shared by several subdomains, this leads to the dependence of the ratio of coefficients. One remedy is to apply the quasi-interpolation on each subdomain and chose a sub-simplex in the quasi-interpolation. Indeed in the original paper [43], a $(d - 1)$ sub-simplex is used. Such modification is suitable for the interior vertex relative to interfaces for which a common $(d - 1)$ sub-simplex on the interface can be used to glue quasi-interpolations in different regions. For vertices on the boundary of the interface, i.e., edges in 3-D and vertices in 2-D, in general there is no common $(d - 1)$ sub-simplex but only $(d - 2)$ sub-simplex. The trace of H^1 functions is not even well defined on $(d - 2)$ sub-simplex. For example, the function value of a H^1 function at a point can be changed without changing this function. In the discrete level, it can be shown that the trace of a finite element function on a $(d - 2)$ sub-simplex can be almost bounded by its Sobolev norm inside. Therefore we can simply set the function values at the vertices of $(d - 2)$ sub-simplex to zero to glue quasi-interpolation operators defined in different domain.

Below, we construct a quasi-interpolation operator by gluing Scott–Zhang operators in each subdomains and interfaces, and show that it is stable uniformly with respect to the jump of coefficients and nearly uniform to the mesh size of the triangulation. We stress that this local quasi-interpolation operator is designed for the analysis only, and is not needed in the practical implementation.

3.1 Notation on triangulations

Let us introduce some notation related to the domain and its triangulations. As we mentioned earlier, we assume that the polygonal or polyhedral subdomains Ω_i ($i = 1, \dots, M$) are open, disjoint to each other, and satisfy $\cup_{i=1}^M \overline{\Omega}_i = \overline{\Omega}$. We denote $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, or simply Γ if without ambiguity, as the *interface* between two subdomains Ω_i and Ω_j . The subdomains Ω_i ($i = 1, \dots, M$) may possibly have complicated geometry but we assume that they are resolved by an initial *conforming* triangulation \mathcal{T}_0 . Recall that a triangulation \mathcal{T} is called *conforming* if the intersection of any two elements τ and τ' in \mathcal{T} either consists of a common vertex, edge, face (when $d = 3$), or empty.

Let \mathcal{N} , \mathcal{E} and \mathcal{F} (when $d = 3$) denote the set of vertices, edges, and faces of \mathcal{T} respectively. For each vertex $p \in \mathcal{N}$, we define local patch $\omega_p := \cup_{\tau \ni p} \tau$ and, for $\tau \in \mathcal{T}$, $\omega_\tau = \cup_{p \in \tau} \omega_p$. Similarly, on the $(d - 1)$ dimensional interface Γ , o_p , o_e and o_f denote the intersection of corresponding local patches and the interface. The linear finite element space associated to \mathcal{T} is denoted by $\mathcal{V}(\mathcal{T})$, or simply \mathcal{V} . More

generally, for any subset $\mathcal{S} \subset \mathcal{T}$, $\mathcal{V}(\mathcal{S})$ denote the finite element subspace restricted to the subset \mathcal{S} . Similarly, we should denote $\mathcal{N}(G) \subset \mathcal{N}$, $\mathcal{E}(G) \subset \mathcal{E}$ and $\mathcal{F}(G) \subset \mathcal{F}$ as the set of vertices, edges, and faces in $\overline{G} \subset \overline{\Omega}$, respectively.

For each element $\tau \in \mathcal{T}$, we define $h_\tau = |\tau|^{1/d}$ and ρ_τ for the radius of its inscribed ball. In the whole paper, we assume that the triangulation is *shape regular* in the sense $h_\tau \approx \rho_\tau$. Let h denote the piecewise constant mesh size function with $h|_\tau = h_\tau$, and $h_{\min} := \min_{\tau \in \mathcal{T}} h_\tau$. We should also denote h_e by the length of an edge $e \in \mathcal{E}$ and h_f by the diameter of a face $f \in \mathcal{F}$. Moreover, we define h_p as the diameter of the local patch ω_p . By the shape regularity assumption, for all $e, f, \tau \subset \omega_p$, we have $h_p \approx h_e \approx h_f \approx h_\tau$.

3.2 Technical lemmas

For completeness here, we quote some technical lemmas from [13], which will be used later for proving the approximation and stability of our local interpolation operator.

In two dimensions, it is well known that $H^1(\Omega)$ is not embedded into $L^\infty(\Omega)$. But for finite element functions, we can control the L^∞ norm by its H^1 -norm with a factor $|\log h_{\min}|^{1/2}$.

Lemma 3.1 ([13, Lemma 2.3]) *For any subdomain $\Omega_i \subset \mathbb{R}^2$, let $\mathcal{V}(\Omega_i)$ be the finite element space based on a shape-regular triangulation \mathcal{T} of Ω_i . Then for all $v \in \mathcal{V}(\Omega_i)$, it satisfies*

$$\|v\|_{L^\infty(\Omega_i)} \lesssim \left| \log \frac{H_i}{h_{\min}} \right|^{1/2} \left(|v|_{H^1(\Omega_i)} + H_i^{-1} \|v\|_{L^2(\Omega_i)} \right),$$

where $H_i = \text{diam}(\Omega_i)$ and $h_{\min} := \min_{\tau \in \mathcal{T}} h_\tau$.

In three dimensions, the trace of an H^1 -function on an edge is not well defined. But for a finite element function, its L^2 -norm on an edge can be bounded by its H^1 -norm with a factor $|\log h_{\min}|^{1/2}$. It is a generalization of Lemma 3.1 to three dimensions in the sense that controlling the norm on a co-dimension 2 boundary manifolds.

Lemma 3.2 ([13, Lemma 2.4]) *Given a polyhedral subdomain $\Omega_i \subset \mathbb{R}^3$, let E be any edge of Ω_i and $\mathcal{V}(\Omega_i)$ be a finite element space based on a shape-regular triangulation of Ω_i . Then for all $v \in \mathcal{V}(\Omega_i)$, there holds*

$$\|v\|_{L^2(E)} \lesssim \left| \log \frac{H_i}{h_{\min}} \right|^{1/2} \left(|v|_{H^1(\Omega_i)} + H_i^{-1} \|v\|_{L^2(\Omega_i)} \right),$$

where $H_i = \text{diam}(\Omega_i)$.

In the analysis of the local quasi-interpolation in Theorem 3.1 below, we should apply Lemmas 3.1 and 3.2 on each subdomain Ω_i , for which the diameter $H_i = \text{diam}(\Omega_i) \simeq 1$ is a fixed generic constant.

3.3 Stable local quasi-interpolation

Given a conforming triangulation \mathcal{T}_h , the Scott-Zhang interpolation operator $\Pi : H^1(\Omega) \rightarrow \mathcal{V}(\mathcal{T}_h)$ can be defined as follows. For any $p \in \mathcal{N}(\mathcal{T}_h)$, we choose a $(d - 1)$ -simplex $\sigma_p \ni p$ in \mathcal{T}_h . We remark that the choice of σ_p is not unique (see Sect. 4.4 for the particular choice of σ_p for our purpose). Let $\{\lambda_{\sigma_p,i} : i = 1, \dots, d\}$ be the barycentric coordinates of σ_p . One can define the L^2 -dual basis $\{\theta_{\sigma_p,i} : i = 1, \dots, d\}$ of $\{\lambda_{\sigma_p,i} : i = 1, \dots, d\}$, namely, $\int_{\sigma_p} \theta_{\sigma_p,i} \lambda_{\sigma_p,j} = \delta_{ij}$. We define a quasi-interpolation Π as

$$\Pi v = \sum_{p \in \mathcal{N}(\mathcal{T}_h)} \left(\int_{\sigma_p} \theta_{\sigma_p} v \right) \phi_p, \tag{3.1}$$

where $\{\phi_p\}_{p \in \mathcal{N}(\mathcal{T}_h)}$ is the set of nodal basis of $\mathcal{V}(\mathcal{T}_h)$, and $\theta_{\sigma_p} = \theta_{\sigma_p,1}$. The following properties of the operator Π can be found in [37,43].

Lemma 3.3 *The interpolation operator Π satisfies the following properties:*

(i) *Stability:*

$$\begin{aligned} \|\Pi v\|_{L^2(\tau)} &\lesssim \|v\|_{L^2(\omega_\tau)}, \\ \|\Pi v\|_{H^1(\tau)} &\lesssim \|v\|_{H^1(\omega_\tau)}; \end{aligned} \tag{3.2}$$

(ii) *Locality:*

$$(\Pi v)|_\tau = v|_\tau \quad \text{if } v \in \mathcal{V}(\omega_\tau); \tag{3.3}$$

(iii) *Approximability:*

$$\|h^{-1}(v - \Pi v)\|_{L^2(\tau)} \lesssim \|v\|_{H^1(\omega_\tau)}. \tag{3.4}$$

We apply the quasi-interpolation (3.1) on each subdomain, and denote $\Pi_i : L^2(\Omega_i) \rightarrow \mathcal{V}(\Omega_i)$ by the Scott-Zhang interpolation restricted to Ω_i . To be able to glue them together, for any vertex on the interior of the interface, we choose a common $(d - 1)$ sub-simplex shared by two sub-domains. By such choice, Π_i and Π_j will match on the vertex interior relative to the interface.

We now define a local interpolation operator \mathcal{I}_h^a which has the desirable local approximation and stability properties in the weighted Sobolev norms. Given a $u \in H^1(\Omega)$, we define $\mathcal{I}_h^a u \in \mathcal{V}(\mathcal{T}_h)$ such that for $p \in \mathcal{N}(\Omega_i)$

$$\mathcal{I}_h^a u(p) := \begin{cases} (\Pi_i u)(p), & \text{otherwise,} \\ 0, & \text{if } p \in \mathcal{N}(\partial\Gamma_i). \end{cases} \tag{3.5}$$

For a vertex p , let σ_p be the $(d - 1)$ -simplex chosen to define the nodal value at p . Then the interpolant \mathcal{I}_h^a is

uniquely determined by the mapping $p \rightarrow \sigma_p$. In (3.5), if p is in the interior of some subdomain Ω_i , then $\sigma_p \subset \Omega_i$ is chosen to be any $(d - 1)$ -simplex in \mathcal{T} containing p ; if p is in the interior of the interface Γ , then $\sigma_p \subset \Gamma$ is chosen to be a $(d - 1)$ -simplex on the interface containing p . The choice of σ_p is not unique. However, in order to preserve the local structure of the adaptive grids, σ_p should be chosen carefully for each vertex p . This will be clear in Sect. 4 when we discuss the geometry of the bisection grids (see Sect. 4.4 for details). Now we are in the position to present the main result in this section:

Theorem 3.1 *Let $\Omega \subset \mathbb{R}^d$ with $d = 2$ or 3 and \mathcal{T}_h be a triangulation of Ω with mesh size h . Then for all $u \in H^1(\Omega)$, we have*

$$\|h^{-1}(u - \mathcal{I}_h^a u)\|_{0,a,\Omega} \lesssim |\log h_{\min}|^{1/2} \|u\|_{1,a,\Omega}.$$

