5

Non-overlapping Domain Decomposition Preconditioners with Inexact Solves

James Bramble, Joseph Pasciak and Apostol Vassilev

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 [BW86], [BPS86b], [Dry89]. To handle the case of cross-points, Bramble, Pasciak and Schatz introduced in [BPS86a] 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 [Wid88] including a Neumann-Dirichlet checkerboard like preconditioner. Subsequently, these techniques were extended to problems in three dimensions in [BPS89] and [Dry88]. A critical ingredient in the three dimensional algorithms was a coarse grid problem involving the solution averages developed in [BPS87]. Related work is contained in [CMW95], [Nep91], [Smi90].

The papers [BPS86b], [BPS86a], [BPS87], [BPS88], and [BPS89] 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 [BW86], [BPX91], [CMW95], and [Smi90]. 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.

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

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 [BPWX91]. However, in parallel implementations, the amount of inter-processor communication is proportional to the amount of overlap. These methods loose some efficiency as the overlap becomes smaller [DW94]. 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 [GW87]. References to other experimental work can be found in [DSW94]. 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 [B8̈9] and [HLM91]. 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 develop new non-overlapping domain decomposition preconditioners with inexact solves. We provide variations of the exact algorithm considered in [BPS87]. 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 [BPS87] and are much more efficient computationally. Our algorithms are alternatives to and in many applications less restrictive than the preconditioners in [B8̈9] and [HLM91]. The convergence estimates developed here are independent of jumps of the operator coefficients across subdomain boundaries.

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 first algorithm of this paper involves a coarse subspace utilizing a simple extension defined in terms of the the average value of the function on the boundary. After preparing this manuscript, it has come to our attention that this extension was also used in a recent paper by Bjørstad, Dryja and Vainikko [BDV96] which was presented in the Eight Domain Decomposition meeting in the summer of 1995. Both the present paper and the one just mentioned rely on the use and analysis of a boundary form defined in terms of boundary averages. This boundary form was also analyzed in [BPS87].

The second algorithm in the present paper is a classical domain decomposition algorithm with inexact solves. It is shown to be an additive Schwarz procedure with special subspace decomposition. The particular decomposition depends on the inexact solve and thus needs to be investigated differently from the standard additive Schwarz approach. Finally, the results and analysis of the current paper were presented by the second author at the Seventh Copper Mountain Multigrid Conference in April of 1995.

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

$$\mathcal{L}u = f \qquad \text{in} \quad \Omega, \tag{1a}$$
$$u = 0 \qquad \text{on} \quad \partial\Omega, \tag{1b}$$

where f is a given function, $\Omega \subset \mathbb{R}^n$ $(n = 1, 2, 3)$ is a bounded polyhedral domain with Lipshitz boundary, and

$$\mathcal{L}v = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial v}{\partial x_j}\right). \tag{2}$$

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.

The generalized Dirichlet form on Ω is given by

$$\mathcal{A}(v, w) = \sum_{i,j=1}^{n} \int_\Omega a_{ij}\partial_i v\, \partial_j w\ dx. \tag{3}$$

This symmetric bilinear 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

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

Let $H_0^1(\Omega)$ be 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 (1) in $H_0^1(\Omega)$ is then given by the following.

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

$$\mathcal{A}(u, \varphi) = (f, \varphi), \quad \text{for all} \quad \varphi \in H_0^1(\Omega). \tag{4}$$

Given a finite dimensional subspace $S_h^0(\Omega)$ of $H_0^1(\Omega)$, the standard Galerkin approximation to (4) is defined by:

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

$$\mathcal{A}(u_h, \varphi) = (f, \varphi), \quad \text{for all} \quad \varphi \in S_h^0(\Omega). \tag{5}$$

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.

