PARALLEL DOMAIN DECOMPOSITION AND ITERATIVE REFINEMENT ALGORITHMS*

PETTER E. BJØRSTAD†

Abstract. Algorithms for the solution of partial differential equations based on a subdivision of the spatial domain, has received much interest in recent years. To a large extent this has been motivated by the new generation of parallel computers. This algorithmic approach can introduce independent parallel tasks of variable granularity, depending on the subdivision and can therefore be adapted to a wide range of parallel computers. We review some of the progress that has been made and supply a few new numerical examples that illustrate the convergence behavior.

Key Words. Schwarz' Method, Iterative Substructuring Methods, Domain Decomposition, Elliptic Equations, Finite Elements, Iterative Refinement, Parallel computing.

1. Introduction. Recently there has been a strong revival of the interest in domain decomposition algorithms for elliptic problems; cf. e.g. Glowinski et al, [17], and Chan et al, [10]. This is to a large extent due to their potential on multiprocessor computing systems. It is therefore not surprising that much of the recent research effort has focused on the case of many subregions; cf. e.g. Bramble et al [8, 9, 5, 7, 6], Dryja [13, 12], Dryja et al [14, 15] and Widlund [29].

The general technique of domain splitting is not new. H. A. Schwarz introduced such a technique in a theoretical context more than hundred years ago [27]. Engineers, in particular within the field of structural analysis, have used the principle extensively, partly to ease the task of modeling complex structures, partly in order to organize the large computational tasks on computers with limited memory [2, 26]. The subdomain solution technique has traditionally been Gaussian elimination and the global solution has been obtained by completing what can be viewed as a block Gaussian elimination process.

We assume that there exists an algorithm for solving the problem on a given subdomain, our main task is to study the process of obtaining the global solution by way of an iterative process. This process typically involves the solution of each subproblem one or two times in every iteration. The iteration should ideally converge in a fixed, very small number of steps, independent of the number of subdomains and of the grid size employed in each subdomain. If the complexity of the subdomain solver is more than linear in the number of unknowns, then the overall complexity is reduced as we "divide and conquer" the domain into more subdomains. The potential gain is even larger if the individual subdomains can be processed in parallel.

In order to achieve fast convergence, the resulting overall algorithm can often be viewed as an accelerated iterative method combined with a good preconditioning. Pre-

* To be presented at Algorithms'89

† Institutt for Informatikk, University of Bergen, Allégaten 55, N-5007 Bergen, Norway. This work was supported in part by the Norwegian Research Council for Science and the Humanities under grant D.01.08.054 and by The Royal Norwegian Council for Scientific and Industrial Research under grant IT2.28.28484.

conditioners often take the form of approximate solvers. Many important approximate solvers can be viewed as solving subproblems that can be formulated as

Find $u_i \in V_i$ such that

$$a(u_i, v_i) = f(v_i) \quad , \quad \forall\, v_i \in V_i, \tag{1}$$

for a number of subspaces $V_i \subset V^h$ such that

$$V^h = V_1 + V_2 + \ \ldots \ + V_n.$$

In general, $V_i \cap V_j \neq \{0\}$. An approximate solution $u_h \in V^h$ to

$$a(u_h, v_h) = f(v_h) \quad , \quad \forall\, v_h \in V^h \tag{2}$$

is then taken to be $\tilde{u} \approx \sum_{i=1}^{n} u_i$. We notice that u_i in 1 is nothing but the orthogonal projection of $u_h \in V^h$ on the subspace V_i, ie. $u_i = P_{V_i} u_h$. This approach corresponds to what we shall call additive algorithms, the approximate solution can be computed in parallel, and the individual contributions need only be added together at the end of every major iteration.

The additive algorithm can be thought of as a block Jacobi type method. If on the other hand, the subdomains are traversed and the information obtained in one subproblem is used as data when solving later subproblems, the resulting method may be viewed as a Gauss Seidel type algorithm. For reasons that will become clear later, this approach is also referred to as a multiplicative algorithm.

