Computing 69, 205–225 (2002)
Digital Object Identifier (DOI) 10.1007/s00607-002-1460-2

Computing
Printed in Austria

A Subspace Cascadic Multigrid Method for Mortar Elements

D. Braess, Bochum, **P. Deuflhard**, Berlin, and **K. Lipnikov**, Houston

Received December 14, 2001; revised September 2, 2002
Published online: November 18, 2002

Abstract

A cascadic multigrid (CMG) method for elliptic problems with strong material jumps is proposed and analyzed. Non–matching grids at interfaces between subdomains are allowed and treated by mortar elements. The arising saddle point problems are solved by a subspace confined conjugate gradient method as smoother for the CMG. Details of algorithmic realization including adaptivity are elaborated. Numerical results illustrate the efficiency of the new subspace CMG algorithm.

AMS Subject Classification: 65N55.

Keywords: cascadic multigrid method, domain decomposition, mortar elements, non-matching grids, material jumps.

1. Introduction

In this paper, we consider linear elliptic problems

$$-\mathrm{div}(a(x)\nabla u) + cu = f$$

on general domains in $\mathbb{R}^d, d = 2$ or $d = 3$, where the focus is on the case when the coefficient $a(x)$ is strongly discontinuous. Standard *multiplicative* multigrid methods [12], [25] or *additive* multilevel methods like KASKADE/BPX [15], [13] are known to converge, but to slow down whenever the material jumps are "too strong". In this case, *cascadic multigrid* (CMG) methods [14], [6], [7], [8], the youngest members of the multigrid family, are known to deteriorate in their performance – see, e.g., the numerical experiments in [16].

In order to overcome such an undesirable effect, domain decomposition seems to be a remedy. In order to allow for non-matching grids and large jumps at interfaces, mortar elements are considered. In this setting, adaptive mesh refinement can be realized on either side of the jump interfaces without penetrating into the "wrong" subdomains. Recently, a CMG algorithm for mortar elements has been suggested in [16]. In this method, however, the CMG appeared only in inner loops for each homogeneous subdomain, realized via a conjugate residual iteration. For strong jumps, however, this approach led to an algorithm not competitive with

the standard KASKADE/BPX algorithm. The purpose of the present paper is to derive a competitive CMG algorithm.

In principle, mortar elements can be implemented and analyzed either as nonconforming elements [4], [5] or as a mixed finite element method [2], [3], [10]. In both cases the Lagrange multipliers enter into the calculation of a posteriori error estimators either directly or in some concealed way. The algorithm to be suggested here provides these multipliers for an easy computation of a posteriori error estimators and thus for controlling the adaptive mesh refinement; cf. also [27] or [29]. The iteration is organized such that the mortaring conditions are automatically satisfied during the cg-iteration, i.e., the iterates stay in the subspace wherein the problem is positive definite.

Although we could follow certain ideas of [11] for the Stokes problem, there were theoretical and practical problems, the answers to which are not straightforward. Specifically, the update of the state variables can be performed in the manner well-known for cg-algorithms, whereas the Lagrange multipliers have to be treated in a different way. In fact, an iterative update of the Lagrange multipliers due to Stevenson [24] is realized. As will be documented by numerical results in Sect. 6, the CMG algorithm with a subspace confined conjugate gradient iteration suggested here, is faster than KASKADE/BPX even in the presence of strong material jumps.

The paper is organized as follows. In Sect. 2, we describe the mortar element setting in the framework of mixed methods [2], [3], [10] with piecewise constant Lagrange multipliers. In Sect. 3, we present a conjugate gradient method with iterates remaining in the subspace of those functions that satisfy the weak matching conditions at the interfaces. The extra treatment of the Lagrange multipliers is elucidated. The smoothing property of the method follows from results in [7], [9], [21]. In Sect. 4, we formulate our new *subspace* CMG method and prove its convergence for the case of quasi-uniform grids. An adaptive version of the method based on an edge-oriented error estimator in the spirit of [15] is discussed in Sect. 5. Finally, comparative numerical results for a notorious material jump problem are given in Sect. 6.

2. Mortar Element Setting

Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be a polygonal Lipschitz domain. We consider the elliptic Dirichlet problem: *find* $u \in H_0^1(\Omega)$ *such that*

$$\int_\Omega [a(x)\nabla u \nabla v + c(x)uv]\mathrm{d}\Omega = \int_\Omega fv\,\mathrm{d}\Omega \quad \forall v \in H_0^1(\Omega). \tag{2.1}$$

Here $f \in L_2(\Omega)$, $a(x)$ is a positive bounded function, and $c(x)$ is a nonnegative bounded function. We specify a non-overlapping partitioning of Ω into subdomains Ω_k:

$$\bar{\Omega} = \bigcup_{k=1}^{K} \bar{\Omega}_k.$$

Each subdomain Ω_k is covered by a triangular ($d = 2$) or tetrahedral ($d = 3$) regular mesh $\mathscr{T}_k$. Let Γ_{kl} denote the interface between the subdomains Ω_k and Ω_l, and assume that Γ_{kl} is simply connected. It is admitted that the grids $\mathscr{T}_k$ and $\mathscr{T}_l$ do not match at the interface Γ_{kl}. Following a commonly used notation, the grid on Γ_{kl} is formed by the grid points of $\mathscr{T}_k$ on the interface, i.e., Ω_k is the *non-mortar side*. The opposite side is the *mortar side* (or master side). As a standard, the mortar side is chosen to be the one with the larger (average) diffusion coefficient $a(x)$ in (2.1). The Lagrange multipliers are associated to the *non-mortar side* Ω_k, and live therefore on the side with the smaller (average) diffusion coefficient.

We denote any finite element spaces on Ω_k and Γ_{kl} by V_k and Λ_{kl}, respectively. (They will be fixed later). Moreover, let

$$V_h := \prod_{k=1}^{K} V_k, \quad \Lambda_h := \prod_{\substack{k<l \\ \Gamma_{kl}\neq\emptyset}} \Lambda_{kl}, \text{ and } X_h := V_h \times \Lambda_h.$$

In this framework, we consider the finite element problem with Lagrange multipliers at the interfaces Γ_{kl}, $k < l$: *find* $(u_h, \lambda_h) \in X_h$ *such that*

$$\left.\begin{array}{lcl} a(u_h, v_h) + b(\lambda_h, v_h) & = & f(v_h), \\ b(\mu_h, u_h) & = & 0 \end{array}\right\} \quad \forall (v_h, \mu_h) \in X_h, \tag{2.2}$$

where

$$a(u, v) := \sum_{k=1}^{K} \int_{\Omega_k} (a(x)\nabla u \nabla v + c(x)uv)\mathrm{d}\Omega,$$

$$b(\lambda, v) := \sum_{k<l} \int_{\Gamma_{kl}} \lambda_{kl}(v_k - v_l)\mathrm{d}s,$$

$$f(v) := \sum_{k=1}^{K} \int_{\Omega_k} fv \,\mathrm{d}\Omega.$$

As usual, $v_k := v|_{\Omega_k}$ and the subscript is depicted in cases of ambiguity.

