Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

DIETRICH BRAESS PETER DEUFLHARD
KONSTANTIN LIPNIKOV

A Subspace Cascadic Multigrid Method for Mortar Elements

Preprint SC 99–07 (December 1999)

A Subspace Cascadic Multigrid Method for Mortar Elements

Dietrich Braess* Peter Deuflhard† Konstantin Lipnikov‡

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.

Keywords: elliptic boundary value problems, mortar elements, domain decomposition, cascadic multigrid method, material jumps, non-matching grids

MSC (1991): 65F10, 65N30, 65N55

*Faculty of Mathematics, Ruhr-University, D-44780 Bochum, Germany, e–mail: `braess@num.ruhr-uni-bochum.de`

†Konrad-Zuse-Zentrum Berlin, Takustr. 7, D-14195 Berlin, e–mail: `deuflhard@zib.de`

‡Department of Mathematics, University of Houston, Houston, TX 77204-3476, USA, e–mail: `lipnikov@math.uh.edu`

Contents

1 Introduction

In this paper we consider linear elliptic problems

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

on general domains with space dimension $d = 2$ or $d = 3$, where typically the coefficient $a(x)$ is strongly discontinuous. Standard *multiplicative* multigrid methods or *additive* multilevel methods such as KASKADE [15] with BPX preconditioner [13] deal quite efficiently with such a situation – apart from certain pathological examples in 3D. However, the recently developed *cascadic* multigrid methods (CMG) such as cascadic conjugate gradient (CCG) methods [6, 7, 14, 8], which are extremely fast for homogeneous problems, tend to exhibit some slow-down whenever the material jumps are "too strong". Moreover, in such a problem *adaptive mesh refinements* may lead to unnecessarily fine grid regions on "coarse grid" subdomains that touch "fine grid" subdomains.

To overcome this undesirable effect, a method combining CCG on homogeneous subdomains with non-overlapping domain decomposition (DD) methods was proposed in [16]. In order to allow for non-matching grids and large jumps at interfaces, mortar elements had been used. In a 2D model problem, that DD/CCG iteration had turned out to be slower than KASKADE/BPX roughly by a factor of two. One reason for that slow performance was that – due to the saddle point structure induced by the mortar elements at interfaces – a *conjugate residual* method had to be applied as iterative solver (smoother).

The present paper aims at overcoming that weak point. Following algorithmic ideas of [11] for the Stokes problem, we organize the iteration for the state variables and Lagrange multipliers such that the iteration never leaves some subspace wherein the problem appears to be positive definite. Consequently, this iteration can be combined with any reasonable preconditioner. As will be shown by numerical experiments, this idea leads to an improved cascadic multigrid (CMG) algorithm that is now faster than KASKADE/BPX.

In Section 2 we describe the mortar element setting in the framework of mixed methods [2, 3, 4, 5] with piecewise constant Lagrangian multipliers. In Section 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 smoothing property of the method follows from results in [7, 9, 20]. In Section 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 Section 5. Finally, comparative numerical results for a notorious material jump problem are given in Section 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*

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

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:

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

For ease of writing, we identify Ω_k also with its triangular ($d = 2$) or tetrahedral ($d = 3$) regular covering relying on the context. Let Γ_{kl} denote the interface between the subdomains Ω_k and Ω_l, and assume that Γ_{kl} is simply connected. Since *non-matching* grids are usually admitted, the traces of the grids Ω_k and Ω_l at the interface Γ_{kl} need not coincide. Following a commonly used notation, we identify Γ_{kl} with the trace of Ω_k at Γ_{kl}, which implies that Ω_l represents the so-called *mortar side*. As a standard, the mortar side is chosen to be the one with the larger (average) diffusion coefficient $a(x)$ as introduced in (2.1) above. 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 the finite element spaces associated with the grids Ω_k and Γ_{kl} by V_k and Λ_{kl}, respectively. Moreover, let

