Analysis and parallel implementation of adaptive mortar element methods

R. HOPPE[*], YU. ILIASH[*,†], YU. KUZNETSOV[†,‡], YU. VASSILEVSKI[†], and B. WOHLMUTH[*,§]

Abstract — We consider the development, analysis, and parallel implementation of domain decomposition techniques on nonmatching grids, also known as the mortar element approach, for the numerical solution of elliptic boundary value problems. The methods are based on a macro-hybrid variational formulation with respect to a non-overlapping decomposition of the computational domain. They feature multilevel block preconditioned Lanczos type algorithms for the efficient iterative solution and efficient and reliable hierarchical type *a posteriori* error estimators for adaptive local grid refinement. Aspects of parallelization such as load balancing are also addressed in some detail. Numerical results are given to illustrate the asymptotically optimal computational complexity and parallel efficiency of the solvers as well as the effectivity of the *a posteriori* error estimators.

Keywords: domain decomposition, mortar element method, *a posteriori* error estimators, asymptotically optimal computational complexity, load balancing.

1. INTRODUCTION

The current research activities in the numerical solution of partial differential equations are largely determined by three important issues:

- adaptive local grid refinement/coarsening by the efficient and reliable *a posteriori* error estimators to keep the number of unknowns as low as possible according to a prespecified tolerance,
- development of efficient iterative solvers by domain decomposition techniques and/or multilevel methods to achieve optimal or suboptimal complexity in terms of the number of arithmetic operations,
- parallelization of the algorithms and their implementation on multiprocessor machines, taking into account the appropriate load balancing and keeping communication costs at a low level.

Among the most powerful iterative solvers meeting the above requirements, domain decomposition methods have aroused considerable interest in the past decade. Originating from Schwarz' early work [39] more than a century ago, which is related to the study of harmonic functions in complex-shaped domains, these techniques have experienced an extensive growth

[*]Math. Institut, Universität Augsburg, Universitätsstr. 14, D–86159 Augsburg, Germany
[†]Institute of Numerical Mathematics of RAS, Gubkina 8, 117333 Moscow, Russia
[‡]Department of Mathematics, University of Houston, Houston 77204-3476, Texas, USA
[§]Courant Institute of Mathematical Sciences, 251 Mercer Str., New York 10012, New York, USA
This work was partially supported by the Deutsche Forschungsgemeinschaft.

since the late seventies of this century in the need for appropriate parallel computing platforms. Actually, the scope of methodology ranges from the theory of partial differential and integral equations, numerical mathematics, and parallel computation to the mathematical modeling and numerical simulation of complex technological processes (cf., e.g. [1–4, 11–14, 16, 17, 20–22, 24, 28, 29–32, 38, 40, 42–45]). A new powerful approach in this class of methods has recently been provided, which has become known as domain decomposition on nonmatching grids (cf. [1–4, 11–13, 16, 21, 22, 28–30, 38, 42–45]). This approach stems from the macro-hybrid formulations of differential problems with Lagrange multipliers at the interfaces of the subdomains. The finite element discretization of the macro-hybrid formulation results in large scale algebraic systems in saddle point form with special block-structured matrices. Efficient iterative methods for the solution of these algebraic systems based on multilevel substructuring with a special coarse grid have recently been developed (cf., e.g. [3, 29]). These techniques are highly amenable to parallelization so that the algorithms have been designed for implementation on parallel machines. On the other hand, as far as self-adaptive discretization techniques are concerned, substantial progress has been made in the past twenty years. Emphasis on *a posteriori* error estimators such that do provide both local and global information on the error of the numerical solution and can be cheaply computed by the available numerical approximation and the data of the problem (cf., e.g. [5–10, 19, 23, 26, 27, 34, 41]). The most common error estimators are residual based estimators arising from an appropriate evaluation of the residual of the computed approximation, hierarchical estimators that can be derived by approximating the error equation, using higher order finite element spaces combined with suitable localization by hierarchical splittings of these higher order spaces, estimators based on the solution of local low-dimensional subproblems, and estimators obtained by local extrapolation or averaging methods. Although *a posteriori* error estimation is the familiar tool in the finite element approximation of partial differential equations, little has been done in the framework of domain decomposition methods.

In this paper, we deal with the development, analysis, and parallel implementation of adaptive mortar finite element approximations of linear elliptic boundary value problems in the form

$$\begin{aligned} -\mathrm{div}(\rho\nabla u) + \varepsilon u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma_0 \\ \mathbf{n}\cdot\rho\nabla u &= g \quad \text{on } \Gamma_1 \end{aligned} \tag{1.1}$$

under the usual assumptions on the data ρ, ε, and f, g, where $\Omega \subset \mathbf{R}^d$, $d = 2$ or $d = 3$, is a bounded polygonal resp. polyhedral domain and $\mathbf{n}$ stands for the outward unit normal vector on Γ_1. Here Γ_0 is a closed subset of $\mathrm{d}\Omega$ with nonzero measure, and $\Gamma_1 = \mathrm{d}\Omega \setminus \Gamma_0$. Then, (1.1) has a unique weak solution.

The paper is organized as follows. In Section 2 we give a very brief outline of the mortar finite element method with Lagrange multipliers. In Section 3 we present the efficient and reliable hierarchical type of *a posteriori* error estimator that is specially designed for the mortar approach. In Section 4 we give block-structured multilevel preconditioners for the saddle point problem arising from the discretization and discuss parallel implementation aspects. Finally in Section 5 we present the results of numerical computations with emphasis on asymptotic optimality and parallel efficiency.

2. MORTAR ELEMENT METHOD WITH LAGRANGE MULTIPLIERS

For the numerical solution of the given elliptic boundary value problem (1.1) we consider the macro-hybrid P1 finite element method with respect to a decomposition of the computational

domain $\Omega \subset \mathbf{R}^d$ into m nonoverlapping polygonal resp. polyhedral subdomains Ω_i , $1 \leq i \leq m$, i.e. $\bar{\Omega} = \cup_{i=1}^m \bar{\Omega}_i$, $\Omega_i \cap \Omega_j = \emptyset$, $1 \leq i \neq j \leq m$. We assume this decomposition to be geometrically conforming, i.e. if $\bar{\Gamma}_{ij} = \bar{\Omega}_i \cap \bar{\Omega}_j \neq \emptyset$, $i \neq j$, then $\bar{\Gamma}_{ij}$ is either a common vertex or a common edge resp. a common face of Ω_i and Ω_j. We refer to

$$\mathcal{S} \;:=\; \bigcup \{\, \bar{\Gamma}_{ij} \,:\, |\, \Gamma_{ij} \,| \neq 0 \;,\; 1 \leq i \neq j \leq m \,\} \tag{2.1}$$

as the skeleton of decomposition. We further decompose the skeleton

$$\mathcal{S} \;=\; \bigcup_{k=1}^{K} \bar{\gamma}_k = \bigcup_{k=1}^{K} \bar{\delta}_k \tag{2.2}$$

into the so-called mortars γ_k and non-mortars δ_k, $1 \leq k \leq K$, where each mortar is the entire open edge resp. face of two adjacent subdomains $\Omega_{M(k)}$ and $\Omega_{\bar{M}(k)}$, $1 \leq M(k) \neq \bar{M}(k) \leq m$, i.e. $\gamma_k = \Gamma_{M(k),\bar{M}(k)}$. The non-mortars δ_k denote the corresponding opposite side of the mortars γ_k. Choosing $H^{1/2}(\delta_k)$ as the trace space of $H^1(\Omega_{\bar{M}(k)})$ on δ_k, we introduce

$$V \;:=\; \prod_{i=1}^{m} H^1(\Omega_i) \;\;,\;\; \Lambda \;:=\; \prod_{k=1}^{K} H^{-1/2}(\delta_k) \;\;.$$

Then, the macro-hybrid primal variational formulation of (1.1) reads:
Find $(u, \lambda) \in V \times \Lambda$ such that

$$\begin{array}{rcl} a(u,v) \;+\; b(\lambda,v) & = & l(v) \;\;,\;\; v \in V \\ b(\mu,u) & = & 0 \;\;\;\;,\;\; \mu \in \Lambda \;\;. \end{array} \tag{2.3}$$

Here the bilinear forms $a(\cdot,\cdot) : V \times V \to \mathbf{R}$, $b(\cdot,\cdot) : \Lambda \times V \to \mathbf{R}$ and the functional $l(\cdot) : V \to \mathbf{R}$ are given by

$$a(v,w) \;:=\; \sum_{i=1}^{m} a_i(v,w) \;\;,\;\; a_i(v,w) \;:=\; \int_{\Omega_i} [\, \rho \, \nabla v \cdot \nabla w \;+\; \varepsilon \, v \, w \,] \, \mathrm{d}x$$

$$b(\mu,v) \;:=\; \sum_{k=1}^{K} b_k(\mu,v) \;\;,\;\; b_k(\mu,v) \;:=\; \langle \mu, [v]_J \rangle_{\delta_k}$$

$$l(v) \;:=\; \sum_{i=1}^{m} \int_{\Omega_i} f \, v \, \mathrm{d}x$$

where $[v]_J|_{\delta_k} \;:=\; v|_{\Omega_{\bar{M}(k)}} - v|_{\Omega_{M(k)}}$, and $\langle \cdot,\cdot \rangle_{\delta_k}$ refers to dual pairing between $H^{-1/2}(\delta_k)$ and $H^{1/2}(\delta_k)$.

We assume that $(\mathcal{T}_i^{(l)})_{l \in \mathbf{N_0}}$ are regular locally quasiuniform nested sequences of simplicial triangulations of Ω_i , $1 \leq i \leq m$. Furthermore, we refer to $\mathcal{N}_l := \cup_{i=1}^m \mathcal{N}_i^{(l)}$ and $\mathcal{E}_l := \cup_{i=1}^m \mathcal{E}_i^{(l)}$ as the sets of vertices and edges of $\mathcal{T}_l = \cup_{i=1}^m \mathcal{T}_i^{(l)}$, respectively. The diameter of a simplex $\tau \in \mathcal{T}_l$ and the length of an edge $e \in \mathcal{E}_l$ are denoted by h_τ and h_e, respectively. The regularity of the sequence implies

$$c \, h_e \;\leq\; h_\tau \;\leq\; C \, h_e \;\;,\;\; e \subset \partial\tau \;,\; \tau \in \mathcal{T}_i^{(l)} \;,\; 1 \leq i \leq m$$

with constants $0 < c < C$ independent of l. We note that any adaptively generated nested sequence $(\mathcal{T}_i^{(l)})_{l \in \mathbf{N_0}}$, $1 \leq i \leq m$, following the refinement rules of [7] or [8], satisfies the

properties of regularity and local quasiuniformity. The macro-hybrid approach allows us to choose the simplicial triangulations $\mathcal{T}_i^{(l)}$ of the subdomains Ω_i independently of one another so that typically nonconforming nodal points arise at the common interface between the adjacent subdomains (see Fig. 1).

The left pattern in Fig. 1 is typical of boundary value problems with discontinuous coefficients, whereas the right pattern occurs naturally in time dependent problems and on sliding meshes.

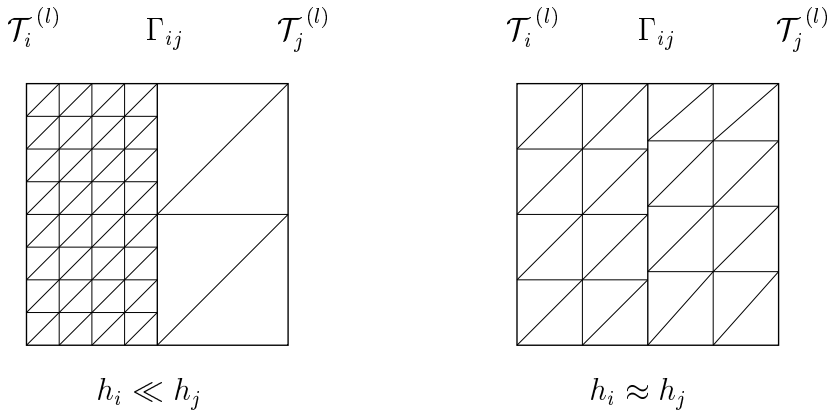


Figure 1. Nonmatching triangulations $\mathcal{T}_i^{(l)}$ and $\mathcal{T}_j^{(l)}$.