Domain decomposition algorithms are sometimes also classified according to whether overlapping subregions are used or not. This distinction often makes sense. Thus, if overlapping is used, certain variables are associated with several subproblems and the cost per iteration normally grows with the overlap. A decision how to balance this effect against an increased rate of convergence has to be made, while for a method which does not use overlapping, no such consideration arises. This distinction does not prevent us from using many of the same technical tools in the analysis, in fact, in [4] it is shown how to relate the classical Schwarz method to some of the iterative substructure methods where no overlap is present.

It should be observed that grid refinement algorithms like the ones introduced by [18, 23, 24] also can be described using this framework. In this context it is more natural to interpret the algorithms as making use of overlapping or direct sums of subspaces.

We emphasize that both domain decomposition algorithms and iterative grid refinement methods may be incorporated into existent large scale industrial codes without the need for a complete restructure of the code. In this setting, the subproblem solver is typically the original code itself. The introduction of subproblems can reduce the complexity of the geometric representation or make the data structure better suited to the given computer configuration. In the case of refinement, one should again be able to employ the original solver restricted to the refined patch, this approach is quite different from the elegant adaptive refinement strategy found in the code PLTMG by

Bank [1], the cost is the introduction of more unknowns than strictly necessary, the potential saving lies in the possibly simpler data structure, ease in vectorization and the potential use of fast, specialized solvers. These methods are perhaps most attractive precisely in the context of nontrivial industrial applications.

2. Preliminaries and notation in a finite element setting. To simplify our presentation, we assume that the elliptic operator is the Laplacian and that we have a zero Dirichlet condition. Thus,

$$\begin{aligned} -\Delta u &= f \qquad in\ \Omega, \\ u &= 0 \qquad on\ \partial\Omega. \end{aligned} \tag{3}$$

The region Ω is bounded, two- or three-dimensional, with a Lipschitz continuous boundary. We use a variational formulation of the problem and write equation 3 as

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx = f(v) \ , \qquad \forall\, v \in H_0^1(\Omega) \ ,$$

where the solution $u \in H_0^1(\Omega)$, the closure in the Sobolev space $H^1(\Omega)$ of the space of smooth functions which vanish in a neighborhood of $\partial\Omega$. In our analysis we work exclusively with the inner product defined by $a(\cdot,\cdot)$. Thus in this paper orthogonality and symmetry always refer to this inner product.

The problem is discretized by finite elements in the customary fashion; cf. Ciarlet [11]. The region Ω is triangulated and a conforming finite element space $V^h \subset H_0^1(\Omega)$ is introduced. We assume that all the elements are shape regular, i.e. there is a bound on h_K/ρ_K which is independent of the number of degrees of freedom and of K. Here h_K is the diameter of the element K and ρ_K the diameter of the largest sphere that can be inscribed in K. The approximate solution $u_h \in V^h$ is defined by

$$a(u_h, v_h) = f(v_h) \ , \qquad \forall\, v_h \in V^h. \tag{4}$$

In domain decomposition algorithms we subdivide Ω into N subdomains $\Omega^i, i = 1, ...N$, we say that Ω^i overlap with Ω^j if their intersection is nonempty. Associated with each subdomain Ω^i we define the space $V_i = V^h \cap H_0^1(\Omega^i)$.

In the analysis of grid refinement algorithms we consider a nested set of domains $\Omega^i \subset \Omega^j$ $i > j$, where we assume that Ω^i has been refined relative to the triangulation of Ω^j. We again associate a space $V_i = V^h \cap H_0^1(\Omega^i)$ with each Ω^i. It is most useful to consider the situation in terms of hierarchical basis functions, see Yserentant [32]. Such a representation provides a direct sum of subspaces V_i (ie. no overlap), while the nodal representation gives rise to subspace overlap corresponding to the coarse mesh functions also being represented in the nodal basis of the refined mesh.

