MATHEMATICS OF COMPUTATION
Volume 67, Number 221, January 1998, Pages 1–19
S 0025-5718(98)00879-5

ANALYSIS OF NON–OVERLAPPING DOMAIN DECOMPOSITION ALGORITHMS WITH INEXACT SOLVES

JAMES H. BRAMBLE, JOSEPH E. PASCIAK, AND APOSTOL T. VASSILEV

Abstract. In this paper we construct and analyze new non-overlapping domain decomposition preconditioners for the solution of second-order elliptic and parabolic boundary value problems. The preconditioners are developed using uniform preconditioners on the subdomains instead of exact solves. They exhibit the same asymptotic condition number growth as the corresponding preconditioners with exact subdomain solves and are much more efficient computationally. Moreover, this asymptotic condition number growth is bounded independently of jumps in the operator coefficients across subdomain boundaries. We also show that our preconditioners fit into the additive Schwarz framework with appropriately chosen subspace decompositions. Condition numbers associated with the new algorithms are computed numerically in several cases and compared with those of the corresponding algorithms in which exact subdomain solves are used.

1. Introduction

In this paper, we consider the solution of the discrete systems of equations which result from finite element or finite difference approximation of second order elliptic and parabolic boundary problems. To effectively take advantage of modern parallel computing environments, algorithms must involve a large number of tasks which can be executed concurrently. Domain decomposition preconditioning techniques represent a very effective way of developing such algorithms. The parallelizable tasks are associated with subdomain solves.

There are two basic approaches to the development of domain decomposition preconditioners. The first is the so-called non-overlapping approach and is characterized by the need to solve subproblems on disjoint subdomains. Early work was applicable to domains partitioned into subdomains without internal cross-points [1], [4], [14]. To handle the case of cross-points, Bramble, Pasciak and Schatz introduced in [5] algorithms involving a coarse grid problem and provided analytic techniques for estimating the conditioning of the domain decomposition boundary preconditioner, a central issue in the subject. Various extensions of these ideas were provided in [23] including a Neumann-Dirichlet checkerboard like preconditioner.

Received by the editor February 21, 1996 and, in revised form, September 6, 1996.
1991 *Mathematics Subject Classification.* Primary 65N30, 65F10.
This manuscript has been authored under contract number DE-AC02-76CH00016 with the U.S. Department of Energy. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. This work was also supported in part under the National Science Foundation Grant No. DMS-9007185 and by the PICS ground water research initiative under contract AS-413-ASD. .

Subsequently, these techniques were extended to problems in three dimensions in [8] and [15]. A critical ingredient in the three dimensional algorithms was a coarse grid problem involving the solution averages developed in [6]. Related work is contained in [13], [20], [21].

The papers [4], [5], [6], [7], and [8] developed domain decomposition preconditioners for the original discrete system. The alternative approach, to reduce to an iteration involving only the unknowns on the boundary, was taken in [1], [11], [13], and [21]. The difference in the two techniques is important in that for the first, it is at least feasible to consider replacing the subproblem solves by preconditioners.

The second approach for developing domain decomposition preconditioners involves the solution of subproblems on overlapping subdomains. For such methods it is always possible to replace the subproblem solution with a preconditioning evaluation [9]. However, in parallel implementations, the amount of inter-processor communication is proportional to the amount of overlap. These methods lose some efficiency as the overlap becomes smaller [17]. Theoretically, they are much worse in the case when there are jumps in coefficients (see, Remark 3.3 below). In contrast, the convergence estimates for correctly designed non-overlapping domain decomposition algorithms are the same as those for smooth coefficients as long as the jumps align with subdomain boundaries.

Thus, it is natural to investigate the effect of inexact solves on non-overlapping domain decomposition algorithms. Early computational results showing that inexact non-overlapping algorithms can perform well were reported in [18]. References to other experimental work can be found in [16]. Analysis and numerical experiments with inexact algorithms of Neumann–Dirichlet and Dirichlet types, under the additional assumption of high accuracy of the inexact solves, were given in [2] and [19]. Their analysis suggests that the inexact preconditioners do not, in general, preserve the asymptotic condition number behavior of the corresponding exact method, even when the forms providing the inexact interior solves are uniformly equivalent to the original.

In this paper, we construct and analyze new non-overlapping domain decomposition preconditioners with inexact solves. We provide variations of the exact algorithm considered in [6]. We develop algorithms based only on the assumption that the interior solves are provided by uniform preconditioning forms. The inexact methods exhibit the same asymptotic condition number growth as the one in [6] and are much more efficient computationally. Our algorithms are alternatives to and in many applications less restrictive than the preconditioners in [2] and [19]. The convergence estimates developed here are independent of jumps of the operator coefficients across subdomain boundaries. The results of this paper were reported by the second author at the Seventh Copper Mountain Multigrid Conference in April of 1995 and by the third author at the Ninth Conference on Domain Decomposition Methods in June of 1996.

An important aspect of the analysis provided in this paper is that the non-overlapping preconditioners are shown to be of additive Schwarz type. Even though the new methods are inspired by and implemented according to the classical non-overlapping methodology, they can be reformulated as additive Schwarz algorithms with appropriately chosen subspace decompositions.

The paper is organized as follows. In Section 2, we formulate the problem and introduce notation. In Section 3, we construct an inexact non-overlapping domain decomposition preconditioner and investigate its properties. Section 4 provides an

application of our preconditioning approach to discretizations of parabolic problems. Computational considerations concerning the preconditioners are given in Section 5. Section 6 considers alternative inexact preconditioners. Finally, the condition number of the preconditioners developed in Section 3 and Section 6 are computed in several cases and presented in Section 7.

2. Preliminaries and notation

In this section we formulate a model elliptic problem and introduce the corresponding finite element discretization. We also outline the guiding principles in constructing our preconditioner.

We consider the Dirichlet problem

$$\begin{aligned} \mathcal{L}u &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{2.1}$$

where Ω is a bounded polyhedral domain in $\mathbb{R}^n$ for $n = 2, 3$ and

$$\mathcal{L}v = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial v}{\partial x_j}\right). \tag{2.2}$$

Here the $n \times n$ coefficient matrix $\{a_{ij}\}$ is symmetric, uniformly positive definite, and bounded above on Ω. This is a classical model problem for a second order uniformly elliptic equation. Generalizations of (2.2) which are needed for time stepping schemes approximating parabolic problems will be discussed in Section 4.

The generalized Dirichlet form on Ω is given by

$$\mathcal{A}(v, w) = \sum_{i,j=1}^{n} \int_{\Omega} a_{ij}\frac{\partial v}{\partial x_i}\frac{\partial w}{\partial x_j}\, dx. \tag{2.3}$$

Clearly, this form is well defined for functions v and w in the Sobolev space $H^1(\Omega)$.

The $L^2(\Omega)$–inner product and the related norm are defined by

$$(v, w)_\Omega = \int_\Omega vw\, dx$$

and

$$\|v\|_\Omega^2 = (v, v)_\Omega.$$

$H_0^1(\Omega)$ is the Sobolev space obtained by the completion of smooth functions with support in Ω with respect to the norm in $H^1(\Omega)$. The weak formulation of (2.1) in $H_0^1(\Omega)$ is then given by the following.

Find $u \in H_0^1(\Omega)$ such that

$$\mathcal{A}(u, \varphi) = (f, \varphi)_\Omega \quad \text{for all} \quad \varphi \in H_0^1(\Omega). \tag{2.4}$$

Given a finite dimensional subspace $S_h^0(\Omega)$ of $H_0^1(\Omega)$, the standard Galerkin approximation to (2.4) is defined by:

Find $U \in S_h^0(\Omega)$ such that

$$\mathcal{A}(U, \varphi) = (f, \varphi)_\Omega \quad \text{for all} \quad \varphi \in S_h^0(\Omega). \tag{2.5}$$

To define $S_h^0(\Omega)$, we partition Ω into triangles $\{\tau_i^h\}$ (or tetrahedra) in the usual way. Here h is the mesh parameter and is defined to be the maximal diameter of all such triangles. By definition, these triangles are closed sets. We assume that the

triangulation is quasi-uniform. The collection of simplex vertices will be denoted by $\{x_i\}$.