We denote by $V_n(\mathcal{T}_i^{(l)})$ the space of P_n conforming finite elements on Ω_i associated with the triangulation $\mathcal{T}_i^{(l)}$ and satisfying homogeneous boundary conditions on $\Gamma_0 \cap \partial\Omega_i$. For discretization we use piecewise linear finite elements $n = 1$. We refer to $W_n(\delta_k)$ as the trace of $V_n(\mathcal{T}_{\bar{M}(k)}^{(l)})$ on δ_k , $1 \le k \le K$. Note that the trace spaces of $V_n(\mathcal{T}_{\bar{M}(k)}^{(l)})$ and $V_n(\mathcal{T}_{M(k)}^{(l)})$ restricted by δ_k are not generally the same. The discrete Lagrange multiplier space $\Lambda_n(\delta_k)$ is the corresponding subspace of $W_n(\delta_k)$ of co-dimension $n_{\partial\delta_k}$, where $n_{\partial\delta_k}$ is the number of nodal points on $\partial\delta_k$. The Lagrange multiplier space in 2D is defined on the non-mortar side δ_k by

$$\begin{aligned}\Lambda_n(\delta_k) \;:=\; \Big\{ & v \in C(\delta_k) \mid v|_e \in P_n(e),\ e \in \mathcal{E}_{\bar{M}(k)}^{(l)} \cap \delta_k, \\ & v|_e \in P_{n-1}(e),\ \text{if } e \text{ has an endpoint of } \delta_k \Big\}\end{aligned}$$

where the 1D triangulation on δ_k is inherited from the triangulation $\mathcal{T}_{\bar{M}(k)}^{(l)}$ on $\Omega_{\bar{M}(k)}$. For the construction of $\Lambda_n(\delta_k)$ in the 3D case see, e.g. [2, 12].

Setting

$$V_n \;:=\; \prod_{i=1}^{m} V_n(\mathcal{T}_i^{(l)}) \;,\; \Lambda_n \;:=\; \prod_{k=1}^{K} \Lambda_n(\delta_k)$$

the mortar finite element approximation of (1.1) requires the computation of $(u_n, \lambda_n) \in V_n \times \Lambda_n$ such that

$$\begin{array}{lcll} a(u_n, v_n) \;+\; b(\lambda_n, v_n) & = & l(v_n) \;, & v_n \in V_n \\ b(\mu_n, u_n) & = & 0 \;, & \mu_n \in \Lambda_n \;. \end{array} \tag{2.4}$$

In the sequel we denote by c, C generic positive constants independent of the refinement level l, and by $A \sim B$ spectral equivalence between the matrices A and B or proportionality between values A and B.

3. HIERARCHICAL BASIS ERROR ESTIMATOR

In this section, we consider the efficient and reliable *a posteriori* error estimator that is cheaply computable by its elementwise contributions and provides lower and upper bounds for the discretization error. Here we use a hierarchical basis estimator based on the appropriate higher order space and restrict ourselves to the 2D case. We note that the concept of error estimator of this type is given in [9, 19] for conforming discretizations. It is further developed and analysed for more general discretization schemes like mixed or nonconforming methods in [10, 26, 27]. We refer the reader to [41] for an overview and additional references. The starting point for hierarchical basis error estimators is generally a saturation assumption. However it can often be shown that the contributions are locally equivalent to those of a residual based *a posteriori* error estimator up to higher order terms. Thus the saturation assumption can be removed [27, 34].

The hierarchical basis estimator can be derived in the framework of the nonconforming formulation or the equivalent saddle point problem. An estimator for both variables (u, λ) or only for the weak solution u can be given. Here we follow the concept proposed in [19], and the *a posteriori* error estimator is based on the saddle point problem. We refer the reader to [21, 28, 44] for the hierarchical basis type error estimator obtained by solving Neumann boundary problems on the subdomains. In this case the boundary data is given by the discrete Lagrange multiplier and a measure for nonconformity at the interface is taken into account.

It is known that the error $|||u - u_n|||^2 + \|\lambda - \lambda_n\|_L^2$ is of order $\mathcal{O}(h^{2n})$ if the weak solution u of (1.1) and $\lambda := \mathbf{n} \cdot \rho \nabla u$ are regular enough, and the discrete Babu ska-Brezzi condition is satisfied [11, 13, 44]. Here, $|||v|||^2 := a(v, v)$ is the energy norm, and $\| \cdot \|_L$ denotes the mesh dependent weighted L^2-norm defined by

$$\|v\|_L^2 := \sum_{\substack{e \in \mathcal{E}_l \cap \mathcal{S} \\ e \subset \text{non-mortar side}}} \frac{h_e}{\rho_e} \|v\|_{0;e}^2, \quad v \in L^2(\mathcal{S}) \tag{3.1}$$

where ρ_e is the coefficient on the non-mortar side. The weight ρ_e^{-1} in the norm for the Lagrange multiplier shows that the error in the weak solution u is measured in the energy norm. Note that the discrete Lagrange multiplier is an approximation of the flux $\mathbf{n} \cdot \rho \nabla u$ rather than the normal derivative $\mathbf{n} \cdot \nabla u$. In the rest of the section we assume that there exists a constant independent of the refinement level l such that locally

$$\frac{\rho_{\text{nmor}}}{h_{\text{nmor}}} \cdot \frac{h_{\text{mor}}}{\rho_{\text{mor}}} \leq C \tag{3.2}$$

where ρ_{nmor} and ρ_{mor} stand for the coefficients of the non-mortar and mortar sides, respectively, and h_{nmor}, h_{mor} denote the corresponding meshsizes. This implies that during the refinement process it is essential that the non-mortar side as compared to the mortar side is not too fine. The numerical results show that this condition is in general automatically satisfied in the adaptive refinement process. We refer the reader to [45] for further results.

We use the saturation assumption

$$|||u - u_2|||^2 + \|\lambda - \lambda_2\|_L^2 \leq \beta_h^2 (|||u - u_1|||^2 + \|\lambda - \lambda_1\|_L^2), \quad 0 < \beta_h^2 \leq \beta^2 < 1. \tag{3.3}$$

If the solution is regular enough, it is motivated by the *a priori* estimates, and β_h tends to zero as h tends to zero. Under this assumption, $(u_2 - u_1, \lambda_2 - \lambda_1)$ provides the upper and lower bounds of the error $(u - u_1, \lambda - \lambda_1)$

$$\begin{aligned} |||u - u_1|||^2 + \|\lambda - \lambda_1\|_L^2 &\leq \frac{1}{(1-\beta)^2} (|||u_2 - u_1|||^2 + \|\lambda_2 - \lambda_1\|_L^2) \\ \frac{1}{(1+\beta)^2} (|||u_2 - u_1|||^2 + \|\lambda_2 - \lambda_1\|_L^2) &\leq |||u - u_1|||^2 + \|\lambda - \lambda_1\|_L^2 \end{aligned} \tag{3.4}$$

and is the solution of the saddle point problem:

$$\begin{aligned} a\,(u_2-u_1,v)+b\,(\lambda_2-\lambda_1,v) &= r_1(v), \quad v\in V_2 \\ b\,(\mu,u_2-u_1) &= r_2(v), \quad \mu\in\Lambda_2. \end{aligned} \tag{3.5}$$

Here, the residuals $r_1(\cdot)$ and $r_2(\cdot)$ are given by $r_1(v) := (f,v)_0 - a\,(u_1,v) - b\,(\lambda_1,v)$ and $r_2(\mu) := -b\,(\mu,u_1)$. We recall that (u_1,λ_1) stands for the solution of the saddle point problem (2.4), where $n=1$. Then it is easy to see that $r_1(v)=0$ for $v\in V_1$ and $r_2(\mu)=0$ for $\mu\in\Lambda_1$. The solution of (3.5) provides the upper and lower bounds for the true error, however it cannot be obtained by solving low dimensional local problems. Thus, in order to obtain a good *a posteriori* error estimator we have to consider a modified saddle point problem whose solution can be easily obtained and is equivalent to the solution of (3.5) at the same time.

In order to introduce simplified bilinear forms $\tilde a(\cdot,\cdot)$ and $\tilde b(\cdot,\cdot)$ we consider the two-level hierarchical splittings which provide direct decompositions of the discrete spaces $V_2\times\Lambda_2$. For V_2 we have:

$$V_2 = V_1\oplus\hat V_2, \quad \hat V_2 := \bigoplus_{i=1}^{m}\bigoplus_{e\in\mathcal{E}_i^{(l)}\setminus\Gamma_0} \operatorname{span}\{\Phi_e\}$$

where Φ_e, $e\in\mathcal{E}_i^{(l)}$ in Ω_i denotes the quadratic bubble function associated with the midpoint of e. For the Lagrange multiplier space we use the same type of decomposition

$$\Lambda_2 = \Lambda_1\oplus\hat\Lambda_2, \quad \hat\Lambda_2 := \bigoplus_{\substack{e\in\mathcal{E}_l\cap\mathcal{S}\\ e\subset\text{non-mortar side}}} \operatorname{span}\{\Psi_e\}.$$

Here, Ψ_e is the one-dimensional quadratic bubble function associated with e if e has no endpoint on the non-mortar side δ_k. Otherwise Ψ_e is a linear hat function associated with the endpoint (see Fig. 2).

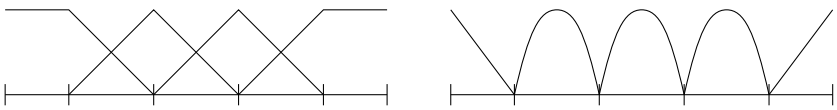


Figure 2. Nodal basis functions of V_1 (left) and $\hat V_2$ (right).

We consider the modified bilinear forms based on this splitting:

$$\begin{aligned} &\hat a(v,w) := a\,(v_1,w_1)+a\,(v_2,w_2)\,,\ v:=v_1+v_2,\ w:=w_1+w_2,\ v_1,w_1\in V_1,\ v_2,w_2\in\hat V_2 \\ &\hat b(\mu,w) := b\,(\mu_1,w_1)+b\,(\mu_2,w_2)\,,\ \mu:=\mu_1+\mu_2,\ \mu_1\in\Lambda_1,\ \mu_2\in\hat\Lambda_2 \end{aligned}$$

and replace (3.5) by:
Find $(u_e,\lambda_e)\in V_2\times\Lambda_2$ such that

$$\begin{aligned} \hat a\,(u_e,v)+\hat b\,(\lambda_e,v) &= r_1(v), \quad v\in V_2 \\ \hat b\,(\mu,u_e) &= r_2(v), \quad \mu\in\Lambda_2. \end{aligned} \tag{3.6}$$

By definition the proposed two-level splitting is orthogonal to the bilinear forms $\hat a(\cdot,\cdot)$ and $\hat b(\cdot,\cdot)$. Then it is easy to see that the solution (u_e,λ_e) of (3.6) is in the subspace $\hat V_2\times\hat\Lambda_2$ and is the solution of the saddle point problem on the hierarchical surplus space:
Find $(u_e,\lambda_e)\in\hat V_2\times\hat\Lambda_2$ such that

$$\begin{aligned} a\,(u_e,v)+b\,(\lambda_e,v) &= r_1(v), \quad v\in\hat V_2 \\ b\,(\mu,u_e) &= r_2(v), \quad \mu\in\hat\Lambda_2. \end{aligned} \tag{3.7}$$

At the next step we show that the solution of the variational problem (3.5) can be replaced by that of (3.6), and (u_e, λ_e) still yields the upper and lower bounds of the error. The main tools to prove the equivalence of the saddle point problems (3.5) and (3.7) are the strengthened Cauchy-Schwarz inequality and the discrete Babu ska-Brezzi condition. Herewe use the mesh dependent norm $\|\cdot\|_L$ for the Lagrange multiplier (see the definition (3.1)). Associated with $\|\cdot\|_L$ is the weighted bilinear form $(\cdot,\cdot)_L$ which is defined by

$$(\mu,\xi)_L := \sum_{\substack{e\in\mathcal{E}_l\cap\mathcal{S}\\ e\subset\text{non}-\text{mortar side}}} \frac{h_e}{\rho_e}\int_e \mu\,\xi\,\mathrm{d}\sigma, \quad \mu,\,\xi\in\Lambda_2.$$

The three Lemmas are stated without proof. We refer the reader to [44] for a detailed analysis.