Proof Using the discrete Sobolev inequality Lemmas 3.1 or 3.2 on $\partial\Gamma$ and the local H^1 -stability (3.2) of Π_i , we have

$$\begin{aligned} \sum_{\Gamma \subset \partial\Omega_i} \|\Pi_i u\|_{L^2(\partial\Gamma)} &\lesssim |\log h_{\min}|^{1/2} \|\Pi_i u\|_{H^1(\Omega_i)} \\ &\lesssim |\log h_{\min}|^{1/2} \|u\|_{H^1(\Omega_i)}. \end{aligned}$$

By the triangle inequality and the approximation property (3.4) of Π_i , we have

$$\begin{aligned} \|h^{-1}(u - \mathcal{I}_h^a u)\|_{L^2(\Omega_i)} &\leq \|h^{-1}(u - \Pi_i u)\|_{L^2(\Omega_i)} + \|h^{-1}(\Pi_i u - \mathcal{I}_h^a u)\|_{L^2(\Omega_i)} \\ &\lesssim \|u\|_{H^1(\Omega_i)} + \sum_{\Gamma \subset \partial\Omega_i} \|\Pi_i u\|_{L^2(\partial\Gamma)} \\ &\lesssim \|u\|_{H^1(\Omega_i)} + |\log h_{\min}|^{1/2} \|u\|_{H^1(\Omega_i)}. \end{aligned}$$

Multiplying by a suitable weight and summing up over all subdomains on both sides, we get the desired estimate. \square

In general, we cannot replace $\|u\|_{1,a}$ by the energy norm $|u|_{1,a}$ in the above lemma; see [53] for a counter example. To be able to use $|u|_{1,a}$ in the estimate, we introduce a subspace $\tilde{H}_D^1(\Omega)$ of $H_D^1(\Omega)$ as follows:

$$\tilde{H}_D^1(\Omega) = \left\{ u \in H_D^1(\Omega) : \int_{\Omega_i} u \, dx = 0 \quad \text{for all } i \in I \right\},$$

where I is the set of indices of all *floating* subdomains:

$$I = \{i : \text{meas}(\partial\Omega_i \cap \Gamma_D) = 0\}.$$

Let $m_0 := \#I$ be the cardinality of I . We emphasize that $m_0 \leq M$ is a constant, depending only on the distribution of the coefficients. In this subspace $\tilde{H}_D^1(\Omega)$, the interpolation \mathcal{I}_h^a has the following properties.

Theorem 3.2 For any $v \in \tilde{H}_D^1(\Omega)$, we have the approximation property of \mathcal{I}_h^a

$$\|h^{-1}(v - \mathcal{I}_h^a v)\|_{0,a} \lesssim |\log h_{\min}|^{\frac{1}{2}} |v|_{1,a}, \tag{3.6}$$

and the stability of \mathcal{I}_h^a in the energy norm

$$|\mathcal{I}_h^a v|_{1,a} \lesssim |\log h_{\min}|^{\frac{1}{2}} |v|_{1,a}. \tag{3.7}$$

Proof For $v \in \tilde{H}_D^1(\Omega)$, it satisfies the Poincaré-Friedrichs inequality on each subdomain Ω_i . Therefore we get $\|v\|_{0,a} \lesssim |v|_{1,a}$. The inequality (3.6) then follows from Lemma 3.1.

To prove inequality (3.7), we use the inequality (3.6) and the local L^2 projection $Q_\tau : L^2(\tau) \rightarrow \mathcal{P}_0(\tau)$ defined by $Q_\tau u|_\tau = |\tau|^{-1} \int_\tau u \, dx$. Then on each element $\tau \in \mathcal{T}_h$, we have

$$\begin{aligned} |\mathcal{I}_h^a v|_{H^1(\tau)}^2 &\lesssim |\mathcal{I}_h^a v - Q_\tau v|_{H^1(\tau)}^2 \lesssim h_\tau^{-2} \|\mathcal{I}_h^a v - Q_\tau v\|_{L^2(\tau)}^2 \\ &\lesssim h_\tau^{-2} \left(\|v - \mathcal{I}_h^a v\|_{L^2(\tau)}^2 + \|v - Q_\tau v\|_{L^2(\tau)}^2 \right) \\ &\lesssim h_\tau^{-2} \|v - \mathcal{I}_h^a v\|_{L^2(\tau)}^2 + |v|_{H^1(\tau)}^2 \end{aligned}$$

where in the last inequality, we used the approximation properties of Q_τ . Multiplying by a suitable weight and summing up over all $\tau \in \mathcal{T}$ on both sides, we get

$$|\mathcal{I}_h^a v|_{1,a}^2 \lesssim \|h^{-1}(v - \mathcal{I}_h^a v)\|_{0,a}^2 + |v|_{1,a}^2 \lesssim |\log h_{\min}| |v|_{1,a}^2$$

where in the last step, we used inequality (3.6). □

Remark 3.1 When the coefficients satisfy the quasi-monotone assumption, the factor $|\log h_{\min}|$ can be removed by arguments on a modified local patch; see [25,39].

4 Bisection grids and space decomposition

In this section, we give a short overview of the framework in the multilevel space decomposition on bisection grids in the recent work [23,52]. Most of the material in this section can be found there.

4.1 Bisection methods

We recall briefly the bisection algorithm for the mesh refinements. Detailed discussions can be found in [11,20,35] and the references cited therein.

Given a conforming triangulation \mathcal{T} of Ω , for each element $\tau \in \mathcal{T}$, we assign an edge of τ to be the *refinement edge* of τ , denoted by $e(\tau)$ or simply e without ambiguity. This procedure is called *labeling*. Given a set of elements marked for refinement, the refinement procedure consists two steps:

- (1) bisect the marked element into two elements by connecting the middle point of the refinement edge to the vertices not contained in the refinement edge;
- (2) assign refinement edges for two new elements.

Given a labeled initial grid \mathcal{T}_0 of Ω and a bisection method, we define

$$\begin{aligned} \mathbb{F}(\mathcal{T}_0) &= \{ \mathcal{T} : \mathcal{T} \text{ is refined from } \mathcal{T}_0 \text{ by bisection method} \}, \\ \mathbb{T}(\mathcal{T}_0) &= \{ \mathcal{T} \in \mathbb{F}(\mathcal{T}_0) : \mathcal{T} \text{ is conforming} \}. \end{aligned}$$

Namely $\mathbb{F}(\mathcal{T}_0)$ contains all triangulations obtained from \mathcal{T}_0 using the chosen bisection method. But a triangulation $\mathcal{T} \in \mathbb{F}(\mathcal{T}_0)$ could be non-conforming and thus we define $\mathbb{T}(\mathcal{T}_0)$ as a subset of $\mathbb{F}(\mathcal{T}_0)$ containing only conforming triangulations.

Given any triangulation \mathcal{T}_0 , we define $\bar{\mathcal{T}}_0 = \mathcal{T}_0$, and the k th uniform refinement $\bar{\mathcal{T}}_k$ ($k \geq 1$) being the triangulation obtained by bisecting all element in $\bar{\mathcal{T}}_{k-1}$ only once. Note that for a conforming initial triangulation \mathcal{T}_0 with arbitrary labeling, $\bar{\mathcal{T}}_k \in \mathbb{F}(\mathcal{T}_0)$ but not necessarily in the set $\mathbb{T}(\mathcal{T}_0)$ in general. Throughout this paper, we shall consider bisection methods which satisfy the following two assumptions:

- (B1) Shape Regularity:** $\mathbb{F}(\mathcal{T}_0)$ is shape regular.
- (B2) Conformity of Uniform Refinement:** $\bar{\mathcal{T}}_k(\mathcal{T}_0) \in \mathbb{T}(\mathcal{T}_0)$ for all $k \geq 0$.

In two dimensions, newest vertex bisection with compatible initial labeling [34] satisfies (B1) and (B2). In three and higher dimensions, the bisection method by Kossaczky [32] and Stevenson [45] will satisfy (B1) and (B2). We note that to satisfy assumption (B2), It might be necessary to modify the initial triangulation by further refinement of each element, which would probably deteriorate the shape regularity. Although (B2) imposes a severe restriction on the initial labeling, it is crucial to control the number of elements added in the completion which is indispensable to establish the optimal complexity of adaptive finite element methods [36].

4.2 Compatible bisections

For a vertex $p \in \mathcal{N}(\mathcal{T})$ or an edge $e \in \mathcal{E}(\mathcal{T})$, we define the *first ring* of p or e to be

$$\mathcal{R}_p = \{ \tau \in \mathcal{T} \mid p \in \tau \}, \quad \mathcal{R}_e = \{ \tau \in \mathcal{T} \mid e \subset \tau \},$$

and the local patch of p or e as $\omega_p = \cup_{\tau \in \mathcal{R}_p} \tau$, and $\omega_e = \cup_{\tau \in \mathcal{R}_e} \tau$. Note that ω_p and ω_e are subsets of Ω , while \mathcal{R}_p and \mathcal{R}_e are subsets of \mathcal{T} which can be thought of as triangulations of ω_p and ω_e , respectively. The cardinality of a set S will be denoted by $\#S$.

Given a labeled triangulation \mathcal{T} , an edge $e \in \mathcal{E}(\mathcal{T})$ is called a *compatible edge* if e is the refinement edge of τ for all $\tau \in \mathcal{R}_e$. For a compatible edge, the ring \mathcal{R}_e is called a *compatible ring*, and the patch ω_e is called a *compatible*

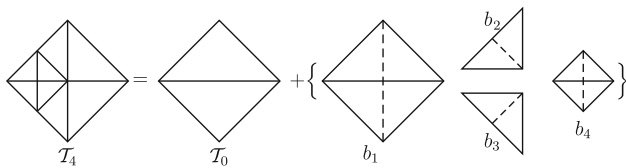


Fig. 1 A decomposition of a bisection grid

patch. Let p be the midpoint of e and \mathcal{R}_p be the ring of p in the refined triangulation. A *compatible bisection* is a mapping $b_e : \mathcal{R}_e \rightarrow \mathcal{R}_p$. We then define the addition

$$\mathcal{T} + b_e := \mathcal{T} \setminus \mathcal{R}_e \cup \mathcal{R}_p.$$

For a compatible bisection sequence $\mathcal{B} := (b_1, \dots, b_k)$, the addition $\mathcal{T} + \mathcal{B}$ is defined as

$$\mathcal{T} + \mathcal{B} = ((\mathcal{T} + b_1) + b_2) + \dots + b_k,$$

whenever the addition is well defined. Note that if \mathcal{T} is conforming, then $\mathcal{T} + b_e$ is conforming for a compatible bisection b_e , whence compatible bisections preserve the conformity of triangulations.

We now present a decomposition of meshes in $\mathbb{T}(\mathcal{T}_0)$ using compatible bisections, which will be instrumental later. We only give a pictorial demonstration in Fig. 1 to illustrate the decomposition. For the proof, we refer to [52].