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

In this framework, we consider the finite element problem with Lagrange multipliers at the interfaces Γ_{kl}, $k < l$, $k, l = 1, 2, \ldots, K$: *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

$$
\begin{aligned}
a(u,v) &:= \sum_{k=1}^{K} \int_{\Omega_k} (a(x)\nabla u \nabla v + c(x)uv)\, \mathrm{d}\Omega, \\
b(\lambda,v) &:= \sum_{k\neq l} \int_{\Gamma_{kl}} \lambda_{kl} v_k \, \mathrm{d}s, \\
f(v) &:= \sum_{k=1}^{K} \int_{\Omega_k} f v \, \mathrm{d}\Omega.
\end{aligned}
$$

By setting $\lambda_{kl} = -\lambda_{lk}$ for $k \neq l$, we assure that the bilinear form b depends exclusively on the jumps at the interfaces.

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 to the grid Ω_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 [23], we here select the space Λ_{kl} as *piecewise constant* functions in $L_2(\Gamma_{kl})$ on the grid Γ_{kl}. 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 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}:

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

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

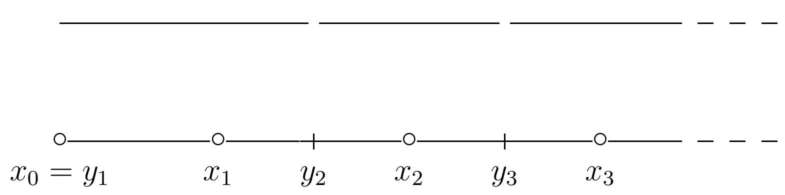

Figure 2.1: Nodal basis functions of Λ_{kl} next to a cross point for a staggered grid.

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:

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

The above matrix 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 B is a matrix with full rank. Therefore problem (2.3) has a unique solution.

3 Subspace confined conjugate gradient iteration

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

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

In principle, the equations (2.3) can be solved by any PCG method realized in the subspace

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

In order to confine all iterates to this subspace, we follow [11] and introduce a preconditioner for the matrix $\mathcal{A}$ as

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

Here D should be some positive definite matrix having a simple structure and satisfy 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 $r^i = F - \mathcal{A}z^i = (r_u^i, r_\lambda^i)$ 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 $\tilde{u} \in U$ is *not* required.

2. *Subspace entering:*

$$\begin{array}{rclcl} z^{00} & = & (u^{00}, \lambda^{00}) & = & \tilde{z} + \mathcal{H}^{-1}(F - \mathcal{A}\tilde{z}), \\ z^0 & = & (u^0, \lambda^0) & = & (u^{00}, \tilde{\lambda}), \\ s^0 & = & (s_u^0, s_\lambda^0) & = & \mathcal{H}^{-1}(F - \mathcal{A}z^0)\,. \\ p^0 & = & s_u^0\,. & & \end{array} \tag{3.4}$$

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

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

   where

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

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 $p^i \in U, i \geq 0$, from which $u^i \in U$ follows by induction.

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 $|\cdot|_{D^{-1}} := (\cdot, D^{-1}\cdot)^{1/2}$. This correction uses only terms that are anyway available.

Remark 3.1 (STEVENSON [22]). *Let u^i, λ^i be approximate solutions of the saddle point problem* (2.3) *and let r^i, s^i be defined by* (3.5). *Then the minimization problem*

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

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

This statement is verified by a simple calculation. We obtain $s^i = \mathcal{H}^{-1} r^i$ in (3.5) by solving the equation $BD^{-1}B^T s_\lambda^i = BD^{-1} r_u^i$. The solution of this equation characterizes the minimum of $|B^T\lambda - r_u^i|_{D^{-1}}$ and the proof is completed by adding λ^i. ■

The preconditioning in the subspace entering part (3.4) and the subspace iteration (3.5) contains terms with $\mathcal{H}^{-1}$. We need to solve interface equations of the type

$$BD^{-1}B^T\lambda = g \tag{3.6}$$

with suitable right hand sides g for their implementation. We will solve this equation directly on the coarsest grid. The computation of the arising matrix is comparatively cheap. Following [23], the solution of such systems just involves *static condensation* towards the cross points in 2D, but still iterative solution at interfaces in 3D. In the present version of our algorithm, we implemented (in 2D and 3D) an inner iteration to be terminated as soon as the condition

$$\|Bu^i\| \leq 10^{-2}\,\|Bu^0\| \tag{3.7}$$

has been passed. This part of computation consumed only 4 – 8 % of the total computing time and turned out to supply sufficient intermediate accuracy in the multigrid setting to be discussed in Section 4.