Lemma 3.1. *There exist constants* $0<\eta_1,\eta_2<1$ *independent of the refinement level such that the strengthened Cauchy-Schwarz inequalities hold*

$$a\,(v,w)^2 \le \eta_1^2\,|||v|||^2\,|||w|||^2, \quad v\in V_1,\ w\in\hat{V}_2$$

$$(\mu,\xi)_L^2 \le \eta_2^2\,\|\mu\|_L^2\,\|\xi\|_L^2, \quad \mu\in\Lambda_1,\ \xi\in\hat{\Lambda}_2.$$

The second basic ingredient to prove equivalence between $(u_2-u_1,\lambda_2-\lambda_1)$ and (u_e,λ_e) is the discrete Babu ska-Brezzi condition.

Lemma 3.2. *Under the assumption* (3.2) *the bilinear form* $b\,(\cdot,\cdot)$ *is continuous on* $\hat{V}_2\times\Lambda_2$

$$b\,(\mu,v)\le C_c\|\mu\|_L|||v|||, \quad v\in\hat{V}_2,\ \mu\in\Lambda_2$$

with constant C_c *independent of* h. *In addition the discrete Babu ska-Brezzi condition holds*

$$\sup_{\substack{v\in V\\ |||v|||\neq 0}}\frac{b\,(\mu,v)}{|||v|||}\ge c_s\|\mu\|_L, \quad \mu\in\Lambda, \quad V\times\Lambda\in\{V_i\times\Lambda_i, 1\le i\le 2, \hat{V}_2\times\hat{\Lambda}_2\}$$

where the constant c_s *is independent of the refinement level.*

Lemma 3.3 provides a measure for the nonconformity of the element $w\in V_2$ satisfying $b(\mu,w)=0$ for all $\mu\in\Lambda_1$.

Lemma 3.3. *Let* $w\in V_2$ *satisfy*

$$b\,(\mu,w)=0, \quad \mu\in\Lambda_1.$$

Then there exists the constant $0<C_J$ *independent of the refinement level such that*

$$\sum_{\substack{e\in\mathcal{E}_l\cap\mathcal{S}\\ e\subset\text{non}-\text{mortar side}}}\frac{\rho_e}{h_e}\|[w]_J\|_{0;e}^2\le C_J|||w-v|||^2, \quad v\in H_0^1(\Omega).$$

The introduction of the error estimator is based on the theorem which states the equivalence of the solutions of (3.5) and (3.7).

Theorem 3.1. *Under the saturation assumption* (3.3) *there exist constants* $0<c_f<C_f$, *such that*

$$c_f(|||u_e|||^2+\|\lambda_e\|_L^2)\le|||u-u_1|||^2+\|\lambda-\lambda_1\|_L^2\le C_f(|||u_e|||^2+\|\lambda_e\|_L^2).$$

Proof. The proof is based on Lemmas 3.1, 3.2, and 3.3. By (3.4) it is sufficient to show the equivalence of $|||u_e|||^2 + \|\lambda_e\|_L^2$ and $|||u_2 - u_1|||^2 + \|\lambda_2 - \lambda_1\|_L^2$. Here we summarize the proof given in [44].

The lower bound is established by the stability of the saddle point problem (3.7), the continuity of the bilinear form $b(\cdot,\cdot)$, and Lemma 3.3

$$\begin{aligned} c(|||u_e|||^2 + \|\lambda_e\|_L^2)^{1/2} &\le \sup_{\substack{v\in\hat V_2\\ |||v|||\le 1}} (a(u_e,v) + b(\lambda_e,v)) + \sup_{\substack{\mu\in\hat\Lambda_2\\ \|\mu\|_L\le 1}} b(\mu,u_e) \\ &= \sup_{\substack{v\in\hat V_2\\ |||v|||\le 1}} r_1(v) + \sup_{\substack{\mu\in\hat\Lambda_2\\ \|\mu\|_L\le 1}} r_2(\mu) \\ &= \sup_{\substack{v\in\hat V_2\\ |||v|||\le 1}} (a(u_2-u_1,v) + b(\lambda_2-\lambda_1,v)) + \sup_{\substack{\mu\in\hat\Lambda_2\\ \|\mu\|_L\le 1}} b(\mu,u_2-u_1) \\ &\le (1+C_J)|||u_2-u_1||| + C_c\|\lambda_2-\lambda_1\|_L. \end{aligned}$$

In order to prove the upper bound we also need the strengthened Cauchy-Schwarz inequality. Let $v = v_1 + v_2$ and $\mu = \mu_1 + \mu_2$, with $v_1 \in V_1$, $v_2 \in \hat V_2$ and $\mu_1 \in \Lambda_1$, $\mu_2 \in \hat\Lambda_2$, respectively. Then,

$$\begin{aligned} c(|||u_2-u_1|||^2 + \|\lambda_2-\lambda_1\|_L^2)^{1/2} &\le \sup_{\substack{v\in V_2\\ |||v|||\le 1}} (a(u_2-u_1,v) + b(\lambda_2-\lambda_1,v)) \\ + \sup_{\substack{\mu\in\Lambda_2\\ \|\mu\|_L\le 1}} b(\mu,u_2-u_1) &= \sup_{\substack{v\in V_2\\ |||v|||\le 1}} r_1(v) + \sup_{\substack{\mu\in\Lambda_2\\ \|\mu\|_L\le 1}} r_2(\mu) = \sup_{\substack{v\in V_2\\ |||v|||\le 1}} r_1(v_2) \\ + \sup_{\substack{\mu\in\Lambda_2\\ \|\mu\|_L\le 1}} r_2(\mu_2) &= \sup_{\substack{v\in V_2\\ |||v|||\le 1}} (a(u_e,v_2) + b(\lambda_e,v_2)) + \sup_{\substack{\mu\in\Lambda_2\\ \|\mu\|_L\le 1}} b(\mu_2,u_e) \\ &\le (|||u_e||| + C_c\|\lambda_e\|_L) \sup_{\substack{v\in V_2\\ |||v|||\le 1}} |||v_2||| + C_c|||u_e||| \sup_{\substack{\mu\in\Lambda_2\\ \|\mu\|_L\le 1}} \|\mu_2\|_L \\ &\le \left(\frac{1}{\sqrt{1-\eta_1}} + \frac{C_c}{\sqrt{1-\eta_2}}\right)|||u_e||| + \frac{C_c}{\sqrt{1-\eta_1}}\|\lambda_e\|_L. \end{aligned}$$

□

Although the saddle point problem (3.7) is defined only on the hierarchical surplus, it is still a global problem. At the next step we follow the approach given in [19] and neglect the coupling between the quadratic bubble functions. The stiffness matrix A associated with $a(\cdot,\cdot)$ in (3.7) is replaced by its spectrally equivalent diagonal matrix. Eventually, we obtain the simplified saddle point problem:
Find $(\tilde u_e, \tilde\lambda_e) \in \hat V_2 \times \hat\Lambda_2$ such that

$$\begin{aligned} \tilde a(\tilde u_e, v) + b(\tilde\lambda_e, v) &= r_1(v), \quad v \in \hat V_2 \\ b(\mu, \tilde u_e) &= r_2(v), \quad \mu \in \hat\Lambda_2. \end{aligned} \tag{3.8}$$

Here the bilinear form $\tilde a(\cdot,\cdot)$ is given by

$$\tilde a(\psi,\phi) := \sum_{i=1}^m \sum_{e\in\mathcal{E}_i^{(l)}\setminus\Gamma_0} \sigma_e\tau_e\, a(\Phi_e,\Phi_e), \quad \phi := \sum_{i=1}^m \sum_{e\in\mathcal{E}_i^{(l)}\setminus\Gamma_0} \sigma_e\Phi_e, \quad \psi := \sum_{i=1}^m \sum_{e\in\mathcal{E}_i^{(l)}\setminus\Gamma_0} \tau_e\Phi_e$$

where σ_e and τ_e, $e \in \mathcal{E}_i^{(l)} \setminus \Gamma_0$ are the coefficients in the representation of ϕ and ψ, respectively. Then the solution $\tilde u_e$ can be written as

$$\tilde u_e = \sum_{i=1}^m \sum_{e\in\mathcal{E}_i^{(l)}\setminus\Gamma_0} \tilde\gamma_e\Phi_e$$

and the coefficients $\tilde{\gamma}_e$ are given by (3.8). The coefficients associated with the internal edges of the subdomains and with Γ_0 can be obtained by the formula

$$\tilde{\gamma}_e = \frac{r_1(\Phi_e)}{a\,(\Phi_e, \Phi_e)}.$$

In order to obtain the coefficients $\tilde{\gamma}_e$ associated with an edge on the skeleton and $\tilde{\lambda}_e$ we have to solve one global system at each interface Γ_{ij}. These systems are associated with 1D triangulation and have a condition number which can be bounded independently of the refinement level. If we start with a conforming coarse triangulation, we have the left pattern in Fig. 1 and can further localize the 1D interface problems.

Then the hierarchical basis error estimator is:

$$\begin{aligned}
\eta_H^2 &:= \sum_{\mathcal{T}_l} \left(\eta_{H;\tau}^2 + w_L \eta_{L;\tau}^2\right) \\
\eta_{H;\tau}^2 &:= \sum_{e \subset \partial\tau \setminus \Gamma_0} w_e |||\tilde{\gamma}_e \Phi_e|||^2 \\
\eta_{L;\tau}^2 &:= \sum_{\substack{e \subset \partial\tau \cap \mathcal{S} \\ e \subset \text{non-mortar side}}} \frac{h_e}{\rho_e} \|\tilde{\lambda}_e\|_{0;e}^2
\end{aligned}$$

where w_e and w_L are the corresponding weighting factors. Note that the term $\|\tilde{\lambda}_e\|_{0;e}^2$ is taken into account only on the non-mortar side.

Theorem 3.2. *Under the assumptions* (3.2) *and* (3.3) *there exist constants* c_H, $C_H > 0$ *independent of the refinement level such that*

$$c_H \eta_H^2 \le |||u - u_1|||^2 + \|\lambda - \lambda_1\|_L^2 \le C_H \eta_H^2.$$

Proof. The proof is an easy consequence of Theorem 3.1 and the equivalence of the bilinear forms $a(\cdot,\cdot)$ and $\tilde{a}(\cdot,\cdot)$ on the hierarchical surplus. □

4. DOMAIN DECOMPOSITION SOLVER

4.1. General framework

The finite element problem (2.4) brings about the problem of solving the system of linear algebraic equations in the saddle-point form:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \tag{4.1}$$

or

$$\begin{bmatrix} A_1 & & & 0 & B_1^T \\ & \cdot & & & \cdot \\ & & \cdot & & \cdot \\ & & & \cdot & \cdot \\ 0 & & & A_m & B_m^T \\ B_1 & \cdot & \cdot & B_m & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_m \\ \lambda \end{bmatrix} = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_m \\ 0 \end{bmatrix}$$

where the block representation of the matrices A and B is associated with the construction of the spaces V_n and Λ_n, while the matrix A and the vector f are specified by the bilinear form

$a(u,v)$ and the functional $l(v)$, respectively. In the sequel we restrict ourselves to piecewise linear finite elements ($n=1$) and drop the subscript n.

In order to solve the linear problem (4.1) we use the generalized Lanczos method with the preconditioner

$$\hat{R} = \begin{bmatrix} \hat{R}_A & 0 \\ 0 & \hat{R}_\lambda \end{bmatrix}. \tag{4.2}$$

Here the symmetric positive definite matrix $\hat{R}_A$ stands for a preconditioner for the subdomain problems specified by the matrix A, and $\hat{R}_\lambda$ for a preconditioner for the interface problem associated with the matrix $BA^{-1}B^T$.