Theorem 4.1 (Decomposition of Bisection Grids). *Let \mathcal{T}_0 be a conforming triangulation. Suppose the bisection method satisfies assumptions (B2), i.e., for all $k \geq 0$ all uniform refinements $\bar{\mathcal{T}}_k$ of \mathcal{T}_0 are conforming. Then for any $\mathcal{T} \in \mathbb{T}(\mathcal{T}_0)$, there exists a compatible bisection sequence $\mathcal{B} = (b_1, \dots, b_N)$ with $N = \#\mathcal{N}(\mathcal{T}) - \#\mathcal{N}(\mathcal{T}_0)$ such that*

$$\mathcal{T} = \mathcal{T}_0 + \mathcal{B}. \tag{4.1}$$

We point out that in practice it is not necessary to store \mathcal{B} explicitly during the refinement procedure. Instead we can apply coarsening algorithms to find the decomposition. We refer to [22] (see also [18]) for a vertex-oriented coarsening algorithm and the application to multilevel preconditioners and multigrid methods.

For a compatible bisection $b_i \in \mathcal{B}$, we use the same subscript i to denote related quantities such as:

- e_i : the refinement edge;
- p_i, p_{r_i} : two end points of e_i ;
- p_i : the midpoint of e_i ;
- ω_i : the patch of p_i i.e. ω_{p_i} ;
- $\tilde{\omega}_i = \omega_{p_i} \cup \omega_{p_i} \cup \omega_{p_{r_i}}$;
- h_i : the diameter of ω_i ;
- \mathcal{R}_i : the first ring of p_i in \mathcal{T}_i .
- $\mathcal{T}_i = \mathcal{T}_0 + (b_1, \dots, b_i)$;

4.3 Generation of compatible bisections

The generation of each element in the initial grid \mathcal{T}_0 is defined to be 0, and the generation of a child is 1 plus that of the father. The generation of an element $\tau \in \mathcal{T} \in \mathbb{F}(\mathcal{T}_0)$ is denoted by g_τ and coincides with the number of bisections needed to create τ from \mathcal{T}_0 . For any vertex $p \in \mathcal{N}(\mathcal{T}_0)$, the generation of p is defined as the minimal integer k such that $p \in \mathcal{N}(\bar{\mathcal{T}}_k)$ and is denoted by g_p . In [52], we show that if $b_i \in \mathcal{B}$ is a compatible bisection, then all elements of \mathcal{R}_i have the same generation g_i . Therefore we can introduce the concept of generation of compatible bisections. For a compatible bisection $b_i : \mathcal{R}_{e_i} \rightarrow \mathcal{R}_{p_i}$, we define $g_i = g(\tau)$, $\tau \in \mathcal{R}_{p_i}$.

Throughout this paper we always assume $h(\tau) \approx 1$ for $\tau \in \mathcal{T}_0$. Then since a bisection of a simplex will reduce the volume by half, we have the following important relation between generation and mesh size

$$h_i \approx \gamma^{g_i}, \text{ with } \gamma = \left(\frac{1}{2}\right)^{1/d} \in (0, 1).$$

In particular, we introduce a “level” (or generation) constant $L := \max_{\tau \in \mathcal{T}} g_\tau$. It is obvious that $L \approx \lceil |\log h_{\min}| \rceil$.

Different bisections with the same generation have disjoint local patches. Namely for two compatible bisections b_i and b_j with $g_j = g_i$, we then have $\omega_i \cap \omega_j = \emptyset$. A simple but important consequence is that, for all $u \in L^2(\Omega)$ and $k \geq 0$,

$$\sum_{g_i=k} \|u\|_{0,a,\tilde{\omega}_i}^2 \lesssim \|u\|_{0,a,\Omega}^2. \tag{4.2}$$

4.4 A local quasi-interpolation

We define a sequence of quasi-interpolation operators recursively. Let $\mathcal{I}_0^a : \mathcal{V}(\mathcal{T}_N) \rightarrow \mathcal{V}_0$ be an arbitrary interpolation operator defined by (3.5). Assume $\mathcal{I}_{i-1}^a : \mathcal{V}(\mathcal{T}_N) \rightarrow \mathcal{V}(\mathcal{T}_{i-1})$ is defined. Let b_i be a compatible bisection, which introduces a new vertex p_i from \mathcal{T}_{i-1} to $\mathcal{T}_i = \mathcal{T}_{i-1} + b_i$. We construct $\mathcal{I}_i^a : \mathcal{V}(\mathcal{T}_N) \rightarrow \mathcal{V}(\mathcal{T}_i)$ as follows. If the new vertex $p_i \in \Gamma_D$, we simply define $(\mathcal{I}_i^a v)(p_i) = 0$ to reflect the vanishing boundary condition of v . Otherwise, if $p_i \notin \Gamma_D$ we define the nodal value at p_i through (3.1) with the choice of σ_{p_i} as follows:

- (i) if p_i is in the interior of some subdomain Ω_i , we choose a $(d - 1)$ -simplex σ_{p_i} containing p_i ;
- (ii) if p_i is in the interior of some interface Γ , we choose a $(d - 1)$ -simplex $\sigma_{p_i} \subset \Gamma$ containing p_i ;
- (iii) otherwise, we simply let $\sigma_{p_i} = \emptyset$ and define $(\mathcal{I}_i^a v)(p_i) = 0$.

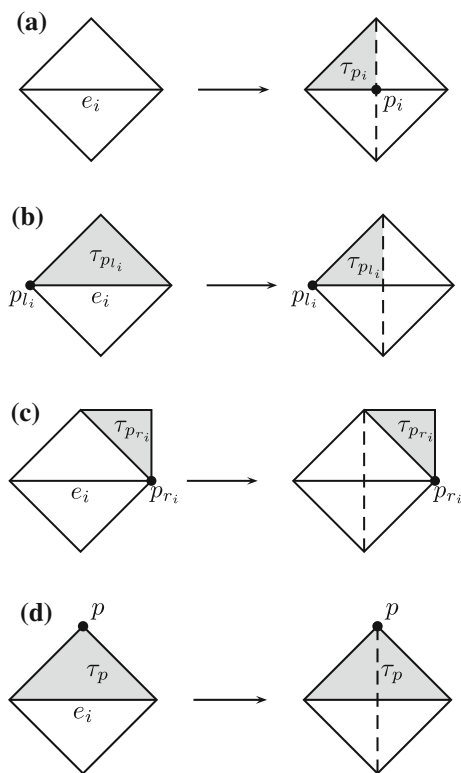


Fig. 2 Update of nodal values \mathcal{I}_{i-1}^a to yield $\mathcal{I}_i^a u$: the element τ chosen to perform the averaging that gives $(\mathcal{I}_i^a u)(p)$ must belong to $\omega_p(\mathcal{T}_i)$. This implies $(\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)u(p) \neq 0$ possibly for $p = p_i, p_{l_i}, p_{r_i}$ and $= 0$ otherwise. **a** Simplex to define $(\mathcal{I}_i^a u)(p_i)$, **b** Simplex to define $(\mathcal{I}_i^a u)(p_{l_i})$, **c** Simplex to define $(\mathcal{I}_i^a u)(p_{r_i})$, **d** Simplex to define $(\mathcal{I}_i^a u)(p)$

For other vertices $p \in \mathcal{N}(\mathcal{T}_{i-1})$, let $\sigma_p \in \mathcal{T}_{i-1}$ be the simplex used to define $(\mathcal{I}_{i-1}^a v)(p)$, we update $(\mathcal{I}_i^a v)(p)$ according to the following two cases:

- (i) if $\sigma_p \subset \overline{\omega_p(\mathcal{T}_i)}$ we keep the nodal value, i.e., $(\mathcal{I}_i^a v)(p) = (\mathcal{I}_{i-1}^a v)(p)$;
- (ii) otherwise we update σ_p as $\sigma_p \leftarrow \overline{\omega_p(\mathcal{T}_i)} \cap \sigma_p$ to define $(\mathcal{I}_i^a v)(p)$.

In either case, we ensure that the simplex $\sigma_p \subset \overline{\omega_p(\mathcal{T}_i)}$. In this way, we obtain a sequence of quasi-interpolation operators

$$\mathcal{I}_i^a : \mathcal{V}(\mathcal{T}_N) \rightarrow \mathcal{V}(\mathcal{T}_i), \quad i = 0 \dots N.$$

Note that in general $\mathcal{I}_N^a v \neq v$ since the simplex used to define nodal values of $\mathcal{I}_N^a v$ may not be in the finest mesh \mathcal{T}_N but in \mathcal{T}_{N-1} . Figure 2 illustrates the choice of σ_p in different cases in 2D.

4.5 Stable space decomposition

Let $\phi_{i,p} \in \mathcal{V}(\mathcal{T}_i)$ denote the nodal basis at node $p \in \mathcal{N}(\mathcal{T}_i)$. Motivated by the stable three-point wavelet construction by Stevenson [44], we define the subspaces $\mathcal{V}_0 = \mathcal{V}(\mathcal{T}_0)$, and

$$\mathcal{V}_i = \text{span}\{\phi_{i,p_i}, \phi_{i,p_{l_i}}, \phi_{i,p_{r_i}}\}.$$

Let $\{\phi_p : p \in \Lambda\}$ be a basis of $\mathcal{V}(\mathcal{T}_N)$, where Λ is the index set of the basis functions, and let \mathcal{V}_p be the 1-dimensional subspace spanned by the nodal bases associated to p in the finest grid. We choose the following space decomposition:

$$\mathcal{V} := \sum_{p \in \Lambda} \mathcal{V}_p + \sum_{i=0}^N \mathcal{V}_i. \tag{4.3}$$

Recall that b_i only changes the local patches of two end points of the refinement edge e_i going from \mathcal{T}_{i-1} to \mathcal{T}_i . By construction $(\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v(p) = 0$ for $p \in \mathcal{N}(\mathcal{T}_i)$, $p \neq p_i, p_{l_i}$ or p_{r_i} , which implies $v_i := (\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v \in \mathcal{V}_i$. Although $\mathcal{I}_N^a v \neq v$ in general, the difference $v - \mathcal{I}_N^a v$ is of high frequency in the finest mesh. Let us write $v - \mathcal{I}_N^a v = \sum_{p \in \Lambda} v_p$ as the basis decomposition. We then obtain a decomposition

$$v = \sum_{p \in \Lambda} v_p + \sum_{i=0}^N v_i, \quad v_i \in \mathcal{V}_i, \tag{4.4}$$

where for convenience we define $\mathcal{I}_{-1}^a := 0$. Moreover, we introduce a subspace $\tilde{\mathcal{V}} := \mathcal{V} \cap \tilde{H}_D^1(\Omega)$. Then we have the following stable decomposition.