Once the iteration is computationally defined, we want to analyze its iterative convergence behavior. For this purpose, we introduce the *D-orthogonal projection* onto U:

$$R := I - D^{-1}B^T(BD^{-1}B^T)^{-1}B. \tag{3.8}$$

It is *symmetric* with respect to the scalar product $(\cdot,\cdot)_D$ since

$$(RD^{-1})^T = RD^{-1} \qquad \text{and} \qquad (DR)^T = DR. \tag{3.9}$$

Recalling (3.3) and using straightforward calculations we obtain

$$|R|_D \leq 1, \qquad |I - D^{-1}A|_D \leq 1, \qquad \text{and} \qquad |I - D^{-1}A|_A \leq 1. \tag{3.10}$$

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:

$$\begin{aligned}
u^0 - u &= R(I - D^{-1}A)(\tilde{u} - u), && (3.11)\\
u^i - u &= (u^{i-1} - u) + \alpha_{i-1}p^{i-1}, && (3.12)\\
p^i &= R(I - D^{-1}A)(u^{i-1} - u) - (u^{i-1} - u) + \gamma_{i-1}p^{i-1} &&\\
&= -RD^{-1}A(u^{i-1} - u) + \gamma_{i-1}p^{i-1}, \qquad i = 1, 2, \ldots &&
\end{aligned}$$

where

$$\alpha_i = \frac{\sigma_i}{(Ap_u^i, p_u^i)}, \quad \gamma_i = \frac{\sigma_i}{\sigma_{i-1}}, \quad \sigma_i = \Big(RD^{-1}A(u^i - u), A(u^i - u)\Big).$$

After m iterative steps we thus arrive at

$$u^m - u = S_m[RD^{-1}A](u^0 - u) \tag{3.13}$$

where $S_m[RD^{-1}A]$ is a nonlinear operator due to the CG iterations. Because of (3.9), the spectrum of $A^{1/2}RD^{-1}A^{1/2}$ is real. From (3.3) we even conclude that it is contained in the interval $[0,1]$. The same holds for the equivalent matrices $RD^{-1}A$ and $D^{-1}AR$. Due to the optimality of the CG iterations in the A-norm, the nonlinear operator S_m can be majorized by a linear operator in the form

$$|u^m - u|_A \le \min_{q_m \in \mathcal{P}_m} |q_m[RD^{-1}A](u^0 - u)|_A$$

where $\mathcal{P}_m$ denotes the subset of those polynomials with degree less or equal to m which satisfy $q_m(0) = 1$. In this context, a spectral analysis in terms of certain Chebyshev polynomials q_m is applicable, which satisfy $(2m+1)\sqrt{t}|q_m(t)| \le 1$ for $t \in [0,1]$. Upon applying the product rule $q_m(XY)X = Xq_m(YX)$, the relation (3.10), and Cauchy's inequality, we obtain

$$\begin{aligned} |q_m[RD^{-1}A]Rv|_A^2 &= (Rv, q_m[D^{-1}AR]D^{-1}AR\, q_m[D^{-1}AR]v)_D \\ &\le \frac{1}{(2m+1)^2}|Rv|_D\,|v|_D \le \frac{1}{(2m+1)^2}|v|_D^2\,. \end{aligned}$$

Thus we have proved the main part of the following lemma.