Lemma 4.1 ([29]). *Let the symmetric positive definite matrices $\hat{R}_A$ and $\hat{R}_\lambda$ be spectrally equivalent to the matrices A and $BA^{-1}B^T$, respectively, with positive constants c_1, c_2, c_3, c_4. Then the boundaries of the segments $[d_1,\, d_2]$, $[d_3,\, d_4]$, $d_1 \le d_2 < 0 < d_3 \le d_4$, that contain the spectrum of the matrix $\hat{R}^{-1}\mathcal{A}$, $\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$, depend only on the values of c_1, c_2, c_3, c_4.*

Corollary 4.1. *Under the above assumption on $\hat{R}$ the convergence rate of the preconditioned Lanczos method of minimal iterations does not depend on the dimension of the matrix $\mathcal{A}$ if it is applied to the problem* (4.1).

4.2. Subdomain preconditioner

Before specifying the subdomains preconditioners we briefly discuss the preconditioning of the first block in (4.1). In general the quality of preconditioning often depends on the shape of the subdomain. It may be that for some subdomains the real condition numbers of the preconditioned stiffness matrix are not small enough to provide fast convergence of the Lanczos method. If this is the case, we introduce a few inner Chebyshev iterations in the first block to enhance the preconditioning:

$$\hat{R}_A = \begin{bmatrix} \hat{R}_{1,A} & & 0 \\ & \ddots & \\ 0 & & \hat{R}_{m,A} \end{bmatrix}, \quad \hat{R}_{i,A} = A_i \left(I_i - \prod_{j=1}^{L_i} (I_i - \gamma_{i,j} \hat{R}_i^{-1} A_i) \right)^{-1}. \tag{4.3}$$

Here the Chebyshev parameters $\gamma_{i,j}$, $j = 1, \dots, L_i$, $i = 1, \dots, m$, are associated with the spectrum of $\hat{R}_i^{-1}A_i$, where $\hat{R}_i$ stands for the actual preconditioner of A_i, I_i for the identity matrix.

4.2.1. Preconditioners for singular perturbed operators

Because of (2.4) the $n_i \times n_i$ matrix A_i may be represented as

$$A_i = \rho_i \overset{\circ}{A}_i + \varepsilon_i M_i$$

where $\overset{\circ}{A}_i$ and M_i are the stiffness and mass matrices associated with the triangulation $\mathcal{T}_i^{(l)}$, respectively. We define the vector $w_{1,i}$:

$$w_{1,i} = \alpha_i e_i, \quad (M_i w_{1,i},\, w_{1,i}) = 1, \quad e_i = [1 \ \dots \ 1]^T \in \mathbf{R}^{n_i}, \quad \alpha_i = \frac{1}{\sqrt{|\Omega_i|}}.$$

Let

$$P_i = w_{1,i} w_{1,i}^T$$

and C_i be spectrally equivalent to $\overset{\circ}{A}_i + \dfrac{1}{d_i^2} M_i$:

$$c_1 C_i \le \overset{\circ}{A}_i + \frac{1}{d_i^2} M_i \le c_2 C_i, \quad c_1 > 0.$$

Then $P_i M_i$ is the M_i-orthogonal projector onto the kernel of $\overset{\circ}{A}_i$, and the spectral decomposition of A_i yields [31] the lemma.

Lemma 4.2. *Let Ω_i be a regular shaped domain, $\varepsilon_i \le \dfrac{c\rho_i}{d_i^2}$, $\hat{R}_i^{-1} := \rho_i^{-1} C_i^{-1} + \frac{1}{\varepsilon_i} P_i$, then*

$$\hat{R}_i \sim A_i \tag{4.4}$$

with constants dependent on c_1, c_2 and independent of ρ_i, ε_i, d_i, n_i.

4.2.2. BPX preconditioner for hierarchical adaptive meshes

In this section we drop the index i of the subdomain Ω_i in the notation. Due to the hierarchy of the grid $\mathcal{T}^{(l)}$ we can split the space $V(\mathcal{T}^{(l)})$ into the nested subspaces $\mathcal{T}^{(k)}$, $k = 1, \ldots, l$,

$$V(\mathcal{T}^{(1)}) \subset V(\mathcal{T}^{(2)}) \subset \ldots \subset V(\mathcal{T}^{(l-1)}) \subset V(\mathcal{T}^{(l)}).$$

Let $\{\phi_{k,i}\}_{i=1}^{n_k}$ be a nodal basis in $V(\mathcal{T}^{(k)})$, and n_k be the dimension of $V(\mathcal{T}^{(k)})$. Due to the hierarchy we have

$$\phi_{k,i} = \sum_{j=1}^{n_k} \pi_{l,j}^{k,i} \phi_{l,j}, \qquad i = 1, \ldots, n_k, \quad k = 1, \ldots, l-1.$$

We define the set of matrices $\mathcal{P}_k \in \mathbf{R}^{n_l \times n_k}$ with elements $(\mathcal{P}_k)_{ij} = \pi_{l,j}^{k,i}$ and the diagonal matrix $\mathcal{R}_k \in \mathbf{R}^{n_k \times n_k}$ with the entries

$$\mathcal{R}_{k,i} = \left(\int_{\mathcal{T}^{(k)}} |\nabla \phi_{k,i}|^2 \, \mathrm{d}x \right)^{-1}.$$

We denote the columns of $\mathcal{P}_k$ by $\mathcal{P}_{k_1}, \ldots, P_{k_{n_k}}$. The BPX preconditioner [15, 35] is to be considered as the algebraic representation of the additive Schwarz operator:

$$C^{-1} = \sum_{n=1}^{l} \sum_{i=1}^{n_k} \mathcal{P}_{n_i} \mathcal{R}_{n_i} \mathcal{P}_{n_i}^T. \tag{4.5}$$

The analysis [35] of C^{-1} may be summarized by the lemma.

Lemma 4.3. *Let the regular hierarchical mesh $\mathcal{T}^{(l)}$ be obtained by the successive nested refinement of triangulations $\mathcal{T}^{(k)}$, $k = 1, \ldots, l-1$. Then*

$$C \sim \overset{\circ}{A} + \frac{1}{d^2} M.$$

The BPX preconditioner may be implemented by the V-cycle of the multigrid method that provides arithmetical complexity of optimal order for any hierarchical grid $\mathcal{T}^{(l)}$.

4.3. Interface preconditioner

We construct the preconditioner $\hat{R}_\lambda$ in (4.2) as an approximate inverse (through the inner iterative process with L_λ Chebyshev parameters) of the matrix $\tilde{B}\tilde{H}\tilde{B}^T$ which is spectrally equivalent to the matrix $BA^{-1}B^T$ [29]:

$$\hat{R}_\lambda = \tilde{B}\tilde{H}\tilde{B}^T \left(I_\lambda - \prod_{j=1}^{L_\lambda} (I_\lambda - \beta_j \tilde{R}_\lambda^{-1} \tilde{B}\tilde{H}\tilde{B}^T) \right)^{-1} \tag{4.6}$$

which is the general form of the preconditioner $\hat{R}_\lambda$. Here $\tilde{B}\tilde{H}\tilde{B}^T$ is assumed to be spectrally equivalent to $BA^{-1}B^T$. To provide spectral equivalence of $\hat{R}_\lambda$ and $BA^{-1}B^T$, we must choose the number of Chebyshev iterations $L_\lambda \sim \sqrt{\nu}$, where ν is the condition number of $\tilde{R}_\lambda^{-1}\tilde{B}\tilde{H}\tilde{B}^T$. In general, given a domain decomposition $\bigcup_{k=1}^{m} \Omega_i$, the larger $\dim \Lambda_h$ is, the larger ν, and the number of Chebyshev iterations L_λ is to be increased as $\dim \Lambda_h$ increases. Thus, in order to provide the efficient preconditioner $\hat{R}_\lambda$, we must minimize the arithmetical complexity of the vector multiplication by the matrix $\tilde{B}\tilde{H}\tilde{B}^T$, choosing corresponding $\tilde{H}, \tilde{B}$. For this purpose we generate the auxiliary sparse grid $\tilde{\mathcal{T}}_i^{(l)}$. The traces of the grids $\mathcal{T}_i^{(l)}$ and $\tilde{\mathcal{T}}_i^{(l)}$ on $\partial\Omega_i$ coincide, both $\mathcal{T}_i^{(l)}$ and $\tilde{\mathcal{T}}_i^{(l)}$ are regular, while the number of nodes $\tilde{n}_i$ in $\tilde{\mathcal{T}}_i^{(l)}$ is proportional to n_{Γ_i} (the number of nodes on $\partial\Omega_i$) (see Fig. 3). We define the matrix $\tilde{B}$ via its block representation, $\tilde{B} = \left[\tilde{B}_1 \dots \tilde{B}_m\right]$, where the matrix $\tilde{B}_i$ is obtained from the matrix B_i by crossing out $n_i - \tilde{n}_i$ zero columns so that B_i and $\tilde{B}_i$ have the same nonzero subblock B_{Γ_i}.

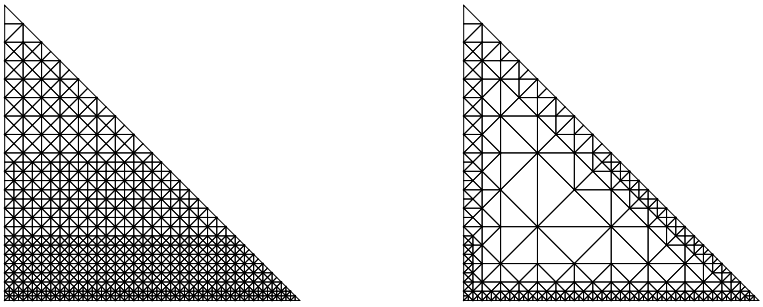


Figure 3. Hierarchical grids used in the preconditioner: original $\mathcal{T}_i^{(l)}$ and sparse $\tilde{\mathcal{T}}_i^{(l)}$.

When $\tilde{\mathcal{T}}_i^{(l)}$ is much sparser than $\mathcal{T}_i^{(l)}$, then $\tilde{B}\tilde{H}\tilde{B}^T$ may be estimated efficiently. The spectral equivalence of $\tilde{B}\tilde{H}\tilde{B}^T$ and $BA^{-1}B^T$ is based on the discrete trace theorem [33]. The particular form of the operators $\tilde{B}\tilde{H}\tilde{B}^T$ is the subject of the next sections.

4.3.1. Multigrid preconditioner on the sparse grid

Let $\tilde{\mathcal{T}}_i^{(l)}$ be a regular hierarchical mesh on Ω_i and let $\tilde{C}_i$ be the BPX preconditioner (4.5) associated with $\tilde{\mathcal{T}}_i^{(l)}$ and M_{Γ_i} be boundary mass matrices, $i = 1, \dots, m$. We introduce the matrix

$$P_{\Gamma_i} = w_{1,\Gamma_i} w_{1,\Gamma_i}^T$$

where

$$w_{1,\Gamma_i} = \beta_i e_{\Gamma_i}, \quad (M_{\Gamma_i} w_{1,\Gamma_i},\, w_{1,\Gamma_i}) = 1, \quad e_{\Gamma_i} = [1 \dots 1]^T \in \mathbf{R}^{n_{\Gamma_i}}, \quad \beta_i = \frac{1}{\sqrt{|\partial\Omega_i|}}.$$

We note that $P_{\Gamma_i} M_{\Gamma_i}$ are M_{Γ_i}-orthogonal projectors, $i = 1, \ldots, m$. The spectral decomposition of the Schur complement for the stiffness matrix $\overset{\circ}{A}_i$ associated with the elimination of the internal unknowns results in the lemma.

Lemma 4.4. *Let the symmetric positive definite matrix $\tilde{C}_i$ be spectrally equivalent to the matrix $\overset{\circ}{\tilde{A}}_i + \dfrac{1}{d_i^2}\tilde{M}_i$, where $\overset{\circ}{\tilde{A}}_i$ and $\tilde{M}_i$ are stiffness and mass matrices specified on $\tilde{\mathcal{T}}_i^{(l)}$, and*

$$\tilde{A}_i := \rho_i \overset{\circ}{\tilde{A}}_i + \varepsilon_i \tilde{M}_i = \begin{bmatrix} \tilde{A}_{\Gamma_i} & \tilde{A}_{\Gamma_i I_i} \\ \tilde{A}_{I_i \Gamma_i} & \tilde{A}_{I_i} \end{bmatrix}, \quad \varepsilon_i \le \frac{c\rho_i}{d_i^2}, \quad \tilde{T}_i := [I_{\Gamma_i}\ 0] \in \mathbf{R}^{\tilde{n}_i \times n_{\Gamma_i}}.$$

Then

$$\frac{1}{\rho_i}\tilde{T}_i \tilde{C}_i^{-1} \tilde{T}_i^T + \frac{1}{\varepsilon_i d_i} P_{\Gamma_i} \sim \left(\tilde{A}_{\Gamma_i} - \tilde{A}_{\Gamma_i I_i} \tilde{A}_{I_i}^{-1} \tilde{A}_{I_i \Gamma_i}^T\right)^{-1}.$$

The spectral equivalence occurs with constants independent of ρ_i, ε_i, d_i.