Theorem 4.2 (Stable Decomposition). *Given a triangulation $\mathcal{T}_N = \mathcal{T}_0 + \mathcal{B}$ in $\mathbb{T}(\mathcal{T}_0)$, let $L = \max_{\tau \in \mathcal{T}_N} g(\tau)$.*

- (i) For any $v \in \mathcal{V}$, there exist $v_p \in \mathcal{V}_p$ ($p \in \Lambda$) and $v_i \in \mathcal{V}_i$ ($i = 1, \dots, N$) such that $v = \sum_{p \in \Lambda} v_p + \sum_{i=0}^N v_i$ and

$$\sum_{p \in \Lambda} h_p^{-2} \|v_p\|_{0,a}^2 + \|v_0\|_{1,a}^2 + \sum_{i=1}^N h_i^{-2} \|v_i\|_{0,a}^2 \lesssim c_d(L) |v|_{1,a}^2, \tag{4.5}$$

$$\text{where } c_d(L) = \begin{cases} L^2, & d = 2 \\ 2^L, & d = 3. \end{cases}$$

- (ii) For any $v \in \tilde{\mathcal{V}}$, there exist $v_p \in \mathcal{V}_p$ ($p \in \Lambda$) and $v_i \in \mathcal{V}_i$ ($i = 1, \dots, N$) such that $v = \sum_{p \in \Lambda} v_p + \sum_{i=0}^N v_i$ and

$$\sum_{p \in \Lambda} h_p^{-2} \|v_p\|_{0,a}^2 + \|v_0\|_{1,a}^2 + \sum_{i=1}^N h_i^{-2} \|v_i\|_{0,a}^2 \lesssim L^2 |v|_{1,a}^2 \tag{4.6}$$

Proof The result of (i) is standard. We may use the standard nodal interpolation operator to define a decomposition using the hierarchical basis (cf. [56]).

Now we prove (ii). Given a $v \in \tilde{\mathcal{V}}$, we define $v_0 := \mathcal{I}_0^a v$ and $v_i := (\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v$. For $v - \mathcal{I}_N^a v = \sum_{p \in \Lambda} v_p$, by the approximability of the quasi-interpolation, cf. (3.6), we have

$$\sum_{p \in \Lambda} h_p^{-2} \|v_p\|_{0,a}^2 \lesssim \|h^{-1}(v - \mathcal{I}_N^a v)\|_{0,a}^2 \lesssim L|v|_{1,a}^2. \tag{4.7}$$

On the other hand, by Theorem 3.2 we obtain

$$\begin{aligned} & \|\mathcal{I}_0^a v\|_{1,a}^2 + \sum_{i=1}^N h_i^{-2} \|(\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v\|_{0,a,\omega_i}^2 \\ &= \|\mathcal{I}_0^a v\|_{1,a}^2 + \sum_{l=1}^L \sum_{g_i=l} h_l^{-2} \|(\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v\|_{0,a,\omega_i}^2 \\ &\lesssim \left(\sum_{i=1}^L |\log h_{\min}| \right) \|v\|_{1,a}^2 \lesssim L^2 |v|_{1,a}^2. \end{aligned}$$

Then (4.6) follows by adding the above inequality to inequality (4.7). \square

Remark 4.1 The estimate (4.5) is not uniform for $d \geq 2$. For $d = 2$, $L \approx |\log h_{\min}|$ and the growth of $c_2(L)$ is acceptable. But for $d = 3$, the constant $c_3(L) = 2^L$ grows exponentially. This is the main reason that the hierarchical basis multilevel method deteriorates rapidly in 3D (cf. [9, 59]). For discontinuous coefficients problems, it seems unlikely to find a better decomposition with a better constants; see the counterexamples in [13, 38].

If the coefficients satisfy certain monotonicity, e.g. quasi-monotonicity (cf. [25, 39]) in the local patches, one can show that the interpolation operator defined above is stable in the energy norm without deterioration.

Remark 4.2 With a close look at the proof of (4.6), we may regroup the $v_i = (\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v$ into groups $\cup_{l=1}^{L'} G(l) = \{1, 2, \dots, N\}$ such that for any $i, j \in G(l)$, $\omega_j \cap \omega_i = \emptyset$ and therefore

$$\begin{aligned} \sum_{i=1}^N h_i^{-2} \|v_i\|_{0,a,\omega_i}^2 &= \sum_{l=1}^{L'} \sum_{j \in G(l)} h_j^{-2} \|v_j\|_{0,a,\omega_j}^2 \\ &\leq L' |\log h_{\min}| |v|_{1,a}^2. \end{aligned}$$

The constant L' could be much smaller than L ; see Sect. 6 for numerical examples.

4.6 Strengthened Cauchy–Schwarz inequality

An important tool in the analysis of the multiplicative preconditioner is the following strengthened Cauchy–Schwarz inequality. A proof can be found in [23, 52].

Lemma 4.1 (Strengthened Cauchy–Schwarz Inequality). *For any $u_i, v_i \in \mathcal{V}_i$, $i = 0, 1, \dots, N$, we have*

$$\left| \sum_{i=0}^N \sum_{j=i+1}^N A(u_i, v_j) \right| \lesssim \left(\sum_{i=0}^N |u_i|_{1,a}^2 \right)^{\frac{1}{2}} \left(\sum_{i=0}^N h_i^{-2} \|v_i\|_{0,a}^2 \right)^{\frac{1}{2}}. \tag{4.8}$$

As a corollary of (4.8) and the inverse inequality, we have

$$\left\| \sum_{i=0}^N u_i \right\|_{1,a}^2 \lesssim \sum_{i=0}^N h_i^{-2} \|u_i\|_{0,a}^2. \tag{4.9}$$

5 Multilevel preconditioners

In this section, we shall analyze the eigenvalue distribution of the BPX preconditioner and the multigrid V -cycle preconditioner on bisection grids, and prove the effective conditioner number is uniformly bounded.

5.1 BPX (Additive) preconditioner

To simplify the notation, we include $\mathcal{V}_{N+1} = \mathcal{V}$ and rewrite our space decomposition as $\mathcal{V} = \sum_{i=0}^{N+1} \mathcal{V}_i$. Based on this space decomposition, we choose SPD smoothers $R_i : \mathcal{V}_i \rightarrow \mathcal{V}_i$ satisfying

$$(R_i^{-1} u_i, u_i)_{0,a} \approx h_i^{-2} (u_i, u_i)_{0,a}, \quad \forall u_i \in \mathcal{V}_i. \tag{5.1}$$

According to [58], both of the standard Jacobi and symmetric Gauss–Seidel smoother satisfy the above assumption. On the coarsest level, i.e. when $i = 0$, we choose the exact solver $R_0 = A_0^{-1}$. Let $Q_i^a : \mathcal{V} \rightarrow \mathcal{V}_i$ be the weighted L^2 projection. Then we can define the BPX-type preconditioner

$$B = \sum_{i=0}^{N+1} R_i Q_i^a. \tag{5.2}$$

It is well known [51, 55, 57] that the operator B defined by (5.2) is SPD, and

$$(B^{-1} v, v)_{0,a} = \inf_{\sum_{i=0}^{N+1} v_i = v} \sum_{i=0}^{N+1} (R_i^{-1} v_i, v_i)_{0,a}. \tag{5.3}$$

We have the following main result for the BPX preconditioner.

Theorem 5.1 *Given a triangulation $\mathcal{T}_N = \mathcal{T}_0 + \mathcal{B}$ in $\mathbb{T}(\mathcal{T}_0)$, let $L = \max_{\tau \in \mathcal{T}_N} g(\tau)$. For the BPX preconditioner defined in (5.2), we have*

$$\kappa(BA) \leq C_1 c_d(L), \text{ and } \kappa_{m_0}(BA) \leq C_0 L^2.$$

Consequently, we have the following convergence estimation of the BPX preconditioned conjugate gradient method:

$$\frac{\|u - u_k\|_A}{\|u - u_0\|_A} \leq 2(C_1 c_d(L) - 1)^{m_0} \left(\frac{C_0 L - 1}{C_0 L + 1} \right)^{k - m_0}.$$

Proof First of all, let us estimate $\lambda_{\max}(BA)$. For any decomposition $v = \tilde{v} + \sum_{i=0}^N v_i$, $\tilde{v} \in \mathcal{V}$, $v_i \in \mathcal{V}_i$, we have

$$\begin{aligned} \|v\|_A^2 &\lesssim \|\tilde{v}\|_A^2 + \left\| \sum_{i=0}^N v_i \right\|_A^2 \\ &\leq \|h^{-1}\tilde{v}\|_{0,a}^2 + \sum_{i=0}^N h_i^{-2} \|v_i\|_{0,a}^2 \\ &\leq \sum_{i=0}^{N+1} (R_i^{-1}v_i, v_i)_{0,a}. \end{aligned}$$

In the second step, we used the inverse inequality and the inequality (4.9). In the third step, we used the assumption (5.1) of R_i . Taking infimum, we get

$$\|v\|_A^2 \lesssim \inf_{\sum_{i=0}^{N+1} v_i = v} \sum_{i=0}^{N+1} (R_i^{-1}v_i, v_i)_{0,a} = (B^{-1}v, v)_{0,a},$$

which implies that $\lambda_{\max}(BA) \lesssim 1$.

To estimate λ_{\min} , in view of (5.3) we choose the decomposition as in the stable decomposition Theorem 4.2 (see (4.5)) to conclude that

$$(B^{-1}v, v)_{0,a} \leq \sum_{i=0}^{N+1} (R_i^{-1}v_i, v_i)_{0,a} \lesssim c_d(L)(Av, v)_{0,a},$$

which implies that $\lambda_{\min}(BA) \gtrsim c_d(L)$. Therefore we have $\kappa(BA) \lesssim c_d(L)$.

On the other hand, if we apply (4.6) in the subspace $\tilde{\mathcal{V}} \subset \mathcal{V}$, we obtain $\lambda_{m_0+1}(BA) \gtrsim L^2$ by the ‘‘min-max’’ Theorem 2.2. Hence we get an estimate of the effective condition number $\kappa_{m_0}(BA) \lesssim L^2$. The convergence rate estimate then follows by Theorem 2.1. This completes the proof. \square

From this convergence result, we can see that the convergence rate will deteriorate a little bit by $c_d(L)$ as L grows. But since m_0 is a fixed number, when k grows, the convergence rate will be controlled by the effective condition number, which is bounded uniformly with respect to the coefficient and logarithmically with respect to the mesh size. Notice that $L \approx |\log h_{\min}|$ and thus the asymptotic convergence rate of the PCG algorithm is $1 - \frac{1}{C|\log h_{\min}|}$ for $h < 1$.

Remark 5.1 The estimate $\kappa(BA) \leq C_1 c_d(L)$ is sharp in the sense that there exists an example on BPX preconditioner such that $\kappa(BA) \approx c_d(L)$ (cf. [38]).

Remark 5.2 Here we should emphasize that the convergence rate estimate in Theorem 5.1 holds for general substructures. In some special circumstance, for example ‘‘edge type’’ or ‘‘exceptional’’ in the terminology in [38], or ‘‘quasi-monotone’’ coefficient in [25], we can sharpen the convergence estimate in Theorem 5.1 by a modification of Theorem 4.2, see [38].

5.2 Multigrid (Multiplicative) preconditioner

We shall use the following symmetric V-cycle multigrid as a preconditioner in the PCG method and prove the efficiency of such a method. Let $A_i := A|_{\mathcal{V}_i}$. Then one step of the standard V-cycle multigrid $B : \mathcal{V} \rightarrow \mathcal{V}$ is recursively defined as follows:

Let $B_0 = A_0^{-1}$, for $i > 0$ and $g \in \mathcal{V}_i$, define $B_i g = w_3$.