There are several ways of choosing finite element spaces V_k and Λ_{kl} that satisfy the inf-sup condition [2], [3], [4], [5], [10]. In this paper, V_k will be the space of piecewise linear finite elements in $H^1(\Omega_k)$ associated with the grid $\mathscr{T}_k$ – *without* any continuity assumptions at the cross points in 2D (and edges in 3D). In contrast to the earlier paper [16], but in the spirit of suggestions due to [27], we

here select the space Λ_{kl} as *piecewise constant* functions in $L_2(\Gamma_{kl})$, and (as usual) we refer to [2] for a proof of the inf-sup condition that applies to these elements; cf. also [28, p. 30].

In Fig. 2.1 we elucidate the basis functions of Λ_{kl} for the case of a 1D interface Γ_{kl}. Let the nodes of the mesh Ω_k be located at points with Cartesian coordinates x_m, $m = 0, 1, \ldots, n_{kl} + 1$ and $0 = x_0 < x_1 < \ldots < x_{n_{kl}+1}$. Basis functions $\phi_m^{(kl)}$ of Λ_{kl} are associated only to the nodes in the interior of Γ_{kl}:

$$\phi_m^{(kl)}(x) := \begin{cases} 1, & y_m \le x \le y_{m+1},\ m = 1, 2, \ldots, n_{kl}, \\ 0, & \text{otherwise} \end{cases}$$

where $y_m := (x_{m-1} + x_m)/2$ for $m = 2, 3, \ldots, n_{kl}$ and $y_1 := x_0$, $y_{n_{kl}+1} := x_{n_{kl}+1}$.

In the 3D case the support of a basis function $\phi_1^{(kl)} \in \Lambda_{kl}$ associated with a vertex of a triangle on a 2D interface is slightly more technical (to be expressed in terms of barycentric coordinates) and is therefore omitted here.

Once the finite element spaces have been fixed, problem (2.2) results in a system of linear algebraic equations in saddle point form:

$$\mathcal{A}z := \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} =: F. \tag{2.3}$$

As usual, the matrix is computed with the (standard) nodal basis functions, and the vector norm is equivalent to the L_2 norm of the finite element function. The submatrix A is a positive definite block diagonal matrix

$$A = \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_K \end{bmatrix}.$$

Let n_k be the size of the matrix A_k, $n_u := \sum_{k=1}^{K} n_k$ the size if u, n_λ the size of λ, and $N := n_u + n_\lambda$ the total problem size of z. We conclude from the inf-sup condition that problem (2.3) has a unique solution. In particular, B is a matrix with full rank.

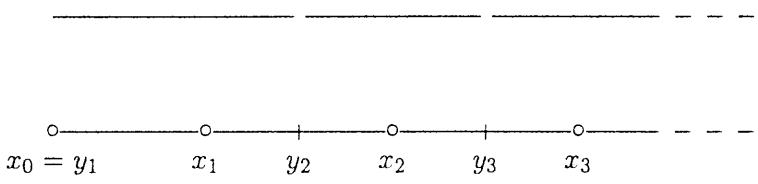

Fig. 2.1. Nodal basis functions of Λ_{kl} next to a cross point for a staggered grid

3. Subspace Confined Conjugate Gradient Iteration

The system of linear equations (2.3) characterizes the solution of the constrained minimization problem

$$\min_v \{(Av, v) - 2(f, v)\} \text{ subject to } Bv = 0. \tag{3.1}$$

As is well–known, this problem is equivalent to an spd–problem on the subspace

$$U := \{v : \ Bv = 0\}. \tag{3.2}$$

Therefore these equations can, in principle, be solved by any preconditioned conjugate gradient (PCG) method realized within the subspace U. In order to confine all iterates to this subspace, we follow [11] and introduce a preconditioner that replaces the matrix $\mathscr{A}$ by

$$\mathscr{H} := \begin{bmatrix} D & B^T \\ B & 0 \end{bmatrix}.$$

Here D should be some positive definite matrix having a simple structure and satisfying the inequality

$$(Dv, v) \geq (Av, v) \qquad \forall v \in \mathbb{R}^{n_u}. \tag{3.3}$$

In the spirit of [11], the 2×2 block system (2.3) is solved iteratively. We introduce the corresponding splitting $z^i = (u^i, \lambda^i)$ for the block variables at iteration step i $(i = 0, 1, \ldots)$ and $r^i = F - \mathscr{A} z^i =: (r_u^i, r_\lambda^i)$ for the block residuals. With this notation, we are now ready to write our suggested PCG method as follows:

(1) *Initial guess:* Fix $\tilde{z} := (\tilde{u}, \tilde{\lambda})$, where $\tilde{u}$ need *not* belong to the subset U.

(2) *Subspace entering:*

$$\begin{aligned} r^0 &= -(0, B\tilde{u}), \\ z^{00} &=: (u^{00}, \lambda^{00}) = \tilde{z} + \mathscr{H}^{-1} \ r^0, \\ z^0 &=: (u^0, \lambda^0) = (u^{00}, \tilde{\lambda}). \end{aligned} \tag{3.4}$$

Moreover, set $p^0 = 0$ and $\gamma_0 = 0$.

(3) *Subspace iteration:* $i = 1, 2, \ldots$

$$\begin{aligned} u^i &= u^{i-1} + \alpha_{i-1} p^{i-1}, \\ \lambda^i &= \lambda^{i-1} + s_\lambda^{i-1}, \\ r^i &= F - \mathscr{A} z^i \text{ with } z^i = (u^i, \lambda^i), \\ s^i &= (s_u^i, s_\lambda^i) = \mathscr{H}^{-1} r^i, \\ p^i &= s_u^i + \gamma_{i-1} p^{i-1}, \end{aligned} \tag{3.5}$$

where

$$\alpha_i = \frac{\sigma_i}{(Ap^i, p^i)}, \quad \gamma_{i-1} = \frac{\sigma_i}{\sigma_{i-1}}, \quad \sigma_i = (s^i, r^i). \tag{3.6}$$

The above iteration actually confines the iterates u^i to the subspace U as desired. In fact, a short calculation verifies that $u^0 \in U$ as well as $s_u^i \in U, i \geq 0$; cf. [11]. The relations $p^i \in U$ and $u^i \in U$ then follow by induction. The formula (3.5) is derived from the well-known procedures for preconditioned conjugate gradients, and s_u^i takes the role of the *preconditioned gradient*. The iteration can be understood as a conjugate gradient method for the u-variable with a preconditioner that is built by the matrix D and a projection.

Note that in the above subspace iteration only the u-components are computed by the rules of cg-iteration, whereas the λ-components are evaluated by the underlying Jacobi iteration. This is motivated by STEVENSON's observation that the Jacobi iteration yields the correction of the Lagrange multiplier that minimizes the residue with respect to the norm $|\cdot|_{D^{-1}} := (\cdot, D^{-1}\cdot)^{1/2}$. The computation uses only terms that are anyway available.

Remark 3.1: (Stevenson [24]): Let u^i, λ^i be approximate solutions of the saddle point problem (3) and let r^i, s^i be defined by (3.5). Then the minimization problem (with $u = u^i$ fixed)

$$\left|Au^i + B^T\lambda - f\right|_{D^{-1}} \to \min_{\lambda}$$

is solved by $\lambda = \lambda^i + s_\lambda^i$.