The above lemma is used for the construction of a preconditioner to $BA^{-1}B^T$ since $B_i A_i^{-1} B_i^T = B_{\Gamma_i}(A_{\Gamma_i} - A_{\Gamma_i I_i} A_{I_i}^{-1} A_{I_i \Gamma_i}^T)^{-1} B_{\Gamma_i}^T$. More precisely, given $\tilde{C}_i \sim \overset{\circ}{\tilde{A}}_i + \dfrac{1}{d_i^2}\tilde{M}_i$, we define $\tilde{B}\tilde{H}\tilde{B}^T$:

$$\tilde{B}\tilde{H}\tilde{B}^T = \sum_{k=1}^{m} B_{\Gamma_i} \left(\frac{1}{\varepsilon_i d_i} P_{\Gamma_i} + \frac{1}{\rho_i}\tilde{T}_i \tilde{C}_i^{-1} \tilde{T}_i^T\right) B_{\Gamma_i}^T. \tag{4.7}$$

In order to eliminate the ρ_i-, ε_i- and m-dependence of the condition number, we introduce in (4.6) the preconditioner $\tilde{R}_\lambda$:

$$\tilde{R}_\lambda = \sum_{i=1}^{m} B_{\Gamma_i} \left(\frac{1}{\varepsilon_i d_i} P_{\Gamma_i} + \frac{d_i}{\rho_i} M_{\Gamma_i}^{-1}\right) B_{\Gamma_i}^T = \sum_{i=1}^{m} \frac{d_i}{\rho_i} B_{\Gamma_i} M_{\Gamma_i}^{-1} B_{\Gamma_i}^T + \sum_{i=1}^{m} \frac{1}{\varepsilon_i d_i} B_{\Gamma_i} P_{\Gamma_i} B_{\Gamma_i}^T. \tag{4.8}$$

The matrix $\tilde{R}_\lambda$ is a low-rank modification of the matrix $D_\lambda = \sum\limits_{i=1}^{m} \dfrac{d_i}{\rho_i} B_{\Gamma_i} M_{\Gamma_i}^{-1} B_{\Gamma_i}^T$, and the rank of the matrix $\tilde{R}_\lambda - D_\lambda$ is equal to m. The solution of the system with matrix $\tilde{R}_\lambda$ involves the inversion of $\sum\limits_{i=1}^{m} \dfrac{d_i}{\rho_i} B_{\Gamma_i} M_{\Gamma_i}^{-1} B_{\Gamma_i}^T$ and the solution of a "coarse mesh" problem generated by the low-rank matrix $\sum\limits_{i=1}^{m} \dfrac{1}{\varepsilon_i d_i} B_{\Gamma_i} P_{\Gamma_i} B_{\Gamma_i}^T$. In practice, we lump the matrix D_λ and invert the resulting diagonal matrix which turns out to be spectrally equivalent to D_λ. In some cases (for example, quasi-uniform meshes) we can prove the validity of this substitution [29]. The "coarse mesh" problem is associated with the domain decomposition $\bar{\Omega} = \bigcup\limits_{i=1}^{m} \bar{\Omega}_i$. The presence of the "coarse mesh" problem provides the m-independence of the boundaries of the spectrum of $\tilde{R}_\lambda^{-1}\tilde{B}\tilde{H}\tilde{B}^T$. Besides the condition number of $\tilde{R}_\lambda^{-1}\tilde{B}\tilde{H}\tilde{B}^T$ is independent of ρ_i, ε_i, d_i. However, the spectrum of $\tilde{R}_\lambda^{-1}\tilde{B}\tilde{H}\tilde{B}^T$ remains sensitive to the condition numbers μ_i of $\tilde{T}_i \tilde{C}_i^{-1} \tilde{T}_i^T$, $i = 1, \ldots, m$ (cf. (4.7)–(4.8)). Thus, to invert $\tilde{B}\tilde{H}\tilde{B}^T$ with fixed accuracy, we must perform L_λ inner Chebyshev iterations in (4.6) with $L_\lambda \sim \left(\max\limits_{i=1,\ldots,m} \mu_i\right)^{1/2}$. Hence the arithmetical complexity of solving the system with resulting preconditioner $\hat{R}_\lambda$ is $O(L_\lambda n_{\Gamma_i})$. In the case of quasiuniform grids, $L_\lambda \sim \left(\max\limits_{i=1,\ldots,m} n_{\Gamma_i}^{\frac{1}{d-1}}\right)^{1/2}$ and we obtain the arithmetical complexity of optimal order both for solving systems with $\hat{R}_\lambda$ and the overall preconditioner $\hat{R}$.

Theorem 4.1. *Let L_i, $i = 1, \dots, m$, be fixed, and $L_\lambda \sim \big(\max\limits_{i=1,\dots,m} \mu_i\big)^{1/2}$, where μ_i is the condition number of the matrix $\tilde{T}_i \tilde{C}_i^{-1} \tilde{T}_i^T$ which depends only on the grid $\tilde{\mathcal{T}}_i^{(l)}$. Besides suppose that the diagonal matrix that results from the lumping of the matrix D_λ is spectrally equivalent to D_λ. Then the convergence rate of the preconditioned Lanczos method of minimal iterations does not depend on n_i, ρ_i, ε_i, d_i, m. If*

$$\Big(\sum_{i=1}^{m} n_i\Big) \Big/ \Big(\sum_{i=1}^{m} n_{\Gamma_i}\Big) > C L_\lambda,$$

the arithmetical complexity of each iteration is of optimal order.

4.3.2. Factorization of the matrix associated with the sparse grid

Due to the shape of subdomains Ω_i, in some cases the actual condition number of the matrix $\hat{R}_\lambda^{-1} B A^{-1} B^T$ is large and leads to slow convergence of the Lanczos method. In these cases we offer the explicit computation of the Schur complement $\tilde{B} \tilde{A}^{-1} \tilde{B}^T$ on the sparse grids $\tilde{\mathcal{T}}_i^{(l)}$

$$\tilde{B} \tilde{H} \tilde{B}^T = \sum_{k=1}^{m} B_{\Gamma_i} \tilde{T}_i \tilde{A}_i^{-1} \tilde{T}_i^T B_{\Gamma_i}^T, \qquad \tilde{A}_i = \rho_i \overset{\circ}{\tilde{A}}_i + \varepsilon_i \tilde{M}_i.$$

To get rid of the ρ_i, ε_i- and m-dependence of L_λ, in (4.6) we use the preconditioner $\tilde{R}_\lambda$ from (4.8). In the case of 2D triangulations the time and storage requirements are not excessive since the dimension $\tilde{n}_i$ of the matrix $\tilde{A}_i$ is not very large (in our computation $\tilde{n}_i$ is hardly larger than 500). The arithmetical complexity of the factorization of the matrix $\tilde{A}_i$ is $O(\tilde{n}_i^{3/2})$ [37] for the regular triangulation $\tilde{\mathcal{T}}_i^{(l)}$, while multiplication of a vector by the factored matrix costs $O(\tilde{n}_i \log \tilde{n}_i)$, which is almost the optimal estimate. The factorization is performed once and does not affect the overall computational time. As is seen, given factored matrices $\tilde{A}_i$, we achieve the arithmetical complexity $O(n_{\Gamma_i}^{3/2} \log n_{\Gamma_i}) = O(n_i^{3/4} \log n_i)$ in the case of quasiuniform grids $\mathcal{T}_i^{(l)}$ to solve the system with $\hat{R}_\lambda$. Thus, we have constructed the fast convergent method with arithmetical complexity of optimal order for each iteration provided that $\mathcal{T}_i^{(l)}$ are quasiuniform 2D triangulations and $\tilde{A}_i$ has been factored. In the case of 3D quasiuniform grids $\mathcal{T}_i^{(l)}$ the matrix $\tilde{A}_i$ may be factored [37] for $O(\tilde{n}_i^2) = O(n_i^{4/3})$ (ops), while multiplication of a vector by the factored matrix requires $O(\tilde{n}_i^{4/3}) = O(n_i^{8/9})$ (ops), and the combined arithmetical complexity for solving the system with the preconditioner $\hat{R}_\lambda$ is $O\Big(\sum\limits_{i=1}^{m} n_i^{19/18}\Big)$.

4.3.3. Parallel implementation aspects

The parallel implementation of the algorithm is based on the following concepts. First, in order to minimize the communications we duplicate the Lagrange multipliers data in both subdomains sharing the respective interface. This slightly increases the data storage but results in only one communication operation in the residual computation. The solution procedure for the system with preconditioner $\hat{R}$ exploits communications only within the evaluation of the interface preconditioner $\hat{R}_\lambda$. We use two types of communications, viz. the interface data exchange and the coarse mesh problem solution on a root processor for which we gather data and broadcast the solution. The above implementation is discussed in [1] in greater detail. The second concept is of great importance in the adaptive parallel computation. The latter results in large variations of the number of nodes in the subdomains. To overcome the problem of

load balancing we combine the subdomains to form clusters, each of them corresponding to a process. We propose the following scheme for the load balance. Given the number of processors P and the domain decomposition $\bar{\Omega} = \bigcup_{i=1}^{m} \bar{\Omega}_i$, execute: allocate (arbitrarily) the subdomain tasks to P processors; perform preliminary measurements; find the optimal allocation of the subproblems based on their computed complexity; redistribute the tasks and perform the main (and cumbersome) computations.

The above scheme for the adaptive parallel solution assumes that a knowledge of the complexity of the tasks is available. In the practical implementation we measure their complexity in terms of the execution time on each processor. The communication time is supposed to be negligible as compared to the calculation time and only the arithmetical job has to be balanced. The structure of the preconditioner implies that the calculations are split into two stages. The first one is preliminary, it measures the bounds of the spectra for $\hat{R}_i^{-1}A_i$ and $\tilde{R}_\lambda^{-1}\tilde{B}\tilde{H}\tilde{B}^T$ to calculate the Chebyshev iterative parameters $\gamma_{i,j}$, $j = 1, \ldots, L_i$, and β_j, $j = 1, \ldots, L_\lambda$. It is where we measure the complexity of each subproblem. The Lanczos iterative method is the second stage. It is more time-consuming than the preparation stage. The preparation stage for 2D computations takes 10–20% of the elapsed time for the problem to be solved completely.

Let us assume that the complexity of the subproblems is given by an array of loads Q, which contains m real numbers $q_1, q_2, \ldots, q_m$. In order to balance the load on P equal processors we solve (approximately) the optimization problem: split the array Q into P subsets P_k, $k = 1, \ldots, P$, so that the load balancing value is as close to 1 as possible. We define the load balancing value as the ratio of the minimal combined load on one processor $\min\limits_{k=1,\ldots,P} Q_k$ and the maximal combined load $\max\limits_{k=1,\ldots,P} Q_k$. We propose the simple heuristic algorithm for attaining optimum:

1. Rearrange the array Q to obtain the monotone sequence: $q_{i_1} \geq q_{i_2} \geq \ldots \geq q_{i_m}$;
2. Allocate the first P elements of the array to P processors;
3. Set $r = P + 1$;
4. Allocate the largest rest subproblem to P_P: $q_{i_r} \in P_P$;
5. Rearrange the array of processors to obtain the monotone sequence of combined loads Q_k:
$$Q_{k_1}(P_{k_1}) \geq Q_{k_2}(P_{k_2}) \geq \ldots \geq Q_{k_P}(P_{k_P});$$
6. If $r < m$, then $r = r + 1$, goto 4; else stop.

This approach to load balancing is worthwhile under the conditions: communication time is small as compared to computation time and does not change considerably as the topology of the links changes; the estimation of the complexity q_i for the subproblems takes less time than computations in the updated allocation of the subproblems; the above heuristic algorithm contributes significantly to the load balancing value. Actually, this implies $m \gg P$. The alternative algorithm is presented in [1].