Lemma 3.2 *There exists a linear operator $L_m = q_m[RD^{-1}A]$ with $q_m \in \mathcal{P}_m$ such that for all $v \in U$ we have*

$$|L_m v|_A \le \frac{1}{2m+1}|v|_D \quad \textit{and} \quad |L_m v|_A \le |v|_A. \tag{3.14}$$

The m-asymptotics in (3.14) is poor compared to the standard Chebyshev estimates, but it has been shown in [20, 7] to open the door for the construction of a *cascadic conjugate gradient* (CCG) algorithm of *optimal* complexity already in 2D. We omit 1D on purpose. An exchange of CG by another smoother like Gauss–Seidel or Gauss–Jacobi would at best lead to some *nearly optimal* complexity in 2D – compare Lemma 1.1 in [7]. In 3D, however, any smoother would be 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. In our present context, however, a more suitable choice appeared to be the still simple *diagonal matrix*

$$D := 2\,\mathrm{diag}(A). \tag{3.15}$$

Remark. Following ideas of Kuznetsov [18, 19], we have also experimented with D as a *small-rank perturbation of a diagonal matrix*: Let $\overset{\circ}{A}_k$ denote the special stiffness matrix in the *Helmholtz-free* case $c_k = 0$, M_k the corresponding specification in the *pure Helmholtz* case; then, with w_k the vector spanning $\ker \overset{\circ}{A}_k$ and normalized according to $w_k^T M_k w_k = 1$, we have the projections $P_k := w_k w_k^T$ and $P_k M_k = (M_k P_k)^T$. In this notation, the suggested preconditioner associated to the subdomain Ω_k reads

$$D_k^{-1} = (I_k - P_k M_k) D_{A_k}^{-1} (I_k - M_k P_k) + \frac{1}{c_k} P_k. \tag{3.16}$$

In our numerical experiments including example (6.1) below, however, our diagonal preconditioner (3.15) has clearly outperformed the variant with (3.16). The reason for this behavior is that the iteration is just required to serve as a *smoother* within a multilevel algorithm rather than as a preconditioner. Moreover we note that the specification (3.16) will cause trouble in the Helmholtz-free case with $c_k = 0$. Therefore, in order to be able to run comparisons between the two preconditioners (3.15) and (3.16), we have selected some "small" Helmholtz term in the test example (6.1) of Section 6.

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, 23]. 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, 23].

Lemma 4.1 *Assume that the exact solution of problem* (2.2) *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 this lemma 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 subdomain index $k = 1, \dots, K$, let Ω_k denote the corresponding triangulation. For refinement levels $j = 0, 1, \dots, J$, a nested family of finite element spaces $X_0 \subset X_1 \subset \dots \subset X_J$ is defined with

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

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])

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

$$\begin{array}{ll} j = 1, \dots, J: & u_j^* = \mathcal{I}_{j,m_j} \mathcal{I}_{U_j} u_{j-1}^* \\ & \text{iterative solution of saddle point problems} \\ & \text{on successively finer grids} \end{array} \tag{4.3}$$

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

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 level j just as defined at the end of Section 2.

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

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

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

$$\frac{1}{C}\bar{h}_j \leq h_{kj} \leq \bar{h}_j\,, \qquad 1 \leq k \leq K\,,$$

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 3.2 that

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

An L_2 estimate of the following kind is typical for the analysis of cascadic multigrid algorithms with nonconforming or mixed elements (cf. [9, 21]).

Lemma 4.2 *There is a linear mapping $F_j : V_j \to U_j$ such that*

$$\|v_j - F_j v_j\|_{L_2(\Omega)} \leq C\bar{h}_j\, \|v_j\|_a \quad \textit{and} \quad \|F_j v_j\|_a \leq \|v_j\|_a \quad \textit{for all} \;\; v_j \in U_{j-1}.$$

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, which has been exemplified for the Stokes problem in [9, chapter 7]. Therefore, the quite similar proof for mortar elements is omitted here.

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*

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

then the final error of the subspace CMG method is bounded by

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

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. For $j = 1, 2, \ldots, J$ we have