3. Multiplicative Algorithms. We return briefly to the continuous case and write down the Schwarz algorithm in its traditional form. There are N fractional steps. Let u^n be the n-th iterate. Then the updated solution u^{n+1} is determined by the successive solution of N subproblems for $i = 1, 2, \ .. \ N$

$$\begin{aligned} -\Delta u^{n+i/N} &= f && \text{in } \Omega^i, \\ u^{n+i/N} &= 0 && \text{on } \partial\Omega^i \cap \partial\Omega, \\ u^{n+i/N} &= u^{n+(i-1)/N} && \text{on } \partial\Omega \setminus \partial\Omega^i. \end{aligned}$$

Note that the solution can be defined globally after each fractional step, by extending the the solution on Ω^i to the entire domain by using the most recently computed values. Following P. L. Lions [19, 20], we transform these equations into variational form. This formulation is equally valid in the discrete case. The exact solution of equation 4 is denoted by u_h.

$$a(u_h^{n+i/N} - u_h^{n+(i-1)/N}, v_h) = a(u_h - u_h^{n+(i-1)/N}, v_h)\ , \qquad \forall\ v_h \in V^h \cap H_0^1(\Omega^i)$$

Here $u_h^{n+i/N} - u_h^{n+(i-1)/N} \in H_0^1(\Omega^i)$ since the boundary values are constant from one fractional step to the next. Following Lions these equations can be restated in terms of orthogonal projections.

$$u_h^{n+i/N} - u_h^{n+(i-1)/N} \quad = \quad P_i(u_h - u_h^{n+(i-1)/N})\ ,$$

where P_i , $i = 1, 2 \ldots N$, is the orthogonal projection of V^h into $V_i = V^h \cap H_0^1(\Omega^i)$.

If we denote the contribution from the subdomain Ω^i, (by subassembly) to the global stiffness matrix K by $K^{(i)}$, and permute the unknowns appropriately, then we see that the projection P_i corresponds to the mapping

$$y = P_i x = \begin{pmatrix} {K^{(i)}}^{-1} & 0 \\ 0 & 0 \end{pmatrix} Kx.$$

The error $e_h^n = u_h^n - u_h$ satisfies the relationship

$$e_h^{n+1} = \prod_{i=0}^{N-1} P_{N-i}^{\perp} e_h^n \tag{5}$$

where $P_j^{\perp} = I - P_j$ is the orthogonal complement of P_j. In a certain sense Schwarz' method is therefore a straightforward iterative method of solving the equation

$$(I - \prod_{i=0}^{N-1} P_{N-i}^{\perp}) u_h = g_m \tag{6}$$

for a certain right hand side g_m. The operator is not symmetric, but it can be made symmetric by introducing more partial steps traversing the subdomains in reverse order back to Ω^1. This symmetric version can be accelerated, for example with the conjugate gradient method.

In the case of iterative refinement, the FAC method [22, 23, 24] can be interpreted as a Schwarz method applied to the subspaces V_i with overlap $V_i \cap V^h$. Let $u_h \in V^h$ be the current approximation and assume we have k levels of refinement.

The FAC algorithm can be stated as follows:

For i=1,2,... k

Compute the solution of

$$u_i \in V_i(\Omega^i): \quad a(u_h + u_i, v_i) = f(v_i) \ \ \forall\, v_i \in V_i(\Omega^i)$$

and set $u_h \leftarrow u_h + u_i$.

Again, one can write similar expressions to 5 and 6 when introducing the appropriate projection operators. The form of these equations, being polynomials in the projections, motivate the term multiplicative algorithms. These algorithms are not perfect for parallelization since the polynomial terms of degree higher than one implies sequential operations on subdomains. It should be noted that the polynomial degree will be quite low in many practical situations, because only terms corresponding to nontrivial overlap situations will enter. Thus, even the multiplicative algorithm may have a considerable number of independent tasks suitable for a multiprocessor system.