As to the technical issues, for completeness we use the Message Passing Interface (MPI) library. The code is "single program – multiple data" (SPMD) written in Fortran-77 and has been ported on IBM SP2, Cray T3E, Dec TruCluster.

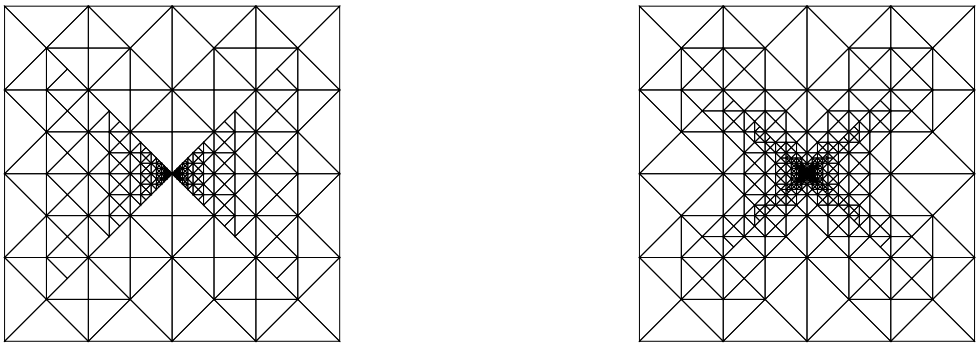
5. NUMERICAL EXPERIMENTS

5.1. Evaluation of adaptivity

We consider the diffusion equation $-\mathrm{div}\,\rho\nabla u = f$, on the unit square, where the coefficient ρ is discontinuous. Here Ω is split into four subdomains $\Omega_1 := \{(x,y) \in \Omega \mid x < y < 1-x\}$, $\Omega_2 := \{(x,y) \in \Omega \mid y < x < 1-y\}$, $\Omega_3 := \{(x,y) \in \Omega \mid x > y > 1-x\}$, and $\Omega_4 := \{(x,y) \in \Omega \mid 1-y < x < y\}$. The coefficient ρ restricted to the subdomains Ω_i is given by the constant ρ_i, $1 \le i \le 4$. The right-hand side f and the Dirichlet boundary conditions are chosen to match the given exact solution.

The solution $u(x,y) = \alpha_i r^{0.1}\sin(0.1\phi + \theta_i)$ has a singularity at $(x,y) = (0.5, 0.5)$. Here $x = r\cdot\cos\phi + 0.5$, $y = r\cdot\sin\phi + 0.5$ and $\phi \in [\frac{\pi}{4}, \frac{9\pi}{4})$. The parameters of the subdomains Ω_i are given by $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = \alpha_4 = \sin(2.1\frac{\pi}{4})(\sin(0.1\frac{\pi}{4}))^{-1}$ and $\rho_1 = \rho_3 = 1$, $\rho_2 = \rho_4 = \alpha_2^2$ and $\theta_1 = 0.9\pi$, $\theta_2 = 1.35\pi$, $\theta_3 = 1.8\pi$, $\theta_4 = 0.45\pi$. Then the solution is continuous and $[\rho_i\nabla u\mathbf{n}]_J$ is equal to zero at the interfaces.

During the adaptive refinement process highly nonconforming triangulations are generated. Figures 4 and 5 show the effect of the choice of the Lagrange multiplier on the adaptive refinement.



Non-mortar side with smaller ρ Non-mortar side with larger ρ

Figure 4. Hierarchical error estimator.

Non-mortar side with smaller ρ Non-mortar side with larger ρ

Figure 5. Hierarchical error estimator - Zoom of the final triangulations.

We see that the triangulation tends to be more conforming when the Lagrange multiplier is chosen on the side, where ρ is larger (see the right pattern in Fig. 4). This is due to the poor approximation property of the Lagrange multiplier space when there would be no refinement on this side of the interface.

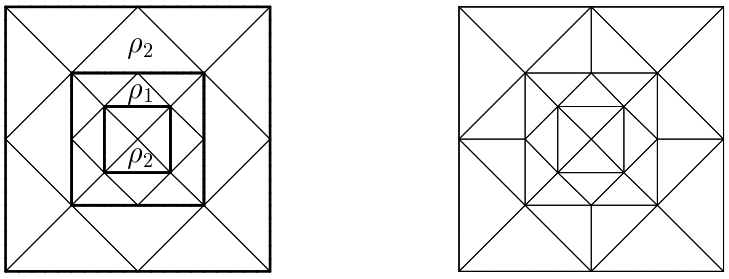
In Table 1 the number of nodes, the estimated and true error are given for the two different choices of the Lagrange multiplier space. The slow decrease of the error is due to the singular

character of the weak solution. If we compare the true error in Table 1, we find that when the non-mortar side is associated with the smaller coefficient ρ, the number of nodes to obtain the given accuracy is smaller. In this case the efficiency index, the ratio between the estimated and true errors is between 0.7 and 0.8. If the Lagrange multiplier is defined for $\rho = 161.44$, we first observe the overestimation and then underestimation of the true error.

Table 1.
Efficiency error for the hierarchical error estimator.

	Lagrange multiplier for $\rho = 161.44$				Lagrange multiplier for $\rho = 1$			
level	nodes	est. err.	real err.	eff. in.	nodes	est. err.	real err.	eff. in.
0	60	16.4	8.43	1.95	60	5.97	8.43	0.708
1	108	10.2	7.28	1.41	132	4.55	6.39	0.712
2	162	9.95	6.41	1.55	168	4.06	5.69	0.715
3	252	8.03	5.73	1.40	220	3.64	5.09	0.715
4	366	4.09	5.14	0.794	274	3.29	4.60	0.716
5	444	3.42	4.65	0.737	344	2.99	4.17	0.718
6	522	2.99	4.23	0.708	416	2.74	3.80	0.721
7	600	2.68	3.86	0.695	488	2.52	3.48	0.726
8	696	2.43	3.54	0.687	560	2.34	3.20	0.731
9	780	2.23	3.25	0.686	632	2.17	2.95	0.737
10	872	2.06	3.00	0.686	710	2.02	2.73	0.742
11	944	1.91	2.78	0.690	814	1.88	2.52	0.747

The second example illustrates the adaptive refinement process for the mortar finite element discretization and the standard one. We again consider the equation $-\mathrm{div}\,\rho\nabla u = f$ on the unit square, where the coefficient ρ is discontinuous. The domain Ω is split into three subdomains and the coefficient ρ is chosen according to Fig. 6, where $\rho_1 = 1$ and $\rho_2 = 10000$. Homogeneous Dirichlet boundary conditions are imposed and f is equal to unity on Ω. In this example the adaptive refinement process is controled by the residual based *a posteriori* error estimator. We refer the reader to [21, 28, 43] for the investigation of this type of the estimator.

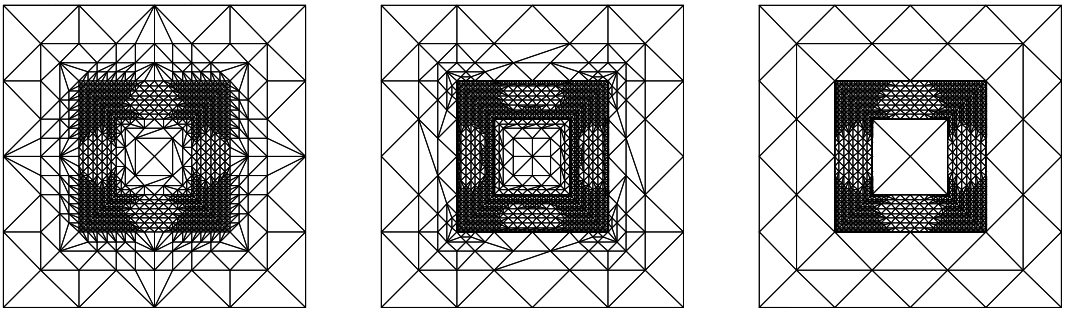


Mortar discretization CR discretization

Figure 6. Decomposition into subdomains and initial triangulations.

Here we compare the adaptively refined grids for three different situations. We apply the nonconforming Crouzeix-Raviart discretization on the whole subdomain (setting I). In this nonconforming approach the simplified residual based error estimator is used. Furthermore we

consider the mortar discretization, where the subdomains are chosen by the piecewise constant coefficients (settings II and III), and conforming P_1 and nonconforming Crouzeix-Raviart elements are coupled. This coupling is realized by the piecewise constant Lagrange multipliers (see [42]). Then we distinguish two different cases. In setting II the Lagrange multiplier space is defined on the side with $\rho = 10000$, whereas in setting III it is taken on the opposite side $\rho = 1$. The adaptively refined triangulations after 4 refinement steps are given in Fig. 7.



Setting I Setting II Setting III

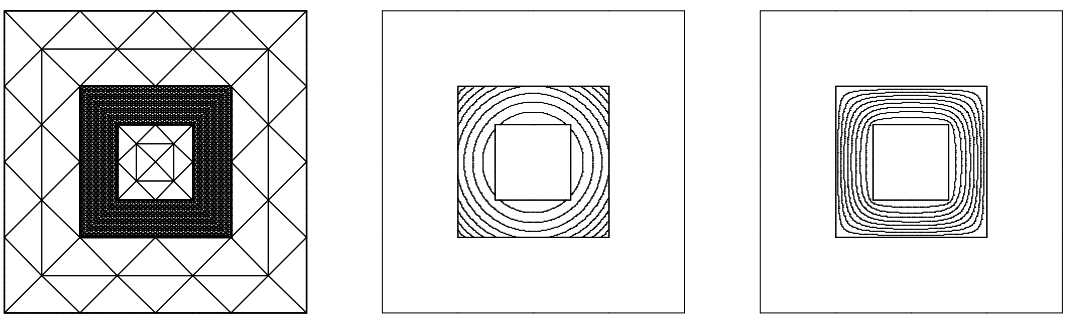
Figure 7. Adaptively refined triangulations.

In setting I we obtain the geometrical conforming triangulation due to the refinement rules. In the subdomain with $\rho = 1$ the local contribution of the error estimator requires the adaptive refinement of the elements. On the other hand, the refinement at the interface in the subdomains with $\rho = 10000$ is a consequence of the refinement rules.

At first glance the situation in setting II is similar to that of setting I. On both sides of the interface we observe strong adaptive refinement. However this is not enforced by the refinement rules. In this setting we need the adaptive refinement at the interface in the subdomains with $\rho = 10000$ because of the Lagrange multiplier. The discrete Lagrange multiplier is nothing but the approximation of the flux along the interface and therefore we observe a strong adaptive refinement on the side with $\rho = 10000$. We recall that this corresponds to the side, where the Lagrange multiplier space is defined.

Finally in setting III we get a sharp interface between the subdomains. In contrast to setting II, we observe the highly nonconforming triangulation at the interface. The Lagrange multiplier space is now defined on the side with $\rho = 1$ and there is no need to refine on both sides of the interface.

The following figure shows how crucial the effect of the choice of the Lagrange multiplier space is. For the given highly nonconforming triangulation, for settings II and III we calculate the mortar finite element solution.



Triangulation Setting II Setting III

Figure 8. Isolines of the mortar finite element solutions.

Figure 8 shows that we obtain the incorrect solution in case of setting II. In this case the consistency error is dominant and the obtained finite element solution is highly discontinuous.

The numerical results indicate that a sharp interface between the strongly refined and almost unrefined subdomains can only be obtained if the Lagrange multiplier is defined on the side with smaller coefficient ρ. Otherwise, if we apply an adaptive refinement strategy controled by the suitable error estimator, we observe adaptive refinement on both sides of the interface. Or if we compute the mortar finite element solution on the given highly nonconforming triangulation, we obtain a poor finite element approximation and the consistency error is dominant.