- (i) Preshooting : $w_1 = R_i g$;
- (ii) Correction: $w_2 = w_1 + B_{i-1} Q_{i-1}(g - A_i w_1)$;
- (iii) Postsmoothing: $w_3 = w_2 + R_i^*(g - A_i w_2)$.

Set $B = B_{N+1}$.

For simplicity, we focus on the case of exact subspace solver, i.e., $R_i = A_i^{-1}$ for $i = 0, \dots, N$ and for the finest level, R_{N+1} is chosen as Gauss–Seidel smoother, which can be also understood as the multiplicative method with exact local solvers applied to the nodal decomposition [54]. Let $P_p : \mathcal{V} \rightarrow \mathcal{V}_p$ and $P_i : \mathcal{V} \rightarrow \mathcal{V}_i$ be the orthogonal projection with respect to the inner product $(\cdot, \cdot)_a$. For our special choices of smoothers, we then have

$$I - R_{N+1}A = \prod_{p \in \Lambda} (I - P_p),$$

$$I - B_N A = \left(\prod_{i=0}^N (I - P_i) \right)^* \left(\prod_{i=0}^N (I - P_i) \right),$$

$$\|I - BA\|_A = \left\| \prod_{i=0}^N (I - P_i) \prod_{p \in \Lambda} (I - P_p) \right\|_A^2.$$

For exact local solvers, we can apply the crucial X-Z identity [21, 57] to conclude

$$\|I - BA\|_A = 1 - \frac{1}{1 + c_0}, \tag{5.4}$$

where

$$\begin{aligned} c_0 &= \sup_{\|v\|_A=1} \inf_{v = \sum_{p \in \Lambda} v_p + \sum_{i=0}^N v_i} \\ &\times \left(\sum_{i=0}^N \left\| P_i \sum_{j=i+1}^N v_j + P_i \sum_{p \in \Lambda} v_p \right\|_A^2 + \sum_{p \in \Lambda} \left\| P_p \sum_{q>p} v_q \right\|_A^2 \right). \end{aligned}$$

Theorem 5.2 Given a triangulation $\mathcal{T}_N = \mathcal{T}_0 + \mathcal{B}$ in $\mathbb{T}(\mathcal{T}_0)$, let $L = \max_{\tau \in \mathcal{T}_N} g(\tau)$. For the multigrid V-cycle preconditioner B , we have

$$\kappa(BA) \lesssim c_d(L), \quad \kappa_{m_0}(BA) \lesssim L^2.$$

Consequently, we have the following the convergence rate estimate of the multigrid V-cycle preconditioned conjugate gradient method:

$$\frac{\|u - u_k\|_A}{\|u - u_0\|_A} \leq 2(C_1 c_d(L) - 1)^{m_0} \left(\frac{C_0 L - 1}{C_0 L + 1} \right)^{k-m_0}.$$

Proof Since $I - BA$ is a non-expansive operator, we conclude $\lambda_{\max}(BA) \leq 1$. Since $I - BA$ is SPD in the A -inner product and $\lambda_{\max}(BA) \leq 1$, we have

$$\begin{aligned} \|I - BA\|_A &= \max\{|1 - \lambda_{\min}(BA)|, |1 - \lambda_{\max}(BA)|\} \\ &= 1 - \lambda_{\min}(BA). \end{aligned}$$

To get an estimate on the minimum eigenvalue of BA , we only need to get a upper bound of the constant c_0 in (5.4).

To do so, for any $v \in \mathcal{V}$, we chose the decomposition in Theorem 4.2. That is,

$$v = \tilde{v} + \sum_{i=1}^N v_i, \text{ with } v_0 = \mathcal{I}_0^a v, \quad v_i = (\mathcal{I}_i^a - \mathcal{I}_{i-1}^a)v,$$

where $\tilde{v} = v - \mathcal{I}_N^a v = \sum_{p \in \Lambda} v_p$. Then by shape regularity of the triangulation, we have

$$c_0 \lesssim \sum_{i=0}^N \left\| P_i \sum_{j=i+1}^N v_j \right\|_A^2 + \sum_{i=0}^N \|P_i \tilde{v}\|_A^2 + \sum_{p \in \Lambda} \left\| P_p \sum_{q>p} v_q \right\|_A^2.$$

We estimate these three terms as follows. For the last term, by the finite overlapping of nodal bases, we have

$$\begin{aligned} \sum_{p \in \Lambda} \left\| P_p \sum_{q>p} v_q \right\|_A^2 &\lesssim \sum_{p \in \Lambda} \left\| \sum_{q>p} v_q \right\|_{A, \omega_p}^2 \\ &\lesssim \sum_{p \in \Lambda} \|v_p\|_{A, \omega_p}^2 \lesssim \sum_{p \in \Lambda} h_p^{-2} \|v_p\|_{0, a, \omega_p}^2 \\ &\lesssim \|h^{-1}(v - \mathcal{I}_N^a v)\|_{0, a}^2 \lesssim \|v\|_A^2. \end{aligned}$$

For the middle term, we regroup by generations and use (4.2) to get

$$\begin{aligned} \sum_{i=0}^N \|P_i \tilde{v}\|_A^2 &= \sum_{k=0}^L \sum_{l, gl=k} \|P_l \tilde{v}\|_A^2 \leq \sum_{k=0}^L \sum_{l, gl=k} \|\tilde{v}\|_{A, \tilde{\omega}_l}^2 \\ &\lesssim \sum_{k=0}^L \|\tilde{v}\|_A^2 = L \|\tilde{v}\|_A^2. \end{aligned}$$

For the first term, we define $u_i = P_i \left(\sum_{j=i+1}^N v_j \right)$ and $u_0 := P_0(v - v_0)$ and apply the strengthened Cauchy Schwarz inequality, cf. Lemma 4.1 to get

$$\begin{aligned} \sum_{i=0}^N \left\| P_i \sum_{j=i+1}^N v_j \right\|_A^2 &= \sum_{i=0}^N \sum_{j=i+1}^N A(u_i, v_j) \\ &\lesssim \|v - v_0\|_A^2 + \sum_{i=1}^N h_i^{-2} \|v_i\|_{0, a}^2 \\ &\lesssim c_d(L) \|v\|_A^2. \end{aligned}$$

Here the constant $c_d(L)$ can be improved to L^2 if we consider the decomposition (4.6) of $v \in \tilde{\mathcal{V}}$. Combined with the Mini-Max Theorem 2.2, yields

$$\lambda_{\min}(BA) \gtrsim c_d(L), \quad \lambda_{m_0+1}(BA) \gtrsim L^{-2},$$

and thus

$$\kappa(BA) \lesssim c_d(L), \quad \kappa_{m_0}(BA) \lesssim L^2.$$

Finally, the convergence rate of the PCG method follows by Theorem 2.1. \square

Follow the same proof as Theorem 5.2, we can also obtain the following convergence result for the local multigrid V -cycle solver.

Corollary 5.1 *For the multigrid V -cycle algorithm defined above on bisection grids, we have*

$$\|E\|_A = \|I - BA\|_A = 1 - \frac{1}{1 + c_0},$$

where $c_0 \lesssim c_d(L)$.

This corollary implies that multigrid alone is not robust, especially in 3D. In this case, the convergence rate of multigrid will be proportional to $1 - 2^{-L} \simeq 1 - h_{\min}^{-1}$, which deteriorates rapidly as the mesh size become small. Remark 5.2 is also applicable here, i.e., all the above estimates are estimates for the worst case. For the special circumstances mentioned in Remark 5.2, the estimates can be improved in the same way.

6 Numerical experiments

In this section, we present some numerical experiments to support the theoretical results in previous sections. In the implementation of the adaptive loop, we use a modification of the error indicator presented in [39]. Some other a posteriori error indicators for jump coefficients problem (1.1) can be found in [10, 15, 19, 46]. The adaptive algorithm using different error indicators will generate different grids. However, we emphasize that the robustness of the local adaptive multilevel preconditioners is independent of how the grids are generated in the refinement procedure.

The implementation of the BPX preconditioner and the multigrid methods are standard, and can be found in, for example, [14, 56]. The implementation of the PCG algorithm can be found in [26, 40]. All numerical examples are implemented by using *i*FEM [18]. We only present three-dimensional examples here and refer to [22] for two-dimensional ones. In the PCG algorithm, we use the stopping criterion

$$\frac{\|u^k - u^{k-1}\|_A}{\|u^k\|_A} \leq 10^{-10}. \tag{6.1}$$

In the implementation of the local multilevel preconditioners, we use an algorithm for coarsening bisection grids introduced by [22] for two dimensional case and [18] for three

dimensional one. The coarsening algorithm will find all compatible bisections and regroup them, with possibly different generations, into groups $\cup_{l=1}^{L'} G(l) = \{1, 2, \dots, N\}$ such that for any $i, j \in G(l), \omega_j \cap \omega_i = \emptyset$. Each coarsening step is corresponding to a *level* in the multilevel terminology, and the total number of levels is L' . There are two major benefits of using this coarsening algorithm.

- (i) We do not need to store the complex bisection tree structure of the refinement procedure explicitly in the algorithm. Instead, we only need the grid information on the finest level and the coarsening subroutine will restore multilevel structure.
- (ii) Our numerical evidence shows that the number of nodes will decrease around one half in one coarsening step. Therefore the constant L' is much smaller than the maximal generation $L \approx |\log h_{\min}|$.

In what follows, we will use some shorthand notation for the different algorithms implemented.

- TPSMG stands for the *V*-cycle multigrid iteration with *Three-Point Smoothing* (TPS), which only performs smoothing on new vertices and their two direct neighbors sharing the same edge.
- TPSMGCG is the PCG algorithm using the TPSMG as preconditioner.
- TPSBPXCG is the PCG algorithm using the additive version of TPSMG as preconditioner.

Among all these algorithms, the main focus of this paper is the behavior of TPSMGCG and TPSBPXCG. In the numerical experiments below, we also report some results for TPSMG for comparison, for which we use the same stopping criterion (6.1) as the PCG algorithms.

Example 1 Inspired by [38, 53, 58], we consider solving the model Eq. (1.1) in the cubic domain $\Omega = (-1, 1)^3$. Let the coefficient $a(x)$ be the constants $a_1 = a_2 = 1$ and $a_3 = \varepsilon$ on the three regions Ω_1, Ω_2 and Ω_3 respectively (see Fig. 3), where

$$\Omega_1 = (-0.5, 0)^3, \Omega_2 = (0, 0.5)^3 \text{ and } \Omega_3 = \Omega \setminus (\overline{\Omega_1} \cup \overline{\Omega_2}).$$

We choose $f = 1$ and impose the following boundary conditions: Dirichlet conditions

$$u_{\{-1\} \times [-1, 1] \times [-1, 1]} = 0, \quad u_{\{1\} \times [-1, 1] \times [-1, 1]} = 1,$$

and homogenous Neumann boundary conditions on the remaining boundary. For this problem, singularities occur along edges of Ω_1 and Ω_2 . Figure 4 shows an adaptive mesh and the corresponding finite element approximation after several iterations of the adaptive algorithm. To view the mesh around the singularity, we only show half of the domain Ω .