4. Additive algorithms. A different motivation for the additive algorithms can be obtained from the discussion of multiplicative methods, by just retaining the linear terms in the polynomial. The resulting linear system will now have the form:

$$(\sum_{i=1}^{N} P_i) u_h = g_a \tag{7}$$

The right hand side g_a can be computed by observing that

$$a(P_i u_h, v_h) = f(v_h) \ \ \forall\, v_h \in V_i(\Omega^i).$$

Therefore, at the expense of solving one subproblem in each domain, the appropriate right hand side can be found.

In the case of iterative refinement algorithms, the non overlapping spaces $V_i(\Omega^i)$ corresponding to a hierarchical representation will be discussed in the next section. Other choices are possible, introducing some degree of overlap, see [25] for details.

5. Convergence. The analysis of these methods is completely understood in the case of two domains. In [3] the relationship between additive and multiplicative Schwarz is fully analyzed and it is shown that the spectra of the iteration operators are related in a very precise way. An eigenvalue of the form $1 - \mu_i^2$ in the multiplicative case corresponds to two eigenvalues of the form $1 \pm \mu_i$ in the additive case. The resulting condition number for the additive method is therefore bounded by 4 times the condition number of the multiplicative method. This is a somewhat pessimistic result since it indicates that we may need twice as many iterations using the additive method. A similar result is given in [25] for the relationship between FAC and AFAC. Some further insights in the two domain Schwarz algorithm can be found in [4].

The multilevel case has been addressed in [16, 21, 23, 24, 31]. These results all show that under reasonable assumptions, the iterative refinement methods are indeed optimal in the sense that they converge in a number of iterations that is independent of both

the number of refinement levels and of the mesh spacing. A more detailed investigation of the two and three level case together with careful numerical experiments in the general case, is also given in [25]. That report indicates that the spectra of the iteration operators behave surprisingly similar also in the multilevel case. The eigenvalues for the additive method are only slightly perturbed from the form $1 \pm \mu$ and approach this form as the grid size parameter h tend to zero. Table 1 displays the extreme eigenvalues for a five level FAC and AFAC test example. The domain is a unit square and we refine a quarter of the square repeatedly towards one corner. Each square has a uniform, piecewise linear triangulation with 7, 15 and 31 interior nodes in each coordinate direction. We note that the eigenvalues behave very similar to the two level theory outlined above. We observe that the condition number of the iteration matrix never exceeds 2.5 in this example. We have never observed μ_1 exceed a half, but this does clearly depend on the ratio of refined to unrefined area.

m =		7	15	31
AFAC	λ_1	1.434421	1.436717	1.439223
	$\lambda_1 + \lambda_N$	2.004588	2.000063	2.000005
FAC	μ_1	0.428042	0.436631	0.439218

Table 1, A 5-level iterative grid refinement

In the case of Schwarz' method applied to the multi domain case, a recent report by Widlund, [30] derives a recursive bound for the condition number of the resulting symmetrized iterative method of the form:

$$\kappa(\Omega^1, \cdots \Omega^k) \leq \kappa(\Omega^1, \Omega^2 \cup \Omega^3 \cup \cdots \Omega^k) \cdots \kappa(\Omega^{k-1}, \Omega^k) \tag{8}$$

Again, this multiplicative bound may appear pessimistic, but a simple argument shows that as long as no mechanism for global exchange of information is present, the condition number must grow at least as fast as the square of the number of subregions that must be traversed in order to connect any two domains. Numerical experiments carried out in a three domain model case [28] indicate that the bound 8 is indeed fairly sharp in most cases. The condition number is only slightly better than the predicted upper bound when the overlapping regions are quite small. We have in our experiment, never seen the estimate more than 20 percent off. As the overlap is increased, the bound becomes very sharp, often within one percent of the bound.

6. Acknowledgments. The author appreciates the assistance provided by Randi Moe and Morten Skogen.

REFERENCES

[1] R. E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0*, SIAM, Philadelphia, PA, 1990.

[2] K. Bell, B. Hatlestad, O. E. Hansteen, and P. O. Araldsen, *NORSAM, a programming system for the finite element method. Users manual, Part 1, General description.*, NTH, Trondheim, 1973.