This statement is verified by a simple calculation. Indeed, we obtain $s^i = \mathscr{H}^{-1} r^i$ in (3.5) by solving the equation

$$BD^{-1}B^T s_\lambda^i = BD^{-1} r_u^i. \tag{3.7}$$

The solution of this equation characterizes the minimum of $|B^T\lambda - r_u^i|_{D^{-1}}$, i.e. $|B^T s_\lambda^i - r_u^i|_{D^{-1}} \leq |B^T\lambda - r_u^i|_{D^{-1}}$ for all λ. Since $r_u^i = f - Au^i - B^T\lambda^i$, the proof is complete. □

In the subspace entering part (3.4) and in the subspace iteration (3.5) there are terms with $\mathscr{H}^{-1}$. We need to solve interface equations of the type

$$BD^{-1}B^T\lambda = g \tag{3.8}$$

for their implementation. The numerical solution of this equation can be done as follows: (a) On the coarsest grid, it is solved directly. (b) On finer grids in 2D, *static condensation* towards the cross points may be applied (cf. [27]), i.e., the values at the cross points are eliminated last in a Gaussian elimination process. (c)

In 3D, iterative solvers are used. – In our algorithm, we have implemented iterative solvers both for 2D and 3D problems. The (inner) iteration was terminated as soon as the condition

$$\left\|Bu^i\right\| \leq 10^{-2}\left\|Bu^0\right\| \tag{3.9}$$

has been passed. This part consumed only 4–8% of the total computing time and is therefore comparatively cheap. The accuracy supplied by the iteration turned out to be sufficient within the multigrid setting to be discussed in Sect. 4. For more investigations of approximate solutions of (3.8) the reader is referred to Zulehner's theory [30] and for the controlled termination of the iteration to [26].

Remark 3.2: The influence of the Lagrange multipliers on the computation is greater than one may imagine from the theoretical considerations. In principle, u^i and λ^i are independent of λ^{i-1} if (3.7) is solved exactly, and one might drop the specifications of λ^0 and λ^i in the definition of the PCG algorithm. Since (3.7) is solved only approximately, however, we are interested in small right hand sides of the equations, i.e. small residues r^i. Good values of λ^i are desired for this computational reason, and when calculating λ^i due to Remark 3.1 we are close to the optimal multiplier with respect to a minimal residue.

The Lagrange multipliers must not be updated in the subspace entering process because of the danger of overshooting. That danger exists in calculations with the matrix $\mathscr{H}$ when we operate with u vectors that are not contained in the subspace U. This follows from the influence of the relaxation parameters on $\mathscr{H}^{-1}$; cf. (3.5) in [7].

Once the iteration is computationally defined, we want to analyze its iterative convergence behavior. For this purpose, we introduce the matrix

$$R := I - D^{-1}B^T\left(BD^{-1}B^T\right)^{-1}B. \tag{3.10}$$

In particular, we have $u^0 = R\tilde{u}$ in the subspace entering procedure. The following properties imply that R is the *D-orthogonal projection* onto U. It is *symmetric* with respect to the scalar product $(\cdot,\cdot)_D$ since

$$\left(RD^{-1}\right)^T = RD^{-1} \text{ and } (DR)^T = DR. \tag{3.11}$$

Recalling (3.3) and using straightforward calculations we obtain

$$|R|_D \leq 1, \quad |I - D^{-1}A|_D \leq 1, \text{ and } |I - D^{-1}A|_A \leq 1. \tag{3.12}$$

in terms of the induced norm $|\cdot|_D^2 := (\cdot, D\cdot)$.

Since $u^i, u \in U$, the iterative errors $u^i - u$, $i \geq 1$, are known to be independent of the error of the Lagrange multiplier λ^i; cf. [11]. They are given by the formulas:

$$u^0 - u = R(\tilde{u} - u), \tag{3.13}$$

$$u^i - u = \left(u^{i-1} - u\right) + \alpha_{i-1} p^{i-1}, \tag{3.14}$$

$$\begin{aligned} p^i &= R(I - D^{-1}A)(u^{i-1} - u) - (u^{i-1} - u) + \gamma_{i-1} p^{i-1} \\ &= -RD^{-1}A(u^{i-1} - u) + \gamma_{i-1} p^{i-1}, \quad i = 1, 2, \ldots \end{aligned}$$

where the factors α_i and γ_i are given by (3.6). After m iterative steps we thus arrive at

$$u^m - u = S_m\left[RD^{-1}A\right]\left(u^0 - u\right), \tag{3.15}$$

where $S_m[\cdot]$ is a polynomial of degree m. Because of (3.11), we have a pcg-algorithm with the symmetric, positive definite matrix RD^{-1} as a preconditioner. From (3.3) we even conclude that the spectrum of $A^{1/2}RD^{-1}A^{1/2}$ is contained in the interval $[0, 1]$. Due to the well-known optimality of the cg iterations in the A-norm,

$$|u^m - u|_A = \min_{q_m \in \mathscr{P}_m} \left|q_m[RD^{-1}A](u^0 - u)\right|_A$$

where $\mathscr{P}_m$ denotes the subset of those polynomials q_m with degree less or equal to m which satisfy $q_m(0) = 1$. In the context of cascadic algorithms, the following polynomial operator is used for upper estimates of the error [21], [7].

Lemma 3.3: There exists a linear operator $L_m = q_m[RD^{-1}A]$ with $q_m \in \mathscr{P}_m$ such that for all $v \in U$ we have

$$|L_m v|_A \leq \frac{1}{2m+1} |v|_D \text{ and } |L_m v|_A \leq |v|_A. \tag{3.16}$$

The m-asymptotics in (3.16) has been shown in [21], [7] to open the door for the construction of a cascadic algorithm of *optimal complexity* already in 2D. A smoother like Gauss–Seidel relaxation or Gauss–Jacobi relaxation would at best lead to some *nearly optimal* code in 2D – compare [7, Lemma 1.1]. In 3D, however, any smoother mentioned above is optimal in terms of the energy norm with differences only in the leading coefficient.

Up to now, we have not yet specified the matrix D. The simplest (local) choice certainly is just $\alpha_k I_k$, a multiple of the identity matrix, as suggested in [9] for the Stokes problem. Since we admit large changes of the diffusion coefficient, however, a more suitable choice appeared to be the (still simple) *diagonal matrix*

$$D := 2\,\mathrm{diag}(A). \tag{3.17}$$

Remark: We also experimented with matrices which are used as preconditioners for one-level cg iterations. Small-rank perturbations of diagonal matrices have been proposed by KUZNETSOV [19], [20] in that context. Let $\mathring{A}_k$ denote the special

stiffness matrix for the elliptic equation (2.1) with the coefficient $c_k = 0$ and M_k be the corresponding matrix for the *pure Helmholtz* case. Then, the vector w_k spanning $\mathring{A}_k$ and being normalized according to $w_k^T M_k w_k = 1$ yields the projections $P_k := w_k w_k^T$ with $P_k M_k = (M_k P_k)^T$. In our notation, the preconditioner suggested in [19, 20] associated to the subdomain Ω_k reads

$$D_k^{-1} = (I_k - P_k M_k) D_{A_k}^{-1} (I_k - M_k P_k) + \frac{1}{c_k} P_k. \tag{3.18}$$

In our numerical experiments including example (6.1) below, however, the diagonal preconditioner (3.17) has clearly outperformed the variant (3.18). The reason is that the iteration is here required to serve *as a smoother* within a multilevel algorithm rather than *as a preconditioner*. Moreover, the specification (3.18) would cause trouble in the (Helmholtz-free) case with $c_k = 0$. In order to be able to run a comparison between the two preconditioners (3.17) and (3.18), we have chosen the test example (6.1) in Sect. 6 with a "small" Helmholtz term.