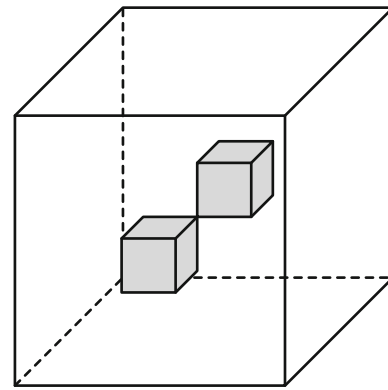


Fig. 3 The coefficients $a_1 = a_2 = 1$ in the gray domains Ω_1 and Ω_2 , and $a_3 = \varepsilon$ in the rest of the domain

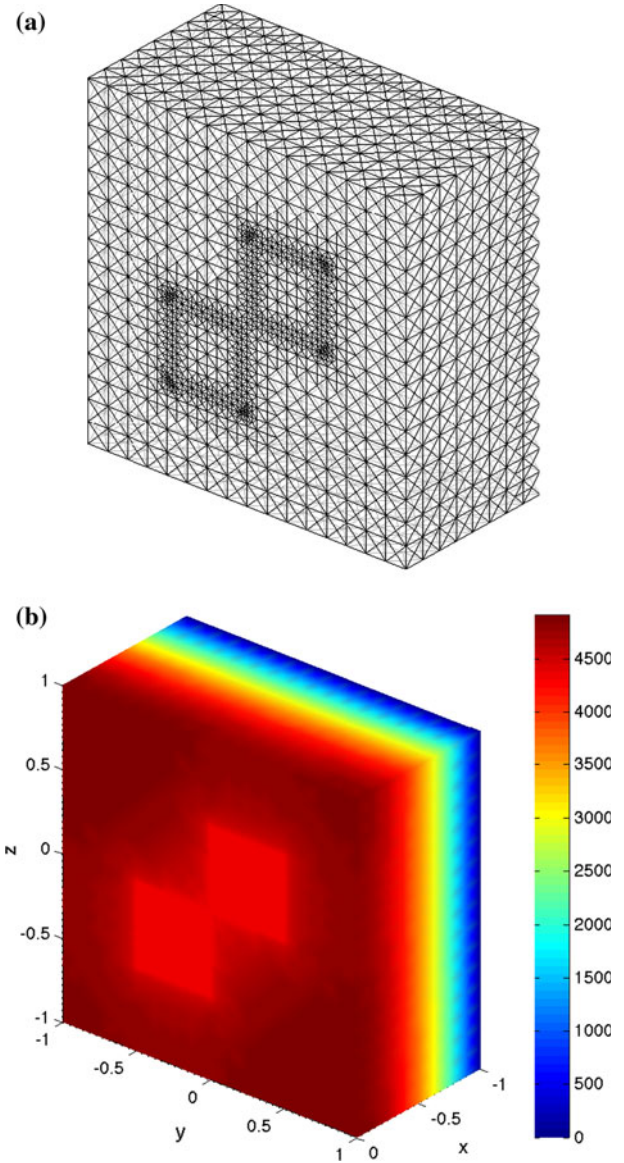


Fig. 4 An adaptive mesh and finite element solution with $\varepsilon = 10^{-4}$ and 36,466 vertices. **a** An adaptive mesh for Example 1, **b** The corresponding finite element solution for Example 1

Tables 1, 2, 3 and 4 give comparisons of the number of iterations for three different algorithms: TPSMG, TPSMGCG and TPSBPXCG algorithms, respectively, with the choice of $\varepsilon = 10^{-4}, 10^{-2}, 10^2$ and 10^4 . As we observe from these tables, the number of iterations for TPSMG algorithm grows rapidly as the mesh is refined when ε is small. On the other hand, the number of iterations for TPSMGCG and TPSBPXCG is very robust and only grows a little bit when the mesh is refined, as we expected from the theory. We also observe that if ε is large, the TPSMG algo-

Table 1 Example 1: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^{-4}$

DOF	TPSMG	TPSMGCG	TPSBPXCG
4,913	41	13	18
5,505	62	15	18
6,617	89	18	21
8,666	99	19	19
10,585	98	19	20
12,411	125	23	25
16,353	154	23	23
21,248	182	22	23
27,755	197	26	32
36,466	178	27	29
43,271	238	25	30
51,163	283	28	36
72,349	395	32	34
89,146	424	31	34
104,747	413	34	38

Table 2 Example 1: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^{-2}$

DOF	TPSMG	TPSMGCG	TPSBPXCG
4,913	46	13	17
5,550	51	15	17
6,743	61	17	20
8,907	65	16	19
10,729	66	17	20
13,281	86	20	24
17,146	90	20	21
23,139	90	20	24
28,613	160	25	29
37,338	175	24	27
43,610	149	22	26
52,715	154	25	31
72,967	238	28	29
89,320	165	25	33
113,131	294	30	38

Table 3 Example 1: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^2$

DOF	TPSMG	TPSMGCG	TPSBPXCG
4,913	16	10	14
5,279	37	15	15
5,867	43	17	18
6,522	48	16	19
7,562	68	17	18
9,493	61	17	18
11,858	49	15	18
15,257	68	15	18
20,649	61	16	19
27,946	49	17	21
36,735	52	16	20
48,890	58	16	22
68,297	71	18	22
89,872	55	16	21
119,109	61	17	23

Table 4 Example 1: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^4$

DOF	TPSMG	TPSMGCG	TPSBPXCG
4,913	16	10	14
5,269	37	15	15
5,863	42	17	18
6,493	45	16	18
7,531	68	17	18
9,419	59	16	17
11,721	46	15	18
14,941	69	15	18
20,065	59	16	19
27,199	47	17	21
35,601	59	16	20
47,743	55	16	22
66,989	71	18	21
88,079	57	16	21
116,739	56	17	23

gorithm will converge uniformly. This is because the coefficient in Ω_3 , which contains the Dirichlet boundary, is dominant. In this case, we could use the standard multi-grid analysis (as in [54]) to show the robustness of the preconditioners.

We use the three-term recursion relation of the PCG coefficients to estimate the eigenvalue distribution of the preconditioned system (see for example [4] or [40, Section 6.7.3]). Figure 5 shows the eigenvalue distributions for the TPSMGCG and TPSBPXCG preconditioned systems. As we can see from the figure, there is one small

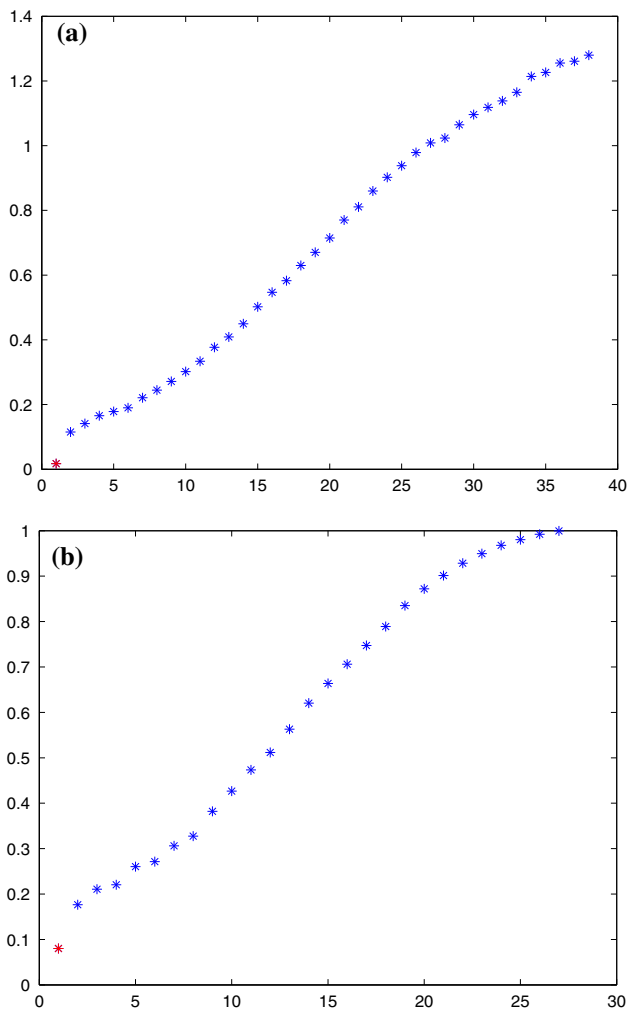


Fig. 5 Example 1: Estimated eigenvalues of BA when $\varepsilon = 10^{-4}$ with 12,411 vertices. The red dots stand for the exceptionally small eigenvalues, due to the large jump in coefficients. **a** Eigenvalues for TPSBPXCG. **b** Eigenvalues for TPSMGCG

eigenvalue for both preconditioned systems. This agrees with the theoretical results, the number of small eigenvalues is bounded by the number of floating subdomains $m_0 \equiv 2$.

Figure 6 shows the condition number and effective condition number of TPSBPXCG and TPSMGCG preconditioned systems. From Fig. 6, we observed that when ε is small, the condition number deteriorates ($\kappa(BA) \in [3, 1100]$ for TPSBPXCG, and $\kappa(BA) \in [3, 125]$ for TPSMGCG as we can see from the figure). On the other hand, if we get rid of the first small eigenvalue, the effective condition number $\kappa_1(BA)$ of TPSBPXCG and TPSMGCG preconditioned systems (the black and red lines, respectively) are almost identical for different ε . This indicates that the effective condition numbers are uniform with respect to the jumps. Moreover, as we can see from Fig. 6, $\kappa_1(BA)$ are mildly increasing with respect to the DOFs ($\kappa_1(BA) \in [1, 80]$

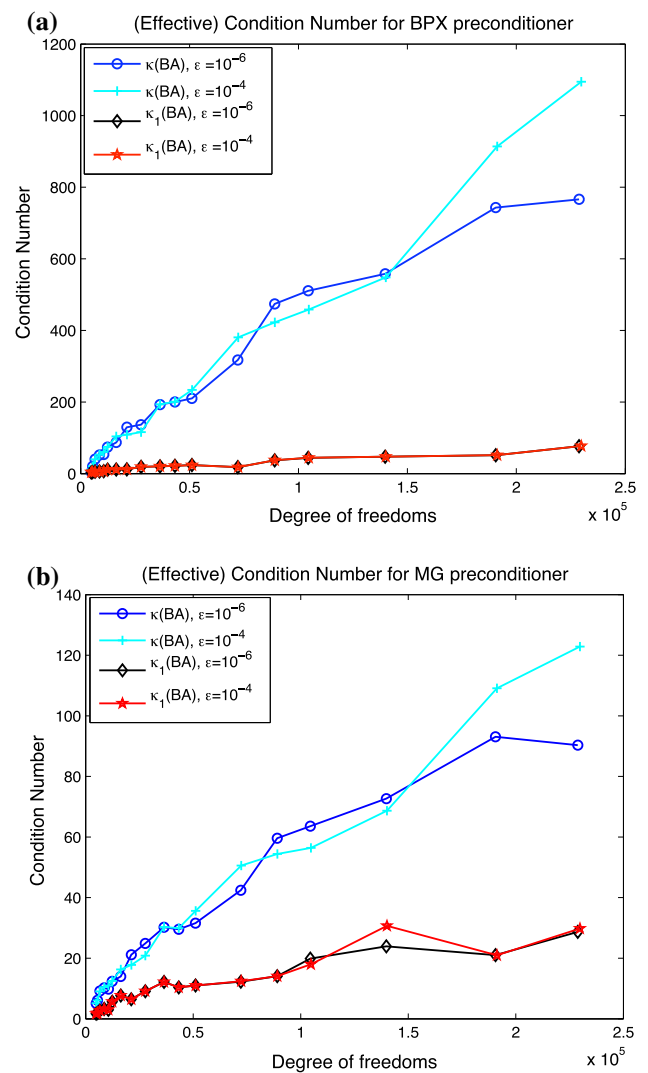


Fig. 6 Example 1: $\kappa(BA)$ and $\kappa_1(BA)$ for the cases $\varepsilon = 10^{-6}, 10^{-4}$ w.r.t the DOFs. **a** TPSBPXCG. **b** TPSMGCG

for TPSBPXCG, and $\kappa_1(BA) \in [1, 30]$ for TPSMGCG). These results agree with our theoretical expectations from Sect. 5.