[3] P. E. Bjørstad, *Multiplicative and Additive Schwarz Methods: Convergence in the 2 domain case*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 147–159.

[4] P. E. Bjørstad and O. B. Widlund, *To overlap or not to overlap: A note on a domain decomposition method for elliptic problems*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 1053–1061.

[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), pp. 103–134.

[6] ——, *An iterative method for elliptic problems on regions partitioned into substructures*, Math. Comp., 46 (1986), pp. 361–369.

[7] ——, *The construction of preconditioners for elliptic problems by substructuring, II*, Math. Comp., 49 (1987), pp. 1–16.

[8] ——, *The construction of preconditioners for elliptic problems by substructuring, III*, Math. Comp., 51 (1988), pp. 415–430.

[9] ——, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp., 53 (1989), pp. 1–24.

[10] T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., *Domain Decomposition Methods*, Philadelphia, PA, 1989, SIAM. Proceedings of the Second International Symposium on Domain Decomposition Methods, Los Angeles, California, January 14 - 16, 1988.

[11] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.

[12] M. Dryja, *A method of domain decomposition for 3-D finite element problems*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 43–61.

[13] ——, *An additive Schwarz algorithm for two- and three-dimensional finite element elliptic problems*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 168–172.

[14] M. Dryja, W. Proskurowski, and O. Widlund, *A method of domain decomposition with crosspoints for elliptic finite element problems*, in Optimal Algorithms, B. Sendov, ed., Sofia, Bulgaria, 1986, Bulgarian Academy of Sciences, pp. 97–111.

[15] M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.

[16] ——, *Some domain decomposition algorithms for elliptic problems*, in Iterative Methods for Large Linear Systems, L. Hayes and D. Kincaid, eds., San Diego, California, 1989, Academic Press, pp. 273–291. Proceeding of the Conference on Iterative Methods for Large Linear Systems held in Austin, Texas, October 19 - 21, 1988, to celebrate the sixty-fifth birthday of David M. Young, Jr.

[17] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., *Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988, SIAM. Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 1987.

[18] L. Hart and S. McCormick, *Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: Computational analysis*, Parallel Computing, 12 (1989), pp. 131–144.

[19] P. L. Lions, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.

[20] ——, *On the Schwarz alternating method. II.*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 47–70.

[21] J. Mandel and S. McCormick, *Iterative solution of elliptic equations with refinement: The model multi-level case*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM.

[22] ——, *Iterative solution of elliptic equations with refinement: The two-level case*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM.

[23] S. McCormick, *Fast adaptive composite grid (FAC) methods*, in Defect Correction Methods: Theory and Applications, K. Böhmer and H. J. Stetter, eds., Computing Supplementum 5, Springer-Verlag, Wien, 1984, pp. 115–121.

[24] S. McCormick and J. Thomas, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., 46 (1986), pp. 439–456.

[25] R. Moe, *Iterative Local Uniform Mesh Refinement Methods and Parallel Processing*, PhD thesis, Department of Informatics, University of Bergen, Høyteknologisenteret, N-5020 Bergen, Norway, June 1992.

[26] J. S. Przemieniecki, *Matrix structural analysis of substructures.*, Am. Inst. Aero. Astro. J., 1 (1963), pp. 138–147.

[27] H. A. Schwarz, *Gesammelte Mathematische Abhandlungen*, vol. 2, Springer, Berlin, 1890, pp. 133–143. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.

[28] M. D. Skogen, *Schwarz Methods and Parallelism*, PhD thesis, Department of Informatics, University of Bergen, Norway, February 1992.

[29] O. B. Widlund, *Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM.

[30] ——, *On the rate of convergence of the classical Schwarz alternating method in the case of more than two subregions*, tech. report, Department of Computer Science, Courant Institute, 1989. A revised version in preparation.

[31] ——, *Optimal iterative refinement methods*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM.

[32] H. Yserentant, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.