By convention, any union of elements τ_j^h in a given triangulation will be called a mesh subdomain. In the sequel Ω is assumed partitioned into n_d mesh subdomains $\{\Omega_k\}_{k=1}^{n_d}$ of diameter less than or equal to d. The notation Ω_k will be used for the set of all points of a subdomain including the boundary $\partial\Omega_k$.

We now define the finite element spaces. Let $S_h^0(\Omega)$ be the space of continuous piecewise linear (with respect to the triangulation) functions that vanish on $\partial\Omega$. Correspondingly, $S_h^0(\Omega_k)$ will be the space of functions whose supports are contained in Ω_k and hence each function in $S_h^0(\Omega_k)$ vanishes on $\partial\Omega_k$. $S_h(\Omega_k)$ will consist of restrictions to Ω_k of functions in $S_h^0(\Omega)$. Let Γ denote $\bigcup_k \partial\Omega_k$ and let $S_h(\Gamma)$ and $S_h(\partial\Omega_k)$ be the spaces of functions that are restrictions to Γ and $\partial\Omega_k$, respectively, of functions in $S_h^0(\Omega)$. We consider piecewise linear functions for convenience since the results and algorithms to be developed extend to higher order elements without difficulty. However, application to h-p methods is beyond the scope of this paper.

The following additional notation will be used. Let the $L^2(\partial\Omega_k)$–inner product be denoted by

$$\langle u, v\rangle_{\partial\Omega_k} = \int_{\partial\Omega_k} uv \; ds$$

and the corresponding norm by

$$|v|_{\partial\Omega_k} = \langle v, v\rangle_{\partial\Omega_k}^{1/2}.$$

On $S_h(\partial\Omega_k)$, the discrete inner product and norm are defined by

$$\langle u, v\rangle_{\partial\Omega_k,h} = h^{n-1} \sum_{x_i\in\partial\Omega_k} u(x_i)v(x_i)$$

and

$$|v|_{\partial\Omega_k,h} = \langle v, v\rangle_{\partial\Omega_k,h}^{1/2}.$$

Because of the mesh quasi-uniformity, the norm equivalence

$$c\,|v|^2_{\partial\Omega_k} \le |v|^2_{\partial\Omega_k,h} \le C\,|v|^2_{\partial\Omega_k} \tag{2.6}$$

holds for function $v \in S_h(\partial\Omega_k)$.

Here and in the remainder of the paper, we shall use c and C to denote generic positive constants independent of discretization and subdivision parameters such as h, n_d, and subdomain index k. The actual values of these constants will not necessarily be the same in any two instances.

Finally, $\mathcal{D}_k(\cdot,\cdot)$ denotes the Dirichlet inner product on Ω_k defined by

$$\mathcal{D}_k(v,w) = \sum_{i=1}^{n} \int_{\Omega_k} \frac{\partial v}{\partial x_i}\frac{\partial w}{\partial x_i}\; dx, \quad \text{for all} \quad v,\, w \,\in\, H^1(\Omega_k). \tag{2.7}$$

The development of a method for efficient iterative solution of (2.5) is the subject of this paper. In particular, using the above described decomposition of Ω, we shall define a bilinear form $\mathcal{B}(\cdot,\cdot)$ on $S_h^0(\Omega)\times S_h^0(\Omega)$ which satisfies the following two basic requirements. First, the solution $W \in S_h^0(\Omega)$ of

$$\mathcal{B}(W,\varphi) = (g,\varphi)_\Omega \quad \text{for all} \quad \varphi \in S_h^0(\Omega), \tag{2.8}$$

with g given, should be more efficient to compute than the solution of (2.5). Second, the two forms should be equivalent in the sense that

$$\lambda_1 \mathcal{B}(V,V) \le \mathcal{A}(V,V) \le \lambda_2 \mathcal{B}(V,V) \quad \text{for all} \quad V \in S_h^0(\Omega), \tag{2.9}$$

for some positive constants λ_1 and λ_2 with λ_2/λ_1 not too large. These conditions, though somewhat vague, serve as guidelines for our construction.

3. The preconditioner $\mathcal{B}(\cdot,\cdot)$ and its analysis

In this section we construct an inexact non-overlapping domain decomposition preconditioner and prove an estimate for the condition number of the preconditioned system. We also show that our preconditioner is of additive Schwarz type with appropriately defined subspace decomposition.

3.1. The preconditioner. To define our domain decomposition preconditioner, we will need boundary extension operators. For each k, let us define linear extension operators $\mathcal{E}_k : S_h(\partial\Omega_k) \to S_h(\Omega_k)$ by

$$\mathcal{E}_k\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \partial\Omega_k, \\ 0 & \text{for } x_i \in \Omega_k \setminus \partial\Omega_k. \end{cases}$$

Correspondingly, let $\mathcal{E} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ be defined by

$$\mathcal{E}\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \Gamma, \\ 0 & \text{for } x_i \in \Omega \setminus \Gamma. \end{cases} \tag{3.1}$$

Clearly, defining these operators at the mesh vertices defines them everywhere.

For each k, let $\mathcal{B}_k(\cdot,\cdot)$ be a bilinear form on $S_h^0(\Omega_k) \times S_h^0(\Omega_k)$ which is uniformly equivalent to $\mathcal{A}_k(\cdot,\cdot)$ where $\mathcal{A}_k(\cdot,\cdot)$ is defined as in (2.3) but with integration only over Ω_k. By this we mean that for each k there are constants c_k and C_k with C_k/c_k bounded independently of h and d such that

$$c_k \mathcal{B}_k(V,V) \le \mathcal{A}_k(V,V) \le C_k \mathcal{B}_k(V,V) \quad \text{for all} \quad V \in S_h^0(\Omega_k). \tag{3.2}$$

The preconditioning form is given by

$$\begin{aligned} \mathcal{B}(U,V) = &\sum_{k=1}^{n_d} \mathcal{B}_k(U - \bar{U}_k - \mathcal{E}_k(U - \bar{U}_k), V - \bar{V}_k - \mathcal{E}_k(V - \bar{V}_k)) \\ &+ h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle U - \bar{U}_k, V - \bar{V}_k \rangle_{\partial\Omega_k,h}. \end{aligned} \tag{3.3}$$

Here, $\bar{U}_k$ denotes the discrete mean value of U on $\partial\Omega_k$, i.e.,

$$\bar{U}_k \equiv \frac{\langle U, 1\rangle_{\partial\Omega_k,h}}{\langle 1, 1\rangle_{\partial\Omega_k,h}}.$$

In (3.3), $\tilde{a}_k$, $k = 1, \ldots, n_d$, are parameters associated with the coefficients a_{ij} in Ω_k. For example, if $\tilde{a}_k$ is taken to be the smallest eigenvalue of $\{a_{i,j}\}$ at some point $x \in \Omega_k$, then

$$C_k^{-1} \tilde{a}_k \mathcal{D}_k(v,v) \le \mathcal{A}_k(v,v) \le C_k \tilde{a}_k \mathcal{D}_k(v,v) \quad \text{for all} \quad v \in S_h(\Omega_k). \tag{3.4}$$

The constant C_k only depends on the local variation of the coefficients $\{a_{ij}\}$ on the subdomain Ω_k. Consequently, we will assume that (3.4) holds with C_k bounded independently of d, h, and k.

3.2. Analysis of the preconditioning form $\mathcal{B}(\cdot,\cdot)$. We introduce some standard assumptions about the domain Ω, the subdomain splitting and the associated finite element spaces which are needed for the analysis.