4. Multigrid Convergence Analysis

As for the finite element solution of problem (2.2), there are quite a number of *a priori error estimates* for the case of mortar elements with *piecewise linear* Lagrange multipliers [3], [4], [5], [10], [27]. For *piecewise constant* multipliers as considered in the present paper, the theory can be easily extended using arguments from the CROUZEIX–RAVIART element. We recall the basic result here; cf. [2], [27], [28].

Lemma 4.1: Assume that the solution of problem (2.1) is in $H_0^1(\Omega) \cap \prod_{k=1}^{K} H^2(\Omega_k)$. Let $\bar{h} := \max_k h_k$. Then

$$\sum_{k=1}^{K} \left(\bar{h}^{-1} \|u - u_h\|_{L_2(\Omega_k)} + \|u - u_h\|_{H^1(\Omega_k)} \right) \leq C \sum_{k=1}^{K} h_k \|u\|_{H^2(\Omega_k)}. \tag{4.1}$$

We will employ Lemma 4.1 later to analyze our *subspace cascadic multigrid algorithm* for mortar elements and its convergence properties for nested *quasi-uniform* grids.

The definition of the multilevel procedure requires some notation. For $k = 1, \ldots, K$, we identify $\mathcal{T}_k$ with the coarse triangulation of Ω_k. For refinement levels $j = 0, 1, \ldots, J$, a nested family of finite element spaces $X_0 \subset X_1 \subset \ldots \subset X_J$ is defined with

$$X_j := V_j \times \Lambda_j \text{ and } V_j := \prod_{k=1}^{K} V_{kj}.$$

As a natural generalization of (3.2), the subspace U_j of V_j will contain those finite element functions on the level j that satisfy the mortar conditions.

In pseudocode formulation our subspace cascadic multigrid (for short SCMG) algorithm reads (compare [9])

$$\begin{aligned} j = 0: \quad & u_0^*,\ \lambda_0^* \\ & \text{direct solution of the saddle point problem} \\ & \text{on the coarse grid;} \end{aligned} \tag{4.2}$$

$$\begin{aligned} j = 1, \ldots, J: \quad & u_j^* = \mathscr{I}_{j,m_j} R_j u_{j-1}^* \\ & \text{iterative solution of saddle point problems} \\ & \text{on successively finer grids.} \end{aligned} \tag{4.3}$$

The first step on each refinement level $j = 1, \ldots, J$ is to prolongate the (approximate) solution $u_{j-1}^*, \lambda_{j-1}^*$ from the previous coarser level for use as the starting point of the iteration on the level j. It is done simply by interpolation. The operator R_j performs the projection to the subspace U_j, as specified by the subspace entering process $\tilde{u} \mapsto u^0$ in (3.4). Similarly, the operator $\mathscr{I}_{j,m_j}$ represents m_j subspace iterations (3.5) on the level j.

The choice of the prolongation of the Lagrange multipliers has very little impact on the vectors in the iteration. We need only reasonable approximations in order to make the inner iteration efficient as discussed in Remark 3.2. Therefore it is sufficient to take the mean value of the multipliers from adjacent nodes whenever a node in the fine grid is not a node on the previous level.

For the analysis of this iteration, let h_{kj} be the discretization parameter associated with V_{kj} and $h_k := h_{kJ}$. As usual $u_j \in V_j$ denotes the finite element solution of the saddle point problem (2.2) and N_j the dimension of the vectors on the level j just as defined at the end of Sect. 2.

Let $\|\cdot\|_a$ denote the energy norm induced by the bilinear form $a(\cdot,\cdot)$, and let $|\cdot|_A$ be the induced norm of its vector representation, i.e.,

$$\|v\|_a = |v|_A, \quad \text{for } v \in V_h.$$

Moreover, since the meshes are shape regular, we have

$$\|v_k\|_{L_2(\Omega_k)} \sim h_k |v_k|_{D_k}. \tag{4.4}$$

Let $\bar{h}_j$ denote the maximal mesh size of the triangles on the level j. The general *quasi-uniformity* assumption is

$$\frac{1}{C}\bar{h}_j \le h_{kj} \le \bar{h}_j, \quad 1 \le k \le K,$$

with a constant $C > 0$. For nested meshes, this assumption is often replaced by the relation $h_{kj} \approx 2^{J-j} h_k$. Recall from Lemma 3.3 that

$$\left\|L_{m_j} v_j\right\|_a \leq \frac{C}{m_j} h_j^{-1} \left\|v_j\right\|_{L_2(\Omega)} \text{ and } \left\|L_{m_j} v_j\right\|_a \leq \left\|v_j\right\|_a \quad \forall v_j \in U_j.$$

An L_2 estimate of the following kind is typical for the analysis of cascadic multigrid algorithms with nonconforming or mixed elements (cf. [9]).

Lemma 4.2: Assume that the given problem (2.1) is H^2 regular. There is a linear mapping $F_j : V_j \to U_j$ such that $\|F_j v_j\|_a \leq \|v_j\|_a$ for all $v_j \in V_j$ and

$$\left\|\left(I - F_j\right) v_j\right\|_{L_2(\Omega)} \leq C \bar{h}_j \left\|v_j\right\|_a \text{ for all } v_j \in U_{j-1}.$$

The idea of the proof of this lemma is to apply the FORTIN interpolation operator for the mapping F_j as in Lemma 2 of [9]. The desired L_2 estimate can then be obtained by a standard duality argument; cf. [9, chapter 7]. For this reason, the quite similar proof for mortar elements is omitted here.

Admittedly, our above assumption of H^2-regulararity is rather strong – in particular, as we aim at discontinuous coefficients. The theoretical problems connected with strong jumps are a well-known subtle issue in all kinds of multigrid convergence analysis and would also deserve more attention in the present context. For the time being, however, the interested reader may recur to the thorough papers by M. Dryja (e.g., [17]) to get an idea of how much extra analytical work would be involved to tackle this question. Moreover the extension to the less regular case has been exemplified for cascadic algorithms in [22], [23]. – The numerical experiments to be reported in Sect. 6 include strong material jumps.

With these preparations, we are now ready to state the main convergence estimate for the SCMG method.