5.2. Evaluation of the algebraic solver

5.2.1. Test problem

As a test problem, we choose the potential flow model which describes the steady inviscid irrotational incompressible flow of an ideal gas around the profile Γ_S:

$$\begin{aligned} \Delta u &= 0 \text{ in } \Omega \\ \mathbf{n} \cdot \nabla u &= g \text{ on } \partial\Omega \end{aligned} \tag{5.1}$$

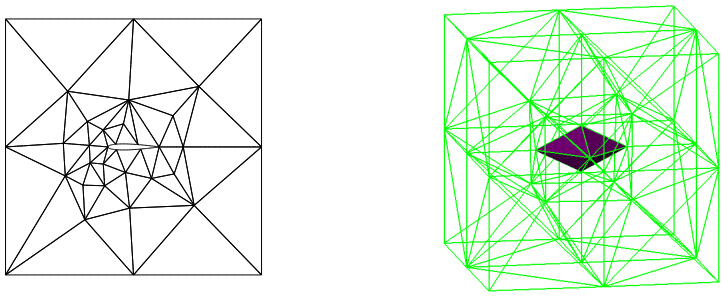
where u is the velocity potential, $\partial\Omega = \Gamma_S \cup \Gamma_\infty$, Γ_∞ is the boundary of a large square containing Γ_S and approximating "infinity", $g = 0$ on Γ_S, $g = (\mathbf{n} \cdot \mathbf{v}_\infty)$ on Γ_∞, $\mathbf{v}_\infty$ is the speed of the flow at infinity. We present here the simplest system for the nonlifting flow. In the case of lifting flows the potential u is assumed to have a constant unknown jump across an artificial slit drawn from Γ_S to Γ_∞, and the system (5.1) is completed with the Kutta-Joukowski condition [36]. Then the solution procedure is to solve two Poisson equations with different right-hand sides.

For the profile Γ_S with curved boundary we use the quasihierarchical grids to approximate (5.1) with the second-order local mesh size, and the hierarchical grids to construct the preconditioner $\hat{R}$. The quasihierarchical grid is obtained from the hierarchical one by shifting all the boundary nodes approximating the profile Γ_S onto Γ_S, at each level of the refinement. If this shifting does not produce degenerate triangles, the stiffness matrices on the hierarchical and quasihierarchical grids are evidently spectrally equivalent and the preconditioning on the hierarchical grids makes sense. In order to obtain positive definite matrices in the subdomains we perturb the Laplace operator by εI, where $\varepsilon > 0$ is small. By the theory of the solver this does not affect the convergence of the iterative process.

5.2.2. Parallel scalability

In order to present the parallel performance of the algorithm we distinguish two settings. The first one exhibits the parallel properties of the method when the number of nodes in each subdomain is the same and the load balance is perfect. In order to have the same number of nodes n_i in Ω_i, we refine each "coarse" triangle uniformly $p-1$ times into 4 equal "finer" ones starting from the triangle-subdomain. (In the 3D case we sequentially split each tetrahedron into 8 tetrahedra, which produces three different similarity classes of tetrahedra and preserves the shape regularity.) In order to simulate nonmatching grids we displace the nodes at the interfaces Γ_{ij}, randomly shifting them along Γ_{ij} from their original positions. Instead of the BPX preconditioner in the subdomains we use the algebraic multigrid preconditioner [25] on the uniformly refined grids. We consider the solution of the problem (5.1), where Ω is the complement of the

profile NACA0012 to a square (an octahedron to a cube in 3D) and is partitioned into 56 (116 in 3D) subdomains (see Fig. 9).



56 subdomains 116 subdomains

Figure 9. Decomposition into subdomains.

In Tables 2, 3 we show the execution time of Lanczos iterations to reduce the initial Euclidean norm of the residual by a factor of 10^6, measured on different numbers of processors of IBM SP2 and Cray T3E, respectively. The reader is referred to the next section for details.

Table 2.
Execution time of Lanczos iterations (sec), and the speed-up with respect to 7 processors for 2D uniformly refined grids on a IBM SP2.

#Processors, P		7	14		28		56	
$level$,p	$N = \sum_{i=1}^{56} n_i$	t_{ex}	t_{ex}	Speed-up	t_{ex}	Speed-up	t_{ex}	Speed-up
6	31416	46.1	26.5	1.74	18.3	2.52	9.6	4.8
7	120120	182.4	95.3	1.91	52.5	3.47	29.4	6.2
8	469560	747.3	379.4	1.97	212.6	3.51	104.6	7.14

Table 3.
Execution time of Lanczos iterations (sec), and the speed-up with respect to 29 processors for 3D uniformly refined grids on a Cray T3E.

#Processors, P		29	39		116	
$level$,p	$N = \sum_{i=1}^{116} n_i$	t_{ex}	t_{ex}	Speed-up	t_{ex}	Speed-up
3	19140	10.1	7.0	1.44	5.9	1.71
4	112404	111.2	83.3	1.34	44.3	2.5
5	759220	1618	1201	1.35	454.1	3.56

According to the above Tables in the case of perfect load balancing the parallel efficiency is 90% only for very large subproblems.

The second setting contains the data obtained in the real adaptive computation when the number of nodes in the subdomains varies up to 10 times. We present the iterations and time measurements for the lifting potential flow, the angle of attack being equal to 3°. The domain

decomposition into 119 subdomains is shown in Fig. 10. The initial grid is a quasiuniform one with mesh size $h_{\tau,i} = d_i/7$. The adaptive refinement is performed by the bisection algorithm [7], and the BPX preconditioner is used in the subdomains. At the third step of the adaptive procedure $\max\limits_{\tau\in\mathcal{T}_i^{(k)}} h_\tau / \min\limits_{\tau\in\mathcal{T}_i^{(k)}} h_\tau$ is as much as 500. The initial and adaptive grids are shown in Fig. 11.

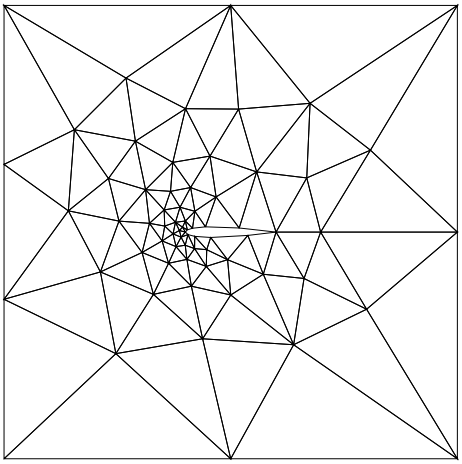


Figure 10. Domain decomposition, 119 subdomains.

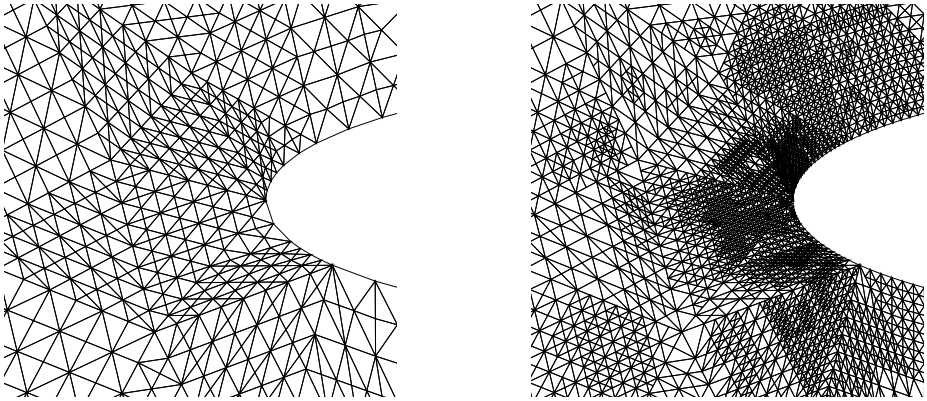


Figure 11. The first and third adaptation step grids.

In Table 4 we exhibit the execution time of the Lanczos iterations applied to two problems with different right-hand sides, on a different number of processors of IBM SP2. The residual was reduced by a factor of 10^5.

If we compare the data in Table 4 and Table 3 we see that when $m \gg P$, the speed-up in the adaptive computation is comparable with that in the 2D computation with perfect load balance and roughly the same number of nodes. The case $P = 48$ illustrates deterioration of the parallel performance in the adaptive computation. The reason is the inefficient load balance. We motivate it by the data in Table 5, where we show the load balancing value defined as the ratio of the minimal combined load on one processor $\min\limits_{i=1,\dots,P} Q_i$ and the maximal combined load $\max\limits_{i=1,\dots,P} Q_i$, obtained before and after the reallocation of the tasks. We see that even at the first step with quasi-uniform grids in the subdomains the reallocation significantly improves the load balance. At the second step of adaptation the load balance drops dramatically without reallocation, while at the third step the deterioration is not as strong due to the reallocation from

the previous step. Further, whatever the adaptive step is, the subtasks reallocation works well for $m/P \gg 1$ and loses its efficiency for small values of m/P.

Table 4.
Execution time of Lanczos iterations (sec), and the speed-up with respect to 6 processors for 2D adaptively refined grids on a IBM SP2.

#Processors, P	6	12		24		48	
$N = \sum_{i=1}^{119} n_i$	t_{ex}	t_{ex}	Speed-up	t_{ex}	Speed-up	t_{ex}	Speed-up
6823	24.3	14.1	1.72	9.0	2.7	8.47	2.87
18934	53.0	30.3	1.75	20.8	2.56	17.0	3.12
20422	62.7	40.9	1.53	24.5	2.56	21.6	2.9

Table 5.
Load balancing values before and after the subtasks reallocation.

#Processors, P	12		24		48	
Adaptive step	before reall.	after reall.	before reall.	after reall.	before reall.	after reall.
1	0.78	0.92	0.7	0.81	0.5	0.68
2	0.58	0.98	0.3	0.91	0.13	0.57
3	0.91	0.99	0.81	0.91	0.55	0.61

5.2.3. Arithmetical scalability

We present the computing time versus the problem size in the same framework as the speed-up. First, we consider the solution of the perturbed problem (5.1) on the uniformly refined grids with the algebraic multigrid preconditioner [25] in the subdomains. The computational domain Ω (the complement of the profile NACA0012 to a square in 2D and of an octahedron to a cube in 3D) is split into 56 and 116 subdomains, respectively (see Fig. 9). The perturbation parameter ε is 10^{-3}, the number of Chebyshev iterations in the first block $\hat{R}_A$, L_i, is $L_A = 5$ ($L_A = 8$ in 3D), while in the second block L_λ is proportional to $2^{p/2}$, where p is the number of refinement levels [29]. To evaluate the matrix $\tilde{B}\tilde{H}\tilde{B}^T$ we use the AMG preconditioner in the 2D case and the factorization technique on the sparse grids, which is motivated by relatively large condition numbers of the matrices $C_i^{-1}A_i$, in the 3D case.

In Tables 6, 7 we give the number of Lanczos iterations needed to reduce the Euclidean norm of the residual by a factor of 10^6 and their execution time measured on IBM SP2 and Cray T3E. The arithmetical scalability in terms of the execution time per unknown per iteration clearly points to the optimal or almost optimal order of arithmetical complexity of the preconditioner. Relatively large numbers of Chebyshev iterations in the first (L_A) and second (L_λ) blocks of the preconditioner $\hat{R}$ are dictated by the poor (but bounded from above) condition number of the matrices $C_i^{-1}A_i$ due to the shape of the subdomains. The growth of the iteration count in Table 7 is explained by the properties of the AMG preconditioner, which provides the saturation of the condition number under a larger number of levels p.