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

$$c\,|v|_{\partial\Omega_k}^2 \le |v|_{\partial\Omega_k,h}^2 \le C\,|v|_{\partial\Omega_k}^2 \tag{6}$$

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 parameters such as h, 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

$$\mathcal{D}_k(v, w) = \sum_{i=1}^{n} \int_{\Omega_k} \partial_i v \partial_i w\ dx, \quad \text{for all} \quad v,\, w \,\in\, H^1(\Omega_k). \tag{7}$$

The development of a method for efficient iterative solution of (5) is the subject of our considerations in this section. In particular, using the decomposition of Ω described above, 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

$$\mathcal{B}(W,\varphi) = (g,\varphi)_\Omega \quad \text{for all} \quad \varphi \in S_h^0(\Omega), \tag{8}$$

with g given, should be more efficient to compute than the solution of (5). Second, the two forms should be equivalent in the sense that

$$\lambda_1 \mathcal{B}(V,V) \leq \mathcal{A}(V,V) \leq \lambda_2 \mathcal{B}(V,V) \quad \text{for all} \quad V \in S_h^0(\Omega), \tag{9}$$

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 $\mathcal{B}(\cdot,\cdot)$

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

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

We remind that the functions in the finite element spaces defined above are fully determined by their values at the grid nodes and thus it is sufficient to define the extensions $\mathcal{E}_k$ at the nodal points x_i. Also, $\mathcal{E}_k$ can be viewed as a linear operator $S_h^0(\Omega) \to S_h^0(\Omega)$ with a trivial modification of the above definition, namely

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

We shall use $\mathcal{E}_k$ in both contexts since it will be easy to determine which is the right one from the functions $\mathcal{E}_k$ is applied to.

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

$$\mathcal{E}\phi(x_i) = \begin{cases} \phi(x_i) & \text{for } x_i \in \Gamma, \\ 0 & \text{for } x_i \in \Omega \setminus \Gamma. \end{cases} \tag{10}$$

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 (3) but with integration only on 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

$$c_k \mathcal{B}_k(V,V) \leq \mathcal{A}_k(V,V) \leq C_k \mathcal{B}_k(V,V), \quad \text{for all} \quad V \in S_h^0(\Omega_k). \tag{11}$$

The preconditioning form is given by

$$\begin{aligned}\mathcal{B}(U,V) = \sum_{k=1}^{n_d} \mathcal{B}_k(U-\bar{U}_k - \mathcal{E}_k(U-\bar{U}_k), V-\bar{V}_k - \mathcal{E}_k(V-\bar{V}_k)) \\ + h^{-1}\sum_{k=1}^{n_d} \tilde{a}_k \langle U-\bar{U}_k, V-\bar{V}_k\rangle_{\partial\Omega_k,h}.\end{aligned} \tag{12}$$

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 (12), $\tilde{a}_k,\ k=1,\dots,n_d$ are parameters. 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

$$C_k^{-1}\tilde{a}_k\mathcal{D}_k(v,v) \le \mathcal{A}_k(v,v) \le C_k\tilde{a}_k\mathcal{D}_k(v,v), \quad \text{for all} \quad v \in S_h(\Omega_k), \tag{13}$$

where C_k depends only on the local variation of the coefficients $\{a_{ij}\}$ on the subdomain Ω_k. Consequently, we will assume that (13) holds with C_k/c_k bounded independently of d, h, and k.

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

$$|u|^2_{\partial\Omega_k} \le C\{\epsilon^{-1}\,\|u\|^2_{\Omega_k} + \epsilon\mathcal{D}_k(u,u)\}, \tag{14}$$

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

$$\|v\|^2_{\Omega_k} \le Cd^2\mathcal{D}_k(v,v) \tag{15}$$

holds for functions v with zero mean value on Ω_k.

The inequalities (14) and (15) 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 theorem establishes bounds for the asymptotic behavior of the preconditioner $\mathcal{B}(\cdot,\cdot)$.

Theorem 3.1 *Let $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ be given by (3) and (12), respectively. Then there exist positive constants c and C not depending on d or h such that*

$$c\mathcal{A}(V,V) \le \mathcal{B}(V,V) \le C\frac{d}{h}\mathcal{A}(V,V), \tag{16}$$

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

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 spatial dimensions. The number of subdomains often equals the number of processors in a parallel implementation and it is now feasible to keep* d *on the order of* $h^{1/2}$. *Applying a conjugate gradient method preconditioned by* $\mathcal{B}(\cdot,\cdot)$ *for solving* (5) *would result in a number of iterations proportional to* $h^{-1/4}$. *In* $\mathbb{R}^3$, *if* Ω *is the unit cube,* $h = 10^{-2}$ *corresponds to a very large computational problem whereas* $10^{1/2} \approx 3.2$. *Also, 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 (see Remark 3.3 below).*

Remark 3.2 *The constants* c *and* C *in Theorem 3.1 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* (13) *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* (11). *Therefore, in applications to problems with large jumps in the coefficients, it is desirable, if possible, to align the subdomain boundaries with the locations of the jumps. In this case the preconditioner* (12) *will be independent of these jumps.*

Remark 3.3 *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. To illustrate that the role of the averages in overcoming difficulties coming from large jumps of the coefficients is essential, we consider a conventional additive Schwarz preconditioner with minimal overlap [DW94]. The asymptotic condition number bound provided in [DW94] 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 [BX91], the condition number of the preconditioned system for the minimal overlap algorithm when* $n = 3$ *can only be bounded by* $(d/h)^2$.