Theorem 4.3: Let $\bar{h}_j = 2^{J-j} \bar{h}_J$, $j = 0, 1, \ldots, J$, and $2 < \beta < 2^d$ for $d = 2, 3$. If the numbers of iteration steps are chosen according to

$$m_j := \left\lceil m_J \beta^{J-j} \right\rceil, \tag{4.5}$$

then the final error of the subspace cascadic multigrid method is bounded by

$$\left\|u_J^* - u_J\right\|_a \leq C \frac{\bar{h}_J}{1 - 2/\beta} \|f\|_{L^2(\Omega)} \tag{4.6}$$

with $C = C(m_J)$, and the computational complexity is bounded by

$$\sum_{j=1}^{J} m_j N_j \leq C \frac{m_J N_J}{1 - \beta/2^d}. \tag{4.7}$$

Proof: The proof follows closely the proof of Proposition 3 in [9], and we have only to generalize it to the more general preconditioner here. For $j = 1, 2, \ldots, J$ we have

$$u_j^* - u_j = \mathcal{I}_{j,m_j} R_j u_{j-1}^* - u_j = S_{j,m_j}\Big(R_j J_j\Big(u_{j-1}^* - u_j\Big)\Big).$$

Here $J_j := I - D_j^{-1} A_j$, and R_j denotes the D_j-orthogonal projection operator onto the subspace U_j as introduced in (3.10). The error propagation operator S_{j,m_j} for the PCG method is found in (3.15). Let L_{j,m_j} be the linear operator appearing in Lemma 3.3. By applying Lemma 3.3, Lemma 4.1, and estimate (3.12) for the operator J_j, we have

$$\begin{aligned}
\Big\|u_j^* - u_j\Big\|_a &\le \Big\|L_{j,m_j}\Big(R_j\Big(u_{j-1}^* - u_j\Big)\Big)\Big\|_a \\
&\le \big\|L_{j,m_j}\big(R_j(u_{j-1} - u_j)\big)\big\|_a + \Big\|L_{j,m_j}\Big(F_j\Big(u_{j-1}^* - u_{j-1}\Big)\Big)\Big\|_a \\
&\quad + \Big\|L_{j,m_j}\big(F_j - R_j\big)\Big(u_{j-1}^* - u_{j-1}\Big)\Big\|_a \\
&\le \frac{C}{m_j}\big|R_j(u_{j-1} - u_j)\big|_{D_j} + \Big\|\Big(u_{j-1}^* - u_{j-1}\Big)\Big\|_a \\
&\quad + \frac{C}{m_j}\Big|\big(R_j - F_j\big)\Big(u_{j-1}^* - u_{j-1}\Big)\Big|_{D_j}.
\end{aligned} \tag{4.8}$$

The difference $u_j - u_{j-1}$ can be expressed in terms of the discretization error at the levels j and $j-1$. Since we have restricted ourselves to the case of full regularity, we obtain from (3.12), (4.4), and Lemma 4.1

$$\begin{aligned}
\big|R_j(u_{j-1} - u_j)\big|_{D_j} &\le |u_{j-1} - u_j|_{D_j} \\
&\le C\bar{h}_j^{-1}\big\|u_{j-1} - u_j\big\|_{L_2(\Omega)} \le C\bar{h}_j \sum_{k=1}^{K}\big\|u_{jk}\big\|_{H_2(\Omega_k)} \\
&\le C\bar{h}_j\|f\|_{L_2(\Omega)}.
\end{aligned}$$

Moreover, by exploiting (3.12) for R_j and J_j, applying an inverse inequality, and using the second inequality in Lemma 4.2 we conclude that

$$\begin{aligned}
\Big|\big(R_j - F_j\big)\Big(u_{j-1}^* - u_{j-1}\Big)\Big|_{D_j} &= |R_j(I - F_j)(u_{j-1}^* - u_{j-1})|_{D_j} \\
&\le \Big|\big(I - F_j\big)\Big(u_{j-1}^* - u_{j-1}\Big)\Big|_{D_j} \\
&\le C\bar{h}_j^{-1}\Big\|\big(I - F_j\big)\Big(u_{j-1}^* - u_{j-1}\Big)\Big\|_{L_2(\Omega)} \\
&\le C\Big\|\Big(u_{j-1}^* - u_{j-1}\Big)\Big\|_a \\
&\le C\Big\|u_{j-1}^* - u_{j-1}\Big\|_a.
\end{aligned}$$

Inserting this into (4.8) we obtain

$$\left\|u_j^* - u_j\right\|_a \leq C\frac{\bar{h}_j}{m_j}\|f\|_{L_2(\Omega)} + \left(1 + \frac{C}{m_j}\right)\left\|u_{j-1}^* - u_{j-1}\right\|_a. \tag{4.9}$$

Before summing up terms by virtue of this recursion relation, we note that the iteration numbers m_j decrease so fast that (4.5) implies $\sum_{j=1}^{J}(1/m_j) \leq 2/m_J$. Hence, $\prod_{j=1}^{J}(1 + C/m_j) \leq \exp(2C/m_J)$. Having these tools we finally estimate

$$\begin{aligned}\left\|u_J^* - u_J\right\|_a &\leq C\sum_{j=1}^{J}\frac{\bar{h}_j}{m_j}\|f\|_{L_2(\Omega)}\prod_{\ell=j+1}^{J}\left(1 + \frac{C}{m_\ell}\right)\\ &\leq C\exp(2C/m_J)\sum_{j=1}^{J}\frac{\bar{h}_j}{m_j}\|f\|_{L_2(\Omega)}.\end{aligned} \tag{4.10}$$

Since $\beta > 2$, the sum in (4.10) is a geometric series that leads to the inequality (4.6). Similarly, the assumption $\beta < 2^d$ guarantees that the sum in (4.7) is also bounded by a convergent geometric series. □

The exponential factor e^C in (4.10) with a problem dependent generic constant C seems to be unavoidable for nonconforming methods, compare [9], [23]. The question is whether this quantity might be "large" in actual problems. Our numerical experiments done so far (not only those presented in Sect. 6) indicate that this factor is typically tolerable and does not lead to a significant slow down of the adaptive method. It is as efficient as in the conforming case.

5. Realization of an Adaptive Version

In this section, we derive an adaptive mesh refinement strategy in close analogy to the derivation in [7]. Assume that up to the level $j-1$ such a strategy has already led to a triangulation satisfying the assumptions

$$h_\tau^{-1}\left\|u_j - u_{j-1}\right\|_{L_2(\tau)} \leq C\left\|u_j - u_{j-1}\right\|_{H^1(\tau)}, \tag{5.1}$$

$$\left\|u_j - u_{j-1}\right\|_a \leq CN_j^{-1/d}\|f\|_{L_2(\Omega)}, \tag{5.2}$$

where τ is an arbitrary element (triangle in 2D or tetrahedron in 3D) and $h_\tau := diam(\tau)$. We refer to [12] for a theoretical justification. Inequality (5.1) means that the finite element correction is locally of high frequency with respect to the finer triangulation. Inequality (5.2) is the assumption of optimal global accuracy. The assumptions above are stronger than inequality (4.1) in Lemma 4.1. Considerations similar to those that led to Theorem 4.3 now yield the result

$$\|u_j^* - u_j\|_a \leq C(m_J)\sum_{j=1}^{J}\frac{N_j^{-1/d}}{m_j}\|f\|_{L_2}. \tag{5.3}$$

With this estimate we are in the setting of [7]. Hence, we can apply the same strategy as suggested there.

The termination criterion developed in [7] is based on a recursion formula similar to (4.9): Let ϵ_{j-1} denote an estimate of the *discretization error* $\|u_{j-1} - u\|_a$, e.g. an a posteriori estimator, which can usually be provided by an adaptive multilevel algorithm. Let δ_j denote an appropriate estimate of the *algebraic error* $\|u_j - u_j^*\|$. Then the threshold for terminating the iteration on the level j appears as

$$\delta_j \le \delta_{j-1} + \rho\left(\frac{\mathrm{TOL}}{\epsilon_{j-1}}\left(\frac{N_j}{N_{j-1}}\right)^{1/d}\right)^{(d+1)/2}\epsilon_{j-1},$$

where ρ is a safety factor, $\rho < 1$, and TOL is some user prescribed error tolerance such that $\epsilon_J \le \mathrm{TOL}$ is to be reached on the final level J.