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

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 (3.8). The error propagation operator R_jJ_j for one step of the Richardson method on the level j is defined as in (3.11). Similarly the error propagation operator S_{j,m_j} for the PCG method is found in (3.13). Let L_{j,m_j} be the linear operator appearing in Lemma 3.2. Upon applying Lemma 3.2, Lemma 4.1, and estimate (3.10) for the operator J_j, we proceed as follows

$$\begin{aligned}
\|u_j^* - u_j\|_a &\leq \|L_{j,m_j}(R_jJ_j(u_{j-1}^* - u_j))\|_a \\
&\leq \|L_{j,m_j}(R_jJ_j(u_{j-1} - u_j))\|_a + \|L_{j,m_j}(F_jJ_j(u_{j-1}^* - u_{j-1}))\|_a \\
&\quad + \|L_{j,m_j}(F_jJ_j - R_jJ_j)(u_{j-1}^* - u_{j-1})\|_a \qquad (4.8)\\
&\leq \frac{C}{m_j}|R_jJ_j(u_{j-1} - u_j)|_{D_j} + \|(u_{j-1}^* - u_{j-1})\|_a \\
&\quad + \frac{C}{m_j}|(R_j - F_j)J_j(u_{j-1}^* - u_{j-1})|_{D_j}.
\end{aligned}$$

Since $u_j - u_{j-1}$ can be expressed in terms of the discretization error at the levels j and $j-1$, we obtain from (3.10), (4.4), and Lemma 4.1

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

Moreover, by exploiting (3.10) and Lemma 4.2 we conclude that

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

Inserting this into (4.8) we obtain

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

Before summing up terms by virtue of this recursion relation, we note that the iteration numbers m_j decrease so fast that by (4.5) we have $\sum_{j=1}^{J}(1/m_j) \leq 2/m_J$.

Hence, $\prod_{j=1}^{J}(1+C/m_j) \le \exp(2C/m_J)$. Having these tools we finally estimate

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

Since $\beta > 2$, the last sum 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 undesirable exponential factor in (4.10), which contains a problem dependent generic constant C, seems to be unavoidable for nonconforming methods, compare [9, 21]. Fortunately, our numerical experiments done so far (including those presented in Section 6) seem to indicate that this factor is typically "not too large".

5 Realization of an adaptive version

In this section we derive an adaptive mesh refinement strategy following mainly the lines of [7]. Assume that up to level $j-1$ such a strategy has already led to a triangulation satisfying the assumptions

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

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

wherein τ is an arbitrary element (triangle in 2D or tetrahedron in 3D). 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 above assumptions are slightly stronger than inequality (4.1) in Lemma 4.1. Considerations similar to those that led to Theorem 4.3 now yield the result

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

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 some estimate of the *discretization error* $\|u_{j-1} - u\|$, 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 [23] 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" 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 obtained by the SCMG method based on linear elements. 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 residual right hand sides

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

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

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

As has been shown in [15] for shape regular triangulations, 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 [18]). 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 just 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 want to 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. That is why we stick to the above definition.

6 Numerical experiments

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

- the best DD/CCG method from [16], and
- 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 (3.7).

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 (3.16) that we abandoned afterwards. Consider the domain $\Omega = [0,1]^d$ and the elliptic equation

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

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 Section 2, the Lagrange multipliers are chosen on 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" 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.

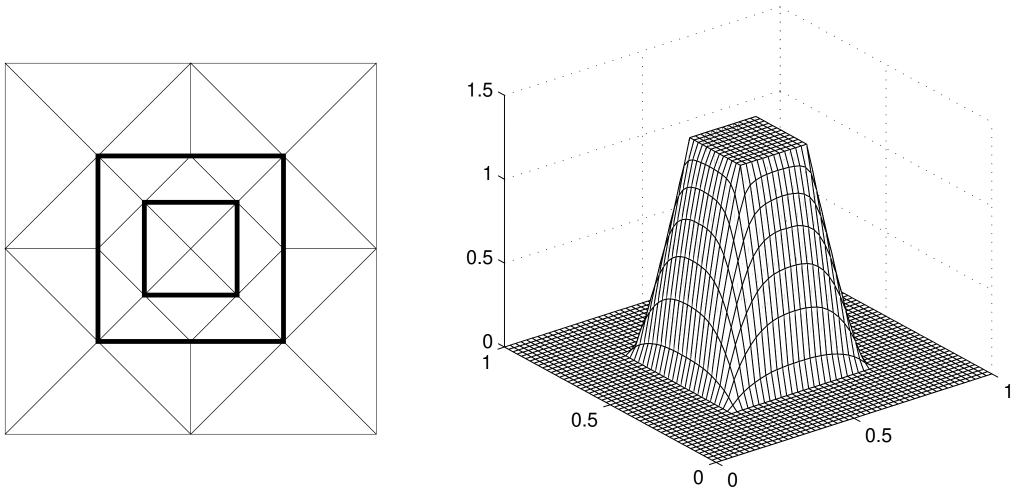

Figure 6.1: Domain decomposition with initial grid and the solution for $a_1 := 10^6$.