We start by requiring that the collection $\{\Omega_k\}$ be quasi-uniform of size d. Also, we shall assume that

$$|u|^2_{\partial\Omega_k} \le C\{\epsilon^{-1}\,\|u\|^2_{\Omega_k} + \epsilon\mathcal{D}_k(u,u)\} \tag{3.5}$$

holds for any ϵ in $(0,d]$ and all k. Finally, we assume that a Poincaré inequality of the form

$$\|v\|^2_{\Omega_k} \le Cd^2\mathcal{D}_k(v,v) \tag{3.6}$$

holds for functions v with zero mean value on Ω_k.

The inequalities (3.5) and (3.6) hold for all but pathological subdomains. A sufficient but by no means necessary condition for the above two inequalities is given in the following assumption.

Each Ω_k is star-shaped with respect to a point. This means that for each Ω_k there is a point $\hat{x}_k$ and a constant $c_k > 0$ such that $(x-\hat{x}_k)\cdot\mathbf{n}(x) \ge c_k d$ for all $x \in \partial\Omega_k$ which are not mesh vertices. We further assume that $c_k \ge c$ for some constant c not depending on d, k or h. Here $\mathbf{n}(x)$ denotes the outward unit normal vector to $\partial\Omega_k$ at a nonvertex point x.

The following lemma will be used in the derivation of our results.

Lemma 3.1. *If $v \in S_h(\Omega_k)$ and vanishes at all interior nodes of Ω_k, then*

$$\mathcal{D}_k(v,v) \le Ch^{-1}\,|v|^2_{\partial\Omega_k,h}\,. \tag{3.7}$$

This lemma is obvious from the local properties of the functions in finite element spaces and we shall omit its proof.

The main result of this paper is contained in the following theorem.

Theorem 3.1. *Let $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ be given by (2.3) and (3.3) respectively. Assume that (3.2), (3.4), (3.5) and (3.6) hold. Then there exist positive constants c and C not depending on d or h such that*

$$c\mathcal{A}(V,V) \le \mathcal{B}(V,V) \le C\frac{d}{h}\mathcal{A}(V,V), \tag{3.8}$$

for all $V \in S^0_h(\Omega)$.

Proof. Because of (3.2), it suffices to prove the theorem for

$$\begin{aligned}\mathcal{B}(U,V) = &\sum_{k=1}^{n_d}\mathcal{A}_k(U-\bar{U}_k-\mathcal{E}_k(U-\bar{U}_k),V-\bar{V}_k-\mathcal{E}_k(V-\bar{V}_k))\\ &+h^{-1}\sum_{k=1}^{n_d}\tilde{a}_k\langle U-\bar{U}_k,V-\bar{V}_k\rangle_{\partial\Omega_k,h}.\end{aligned} \tag{3.9}$$

We first prove the left inequality in (3.8). The arithmetic–geometric mean inequality shows that for any constant α we have

$$\begin{aligned}\frac{1}{2}\mathcal{A}_k(V,V) &= \frac{1}{2}\mathcal{A}_k(V-\alpha,V-\alpha)\\ &\le \mathcal{A}_k(V-\alpha-\mathcal{E}_k(V-\alpha),V-\alpha-\mathcal{E}_k(V-\alpha))\\ &\quad+\mathcal{A}_k(\mathcal{E}_k(V-\alpha),\mathcal{E}_k(V-\alpha)).\end{aligned} \tag{3.10}$$

The left inequality in (3.8) is a simple consequence of (3.7), (3.4), (3.10), and the definition of $\mathcal{E}_k$ with $\alpha = \bar{V}_k$.

In order to prove the right inequality, we apply the arithmetic–geometric mean inequality to the terms in the first sum in (3.9) and get

$$\begin{aligned}\mathcal{B}(V,V) \le 2\mathcal{A}(V,V) &+ 2\sum_{k=1}^{n_d} \mathcal{A}_k(\mathcal{E}_k(V-\bar{V}_k),\mathcal{E}_k(V-\bar{V}_k)) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}.\end{aligned} \tag{3.11}$$

By (3.4) and (3.7), we obtain

$$\mathcal{B}(V,V) \le 2\mathcal{A}(V,V) + Ch^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}. \tag{3.12}$$

Let $\bar{\bar{V}}_k$ be the mean value of V on Ω_k. Using the definition of $\bar{V}_k$ yields

$$\langle V-\bar{V}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h} \le \langle V-\bar{\bar{V}}_k, V-\bar{\bar{V}}_k\rangle_{\partial\Omega_k,h}.$$

We combine the above inequality with (2.6) and obtain

$$\left|V-\bar{V}_k\right|^2_{\partial\Omega_k,h} \le C\left|V-\bar{\bar{V}}_k\right|^2_{\partial\Omega_k}.$$

Applying (3.5) with $\epsilon = d$ and (3.6) to the right hand side of the last inequality gives

$$\left|V-\bar{V}_k\right|^2_{\partial\Omega_k,h} \le Cd\mathcal{A}_k(V,V). \tag{3.13}$$

Using this estimate in (3.12) proves (3.8). □

Remark 3.1. The preconditioning form $\mathcal{B}(\cdot,\cdot)$ defined above is not uniformly equivalent to $\mathcal{A}(\cdot,\cdot)$. Nevertheless, its preconditioning effect is very close to that of a uniform preconditioner for many practical problems, particularly in three space dimensions. The number of subdomains often equals the number of processors in a parallel implementation and it is feasible to keep d on the order of $h^{1/2}$. Applying a conjugate gradient method preconditioned by $\mathcal{B}(\cdot,\cdot)$ for solving (2.5) would result in a number of iterations proportional to $h^{-1/4}$. In $\mathbb{R}^3$, $h = 10^{-2}$ corresponds to a very large computational problem whereas $10^{1/2} \approx 3.2$.

Remark 3.2. The constants c and C in (3.8) depend on the local (with respect to the subdomains) behavior of the operator and the preconditioner. Clearly, one of the most influential factors on the local properties of $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ is the coefficient matrix $\{a_{i,j}\}|_{\Omega_k}$. In fact, the constants C_k in (3.4) depend on the local lower and upper bounds for the eigenvalues of $\{a_{i,j}\}|_{\Omega_k}$ and in general so do the constants c_k and C_k in (3.2). Therefore, in applications to problems with large jumps in the coefficients, it is desirable to align the subdomain boundaries with the locations of the jumps. In this case the preconditioner (3.3) will be independent of these jumps.

Remark 3.3. It is well known that classical overlapping domain decomposition algorithms with small overlap exhibit the same condition number growth but in contrast to our method the overlapping preconditioners are adversely sensitive to large jumps in the operator coefficients. The utilization of the averages $\bar{U}_k$ plays the role of a coarse problem especially designed to take into account cases with interior

subdomains and also applications with large jumps in the operator coefficients provided that the locations of the jumps are aligned with the subdomain boundaries. The numerical calculations in Section 7 indicate the effectiveness of our preconditioner when such problems are solved. To illustrate that the role of the averages is essential in overcoming difficulties coming from large jumps of the coefficients, we consider a conventional additive Schwarz preconditioner with minimal overlap [17]. The asymptotic condition number bound provided in [17] is the same as that of our theorem in the case of smooth coefficients. However, because of the deterioration in the approximation and boundedness properties of the weighted L^2 projection into the coarse subspace [12], the condition number of the preconditioned system for the minimal overlap algorithm when $n = 3$ can only be bounded by $(d/h)^2$.

Remark 3.4. It is possible to apply the above preconditioner to the discrete systems which arise from other types of numerical approximation. For example, it is straightforward to apply the technique to finite difference approximations. In addition, its application to non-conforming finite element discretizations as well as mixed finite element approximations is given in [22].

Our preconditioner is very economical computationally. In fact, it allows the use of efficient subdomain preconditioners such as one multigrid V–cycle. The use of the simple extension $\mathcal{E}$ also results in enhanced efficiency. We shall discuss the computational aspects of this algorithm in detail in the Section 5.