Our preconditioner is very economical computationally. In fact, it allows the use of efficient subdomain preconditioners such as one multigrid V-cycle (cf. [Bra93]). The use of the simple extension $\mathcal{E}$ also results in enhanced efficiency.

4 An Additive Schwarz Reformulation of the Domain Decomposition Algorithm

A very important observation for the subsequent analysis is that the preconditioner $\mathcal{B}(\cdot,\cdot)$ can be viewed as an additive subspace correction method (cf. [BPX90] and [Xu92]) with judiciously chosen subspaces. Let the linear operator $\tilde{\mathcal{E}} : S_h^0(\Omega) \mapsto S_h^0(\Omega)$ be defined by

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

In the above definition, $\bar{V}_k$ is a constant function with support in the closed subdomain Ω_k.

Furthermore, define

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

and

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

Thus $\hat{S}_h^0(\Omega)$ 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

$$\mathcal{B}_0(Y_0, \phi) = (g, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \tag{17}$$

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

$$\mathcal{B}_\Gamma(Y_\Gamma, \phi) = (g, \phi), \text{ for all } \phi \in S_\Gamma(\Omega). \tag{18}$$

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 (12) is equivalent to the additive Schwarz method above when

$$\mathcal{B}_0(\varphi, \phi) = \sum_{k=1}^{n_d} \mathcal{B}_k(\varphi, \phi) \tag{19}$$

and

$$\mathcal{B}_\Gamma(\varphi, \phi) = h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle \varphi - \bar{\varphi}_k, \phi - \bar{\phi}_k \rangle_{\partial\Omega_k, h}. \tag{20}$$

Let W be the solution of (8). Then

$$\mathcal{B}(W, \varphi) = \mathcal{B}_k(W^{(k)}, \varphi) = (g, \varphi)_\Omega, \quad \text{for all} \quad \varphi \in S_h^0(\Omega_k), \tag{21}$$

where $W^{(k)} \equiv W - \bar{W}_k - \mathcal{E}_k(W - \bar{W}_k)$. The function Y_0 satisfying (17) is given by

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

The form given by (20) depends only on the boundary values of φ and ϕ. Also, the function Y_Γ solving (18) equals the solution W on Γ. From the definition of $\tilde{\mathcal{E}}$,

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

Thus, the solution W of (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 (19) and (20).

5 Alternative Inexact Additive Preconditioners

We now consider a classical technique for developing nonoverlapping 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 [B̈89], [HLM91]. 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 [B̈89], [HLM91].

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., [BPS86a],[BPS87],[BPS88], [BPS89]) 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. Inverting $\hat{\mathbf{A}}$ is 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

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \tilde{\mathbf{Z}} \end{pmatrix}. \tag{22}$$

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 5.1).

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

$$\begin{aligned} \hat{S}_h(\Gamma) = \Big\{ & \mathcal{E}\varphi + \varphi_0 \mid \varphi \in S_h^0(\Omega); \\ & \mathcal{B}_0(\varphi_0, \phi) = -\mathcal{A}(\mathcal{E}\varphi, \phi), \text{ for all } \phi \in \hat{S}_h^0(\Omega) \Big\}. \end{aligned} \tag{23}$$

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 $\mathbf{B}_{22}$. $\mathcal{B}_\Gamma(u,v)$ depends only on the boundary nodal values of u and v and thus naturally extends to $S_h^0(\Omega)\times S_h^0(\Omega)$.

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.

The preconditioner defined by (22) 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 it corresponds to the preconditioning operator defined in the following algorithm.