Example 2 To test the case with more floating subdomains, we consider the following settings in the coefficients. We divide the computational domain $\Omega = (-1, 1)^3$ into 8 small cubes of the same size, and in each of these 8 small cubes, we set the coefficient same as Example 1 (see Fig. 3). Therefore, Ω contains 16 small “gray” cubes, which are pairwise touched at one point. The coefficient $a(x) = 1$ in these small cubes, and $a(x) = \varepsilon$. To resolve the discontinuity, we used a uniform tetrahedra mesh with 729 nodes and 3072 elements as the initial triangulation \mathcal{T}_0 . The right hand and boundary conditions are the same as Example 1.

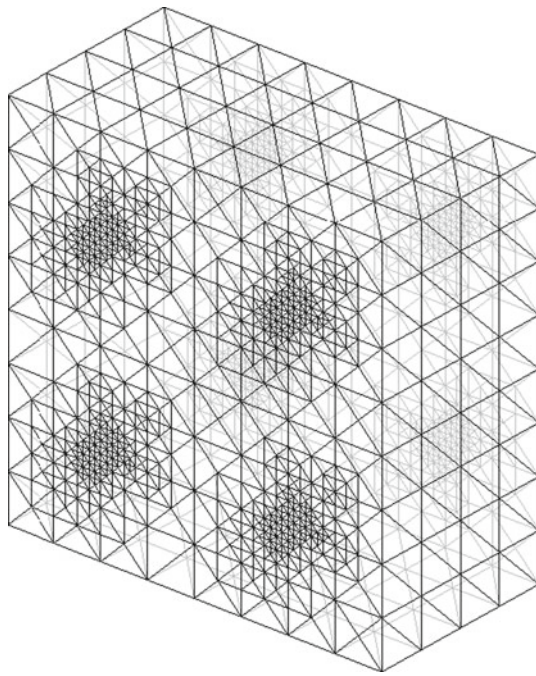


Fig. 7 Example 2: an adaptive mesh with $\varepsilon = 10^{-4}$ and 11,233 vertices

Table 5 Example 2: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^{-4}$

DOF	TPSMG	TPSMGCG	TPSBPXCG
729	3	2	2
1,042	26	9	10
1,977	65	15	17
2,803	136	16	20
4,095	165	19	23
6,195	221	19	25
7,307	349	23	28
11,233	*	27	37
17,023	*	30	42
25,809	*	35	58
28,727	*	33	65
37,473	*	32	62
44,402	*	33	64
49,553	*	35	67

Here “*” means the algorithm doesn’t converge to the required tolerance within 500 iterations (same meaning below)

Figure 7 shows an adaptive mesh after several iterations of the adaptive algorithm. In order to view the mesh around the singularity, we only show the triangulation of the subset $[-0.5, 0.5] \times [-1, 1] \times [-1, 1] \subset \Omega$. As we can see from the figure, the refinement indeed goes around the interesting jump regions.

Table 6 Example 2: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^{-2}$

DOF	TPSMG	TPSMGCG	TPSBPXCG
729	3	2	2
913	11	7	8
1,218	24	9	12
1,848	40	15	15
2,481	54	16	18
3,467	89	19	21
5,730	110	21	24
7,654	132	21	28
11,210	198	26	35
18,844	280	29	39
27,601	306	28	46
42,931	404	36	55
67,598	459	38	60
102,216	*	44	73

Table 7 Example 2: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^2$

DOF	TPSMG	TPSMGCG	TPSBPXCG
729	3	3	3
859	7	7	8
1,219	9	8	10
1,793	14	10	12
3,145	19	12	14
4,609	23	13	16
8,550	32	16	20
12,825	36	17	22
20,901	52	20	27
37,697	69	22	30
57,348	72	22	36
90,812	73	24	40
146,560	79	25	45

As in Example 1, we present the comparisons of the number of iterations for the algorithms: TPSMG, TPSMGCG, and TPSBPXCG in Tables 5, 6, 7 and 8, respectively, with the choice of $\varepsilon = 10^{-4}$, 10^{-2} , 10^2 and 10^4 . When $\varepsilon = 10^{-4}$ or $\varepsilon = 10^{-2}$ (cf. Tables 5, 6), we observe that the number of iterations for TPSMG algorithm grows rapidly as the mesh is refined for both cases. In particular, for sufficiently fine mesh the TPSMG algorithm even does not converge to the required error tolerance in 500 iterations. However, the number of iterations for TPSMGCG and TPSBPXCG is very robust and only grows moderately when the mesh is refined, as we expected from the theory. On the other hand, from Tables 7 and 8 we also observe that in case of $\varepsilon = 10^2$

Table 8 Example 2: Comparison of number of iterations for TPSMG, TPSMGCG and TPSBPXCG when $\varepsilon = 10^4$

DOF	TPSMG	TPSMGCG	TPSBPXCG
729	3	3	3
857	7	7	8
1,225	9	8	10
1,785	14	10	12
3,145	19	12	14
4,591	23	13	16
8,585	32	16	19
12,868	36	17	22
20,687	51	20	27
37,593	69	22	30
57,154	72	22	36
90,293	75	24	41
145,395	80	25	45

or $\varepsilon = 10^4$, the TPSMG algorithm performs well. But we can still see the improvements of the TPSMGCG and TPSBPXCG algorithms. This is consistent with the results in Example 1 (cf. Tables 3, 4).

Figure 8 shows the eigenvalue distributions for the TPSMGCG and TPSBPXCG preconditioned systems, which again are obtained using the relationship between the PCG coefficients and the eigenvalues of the preconditioned systems. As we can see from the figure, there are only 4 small eigenvalues for both preconditioned systems. This agrees with the theoretical results, the number of small eigenvalues is bounded by the number of floating subdomains $m_0 \equiv 16$.

7 Conclusion

In this paper, we designed local multilevel preconditioners based on the decomposition of the finite element space into 3-point subspaces for the highly graded mesh obtained from adaptive bisection algorithms. To analyze the behavior of the local multilevel preconditioners, we introduced a local interpolation operator and proved some approximation and stability properties of it. Based on these properties, we showed the decomposition of the finite element space is stable, which is a key ingredient in the multilevel analysis. This enabled us to analyze the eigenvalue distributions of the preconditioned systems. In particular, we showed that there are only a small fixed number of eigenvalues that are deteriorated by the coefficients and mesh size, and the other eigenvalues are uniformly bounded with respect to the coefficients and logarithmically depends on the mesh size. As a result, we proved the asymptotic convergence rate of the PCG algorithm is

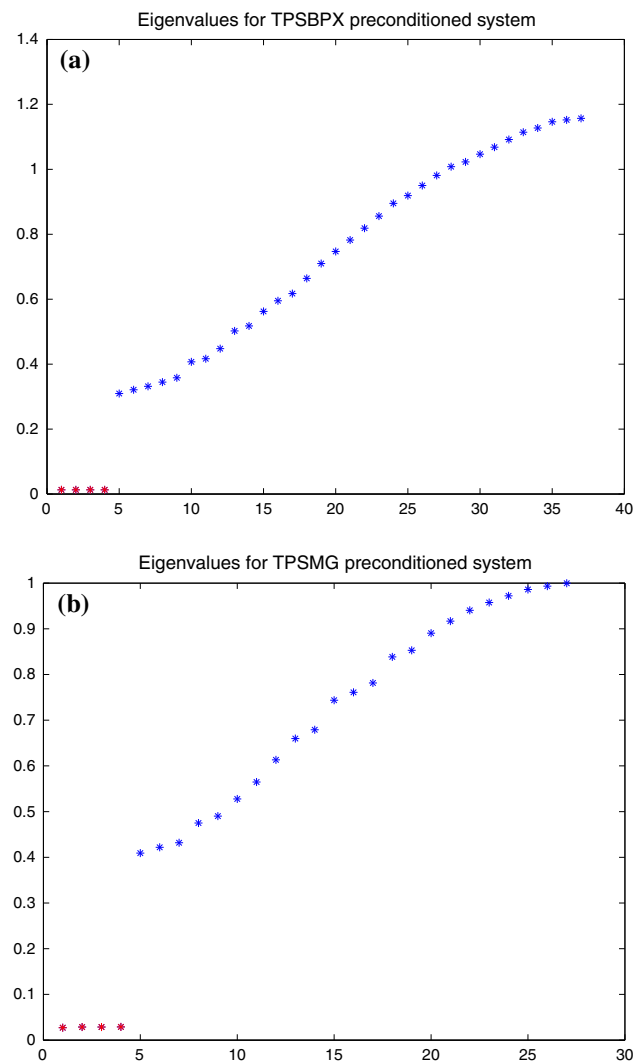


Fig. 8 Example 2: estimated eigenvalues of BA when $\varepsilon = 10^{-4}$ with 11,233 vertices. The red dots stand for the exceptionally small eigenvalues, due to the large jump in coefficients. **a** Eigenvalues for TPSBPXCG. **b** Eigenvalues for TPSMGCG

uniform with respect to the coefficient and nearly uniform with respect to the mesh size. Moreover, the overall computation complexity of these multilevel preconditioner are nearly optimal. Numerical experiments justified our theoretical results.

Acknowledgments The first author is supported in part by NSF Grant DMS-0811272, NSF Grant DMS-1115961, and in part by Department of Energy prime award #DE-SC0006903. The second and fourth authors were supported in part by NSF Awards 1065972 and 1217175, by DTRA Award HDTRA-09-1-0036, and by subcontract to DOE Award #DE-SC0006903. The third author was supported in part by NSF DMS 1217142 and DOE Award #DE-SC0006903. The fourth author was supported in part by NSF DMS 1319110 and the University Research Committee Grant No. F119 at Idaho State University, Pocatello, Idaho. This work is also partially supported by the Beijing International Center for Mathematical Research.