3.3. An additive Schwarz reformulation of the domain decomposition algorithm. We shall demonstrate that the preconditioner $\mathcal{B}(\cdot,\cdot)$ can be viewed as an additive subspace correction method (cf. [10] and [24]) with judiciously chosen subspaces. Let the linear operator $\tilde{\mathcal{E}} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ be defined by

$$\tilde{\mathcal{E}}V = \mathcal{E}V + \sum_{k=1}^{n_d} (\bar{V}_k - \mathcal{E}_k \bar{V}_k).$$

Furthermore, define

$$\hat{S}_h^0(\Omega) = \left\{ v \in S_h^0(\Omega) \,|\, v = 0 \quad \text{on}\, \Gamma \right\}$$

and

$$S_\Gamma(\Omega) = \{ \tilde{\mathcal{E}} v \mid v \in S_h^0(\Omega) \}.$$

It is immediate that $\hat{S}_h^0(\Omega)$ and $S_\Gamma(\Omega)$ provide a direct sum decomposition of $S_h^0(\Omega)$.

The additive Schwarz preconditioner applied to $g \in S_h^0(\Omega)$ based on the above two spaces results in a function $Y = Y_0 + Y_\Gamma$ where $Y_0 \in \hat{S}_h^0(\Omega)$ satisfies

$$\mathcal{B}_0(Y_0, \phi) = (g, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \tag{3.14}$$

and $Y_\Gamma \in S_\Gamma(\Omega)$ satisfies

$$\mathcal{B}_\Gamma(Y_\Gamma, \phi) = (g, \phi), \text{ for all } \phi \in S_\Gamma(\Omega). \tag{3.15}$$

Here $\mathcal{B}_0(\cdot,\cdot)$ and $\mathcal{B}_\Gamma(\cdot,\cdot)$ are symmetric and positive definite bilinear forms.

We shall see that the preconditioner in (3.3) is equivalent to the additive Schwarz method above when

$$\mathcal{B}_0(\varphi, \phi) = \sum_{k=1}^{n_d} \mathcal{B}_k(\varphi, \phi) \tag{3.16}$$

and

$$\mathcal{B}_\Gamma(\varphi,\phi) = h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle \varphi - \bar{\varphi}_k, \phi - \bar{\phi}_k\rangle_{\partial\Omega_k,h}. \tag{3.17}$$

Indeed, let W be the solution of (2.8). Then

$$\mathcal{B}(W,\varphi) = \mathcal{B}_k(W^{(k)},\varphi) = (g,\varphi)_\Omega, \text{ for all } \varphi \in S_h^0(\Omega_k), \tag{3.18}$$

where $W^{(k)} \equiv W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k)$. It follows that the function Y_0 satisfying (3.14) (with $\mathcal{B}_0(\cdot,\cdot)$ given by (3.16)) can be written

$$Y_0 = W - \tilde{\mathcal{E}}W \quad \text{on} \quad \Omega_k.$$

Taking $\varphi = \tilde{\mathcal{E}}V$ in (2.8) shows that for $\mathcal{B}_\Gamma(\cdot,\cdot)$ given by (3.17),

$$\mathcal{B}_\Gamma(W,\tilde{\mathcal{E}}V) = (g,\tilde{\mathcal{E}}V).$$

Thus, $Y_\Gamma = W$ on Γ. From the definition of $S_\Gamma(\Omega)$ it follows that $Y_\Gamma = \tilde{\mathcal{E}}W$, i.e, $W = Y_0 + Y_\Gamma$. Thus, the solution W of (2.8) is the result of the additive Schwarz algorithm with subspace decomposition given by $\hat{S}_h^0(\Omega)$ and $S_\Gamma(\Omega)$ with forms defined by (3.16) and (3.17).

4. Application to parabolic problems

Our preconditioning approach can be extended to more general bilinear forms of the type

$$\mathcal{A}(v,w) = \delta \sum_{i,j=1}^{n} \int_\Omega a_{ij} \frac{\partial v}{\partial x_i}\frac{\partial w}{\partial x_j}\, dx + (v,w)_\Omega.$$

Such forms arise from implicit time–stepping numerical approximations of parabolic problems. In such settings δ is related to the time step and is usually small. We shall consider the case when $ch^2 \le \delta \le Cd^2$.

We define our preconditioner $\mathcal{B}(\cdot,\cdot)$ by

$$\mathcal{B}(v,w) = \sum_{k=1}^{n_d} \mathcal{B}_k(v - \mathcal{E}_k v, w - \mathcal{E}_k w) + \frac{\delta}{h}\sum_{k=1}^{n_d} \langle w, v\rangle_{\partial\Omega_k,h},$$

where $\mathcal{B}_k(\cdot,\cdot)$ are the subdomain preconditioning forms satisfying (3.2). Note that the above form no longer includes the average values on the subdomain boundaries.

It is easy to see that

$$\begin{aligned} \mathcal{A}(v,v) &\le 2[\mathcal{A}(v - \mathcal{E}v, v - \mathcal{E}v) + \mathcal{A}(\mathcal{E}v, \mathcal{E}v)] \\ &\le C\left\{\sum_{k=1}^{n_d} \mathcal{B}_k(v - \mathcal{E}_k v, v - \mathcal{E}_k v) + (h + \frac{\delta}{h})\sum_{k=1}^{n_d}\langle v, v\rangle_{\partial\Omega_k,h}\right\} \\ &\le C\mathcal{B}(v,v). \end{aligned} \tag{4.1}$$

Moreover, applying (3.5) gives

$$\frac{\delta}{h}\langle v, v\rangle_{\partial\Omega_k,h} \le C\frac{\delta}{h}\left(\frac{1}{\epsilon}(v,v)_{\Omega_k} + \epsilon\mathcal{D}_k(v,v)\right).$$

Choosing $\epsilon = \max(\delta^{1/2}, d)$ in the last inequality yields

$$\frac{\delta}{h}\langle v, v\rangle_{\partial\Omega_k,h} \le C\frac{\delta^{1/2}}{h}\mathcal{A}_k(v,v). \tag{4.2}$$

Using (4.2) for each k as in the proof of Theorem 3.1, we obtain

$$\mathcal{B}(v,v) \leq C\frac{\delta^{1/2}}{h}\mathcal{A}(v,v), \tag{4.3}$$

Combining (4.1) and (4.3) shows that

$$c\mathcal{A}(v,v) \leq \mathcal{B}(v,v) \leq C\frac{\delta^{1/2}}{h}\mathcal{A}(v,v) \quad \text{for all} \quad v \in S_h^0(\Omega). \tag{4.4}$$

The resulting condition number depends on δ in a natural way. Smaller time steps correspond to better conditioning. Obviously, the preconditioner would be uniform if $\delta = h^2$ but such time stepping is too restrictive for the vast majority of applications. On the other hand, $\delta = h$ corresponds to a very reasonable time-stepping scheme whose condition number is governed by $h^{-1/2}$. Again, although not uniform, such rate of growth is often acceptable in practice for reasons already mentioned.

5. Computational aspects of the preconditioning problem

In this section, we provide an algorithm for applying the preconditioning operator correspnding to the form $\mathcal{B}(\cdot,\cdot)$. This consists of two main steps, solution of the approximate subdomain problems and inversion of the boundary form. As we shall see, these steps are independent and can be carried out in parallel.

5.1. The domain decomposition algorithm. The action of the preconditioner corresponding to $\mathcal{B}(\cdot,\cdot)$ is obtained by computing the solution of (2.8) for a given g. The first step involves the computation of $W^{(k)} \in S_h^0(\Omega_k)$ satisfying (3.18) and reduces to the solution of subdomain preconditioning problems which can be performed in parallel.