Algorithm 5.1 *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*
$$\mathcal{B}_0(U_0,\varphi)=(g,\varphi)\quad \textit{for all}\quad \varphi\in\hat{S}_h^0(\Omega). \tag{24}$$
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. It is easy to see that the problem solved in Step 2 of Algorithm 5.1 is independent of U_0. Indeed, for any $\phi\in\hat{S}_h(\Gamma)$, we decompose $\phi=\mathcal{E}\phi+\phi_0$ as in (23) and observe
$$-\mathcal{A}(\mathcal{E}\phi,U_0)=\mathcal{B}_0(\phi_0,U_0)=(g,\phi_0).$$
Thus, Steps 2 and 3 of the above algorithm reduce to finding $U_\Gamma\in\hat{S}_h(\Gamma)$ such that
$$\mathcal{B}_\Gamma(U_\Gamma,\phi)=(g,\phi)\quad\text{for all}\quad \phi\in\hat{S}_h(\Gamma). \tag{25}$$
Hence, $\mathbf{B}^{-1}g=U=U_0+U_\Gamma$ where U_0 and U_Γ satisfy (24) and (25) respectively, i.e., Algorithm 5.1 is an implementation of an additive Schwarz procedure.

Notice that Algorithm 5.1 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 [BPS86a], [BPS87], [BPS88], and [BPS89].

It follows that the preconditioning form $\mathcal{B}(\cdot,\cdot)$ corresponding to the operator defined in Algorithm 5.1 is given by
$$\mathcal{B}(V,V)=\mathcal{B}_0(V_0,V_0)+\mathcal{B}_\Gamma(V_\Gamma,V_\Gamma). \tag{26}$$

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 provide bounds for (26). 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 (with C_k/c_k in (11) bounded independently of h, k, and d). The first theorem in this section was given by Börgers [B$\ddot{8}$9] and Haase at al. [HLM91] 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

$$\inf_{\phi \in \hat{S}_h^0(\Omega)} A(u + \phi, u + \phi), \quad \text{for all} \quad u \in \hat{S}_h(\Gamma). \tag{27}$$

Theorem 5.1 *Let $\mathcal{A}(\cdot, \cdot)$ be given by (3) and $\mathcal{B}(\cdot, \cdot)$ by (26). Assume that the quadratic form $\mathcal{B}_\Gamma(\cdot, \cdot)$ is uniformly equivalent to the quadratic from induced by (27). In addition, let γ be the smallest positive constant such that*

$$|\mathcal{A}(\varphi, \varphi) - \mathcal{B}(\varphi, \varphi)| \le \gamma \mathcal{A}(\varphi, \varphi) \quad \textit{for all} \quad \varphi \in \hat{S}_h^0(\Omega). \tag{28}$$

Then

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

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

Remark 5.1 *Condition (28) requires that $\mathcal{B}_0(\cdot, \cdot)$ should be a good approximation to $\mathcal{A}(\cdot, \cdot)$ for the preconditioner (26) to be efficient. The result of the theorem shows that if (28) 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 (28) 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. However, 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. [B$\ddot{8}$9]) 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

$$\mathcal{B}_\Gamma(u, v) = h^{-1} \sum_{k=1}^{n_d} \tilde{a}_k \langle u - \bar{u}_k, v - \bar{v}_k \rangle_{\partial\Omega_k, h}, \quad \text{for all} \quad u, v \in \hat{S}_h(\Gamma). \tag{29}$$

Theorem 5.2 *Let $\mathcal{A}(\cdot, \cdot)$ be given by (3), $\mathcal{B}(\cdot, \cdot)$ be given by (26), and $\mathcal{B}_\Gamma(\cdot, \cdot)$ defined by (29). Then*

$$c\mathcal{A}(U, U) \le \mathcal{B}(U, U) \le C\frac{d}{h}\mathcal{A}(U, U) \tag{30}$$

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

Remark 5.2 *The result of Theorem 5.2 shows that introducing inexact solves in the interior of the subdomains does not deteriorate the overall preconditioning effect of the corresponding exact method analyzed in [BPS87]. As we have pointed out in Remark 3.1, the adverse effect of h approaching zero on the condition number can be compensated for easily by adjusting the parameter d. This balance is an alternative to* (28) *and could be a better choice when h is small relative to γ. In fact, the utilization of the bilinear form* (29) *leads to computationally efficient algorithms, unconstrained by accuracy conditions like* (28). *The differences in the preconditioning effect of the inexact (Algorithm 5.1) and exact (cf. [BPS87]) methods are negligible. However, the savings of computational time are significant in favor of Algorithm 5.1.*

REFERENCES

[B$\ddot{8}$9] Börgers C. (1989) The Neumann–Dirichlet domain decomposition method with inexact solvers on the subdomains. *Numer. Math.* 55: 132–136.