References

1. Aksoylu, B., Holst, M.: Optimality of multilevel preconditioners for local mesh refinement in three dimensions. *SIAM J. Numer. Anal.* **44**(3), 1005–1025 (2006)
2. Aksoylu, B., Graham, I., Klie, H., Scheichl, R.: Towards a rigorously justified algebraic preconditioner for high-contrast diffusion problems. *Comput. Vis. Sci.* **11**(4), 319–331 (2008)
3. Alcouffe, R.E., Brandt, A., Dendy, J.E., Painter, J.W.: The multigrid methods for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.* **2**, 430–454 (1981)
4. Ashby, S.F., Holst, M.J., Manteuffel, T.A., Saylor, P.E.: The role of the inner product in stopping criteria for conjugate gradient iterations. *BIT* **41**(1), 026–052 (2001)
5. Axelsson, O.: Iteration number for the conjugate gradient method. *Math. Comput. Simul.* **61**(3–6), 421–435 (2003). *MODELLING 2001* (Pilsen)
6. Axelsson, O.: *Iterative Solution Methods*. Cambridge University Press, Cambridge (1994)
7. Bai, D., Brandt, A.: Local mesh refinement multilevel techniques. *SIAM J. Sci. Stat. Comput.* **8**(2), 109–134 (1987)
8. Bank, R.E., Dupont, T., Yserentant, H.: The hierarchical basis multigrid method. *Numer. Math.* **52**, 427–458 (1988)
9. Bank, R.E.: Hierarchical bases and the finite element method. *Acta Numer.* **5**, 1–43 (1996)
10. Bernardi, C., Verfürth, R.: Adaptive finite element methods for elliptic equations with non-smooth coefficients. *Numer. Math.* **85**(4), 579–608 (2000)
11. Binev, P., Dahmen, W., DeVore, R.: Adaptive finite element methods with convergence rates. *Numer. Math.* **97**(2), 219–268 (2004)
12. Bramble, J.H., Pasciak, J.E., Xu, J.: Parallel multilevel preconditioners. *Math. Comput.* **55**(191), 1–22 (1990)
13. Bramble, J.H., Xu, J.: Some estimates for a weighted L^2 projection. *Math. Comput.* **56**, 463–476 (1991)
14. Briggs, W.L., Henson, V.E., McCormick, S.F.: *A Multigrid Tutorial*, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2000)
15. Cai, Z., Zhang, S.: Recovery-based error estimator for interface problems: conforming linear elements. *SIAM J. Numer. Anal.* **47**(3), 2132–2156 (2009)
16. Cascon, J.M., Kreuzer, C., Nochetto, R.H., Siebert, K.G.: Quasi-optimal convergence rate for an adaptive finite element method. *SIAM J. Numer. Anal.* **46**(5), 2524–2550 (2008)
17. Chan, T.F., Wan, W.L.: Robust multigrid methods for nonsmooth coefficient elliptic linear systems. *J. Comput. Appl. Math.* **123**(1–2), 323–352 (2000)
18. Chen, L.: iFEM: an integrate finite element methods package in MATLAB. Technical report, University of California at Irvine (2009)
19. Chen, Z., Dai, S.: On the efficiency of adaptive finite element methods for elliptic problems with discontinuous coefficients. *SIAM J. Sci. Comput.* **24**(2), 443–462 (2002)
20. Chen, L.: Short implementation of bisection in MATLAB. In: Jorgensen, P., Shen, X., Shu, C.-W., Yan, N. (eds.) *Recent Advances in Computational Sciences—Selected Papers from the International Workshop on Computational Sciences and Its Education*, pp. 318–332. World Scientific Pub Co Inc, Singapore (2007)
21. Chen, L.: Deriving the X-Z Identity from auxiliary space method. In: Yunqing, Huang, Ralf, Kornhuber, Olof, Widlund, Xu, Jinchao (eds.) *Domain Decomposition Methods in Science and Engineering XIX*, pp. 309–316. Springer, Berlin (2010)
22. Chen, L., Zhang, C.-S.: A coarsening algorithm and multilevel methods on adaptive grids by newest vertex bisection. *J. Comput. Math.* **28**(6), 767–789 (2010)
23. Chen, L., Nochetto, R.H., Xu, J.: Optimal multilevel methods for graded bisection grids. *Numer. Math.* **120**(1), 1–34 (2011)
24. Coomer, R.K., Graham, I.G.: Massively parallel methods for semiconductor device modelling. *Computing* **56**(1), 1–27 (1996)
25. Dryja, M., Sarkis, M.V., Widlund, O.B.: Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.* **72**(3), 313–348 (1996)
26. Golub, G.H., Van Loan, C.F.: *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
27. Graham, I.G., Hagger, M.J.: Unstructured additive schwarz-conjugate gradient method for elliptic problems with highly discontinuous coefficients. *SIAM J. Sci. Comput.* **20**, 2041–2066 (1999)
28. Graham, I., Lechner, P., Scheichl, R.: Domain decomposition for multiscale pdes. *Numer. Math.* **106**(4), 589–626 (2007)
29. Heise, B., Kuhn, M.: Parallel solvers for linear and nonlinear exterior magnetic field problems based upon coupled FE/BE formulations. *Computing*, **56**(3), 237–258 (1996). *International GAMM-Workshop on Multi-level Methods* (Meisdorf, 1994)
30. Hiptmair, R., Zheng, W.: Local multigrid in $H(\text{curl})$. *J. Comput. Math.* **27**(5), 573–603 (2009)
31. Kees, C.E., Miller, C.T., Jenkins, E.W., Kelley, C.T.: Versatile two-level Schwarz preconditioners for multiphase flow. *Comput. Geosci.* **7**(2), 91–114 (2003)
32. Kossaczky, I.: A recursive approach to local mesh refinement in two and three dimensions. *J. Comput. Appl. Math.* **55**, 275–288 (1994)
33. Meza, J., Tuminaro, R.: A multigrid preconditioner for the semiconductor equations. *SIAM J. Sci. Comput.* **17**, 118–132 (1996)
34. Mitchell, W.F.: A comparison of adaptive refinement techniques for elliptic problems. *ACM Trans. Math. Softw. (TOMS) Arch.* **15**(4), 326–347 (1989)
35. Mitchell, W.F.: Optimal multilevel iterative methods for adaptive grids. *SIAM J. Sci. Stat. Comput.* **13**, 146–167 (1992)
36. Nochetto, R., Siebert, K., Veeser, A.: Theory of adaptive finite element methods: an introduction. In: DeVore, R., Kunoth, A. (eds.) *Multiscale, Nonlinear and Adaptive Approximation*, pp. 409–542. Springer, 2009. Dedicated to Wolfgang Dahmen on the Occasion of His 60th Birthday
37. Oswald, P.: *Multilevel Finite Element Approximation. Theory and Applications*. Teubner Skripten zur Numerik. Teubner Verlag, Stuttgart (1994)
38. Oswald, P.: On the robustness of the BPX-preconditioner with respect to jumps in the coefficients. *Math. Comput.* **68**, 633–650 (1999)
39. Petzoldt, M.: A posteriori error estimators for elliptic equations with discontinuous coefficients. *Adv. Comput. Math.* **16**(1), 47–75 (2002)
40. Saad, Y.: *Iterative Methods for Sparse Linear Systems*, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
41. Sarkis, M.: Nonstandard coarse spaces and schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. *Numer. Math.* **77**(3), 383–406 (1997)
42. Scheichl, R., Vainikko, E.: Additive schwarz with aggregation-based coarsening for elliptic problems with highly variable coefficients. *Computing* **80**(4), 319–343 (2007)
43. Scott, R., Zhang, S.: Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Math. Comput.* **54**, 483–493 (1990)
44. Stevenson, R.: Stable three-point wavelet bases on general meshes. *Numer. Math.* **80**(1), 131–158 (1998)
45. Stevenson, R.: The completion of locally refined simplicial partitions created by bisection. *Math. Comput.* **77**, 227–241 (2008)

46. Vohralik, M.: Guaranteed and fully robust a posteriori error estimates for conforming discretizations of diffusion problems with discontinuous coefficients. Technical Report Preprint R08009, Laboratoire Jacques-Louis Lions (2008)
47. Vuik, C., Segal, A., Meijerink, J.A.: An efficient preconditioned cg method for the solution of a class of layered problems with extreme contrasts in the coefficients. *J. Comput. Phys.* **152**(1), 385–403 (1999)
48. Wang, J., Xie, R.: Domain decomposition for elliptic problems with large jumps in coefficients. In: *Proceedings of Conference on Scientific and Engineering Computing*, pp. 74–86. National Defense Industry Press (1994)
49. Wang, J.: New convergence estimates for multilevel algorithms for finite-element approximations. *J. Comput. Appl. Math.* **50**, 593–604 (1994)
50. Wang, Z., Wang, C., Chen, K.: Two-phase flow and transport in the air cathode of proton exchange membrane fuel cells. *J. Power Sources* **94**, 40–50 (2001)
51. Widlund, O.B.: Some Schwarz methods for symmetric and non-symmetric elliptic problems. In: Keyes, D.E., Chan, T.F., Meurant, G.A., Scroggs, J.S., Voigt, R.G. (eds.) *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pp. 19–36. SIAM, Philadelphia (1992)
52. Xu, J., Chen, L., Nochetto, R.: Optimal multilevel methods for H (grad), H (curl), and H (div) systems on graded and unstructured grids. In: *Multiscale, Nonlinear and Adaptive Approximation*, pp. 599–659. Springer, Berlin (2009)
53. Xu, J.: Counter examples concerning a weighted L^2 projection. *Math. Comput.* **57**, 563–568 (1991)
54. Xu, J.: Iterative methods by space decomposition and subspace correction. *SIAM Rev.* **34**, 581–613 (1992)
55. Xu, J.: A new class of iterative methods for nonselfadjoint or indefinite problems. *SIAM J. Numer. Anal.* **29**, 303–319 (1992)
56. Xu, J.: An introduction to multigrid convergence theory. In: Chan, R., Chan, T., Golub, G. (eds.) *Iterative Methods in Scientific Computing*. Springer, Berlin (1997)
57. Xu, J., Zikatanov, L.: The method of alternating projections and the method of subspace corrections in Hilbert space. *J. Am. Math. Soc.* **15**, 573–597 (2002)
58. Xu, J., Zhu, Y.: Uniform convergent multigrid methods for elliptic problems with strongly discontinuous coefficients. *Math. Model. Methods Appl. Sci.* **18**(1), 77–105 (2008)
59. Yserentant, H.: Old and new convergence proofs for multigrid methods. *Acta Numer.*, pp. 285–326 (1993)
60. Yserentant, H.: Two preconditioners based on the multi-level splitting of finite element spaces. *Numer. Math.* **58**, 163–184 (1990)
61. Zhu, Y.: Domain decomposition preconditioners for elliptic equations with jump coefficients. *Numer. Linear Algebra Appl.* **15**(2–3), 271–289 (2008)