In [27] the *edge-oriented* a posteriori error estimator due to [15] is naturally transferred to the case of mortar elements. It is based on a hierarchical extension of the space of *linear* finite elements, say V_{kL}, by a space of *quadratic functions*, say V_{kQ}, living on the edges of the grid Ω_k. Each quadratic "bubble function" in V_{kQ} vanishes at the vertices of Ω_k and is parametrized by its midpoint values on the edge. Let

$$V_L := \prod_{k=1}^{K} V_{kL} \quad V_Q := \prod_{k=1}^{K} V_{kQ}, \quad V_h := V_L \oplus V_Q, \quad X_h = V_h \times \Lambda_h.$$

Note that an extension for the Lagrange multipliers is not needed since these are anyway defined via the traces of the associate subdomain grids.

Then the finite element problem in X_h with $u_L \in V_{kL}$, $u_Q \in V_{kQ}$ leads to the algebraic equations:

$$\begin{bmatrix} A_{LL} & A_{LQ} & B_L^T \\ A_{QL} & A_{QQ} & B_Q^T \\ B_L & B_Q & 0 \end{bmatrix} \begin{bmatrix} u_L \\ u_Q \\ \lambda \end{bmatrix} = \begin{bmatrix} f_L \\ f_Q \\ 0 \end{bmatrix}.$$

Let $x_L^* = (u_L^*, \lambda^*)$ be an approximate solution with linear elements obtained by the SCMG method. Upon introducing the defects $d_Q := u_Q$ for the discretization error and $d_L := u_L - u_L^*$, $d_\lambda := \lambda - \lambda^*$ for the iterative errors, we arrive at the system

$$\begin{bmatrix} A_{LL} & A_{LQ} & B_L^T \\ A_{QL} & A_{QQ} & B_Q^T \\ B_L & B_Q & 0 \end{bmatrix} \begin{bmatrix} d_L \\ d_Q \\ d_\lambda \end{bmatrix} = \begin{bmatrix} r_L \\ r_Q \\ r_\lambda \end{bmatrix}, \tag{5.4}$$

with the residues

$$r_L := f_L - A_{LL}u_L^* - B_L^T\lambda^*, \quad r_Q := f_Q - A_{QL}u_L^* - B_Q^T\lambda^*, \text{ and } r_\lambda := -B_L u_L^*.$$

Of course, we do not aim at an exact solution of equation (5.4), but only at a rough approximation for the mere purpose of mesh refinement. An appropriate estimator can be obtained from the simpler system

$$\begin{bmatrix} A_{LL} & 0 & 0 \\ 0 & D_{QQ} & 0 \\ 0 & 0 & S_Q \end{bmatrix} \begin{bmatrix} \tilde{d}_L \\ \tilde{d}_Q \\ \tilde{d}_\lambda \end{bmatrix} = \begin{bmatrix} r_L \\ r_Q \\ r_\lambda \end{bmatrix}, \tag{5.5}$$

where D_{QQ} is just the diagonal part of A_{QQ} and

$$S_Q = \begin{bmatrix} B_L & B_Q \end{bmatrix} \begin{bmatrix} A_{LL} & A_{LQ} \\ A_{QL} & A_{QQ} \end{bmatrix}^{-1} \begin{bmatrix} B_L^T \\ B_Q^T \end{bmatrix}.$$

As has been shown for shape regular triangulations in [15], the block diagonal matrix diag $\{A_{LL}, D_{QQ}\}$ is spectrally equivalent to the corresponding 2×2 block matrix in (5.4). Therefore the stiffness matrices in (5.4) and (5.5) are also spectrally equivalent; see [19]. As a consequence, the energy norm $\|d_Q\|_a$ of the discretization error can be estimated roughly by

$$\|d_Q\|_a \approx \|\tilde{d}_Q\|_{D_{QQ}}.$$

As usual, the global discretization error estimator is given as the sum over all local contributions on the edges of the triangulations Ω_k.

On the basis of this error estimation technique, we suggest the following *first step* of our mesh refinement strategy. Let η_e be a local error estimator living on the edge e of Ω_k, which can be either matching or non-matching. Then those edges with

$$\eta_e \geq \frac{1}{4} \max_{e'} \eta_{e'}.$$

are marked for refinement.

Note that due to the decoupling of the defects in (5.5), the defect estimate $\tilde{d}_\lambda$ need not be computed at all. Hence, the adaptive strategy so far does not monitor the mortar edges in particular. This led us to propose a *second step* of mesh refinement strategy. For this purpose, consider the functional

$$\Phi(u) := (A_{LL}u, u) - 2(f_L, u)$$

that had already appeared in (3.1). Recall that the solution u_L of a saddle point problem is a minimizer of $\Phi(u)$ subject to the constraint $Bu = 0$. From variational calculus, we know that

$$B^T \lambda_L = \left. \frac{\partial \Phi(u)}{\partial u} \right|_{u=u_L}$$

is the *sensitivity* of the functional with respect to local changes of u_L. Let θ_e be a *sensitivity measure* at $u = u_L$ related to an edge e: for piecewise constant Lagrange multipliers as used here we may set

$$\theta_e := B^T \lambda_L \big|_e [u_L]_e, \tag{5.6}$$

where $[u]_e$ denotes an average absolute value of the jump of u_L at e. In order to select the "most sensitive" edges (with respect to changes in the constrained functional), we mark those edges for additional refinement which satisfy

$$\theta_e \geq 0.95 \max_{e'} \theta_{e'}.$$

For the sake of completeness, we mention that we had also experimented with the quadratic bubbles $d_Q\big|_e = u_Q\big|_e$ replacing the jumps $[u_L]_e$ in (5.6). We obtained nearly the same numerical results. For this reason we stick to the concept above.

6. Numerical Experiments

In this section, we want to illustrate the performance of our adaptive subspace cascadic multigrid algorithm (SCMG) with CG as selected smoother. An implementation of this algorithm is compared with the following two adaptive multilevel methods:

(i) the best DD/CCG method from [16], and

(ii) the code Kaskade with BPX as preconditioner.

The DD/CCG method is a domain decomposition method combined with *cascadic* multigrid methods on *convex* subdomains with homogeneous materials; it also allows for *non-matching* grids as the method presented herein, but it uses an indefinite iterative solver. The Kaskade code is an implementation of an additive multilevel method on matching grids. In 2D the BPX preconditioner could, in principle, be replaced by a hierarchical basis preconditioner – which has not been done since our intention is the design of an efficient 3D code.

The outer iterations in SCMG were terminated by the condition $\|u - u_h\|_a \leq 0.02\|u\|_a$, whereas the inner iterations were terminated by the requirement (3.9).

Remark: We wish to explicitly mention that in our actual numerical computations we have used a subspace entering procedure differing slightly from (3.4): A relaxation step before the application of the projection R_j is included. Later the

referees noted that this extra step spoiled our former proof. As a consequence, we changed the convergence theory slightly in this respect, but did not want to change all our numerical experiments since the changes are too marginal.