The second step involves the solution of a problem on Γ which we shall now describe. For $\psi \in S_h^0(\Omega)$ set $\varphi = \mathcal{E}\psi$. Notice that $\varphi = \mathcal{E}_k\psi$ on Ω_k, $(\overline{\mathcal{E}\psi})_k = \bar{\psi}_k$, and $\mathcal{E}_k^2\psi = \mathcal{E}_k\psi$. For this choice of φ, (2.8) becomes

$$\begin{aligned}\mathcal{B}(W,\varphi) = &\sum_{k=1}^{n_d} \mathcal{B}_k(W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k), \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k) \\ &+ h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k\langle W - \bar{W}_k, \psi\rangle_{\partial\Omega_k,h} = (g, \mathcal{E}\psi)_\Omega.\end{aligned} \tag{5.1}$$

Here we have used the fact the $W - \bar{W}_k$ has zero discrete mean value on $\partial\Omega_k$ and therefore is orthogonal to constants with respect to the inner product on $\partial\Omega_k$.

Since $\mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k$ vanishes on $\partial\Omega_k$,

$$\sum_{k=1}^{n_d} \mathcal{B}_k(W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k), \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k) = \sum_{k=1}^{n_d}(g, \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k)_{\Omega_k}$$

and hence

$$h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k\langle W - \bar{W}_k, \psi\rangle_{\partial\Omega_k,h} = (g, \mathcal{E}\psi)_\Omega - \sum_{k=1}^{n_d}(g, \mathcal{E}_k\bar{\psi}_k - \bar{\psi}_k)_{\Omega_k}. \tag{5.2}$$

Notice that because of the explicit extensions used in the definition of $\mathcal{B}(\cdot,\cdot)$, the setup of the right hand side in (5.2) involves minimal computational cost. Clearly, this step is independent of the previous one and thus the procedure solving the preconditioning problem with $\mathcal{B}(\cdot,\cdot)$ given by (3.3) decouples into two independent

tasks. Once $W^{(k)}$ and $W|_\Gamma$ are known then the assembly of the solution in Ω is easy. The actual implementation of the solution procedure for (5.2) is an important issue for the overall computational efficiency of the proposed preconditioner. We shall give a detailed description how to solve this problem in the next subsection.

The above discussion can be summarized in the following algorithm.

Algorithm 5.1. *Solve the preconditioning problem (2.8) by*

(1) *Compute the solution $W^{(k)}$ of (3.18) for each k.*
(2) *Compute the trace of W on Γ from (5.2).*
(3) *The solution of (2.8) is given by*

$$W = \mathcal{E}W + \sum_{k=1}^{n_d} (W^{(k)} + \bar{W}_k - \mathcal{E}_k \bar{W}_k).$$

5.2. The algorithm for inverting the boundary form. In this subsection we describe the algorithm for solving (5.2). As it was observed in the previous subsection (this applies also to Section 5 below), the algorithm for inverting the preconditioner (3.3) requires an efficient method for determining the averages $\bar{W}_k$ and finding the solution to (5.2). The implementation details of this method are described below. The algorithm for solving (5.2) was originally developed in [6] and it is included here for completeness.

We start by observing that the solution of (5.2) is trivial provided that $\bar{W}_k$ is known for each k. In fact, the resulting matrix is diagonal using the usual nodal basis for $S_h(\Gamma)$ and thus inverting it is straightforward. Therefore, we only have to describe how to solve for $\bar{W}_k$.

For $\ell = 1, \ldots, n_d$, let ψ^ℓ be the unique function in $S_h(\Gamma)$ which satisfies

$$h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle V, \psi^\ell \rangle_{\partial\Omega_k, h} = \bar{V}_\ell, \quad \text{for all } V \in S_h(\Gamma). \tag{5.3}$$

That such functions are uniquely defined follows from the Riesz Representation Theorem. Substituting ψ^ℓ in (5.2) gives

$$h^{-1} \left\{ \bar{W}_\ell - \sum_{k=1}^{n_d} \tilde{a}_k \bar{W}_k \langle 1, \psi^l \rangle_{\partial\Omega_k} \right\} = (g, \mathcal{E}\psi^\ell)_\Omega - \sum_{k=1}^{n_d} (g, \mathcal{E}_k \bar{\psi}_k^\ell - \bar{\psi}_k^\ell)_{\Omega_k}. \tag{5.4}$$

Setting $\bar{W} = [\bar{W}_1, \ldots, \bar{W}_{n_d}]^T$, (5.4) can be rewritten in a matrix form as

$$\mathbf{M}\bar{W} = G. \tag{5.5}$$

It was observed in [6] that the matrix $\mathbf{M}$ is symmetric and irreducibly diagonally dominant and hence positive definite. Thus (5.4) is solvable. One efficiently implements the above algorithm by explicitly computing the functions $\{\psi^\ell\}$. We illustrate this construction in the case when the operator $\mathcal{L}$ from (2.2) is the Laplacian in two spatial dimensions and $\tilde{a}_k = 1$, $k = 1, \ldots, n_d$. To do this we need to define some additional notation. The nodes on $\Gamma \setminus \partial\Omega$ that are shared by exactly m subdomains will be called m-**edge** nodes. For example, in the picture shown in Fig. 5.1, all nodes on $\Gamma \setminus \partial\Omega$ but node E are 2-**edge** nodes. Node E of this example is a 4-**edge** node. With this terminology in mind, we define ψ^ℓ by

$$\psi^\ell(x_i) = \begin{cases} \frac{1}{mN_\ell}, & \text{if } x_i \in \partial\Omega_\ell \setminus \partial\Omega \text{ and } x_i \text{ is an } m\text{-}\mathbf{edge}, \\ 0, & \text{elsewhere.} \end{cases} \tag{5.6}$$

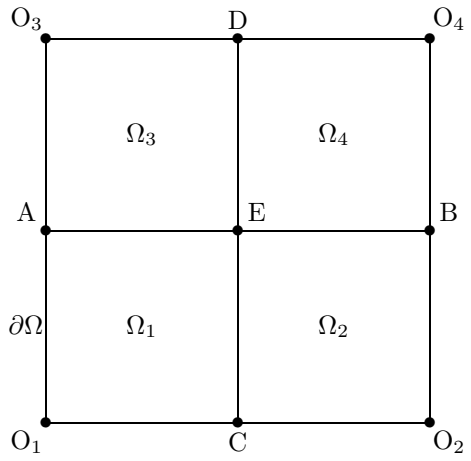

FIGURE 5.1. A simple example with four subdomains.

Here N_ℓ is the number of nodes on $\partial\Omega_\ell$. For the example shown in Fig. 5.1, there will be four such basis functions. The function associated with the first subdomain is ψ^1 such that $\psi^1(E) = 1/(4N_1)$; $\psi^1 \equiv 1/(2N_1)$ at all remaining nodes on the edges AE and EC, the points A and C excluded; $\psi^1 \equiv 0$ on the edges AO_1, O_1C, and everywhere in the exterior and interior of Ω_1.

This approach to solving the problem for the averages extends to three dimensional problems as well as the case when $\tilde{a}_k \neq 1$. The reader is referred to [6] for further details.

6. Alternative additive preconditioners with inexact solves

In this section, we consider a classical technique for developing non-overlapping domain decomposition preconditioners. The behavior of such methods has been investigated in the case when the boundary form is uniformly equivalent to the corresponding Schur complement subsystem [2], [19]. Here, we show that this method also reduces to an additive Schwarz preconditioner. In addition, we show that the inexact solve technique combined with the boundary form discussed earlier provides an effective preconditioner. Indeed, our results are much better than what would be expected from the analysis of [2], [19].

6.1. Matrix form of the inexact solve domain decomposition algorithm. The classical inexact domain decomposition preconditioners are easily understood from the matrix point of view. In this case, one orders the unknowns so that the stiffness matrix corresponding to $\mathcal{A}(\cdot,\cdot)$ can be written in a block form as

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}.$$

Here $\mathbf{A}_{22}$ corresponds to the nodes on Γ and $\mathbf{A}_{11}$ to the remaining nodes. With this ordering, the form corresponding to a typical domain decomposition preconditioner

(e.g., [5],[6],[7],[8]) has a stiffness matrix of the form

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{Z} \end{pmatrix},$$

where $\mathbf{Z} = \mathbf{B}_{22} + \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$ and $\mathbf{B}_{22}$ is the domain decomposition boundary preconditioning matrix. Computing the inverse of $\hat{\mathbf{A}}$ applied to a vector reduces to a three step block Gaussian elimination procedure.