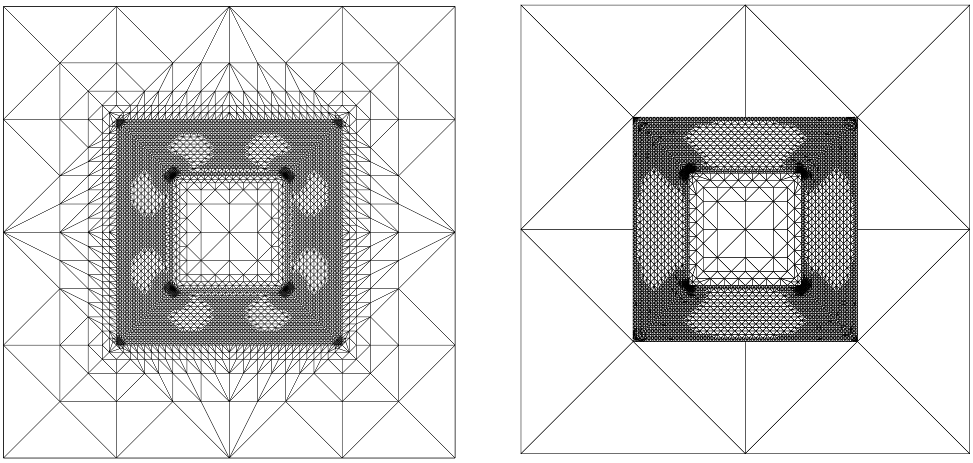

Figure 6.2: Adaptive grids from KASKADE/BPX (left) and from the SCMG method (right).

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 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.

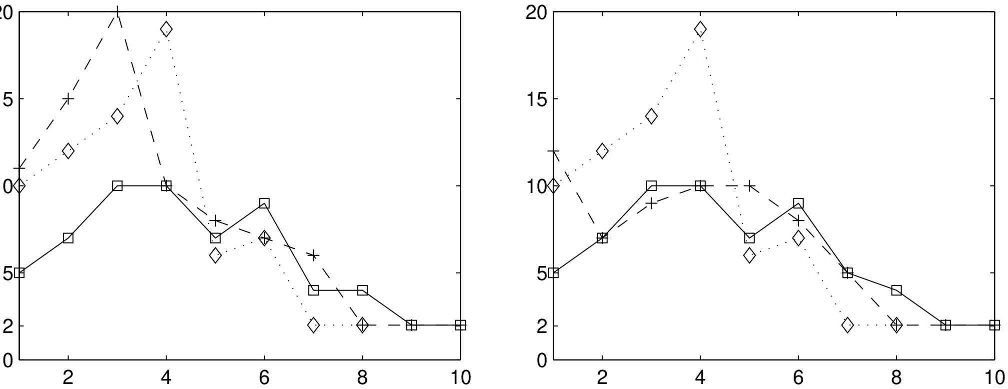

Figure 6.3: Number of outer iterations versus level for the SCMG method. Comparison of two adaptive mesh refinement strategies from Section 5: one step strategy only (left), two step strategy (right). Material jumps: ($\diamond$) $a_1 = 10^0$ ($+$) $a_1 = 10^3$ ($\square$) $a_1 = 10^6$.

	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

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.]

Finally, we provide results of the SCMG method 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 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 using inner iterations as well. Moreover we have more than half a million unknowns on the finest level.

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

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

Acknowledgement. The authors are grateful to Bodo Erdmann (ZIB) for his computational assistance.

References

[1] Achdou, Y., Kuznetsov, Yu.A., and Pironneau, O. (1995): Substructuring preconditioners for Q_1 mortar element method. *Numer. Math.* **71**, 419–449.