Notorious test problem: We have chosen a relatively simple test problem from the literature [15] – adding a "small" perturbation term as in [16] in view of possible comparisons with the alternative preconditioner (3.18) that we abandoned afterwards. Consider the domain $\Omega = [0,1]^d$ and the elliptic equation

$$\begin{aligned} -\mathrm{div}(a(x)\nabla u) + 10^{-4}u &= 100 \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega \end{aligned} \tag{6.1}$$

with material jumps modelled by

$$a(x) = \begin{cases} a_0 := 1 & \text{if } x \in [0.25, 0.75]^d \setminus [0.375, 0.625]^d, \\ a_1 := 10^6 & \text{otherwise.} \end{cases}$$

In order to study the influence of jumps, variations of the coefficient a_1 were also included in our computations.

For the application of mortar elements, the domain Ω is decomposed into three subdomains according to the jumps of the diffusion coefficient – see Fig. 6.1. Note that for the earlier DD/CCG method, these subdomains have to be decomposed further into convex subdomains; see [16]. As already stated in Sect. 2, the Lagrange multipliers are attributed to the sides with the smaller diffusion coefficients. In Fig. 6.2, we compare the non-matching grids arising from the SCMG method with the matching grids from `Kaskade/BPX`. Obviously, `Kaskade/BPX` produces excess refinements near interfaces between "fine grid" and "coarse grid"

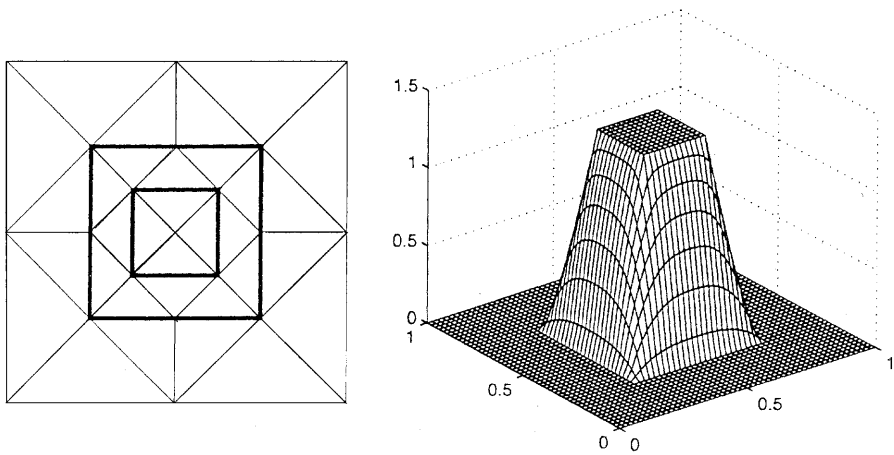

Fig. 6.1. Domain decomposition with initial grid and the solution for $a_1 := 10^6$

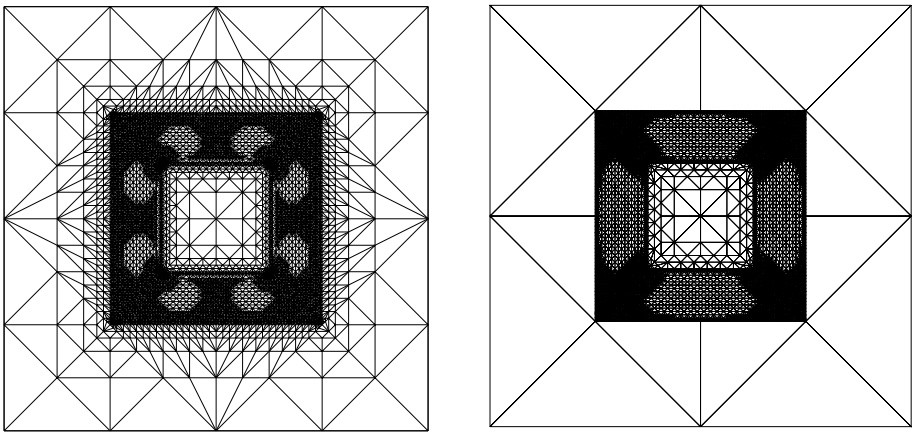

Fig. 6.2. Adaptive grids from KASKADE/BPX (left) and from the SCMG method (right)

subdomains – an effect that is even more severe in 3D problems. The non-matching grids and the new SCMG algorithm, however, lead to a more flexible adaptive algorithm which is also better parallelizable.

In Table 6.1 below we list the numerical results for this problem when solved by the three adaptive multilevel methods above. Obviously, the new SCMG version gains a factor of about 3 compared to the older CCG method. The new method turns out to be also faster than `Kaskade/BPX` with non-matching grids. We note that about 60% of the computation time is spent on the two finest grids where only 2 iteration steps are required.

Next, we studied the dependence of the new method (for $d = 2$) upon *variations of the jumps* in the diffusion coefficients. Results for $a_1 = 10^6$, 10^3, and 10^0, resp., are presented in Fig. 6.3.

Table 6.1. Comparison of three adaptive multilevel methods to solve problem (6.1) with $a_1 = 10^6$. [*j*: level – *N*: number of variables – *acc*: energy norm accuracy – *itr*: number of outer iterations – *inn*: number of inner iterations – *time*: computation time.]

	Mortar elements								Standard elements			
	DD /CCG			SCMG					KASKADE/BPX			
j	itr	*N*	time	Itr	*N*	acc	time	inn	itr	*N*	acc	time
1	15	87	0.17	5	87	0.195	0.06	6	4	45	0.386	0.02
2	14	163	0.41	7	173	0.126	0.09	8	4	145	0.256	0.07
3	12	319	0.78	10	333	0.100	0.17	13	5	257	0.120	0.17
4	10	521	1.08	10	537	0.077	0.31	14	4	489	0.081	0.34
5	12	759	1.63	7	771	0.060	0.49	15	6	685	0.086	0.62
6	10	1021	2.33	9	939	0.053	0.72	12	4	973	0.048	1.01
7	6	1657	3.22	5	1559	0.040	1.06	12	4	1821	0.032	1.74
8	12	2717	5.47	4	2435	0.030	1.67	11	3	2477	0.031	2.77
9	6	3463	7.42	2	3455	0.025	2.51	14	3	4153	0.024	4.51
10	6	7109	12.2	2	5683	0.019	4.74	14	2	6313	0.016	7.22

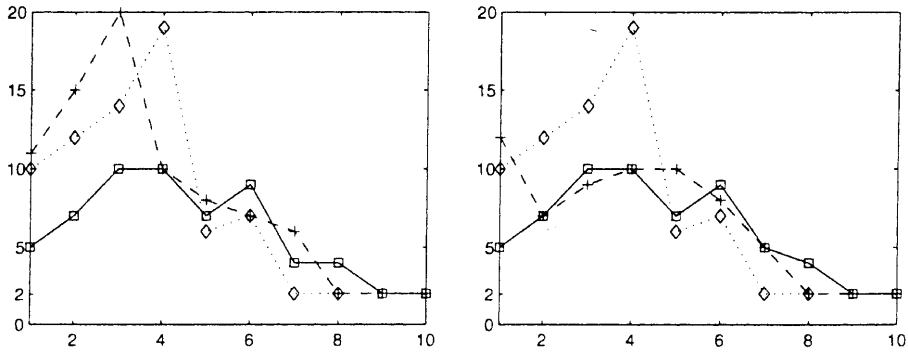

Fig. 6.3. Number of outer iterations versus level for the SCMG method. Comparison of two adaptive mesh refinement strategies from Sect. 5: one step strategy only (left), two step strategy (right). Material jumps: ($\diamond$) $a_1 = 10^0$, ($+$) $a_1 = 10^3$, ($\square$) $a_1 = 10^6$

Table 6.2. Performance of the SCMG method for problem (6.1) with $d = 3$ and $a_1 = 10^6$

j	itr	N	acc	inn	n_λ
0	32	1543	0.755	41	690
1	6	2553	0.524	15	1122
2	8	8005	0.354	14	2706
3	13	35506	0.254	14	8480
4	14	137036	0.184	9	30047
5	8	545531	0.137	12	93043

Finally, we provide results for problem (6.1) with $d = 3$. The domain Ω is now decomposed into 3 subdomains by 2D interfaces that match the jumps of the diffusion coefficient. As shown in Table 6.2, there is no significant difference in the behavior of the SCMG method in two and three space dimensions. The only slight difference is that due to the large size of the algebraic equations on the coarsest grid, that part is no longer solved directly, but also iteratively. There are more than half a million unknowns on the finest level.