The classical inexact method is defined by replacing $\mathbf{A}_{11}$ with $\mathbf{B}_{11}$ where $\mathbf{B}_{11}$ is another symmetric and positive definite matrix. This defines a new preconditioning operator $\mathbf{B}$ given by

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \tilde{\mathbf{Z}} \end{pmatrix}. \tag{6.1}$$

Here $\tilde{\mathbf{Z}}$ is given by $\tilde{\mathbf{Z}} = \mathbf{B}_{22} + \mathbf{A}_{21}\mathbf{B}_{11}^{-1}\mathbf{A}_{12}$.

Generally, the inexact algorithm may not converge as well as the exact version. Even if one takes $\mathbf{B}_{22}$ to be the Schur complement, $\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{B}_{11}^{-1}\mathbf{A}_{12}$, the inexact preconditioner may perform poorly unless the difference between the two matrices $\mathbf{B}_{11}$ and $\mathbf{A}_{11}$ is sufficiently small in an appropriate sense (see Theorem 6.1).

6.2. The inexact algorithm as a two level additive Schwarz procedure. We now show that the inexact preconditioners correspond to additive Schwarz methods. The first subspace in this decomposition is $\hat{S}_h^0(\Omega)$. Let $\mathcal{B}_0(\cdot,\cdot)$ be the form on $\hat{S}_h^0(\Omega) \times \hat{S}_h^0(\Omega)$ with stiffness matrix $\mathbf{B}_{11}$. The second subspace is given by

$$\begin{aligned} \hat{S}_h(\Gamma) = \Big\{ & \mathcal{E}\varphi + \varphi_0 \,|\, \varphi \in S_h^0(\Omega) \text{ and } \varphi_0 \in \hat{S}_h^0(\Omega) \text{ such that} \\ & \mathcal{B}_0(\varphi_0, \phi) = -\mathcal{A}(\mathcal{E}\varphi, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \Big\}. \end{aligned} \tag{6.2}$$

Clearly, the functions in $\hat{S}_h(\Gamma)$ are completely determined by their traces on Γ. Let $\mathcal{B}_\Gamma(\cdot,\cdot)$ be the form on $\hat{S}_h(\Gamma) \times \hat{S}_h(\Gamma)$ with stiffness matrix $\mathbf{B}_{22}$. $\mathcal{B}_\Gamma(u,v)$ only depends on the boundary nodal values of u and v and is thus defined on $S_h^0(\Omega) \times S_h^0(\Omega)$ by restriction.

Clearly, $\hat{S}_h^0(\Omega)$ and $\hat{S}_h(\Gamma)$ provide a direct sum decomposition of $S_h^0(\Omega)$. This decomposition is tied strongly to the bilinear form $\mathcal{B}_0(\cdot,\cdot)$. In particular, if $\mathcal{B}_0(\cdot,\cdot) \equiv \mathcal{A}(\cdot,\cdot)$ on $\hat{S}_h^0(\Omega) \times \hat{S}_h^0(\Omega)$, then the space $\hat{S}_h(\Gamma)$ consists of discrete harmonic functions and the decomposition is $\mathcal{A}(\cdot,\cdot)$–orthogonal. In general, the decomposition is not $\mathcal{A}(\cdot,\cdot)$–orthogonal.

6.3. Conditioning estimates for the inexact algorithms. The preconditioner defined by (6.1) can be restated as an operator $\mathbf{B} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$. In fact, it is a straightforward exercise to check that the block Gaussian elimination procedure applied to the matrix $\mathbf{B}$ of (6.1) corresponds to the preconditioning operator defined in the following algorithm.

Algorithm 6.1. *Given $g \in S_h^0(\Omega)$ we define $\mathbf{B}^{-1}g = U$ where U is computed as follows:*

(1) *Compute $U_0 \in \hat{S}_h^0(\Omega)$ by solving*

$$\mathcal{B}_0(U_0, \varphi) = (g, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega). \tag{6.3}$$

(2) *Compute the trace* U_Γ *on* Γ *by solving*

$$\mathcal{B}_\Gamma(U_\Gamma, \mathcal{E}\phi) = (g, \mathcal{E}\phi) - \mathcal{A}(U_0, \mathcal{E}\phi) \quad \textit{for all} \quad \phi \in \hat{S}_h(\Gamma).$$

(3) *Compute* $U_{\Gamma 0}$ *by solving*

$$\mathcal{B}_0(U_{\Gamma 0}, \varphi) = -\mathcal{A}(\mathcal{E}U_\Gamma, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega).$$

(4) *Set* $U = U_0 + \mathcal{E}U_\Gamma + U_{\Gamma 0}$.

Although the above algorithm appears as a multiplicative procedure, we shall now demonstrate that it is equivalent to an additive Schwarz method. The problem solved in Step (2) of Algorithm 6.1 is independent of U_0. Indeed, for any $\phi \in \hat{S}_h(\Gamma)$, we decompose $\phi = \mathcal{E}\phi + \phi_0$ as in (6.2) and observe

$$-\mathcal{A}(\mathcal{E}\phi, U_0) = \mathcal{B}(\phi_0, U_0) = (g, \phi_0).$$

Thus, Steps (2) and (3) of the above algorithm reduce to finding $U_\Gamma = \mathcal{E}U_\Gamma + U_{\Gamma_0} \in \hat{S}_h(\Gamma)$ such that

$$\mathcal{B}_\Gamma(U_\Gamma, \phi) = (g, \phi) \text{ for all } \phi \in \hat{S}_h(\Gamma). \tag{6.4}$$

Hence, $\mathbf{B}^{-1}g = U = U_0 + U_\Gamma$ where U_0 and U_Γ satisfy (6.3) and (6.4) respectively, i.e., Algorithm 6.1 is an implementation of an additive Schwarz procedure.

Notice that Algorithm 6.1 avoids the need of knowing explicitly a basis for the space $\hat{S}_h(\Gamma)$ which could be either a computationally expensive problem or a significant complication of the overall algorithm. Obviously this procedure provides inexact variants of the methods given in [5], [6], [7], [8], [14] and [23].

Since $\hat{S}_h^0(\Omega)$ and $\hat{S}_h(\Gamma)$ give a direct sum decomposition of $S_h^0(\Omega)$, the preconditioning form $\mathcal{B}(\cdot,\cdot)$ corresponding to the operator defined in Algorithm 6.1 is given by

$$\mathcal{B}(V, V) = \mathcal{B}_0(V_0, V_0) + \mathcal{B}_\Gamma(V_\Gamma, V_\Gamma). \tag{6.5}$$

Here $V = V_0 + V_\Gamma$ with $V_0 \in \hat{S}_h^0(\Omega)$ and $V_\Gamma \in \hat{S}_h(\Gamma)$. In the remainder of this section we analyze the above preconditioner by providing bounds for (6.5). We take

$$\mathcal{B}_0(u, v) = \sum_{k=1}^{n_d} \mathcal{B}_k(u, v)$$

where $\mathcal{B}_k(\cdot,\cdot)$ is defined as in Section 3.

The first theorem in this section was given by Börgers [2] and Haase at al. [19] and provides a result when $\mathbf{B}_{22}$ is uniformly equivalent to the Schur complement $\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$. This is the same as assuming that the quadratic form $\mathcal{B}_\Gamma(\cdot,\cdot)$ is equivalent to the boundary form with diagonal

$$\inf_{\phi \in \hat{S}_h^0(\Omega)} A(u + \phi, u + \phi), \text{ for all } u \in \hat{S}_h(\Gamma). \tag{6.6}$$

Theorem 6.1. *Let* $\mathcal{A}(\cdot,\cdot)$ *be given by (2.3) and* $\mathcal{B}(\cdot,\cdot)$ *by (6.5) respectively. Assume that the quadratic form* $\mathcal{B}_\Gamma(\cdot,\cdot)$ *is uniformly equivalent to the quadratic form induced by (6.6). In addition, let* γ *be the smallest positive constant such that*

$$|\mathcal{A}(\varphi, \varphi) - \mathcal{B}(\varphi, \varphi)| \le \gamma \mathcal{A}(\varphi, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega). \tag{6.7}$$

Then

$$c\left(\frac{\gamma^2}{h}\right)^{-1} \mathcal{A}(U, U) \le \mathcal{B}(U, U) \le C\frac{\gamma^2}{h}\mathcal{A}(U, U) \tag{6.8}$$

holds for all $U \in S_h^0(\Omega)$ *with constants* c *and* C *independent of* d *and* h.