[2] Agouzal, A. and Thomas, J-M. (1995): Une méthode d'éléments finis hybrides en décomposition de domaines. *M²AN* **29**, 749–764.

[3] Ben Belgacem, F. (1999): The mortar finite element method with Lagrange multipliers. *Numer. Math.* **84**, 173–199

[4] Bernardi, C. and Maday, Y. (1995): Mesh adaptivity in finite elements by the mortar method. R 94029, Université Pierre et Marie Curie, Paris, France, 1–12.

[5] Bernardi, C., Maday, Y., and Patera, A. (1994): A new nonconforming approach to domain decomposition: the mortar finite element method, In: Pitman, H. Brezis, J.L. Lions (eds.) Nonlinear partial differential equations and their applications, 13–51.

[6] Bornemann, F.A. and Deuflhard, P. (1997): Cascadic multigrid methods. In: R. Glowinski, J. Périaux, Z–C. Shi, O. Widlund (eds), Domain Decomposition Methods in Sciences and Engineering, J. Wiley & Sons, 205–212.

[7] Bornemann, F.A. and Deuflhard, P. (1996): The cascadic multigrid method for elliptic problems. *Numer. Math.* **75**, 135–152.

[8] Bornemann, F.A. and Krause, R.(1998): Classical and cascadic multigrid – a methodogical comparison. In: P. Bjørstad, M. Espedal, D. Keyes, (eds.) Domain Decomposition Methods in Sciences and Engineering, DD Press, 64–71.

[9] Braess, D. and Dahmen, W. (1999): A cascadic multigrid algorithm for the Stokes problem. *Numer. Math.* **82**, 179–191

[10] Braess, D., Dahmen, W., and Wieners, C.(1999/2000): A multigrid algorithm for the mortar finite element method. *SIAM J. Numer. Anal.* **37**, 48–69.

[11] Braess, D. and Sarazin, R. (1996): An efficient smoother for the Stokes problem. *Applied Numerical Mathematics* **23**, 3–19.

[12] Bramble, J.H., Pasciak, J., Wang, J., and Xu, J. (1991): Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.* **57**, 23–45

[13] Bramble, J., Pasciak, J., and Xu, J. (1990): Parallel multilevel preconditioners. *Math. Comp.* **55**, pp. 1–22.

[14] Deuflhard, P. (1994): Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results. In: D. Keyes and J. Xu (eds.), Domain Decomposition Methods in Scientific and Engineering Computing. AMS Series, **180**, pp. 29–42.

[15] Deuflhard, P., Leinen, P., and Yserentant, H. (1989): Concepts of an adaptive hierarchical finite element code. IMPACT Comp. Sci. Eng. **1**, 3–35.

[16] Deuflhard, P. and Lipnikov, K. (1998): Domain decomposition with subdomain CCG for material jump elliptic problems. *East-West J. Numer. Anal.* **6**, 81–100.

[17] Godunov, S.K., Prokopov, G.P. (1969): On the solution of the Laplace difference equation (Russian). *Zh.vychisl. Mat. Fiz.* **9**, 462–468.

[18] Kuznetsov, Yu.A. (1995): Efficient iterative solvers for elliptic finite element problems on non-matching grids. *Russ. J. Numer. Anal. Math. Modelling* **10**, 187–211.

[19] Kuznetsov, Yu.A. (1997): Iterative analysis of finite element problems with Lagrange multipliers. In: Computational Sciences for the 21st Century, John Wiley & Sons Ltd, Chichester, 170–178.

[20] Shaidurov, V.V. (1996): Some estimates of the rate of convergence for the cascadic conjugate-gradient method. *J. Comput. Math. Appl.* **31** No. 4-5, 161–171.

[21] Stevenson, R., The Cascade iteration for nonconforming discretizations, Report 9803 University of Nijmegen – Numer. Math., to appear.

[22] Stevenson, R.(1997): private communication.

[23] Wohlmuth, B. (1999): Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers. *SIAM J. Numer. Anal.* **36**, 1636–1658.