Acknowledgement

The authors are grateful to Bodo Erdmann (ZIB) for his computational assistance. Moreover, the authors are indebted to the unknown referees for indicating a slight incorrectness in a proof in the first version of the manuscript.

References

[1] Achdou, Y., Kuznetsov, Yu. A., Pironneau, O.: Substructuring preconditioners for Q_1 mortar element method. Numer. Math. *71*, 419–449 (1995).
[2] Agouzal, A., Thomas, J.-M.: Une méthode d'éléments finis hybrides en décomposition de domaines. M^{2}AN *29*, 749–764 (1995).

[3] Ben Belgacem, F.: The mortar finite element method with Lagrange multipliers. Numer. Math. *84*, 173–199 (1999).
[4] Bernardi, C., Maday, Y.: Mesh adaptivity in finite elements by the mortar method. R 94029, Université Pierre et Marie Curie, Paris, France, (1995), 1–12.
[5] Bernardi, C., Maday, Y., Patera A.: A new nonconforming approach to domain decomposition: the mortar finite element method. In: Nonlinear partial differential equations and their applications Pitman, H. Brezis, J.L. Lions (eds.) (1994), pp. 13–51.
[6] Bornemann, F. A., Deuflhard, P.: Cascadic multigrid methods. In: Domain Decomposition Methods in Sciences and Engineering, R. Glowinski, J. Périaux, Z–C. Shi, O. Widlund (eds) New York: Wiley, (1997) pp. 205–212.
[7] Bornemann, F.A., Deuflhard, P.: The cascadic multigrid method for elliptic problems. Numer. Math. *75*, 135–152 (1996).
[8] Bornemann, F. A., Krause, R.: Classical and cascadic multigrid – a methodogical comparison. In: Domain Decomposition Methods in Sciences and Engineering, (Bjørstad, P., Espedal, M., Keyes, D. eds.), DD Press (1998), pp. 64–71.
[9] Braess, D., Dahmen, W.: A cascadic multigrid algorithm for the Stokes problem. Numer. Math. *82*, 179–191 (1999).
[10] Braess, D., Dahmen, W., Wieners, C.: A multigrid algorithm for the mortar finite element method. SIAM J. Numer. Anal. *37*, 48–69 (2000).
[11] Braess, D., Sarazin, R.: An efficient smoother for the Stokes problem. Appl. Numer. Math. *23*, 3–19 (1996).
[12] Bramble, J.H., Pasciak, J., Wang, J., Xu, J.: Convergence estimates for multigrid algorithms without regularity assumptions. Math. Comp. *57*, 23–45 (1991).
[13] Bramble, J., Pasciak, J., Xu, J.: Parallel multilevel preconditioners. Math. Comp. *55*, 1–22 (1990).
[14] Deuflhard, P.: Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results. In: Domain Decomposition Methods in Scientific and Engineering Computing (Keyes D., Xu, J., eds.) AMS Series 180 (1994) pp. 29–42.
[15] Deuflhard, P., Leinen, P., Yserentant, H.: Concepts of an adaptive hierarchical finite element code. IMPACT Comp. Sci. Eng. *1*, 3–35 (1989).
[16] Deuflhard, P., Lipnikov, K.: Domain decomposition with subdomain CCG for material jump elliptic problems. East-West J. Numer. Anal. *6*, 81–100 (1998).
[17] Drya, M.: An iterative substructuring method for elliptic mortar finite element problems with discontinuous coefficients. In: ''Domain Decomposition Methods 10'', (Mandel, J., Farhat, C., Cai, X.-C. eds.), AMS Series 218 (1998), pp. 94–103.
[18] Godunov, S. K., Prokopov, G. P.: Solution of the Laplace difference equation. Zh.vychisl. Mat. Fiz. 9 (1969), 462–468 (Russian) and (English) U.S.S.R. Comput. Math. Math. Phys. 9 No. 2 285–292 (1971).
[19] Kuznetsov, Yu. A.: Efficient iterative solvers for elliptic finite element problems on non-matching grids. Russ. J. Numer. Anal. Math. Modelling *10*, 187–211 (1995).
[20] Kuznetsov, Yu. A.: Iterative analysis of finite element problems with Lagrange multipliers. In: Computational Sciences for the 21st Century, pp. 170–178, John Wiley & Sons Ltd, Chichester (1997).
[21] Shaidurov, V.V.: Some estimates of the rate of convergence for the cascadic conjugate-gradient method. J. Comput. Math. Appl. 31 No. 4–5 (1996), 161–171.
[22] Shi, Z., Xu, X.: Cascadic multigrid for elliptic problems. East-West J. Numer. Math. *7*, 199–209 (1999).
[23] Stevenson, R.: Nonconforming finite elements and the cascadic iteration, Numer. Math. *91*, 351–387 (2002).
[24] Stevenson, R.: private communication (1997).
[25] Wang, J.: New convergence estimates for multilevel algorithms for finite-element approximations. J. Comp. Appl. Math. *50*, 593–604 (1994).
[26] Wieners, C., Wohlmuth, B.: Duality estimates and multigrid analysis for saddle point problems arising from mortar discretizations. Report 2002/02 University Stuttgart.
[27] Wohlmuth, B.: Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers. SIAM J. Numer. Anal. *36*, 1636–1658 (1999).
[28] Wohlmuth, B.: Discretization Methods and Iterative Solvers Based on Domain Decomposition. Springer-Verlag Berlin (2001).
[29] Wohlmuth, B., Krause, R.: Multigrid methods based on the unconstrained product space arising from mortar finite element discretizations. SIAM J. Numer. Anal. *39*, 192–219 (2001).

[30] Zulehner, W.: A class of smoothers for saddle point problems. Computing *65*, 227–246 (2000).

Dietrich Braess
Faculty of Mathematics
Ruhr-University
D-44780 Bochum
Germany
e-mail: braess@num.ruhr-uni-bochum.de

Peter Deuflhard
Konrad-Zuse-Zentrum Berlin (ZIB) &
Freie Universität Berlin
Takustrasse 7 & Arnimallee 2–6
D-14195 Berlin Germany
e-mail: deuflhard@zib.de

Konstantin Lipnikov
Department of Mathematics
University of Houston
Houston
TX 77204-3476
USA
e-mail: lipnikov@math.uh.edu