Remark 6.1. Condition (6.7) requires $\mathcal{B}_0(\cdot,\cdot)$ to be a good approximation to $\mathcal{A}(\cdot,\cdot)$ for the preconditioner (6.5) to be efficient. The result of the theorem shows that if (6.7) holds with γ on the order of $h^{1/2}$, then the preconditioner $\mathcal{B}(\cdot,\cdot)$ is uniform. However, the development of a form $\mathcal{B}_0(\cdot,\cdot)$ satisfying (6.7) usually involves significant additional computational work since γ must tend to zero as h becomes small. Alternatively keeping γ fixed independent of h may result in a rather ill–conditioned method when h is small. There are examples of reasonably accurate preconditioners $\mathcal{B}_0(\cdot,\cdot)$, e.g. multigrid V– or W–cycles, which appear to perform well when h is not very small (cf. [2]) due to the fact that the corresponding γ's are comparable to $h^{1/2}$.

The main result of this section is given in the next theorem. It is for the case when

$$\mathcal{B}_\Gamma(u,v) = h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle u-\bar{u}_k, v-\bar{v}_k\rangle_{\partial\Omega_k,h}, \quad \text{for all} \quad u,\, v \in \hat{S}_h(\Gamma). \tag{6.9}$$

Theorem 6.2. *Let* $\mathcal{A}(\cdot,\cdot)$ *be given by (2.3),* $\mathcal{B}(\cdot,\cdot)$ *be given by (6.5), and* $\mathcal{B}_\Gamma(\cdot,\cdot)$ *defined by (6.9). Then*

$$c\mathcal{A}(U,U) \le \mathcal{B}(U,U) \le C\frac{d}{h}\mathcal{A}(U,U) \tag{6.10}$$

holds for all $U \in S_h^0(\Omega)$ *with constants* c *and* C *independent of* d *and* h.

Remark 6.2. The result of Theorem 6.2 shows that introducing inexact solves in the interior of the subdomains does not degrade the overall preconditioning effect of the corresponding exact method analyzed in [6]. As we have pointed out in Section 3, the adverse effect on the condition number of h approaching zero can be compensated easily by adjusting the parameter d. This balance is an alternative to (6.7) and could be a better choice when h is small relative to γ. In fact, the utilization of the bilinear form (6.9) leads to computationally efficient algorithms, unconstrained by accuracy conditions like (6.7). We shall see in Section 7 that for this boundary form the differences in the preconditioning effect of the inexact (Algorithm 6.1) and exact (cf. [6]) methods are negligible. However, the saving of computational time is significantly in favor of Algorithm 6.1.

We conclude this section with the proof of Theorem 6.2.

Proof of Theorem 6.2. Because of (6.5), the technique for establishing (6.10) is similar to the one used in the proof of Theorem 3.1.

Let $U_\Gamma = \mathcal{E}U_\Gamma + U_{\Gamma 0}$ as in (6.2) and write $U = U_0 + U_\Gamma$. The first inequality in (6.10) follows from the arithmetic–geometric mean inequality and the assumptions on $\{\mathcal{B}_k(\cdot,\cdot)\}$. Indeed, we have

$$\begin{aligned} \mathcal{A}(U,U) &= \mathcal{A}(U_0+U_\Gamma, U_0+U_\Gamma) \\ &\le C\left(\mathcal{B}_0(U_0,U_0) + \mathcal{B}_0(U_{\Gamma 0},U_{\Gamma 0}) + \mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma)\right). \end{aligned} \tag{6.11}$$

If follows from the definition of $U_{\Gamma 0}$ that

$$\mathcal{B}_0(U_{\Gamma 0},U_{\Gamma 0}) \le C\mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma). \tag{6.12}$$

Using (6.12) together with (3.7) and (3.4) in (6.11) yields

$$\mathcal{A}(U,U) \le C\mathcal{B}(U,U).$$

To prove the right hand inequality in (6.10), we use again the decomposition of U. Thus,

$$\begin{aligned} \mathcal{B}_0(U_0,U_0) &\le C\mathcal{A}(U-U_\Gamma,U-U_\Gamma) \le C(\mathcal{A}(U,U)+\mathcal{A}(\mathcal{E}U_\Gamma,\mathcal{E}U_\Gamma)) \\ &\le C\left(\mathcal{A}(U,U)+\mathcal{B}_\Gamma(U_\Gamma,U_\Gamma)\right). \end{aligned} \tag{6.13}$$

Hence, we need to estimate $\mathcal{B}_\Gamma(U_\Gamma,U_\Gamma)$ from above by $\mathcal{A}(U,U)$. Applying the reasoning used to show (3.13) in (6.13) gives the desired bound. □

7. Numerical examples

In this section we present numerical calculations involving the non-overlapping domain decomposition preconditioners developed in Section 3 and Section 5. We report results obtained from examples with Algorithm 5.1 and Algorithm 6.1 with boundary form given by (6.9). We tested two main aspects of these preconditioners, namely the computational efficiency of the method, in terms of the condition numbers obtained, and the independence of the jumps in the operator coefficients $\{a_{ij}\}$. Comparisons between the inexact algorithms and the corresponding exact methods are included as well.

The numerical results presented in this section are applied to

$$\mathcal{L} = -\nabla\cdot a\nabla, \tag{7.1}$$

where a is a piecewise constant function in Ω and constant on each subdomain. In all of our calculations Ω is the unit cube in three spatial dimensions. The subdomains are obtained by subdividing Ω into regions by slicing it parallel to the coordinate axes. Here we shall consider only cases where the unit cube is split into m^3 equal sub-cubes, which implies $d = 1/m$. In the examples below, $S_h^0(\Omega)$ is the space of piecewise linear functions with respect to a uniform mesh of size h. Also, the action of one multigrid V-cycle is used as an inexact solver in the interior of the subdomains.

The multigrid algorithm is variational and based on a trilinear finite element approximation. A nested sequence of approximation subspaces is defined by successively doubling the mesh size. For computational efficiency, the fine grid form is defined by numerical quadrature utilizing a quadrature which gives rise to a seven point operator. The operators on the coarser grids are twenty seven point and determined variationally from the fine gird operator. The analysis of variational multigrid procedures based on a fine grid operator defined by numerical quadrature can be found in [3]. Pointwise forward and backward Gauss–Seidel sweeps are used as pre- and post-smoothing iterations respectively. On the coarsest level we apply five pairs of forward and backward Gauss–Seidel sweeps. Obviously, if we have only one degree of freedom on the coarsest level, then this is equivalent to an exact solve on that level. This multigrid procedure results in a symmetric and positive definite operator whose action provides an inexact interior solve. The corresponding $\mathcal{B}_k(\cdot,\cdot)$ satisfies (3.2) with uniform constants c_k and C_k for each k. Also, the evaluation of the action of this operator is proportional to the number of grid points on the mesh used for the discretization of Ω_k.

The first cases which we report are intended to confirm numerically the d/h-like behavior of the condition number K, established in Theorem 3.1. We consider the model problem (2.1) with $\mathcal{L} \equiv -\Delta$. The results are presented in Table 7.1. According to our theory, the condition number K should be bounded if d/h is fixed. This is clearly indicated in the computational results of Table 7.1.