[BPS86a] Bramble J., Pasciak J., and Schatz A. (1986) The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.* 47: 103–134.

[BPS86b] Bramble J., Pasciak J., and Schatz A. (1986) An iterative method for elliptic problems on regions partitioned into substructures. *Math. Comp.* 46: 361–369.

[BPS87] Bramble J., Pasciak J., and Schatz A. (1987) The construction of preconditioners for elliptic problems by substructuring, II. *Math. Comp.* 49: 1–16.

[BPS88] Bramble J., Pasciak J., and Schatz A. (1988) The construction of preconditioners for elliptic problems by substructuring, III. *Math. Comp.* 51: 415–430.

[BPS89] Bramble J., Pasciak J., and Schatz A. (1989) The construction of preconditioners for elliptic problems by substructuring, IV. *Math. Comp.* 53: 1–24.

[BPWX91] Bramble J., Pasciak J., Wang J., and Xu J. (1991) Convergence estimates for product iterative methods with applications to domain decomposition. *Math. Comp.* 57: 1–21.

[BPX90] Bramble J., Pasciak J., and Xu J. (1990) Parallel multilevel preconditioners. *Math. Comp.* 55: 1–22.

[BPX91] Bramble J., Pasciak J., and Xu J. (1991) A multilevel preconditioner for domain decomposition boundary systems. In *Proceedings of the 10'th International Conference on Computational Methods in Applied Sciences and Engineering.* Nova Sciences, New York.

[Bra93] Bramble J. (1993) *Multigrid Methods*, volume 294 of *Pitman Research Notes in Mathematics Series.* Longman Scientific & Technical, London.

[BW86] Bjørstad P. E. and Widlund O. (1986) Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* 23: 1097–1120.

[BX91] Bramble J. and Xu J. (1991) Some estimates for weighted L^2 projections. *Math. Comp.* 56: 463–476.

[CMW95] Cowsar L., Mandel J., and Wheeler M. (1995) Balancing domain decomposition for mixed finite elements. *Math. Comp.* 64: 989–1015.

[Dry82] Dryja M. (1982) A capacitance matrix method for the Dirichlet problem on a plygonal region. *Numer. Math.* 39: 51–64.

[Dry88] Dryja M. (1988) A method of domain decomposition for three-dimensional finite element elliptic problems. In Glowinski R., Golub G., Meurant G., and Périaux J. (eds) *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 43–61. SIAM, Philadelphia, PA.

[DSW94] Dryja M., Smith B., and Widlund O. (1994) Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.* 31(6): 1662–1694.

[DW91] Dryja M. and Widlund O. (1991) Additive Schwarz methods for elliptic finite element problems in three dimensions. Technical Report 570, Courant Institute of Mathematical Sciences, New York, NY.

[DW94] Dryja M. and Widlund O. (1994) Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comp.* 15: 604–620.

[GW87] Gonzalez R. and Wheeler M. (1987) Domain decomposition for elliptic partial differential equations with neumann boundary conditions. *Parallel Comput.* 5: 257–263.

[HLM91] Haase G., Langer U., and Meyer A. (1991) The approximate Dirichlet domain decomposition method. Part I: An algebraic approach. *Computing* 47: 137–151.

[Nep91] Nepomnyaschikh S. (1991) Application of domain decomposition to elliptic problems with discontinuous coefficients. In Glowinski R., Kuznetzov Y., Meurant G., and Périaux J. (eds) *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 242–251. SIAM, Philadelphia, PA.

[BDV96] Bjørstad P. E., Dryja M., and Vainikko E. (1996) Additive schwarz methods without subdomain overlap and with new coarse spaces. In Glowinski R., Périaux J., Shi Z., and Widlund O. (eds) *Domain Decomposition Methods in Science and Engineering*.

[Smi90] Smith B. (1990) *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*. PhD thesis, Courant Institute of Mathematical Sciences, New York, NY.

[Xu92] Xu J. (1992) Iterative methods by space decomposition and subspace correction. *SIAM Review* 34: 581–613.