Table 6.
The number of iterations, execution time t_{ex}, and arithmetical scalability for 3D uniformly refined grids on 56 processors of Cray T3E.

p	$N = \sum_{i=1}^{56} n_i$	L	#iter	t_{ex}	$t_{ex}/(N \cdot \#\text{iter}) \cdot 10^{-6}$
5	8568	16	77	4.7	7.1
6	31416	24	71	9.6	4.3
7	120120	34	70	29.4	3.5
8	469560	48	66	104.6	3.4

Table 7.
The number of iterations, execution time t_{ex}, and arithmetical scalability for 3D uniformly refined grids on 116 processors of Cray T3E.

p	$N = \sum_{i=1}^{116} n_i$	L	#iter	t_{ex}	$t_{ex}/(N \cdot \#\text{iter}) \cdot 10^{-6}$
3	19140	34	39	5.9	7.9
4	112404	48	43	44.3	9.2
5	759220	68	58	454.1	10.3

Finally we consider the arithmetical scalability of the 2D adaptive computation. We solve the perturbed problem (5.1), $\varepsilon = 10^{-2}$, and the function g corresponds to the angle of attack 3°. Since the flow is lifting, we must solve two Poisson equations. The domain Ω is partitioned into 119 subdomains (see Fig. 10). The BPX preconditioner is used in the subdomains. Due to the shape of the subdomains Ω_i the condition number of $\hat{R}_i^{-1} A_i$ is rather large (up to 100) and we use the fixed numbers of inner iterations $L_i = L_A = 16$ in (4.3) to reduce cond $\hat{R}_A^{-1} A$ to about 2. As a consequence, the approximate inversion of $\tilde{B}\tilde{H}\tilde{B}^T$ does not approximate $BA^{-1}B^T$ properly. To avoid slow convergence, we apply the approach of Section 4.3.2. Namely, we explicitly factor the matrices $\tilde{A}_i$ which are finite element discretizations of $a(u, v)$ on the sparse grids $\tilde{\mathcal{T}}_i^{(l)}$ adapted to the boundary of $\partial\Omega_i$. We note that the dimensions $\tilde{n}_i$ of the matrices $\tilde{A}_i$ are relatively small even for a large number of nodes n_i in $\mathcal{T}_i^{(l)}$. Moreover, since $\mathcal{T}_i^{(l)}$ is adaptive, we cannot estimate the condition number of $\tilde{B}\tilde{A}^{-1}\tilde{B}^T$ in advance, and consequently, the number of inner iterations L_λ for the approximate inversion of $\tilde{B}\tilde{A}^{-1}\tilde{B}^T$ must be obtained in the computations. We choose L_λ such that the condition number of $\hat{R}_\lambda^{-1}\tilde{B}\tilde{A}^{-1}\tilde{B}^T$ reduces to 2. In Table 8 we present the total number of Lanczos iterations and their execution time needed for two solutions of the Poisson equation with different right-hand sides, on 6 processors of IBM SP2. The reduction of the residual norm by a factor of 10^5 terminated the iterations. The arithmetical scalability in terms of the execution time per iteration per unknown shows the optimal order of arithmetical complexity. Comparing the arithmetical scalabilities in Tables 6 and 8, we conclude that the arithmetical scalabilities converted to one processor are approximately the same in spite of many differences in the approaches (different types of grids, subdomain preconditioners, evaluation of the interface preconditioner).

Table 8.
The number of iterations for two Poisson equations, execution time t_{ex}, and arithmetical scalability for 2D adaptively refined grids on 6 processors of a IBM SP2.

$N = \sum_{i=1}^{119} n_i$	#iter	t_{ex}	$t_{ex}/(N \cdot \#\text{iter}) \cdot 10^{-5}$
6823	53	24.3	6.7
18934	58	53	4.8
20422	57	62.7	5.4

Acknowledgments. This work has been supported by a grant from the "Volkswagen-Stiftung" No. I/71312. The authors are also indebted to the Institute of Mathematics at the University of Augsburg and the Institute of Numerical Mathematics at the Russian Academy of Sciences in Moscow for support. They further want to express their sincere thanks to Prof. Chr. Zenger at the Technical University of Munich for various helpful discussions and valuable advice and to Dipl. Math. B. Engelmann for his assistance in carrying out the numerical computations.

REFERENCES

1. G. Abdoulaev, Y. Achdou, Yu. A. Kuznetsov, and C. Prud'homme, On a Parallel Implementation of the Mortar Element Method. *M^2AN* (to appear).
2. G. Abdoulaev, Yu. A. Kuznetsov, and O. Pironneau, The numerical implementation of the domain decomposition method with mortar finite elements for a 3D problem. *Preprint de Laboratoire d'Analyse Numérique*, Université Pierre et Marie Curie, Paris, 1996.
3. Y. Achdou, Yu. A. Kuznetsov, and O. Pironneau, Substructuring preconditioners for the Q_1 mortar element method. *Numer. Math.* (1995) **71**, 419–449.
4. Y. Achdou, Y. Maday, and O. Widlund, Méthode iterative de sous-structuration pour les éléments avec joints. *C. R. Acad. Sci., Paris, Sér. I* (1996) **322**, 185–190.
5. I. Babu ska and W. Rheinboldt, Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.* (1978) **15**, 736–754.
6. I. Babu ska and W. Rheinboldt, *A posteriori* error estimates for the finite element method. *Int. J. Numer. Methods Eng.* (1978) **12**, 1597–1615.
7. E. Bänsch, Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engrg.* (1991) **3**, 181–191.
8. R. E. Bank, A. H. Sherman, and A. Weiser, Refinement algorithm and data structures for regular local mesh refinement. In: *Scientific Computing* (Eds. R. Stepleman et al.). IMACS North-Holland, Amsterdam, 1983, pp. 3–17.
9. R. E. Bank and A. Weiser, Some *a posteriori* error estimators for elliptic partial differential equations. *Math. Comp.* (1985) **44**, 283–301.
10. R. E. Bank and R. K. Smith, *A posteriori* error estimates based on hierarchical bases. *SIAM J. Numer. Anal.* (1993) **30**, 921–935.
11. F. Ben Belgacem, The Mortar finite element method with Lagrange multipliers. *Preprint de Laboratoire d'Analyse Numérique*, Université Pierre et Marie Curie, Paris, 1995.
12. F. Ben Belgacem and Y. Maday, The mortar element method for three dimensional finite elements. *Report No. 94-19,* Université Paul Sabatier, 1994.
13. C. Bernardi, Y. Maday, and A. T. Patera, A new nonconforming approach to domain decomposition: The mortar element method. In: *Nonlinear partial differential equations and their applications* (Eds. H. Brezis et al.). Paris, 1994, pp. 13–51.
14. J. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu, Variational formulation and algorithm for trace

operator in domain decomposition calculations. In: *Domain Decomposition Methods for PDEs* (Eds. T. Chan et al.). SIAM, Philadelphia, 1989, pp. 3–17.

15. J. Bramble, J. Pasciak, and J. Xu, Parallel multilevel preconditioners. *Math. Comp.* (1990) **55**, 1–22.
16. F. Brezzi and L. Marini, A three-field domain decomposition method. In: *Domain Decomposition Methods for PDEs* (Eds. Yu. Kuznetsov, J. Periaux, A. Quarteroni, and O. Widlund). Contemporary Math., AMS, 1994, pp. 27–34.
17. T. F. Chan and T. P. Mathew, Domain decomposition algorithms. *Acta Numerica* (1994), 61–143.
18. P. Ciarlet, *The Finite Element Method for Elliptic Problems.* North-Holland, Amsterdam, 1978.
19. P. Deuflhard, P. Leinen, and H. Yserentant, Concepts of an adaptive hierarchical finite element code. *IMPACT of Computing in Science and Engrg.* (1989) **1**, 3–35.
20. M. Dryja and O. B. Widlund, Towards a unified theory of domain decomposition algorithms for elliptic problems. In: *The 3rd Int. Symp. on Domain Decomposition Methods for PDEs* (Eds. T. F. Chan et al.). SIAM, Philadelphia, 1990, pp. 3–21.
21. B. Engelmann, R. H. W. Hoppe, Yu. Iliash, Yu. Kuznetsov, Yu. Vassilevski, and B. Wohlmuth, Adaptive finite element methods for domain decomposition on nonmatching grids. In: *Parallel Solution of PDEs*, IMA Volume in Mathematics and its Applications, Springer, Berlin-Heidelberg-New York, 1998 (to appear).
22. B. Engelmann, R. H. W. Hoppe, Yu. Iliash, Yu. Kuznetsov, Yu. Vassilevski, and B. Wohlmuth, Adaptive macro-hybrid finite element methods. In: *Proc. 2nd European Conference on Numerical Methods (ENUMATH),* Heidelberg, Sept. 29 – Oct. 3, 1997 (Eds. R. Rannacher et al.). World Scientific, Singapore, 1998 (to appear).
23. D. Estep, K. Eriksson, P. Hansbo, and C. Johnson, Introduction to adaptive methods for differential equations. *Acta Numerica* (1995) **3**, 105–158.
24. R. Glowinski and M. Wheeler, Domain decomposition and mixed finite element methods for elliptic problems. In: *Domain Decomposition Methods for Partial Differential Equations* (Eds. R. Glowinski et al.). SIAM, Philadelphia, 1988, pp. 144–172.
25. Yu. Hakopian and Yu. Kuznetsov, Algebraic multigrid/substructuring preconditioners. *Sov. J. Numer. Anal. Math. Modelling* (1991) **6**, 453–483.
26. R. H. W. Hoppe and B. Wohlmuth, Element-oriented and edge-oriented local error estimators for nonconforming finite element methods. *RAIRO, Modelisation Math. Anal. Numer.* (1996) **30**, 237–263.
27. R. Hoppe and B. Wohlmuth, A comparison of *a posteriori* error estimators for mixed finite element discretizations. *Math. Comp.* (to appear).
28. R. H. W. Hoppe and B. Wohlmuth, Adaptive mixed hybrid and macro-hybrid finite element methods. In: *Proc. Conf. ALGORITHMY'97* Zuberec, West Tatra Mountains, September 1-5, 1995 (Eds. J. Kacur and K. Mikula), 1998 (to appear).
29. Yu. Kuznetsov, Efficient iterative solvers for elliptic finite element problems on nonmatching grids. *Russ. J. Numer. Anal. Math. Modelling* (1995) **10**, No. 3, 187–211.
30. Yu. A. Kuznetsov and M. F. Wheeler, Optimal order substructuring preconditioners for mixed finite element methods on nonmatching grids. *East-West J. Numer. Math.* (1995) **3**, 127–143.
31. Yu. Kuznetsov, Iterative analysis of finite element problems with Lagrange multipliers. In: *Computational Sciences for the 21st Century*. John Wiley & Sons Ltd, Chichester, 1997, pp. 170–178.
32. P. Le Tallec, Neumann-Neumann domain decomposition algorithms for solving 2D elliptic problems with nonmatching grids. *East-West J. Numer. Math.* (1993) **1**, 129–146.
33. S. V. Nepomnyaschikh, Mesh theorems of traces, normalizations of functions traces and their inversion. *Sov. J. Numer. Anal. Math. Modelling* (1991) **6**, No, 3, 223–242.
34. R. H. Nochetto, Removing the saturation assumption in *a posteriori* error analysis. *Istituto Lombardo A 127,* 1993, pp. 67–82.
35. P. Oswald, *Multilevel finite element approximation: Theory and Application.* Teubner-Skripten zur Numerik, Teubner, Stuttgart, 1994.
36. O. Pironneau, *Finite element methods for fluids.* Masson, Paris, 1989.
37. S. Pissanetzky, *Sparse matrix technology.* Academic Press Inc. Ltd., London, 1984.
38. J. Poussin and T. Sassi, Adaptive finite element and domain decomposition with nonmatching grids. In: *Proc. 2nd ECCOMAS Conf. on Numer. Math. in Engrg.* (Eds. J. Désidéri et al.). Wiley, Chichester, 1996,

pp. 476–481.

39. H. A. Schwarz, Gesammelte Mathematische Abhandlungen. *Vierteljahrsschrift der Naturforschenden Gesellschaft Zürich* (1870) **15**, 272–286.
40. B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain Decomposition and Multilevel Methods for Elliptic PDEs: Algorithms, Implementations and Theory.* Cambridge University Press, Cambridge, 1995.
41. R. Verfürth, *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Teubner-Verlag, Stuttgart, 1996.
42. C. Wieners and B. I. Wohlmuth, The coupling of mixed and conforming finite element discretizations. In: *Domain Decomposition Methods 10,* Boulder, August 1997 (Eds. J. Mandel, C. Farhat, and X.-C. Cai). American Mathematical Society 218, 1998, pp. 453–459.
43. B. Wohlmuth, A residual based error estimator for mortar finite element discretizations. *Preprint No. 370*, Institute of Mathematics, University of Augsburg, 1997.
44. B. Wohlmuth, Hierarchical *a posteriori* error estimators for mortar finite element methods with Lagrange multipliers. *Preprint TR-749*, Courant Institute of Mathematical Sciences. New York University, 1997.
45. B. Wohlmuth, Mortar finite element methods for discontinuous coefficients. *ZAMM*, 1998 (to appear).