TABLE 7.1. Condition numbers with the inexact preconditioner (3.3).

h	$d = 1/3$	$d = 1/6$
1/12	21.46	8.12
1/24	55.70	23.20
1/48	131.19	59.33

TABLE 7.2. Condition numbers with the inexact preconditioner (3.3); $d = 1/4$.

h	Variable a	$a \equiv 1$
1/12	15.71	13.87
1/24	42.94	39.79
1/48	106.76	95.38

TABLE 7.3. Comparison of the inexact and the exact methods; $d = 1/3$.

h	K_{exact}	K-Algorithm 5.1	K-Algorithm 6.1
1/6	6.27	6.73	6.27
1/12	15.23	21.96	15.40
1/24	32.55	57.01	33.83
1/48	66.12	130.88	70.76

The second set of calculations illustrate that the condition number for the preconditioner defined in (3.3) can be bounded independently of large jumps in the operator coefficients. The data in Table 7.2 represent experimental results where Ω is split into $4 \times 4 \times 4$ subdomains. The coefficient a in (7.1) is defined as follows: $a_{222} = a_{333} = 10^5$, a is a constant in the interval [0.1, 21.1] for the remaining subdomains. Here a_{ijk} is the operator coefficient in the subdomain with integer coordinates i, j, k. The largest jump in the operator coefficient between two neighboring subdomains in this case is 10^6. For comparison, we have included the corresponding condition numbers for the case when $a \equiv 1$ in Ω. Clearly, the results in Table 7.2 are in good agreement with Remark 3.2.

Our final numerical example is a comparison of the performance of the inexact preconditioners (3.3) and (6.5) with $\mathcal{B}_\Gamma(\cdot,\cdot)$ given by (6.9), and the exact method analyzed in [6]. The piecewise constant coefficient a in this case is defined according to the data for μ in Example 3 in [6]. We note that the condition numbers for the exact method reported in Table 7.3 are better than the ones reported in Table 4.5 in [6] due to the different scaling of the boundary form (cf. Remark 2.5, [6]). The data in Table 4.5, [6] are obtained when the boundary form is scaled by d^{-1} whereas the results in Table 7.3 are obtained with the scaling h^{-1}. Clearly, the exact preconditioner and the inexact method implemented by Algorithm 6.1 exhibit almost the same condition numbers which is in good agreement with Remark 6.2.

Although the condition numbers reported for these two methods are better than those for Algorithm 5.1, one application of the inexact preconditioner (3.3) requires substantially less computer time thus resulting in a more efficient computational algorithm. We illustrate this with some timing statistics made on a SUN Sparc 20/502 workstation. For mesh sizes between 1/12 and 1/48, the inexact preconditioner (Algorithm 5.1) was more than 4.5 times faster to evaluate than the

exact method which used the Fast Fourier transform to diagonalize the stiffness matrix. This translates into an overall factor of three reduction in computing time in the case of the grid with $h = 1/48$ and a problem solved by a preconditioned conjugate gradient iteration. A similar comparison of Algorithm 5.1 and Algorithm 6.1 indicates that Algorithm 5.1 is about 25 percent more efficient.

References

1. P.E. Bjørstad and O.B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal. **23** (1986), 1097–1120. MR **88h:**65188
2. C. Börgers, *The Neumann–Dirichlet Domain Decomposition Method with Inexact Solvers on the Subdomains*, Numerische Mathematik **55** (1989), 123–136. MR **90f:**65191
3. J.H. Bramble, C.I. Goldstein, and J.E. Pasciak, *Analysis of V–cycle multigrid algorithms for forms defined by numerical quadrature*, SIAM Sci. Stat. Comp. **15** (1994), 566–576. MR **95b:**65047
4. J.H. Bramble, J.E. Pasciak and A.H. Schatz , *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp. **46** (1986), 361–369. MR **88a:**65123
5. J.H. Bramble, J.E. Pasciak and A.H. Schatz , *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), 103–134. MR **87m:**65174
6. J.H. Bramble, J.E. Pasciak and A.H. Schatz , *The construction of preconditioners for elliptic problems by substructuring, II*, Math. Comp. **49** (1987), 1–16. MR **88j:**65248
7. J.H. Bramble, J.E. Pasciak and A.H. Schatz , *The construction of preconditioners for elliptic problems by substructuring, III*, Math. Comp. **51** (1988), 415–430. MR **89e:**65118
8. J.H. Bramble, J.E. Pasciak and A.H. Schatz , *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp. **53** (1989), 1–24. MR **89m:**65098
9. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition*, Math. Comp. **57** (1991), 1–21. MR **92d:**65094
10. J.H. Bramble, J.E. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp. **55** (1990), 1–22. MR **90k:**65170
11. J.H. Bramble, J.E. Pasciak and J. Xu, *A multilevel preconditioner for domain decomposition boundary systems*, Proceedings of the 10'th Inter. Conf. on Comput. Meth. in Appl. Sci. and Engr., Nova Sciences, New York, 1992.
12. J.H. Bramble and J. Xu, *Some estimates for weighted L^2 projections*, Math. Comp. **56** (1991), 463–476. MR **91k:**65140
13. L.C. Cowsar, J. Mandel, and M.F. Wheeler, *Balancing Domain Decomposition for Mixed Finite Elements*, Math. Comp. **64** (1995), 989–1015. MR **95j:**65161
14. M. Dryja, *A capacitance matrix method for the Dirichlet problem on a polygonal region*, Numer. Math. **39** (1982), 51–64. MR **83g:**65102
15. M. Dryja, *A method of domain decomposition for three-dimensional finite element elliptic problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, (eds, R. Glowinski, G.H. Golub, G.A. Meurant, and J. Périaux) SIAM, Phil. PN, 1988, pp. 43–61. MR **90b:**65200
16. M. Dryja, B.F. Smith, and O.B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, SIAM J. Num. Anal. **31** (1994), 1662–1694. MR **95m:**65211
17. M. Dryja and O.B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), 604–620. MR **95d:**65102
18. R. Gonzalez and M.F. Wheeler, *Domain decomposition for elliptic partial differential equations with Neumann boundary conditions.*, Proceedings, International conference on vector and parallel computing—issues in applied research and development, Loen, 1986., vol. 5, Parallel Comput., 1987, pp. 257–263. MR **88d:**65148
19. G. Haase, U. Langer, and A. Meyer, *The Approximate Dirichlet Domain Decomposition Method. Part I: An Algebraic Approach*, Computing **47** (1991), 137–151. MR **93e:**65146a
20. S.V. Nepomnyaschikh, *Application of domain decomposition to elliptic problems with discontinuous coefficients*, Fourth International Symposium on Domain Decomposition Methods for

Partial Differential Equations, (eds, R. Glowinski, Y.A. Kuznetzov, G.A. Meurant, and J. Périaux) SIAM, Phil. PN, 1991, pp. 242–251. CMP 91:12

21. B.F. Smith, *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*, Ph.D. Thesis, Courant Institute of Mathematical Sciences, Dept. of Computer Science Tech. Rep. 517, New York, 1990.
22. A.T. Vassilev, *On Discretization and Iterative Techniques for Second–Order Problems with Applications to Multiphase Flow in Porous Media*, Ph.D. Thesis, Texas A&M University, College Station, Texas, 1996.
23. O. Widlund, *Iterative substructuring methods: algorithms and theory for elliptic problems in the plane*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations (R. Glowinski, G.H. Golub, G.A. Meurant, and J. Périaux, eds.), SIAM, Phil. PN, 1988, pp. 113–128. MR **90c:**65138
24. J. Xu, *Iterative methods by space decomposition and subspace correction*, vol. 34, SIAM Review, 1992, pp. 581–613. MR **93k:**65029

Department of Mathematics, Texas A&M University, College Station, Texas 77843
E-mail address: bramble@math.tamu.edu

Department of Mathematics, Texas A&M University, College Station, Texas 77843
E-mail address: pasciak@math.tamu.edu

Schlumberger, 8311 N. FM 620 Rd., Austin, Texas 78726
E-mail address: vassilev@slb.com