CHAPTER 9

Domain Decomposition Methods for Problems with Uniform Local Refinement in Two Dimensions*

James H. Bramble†
Richard E. Ewing‡
Rossen R. Parashkevov‡
Joseph E. Pasciak§

Abstract. In this talk, we first present a flexible mesh refinement strategy for the approximation of solutions of elliptic boundary value problems in two dimensional domains. Coupled with this approximation scheme, we shall describe preconditioners for the resulting discrete system of algebraic equations. These techniques lead to efficient computational procedures in serial as well as parallel computing environments. The preconditioners are based on overlapping domain decomposition and involve solving (or preconditioning) subproblems on regular subregions. These techniques are analyzed in a forthcoming paper [2]. We present the results of numerical experiments illustrating the preconditioning algorithms.

INTRODUCTION

To provide the required accuracy in many applications involving large scale scientific computation, it becomes necessary to use local mesh refinement techniques. These techniques allow the use of finer meshes in regions of the computational domain where the solution exhibits large gradients. This remains practical only if efficient techniques for the solution of the resulting discrete systems are available. In this talk, we will give a flexible scheme for refinement as well as develop effective iterative methods for the solution of the resulting systems of discrete equations. The analysis for the methods discussed in this talk is given in [2].

* 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. DMS84-05352 and by the U.S. Army Research Office through the Mathematical Science Institute, Cornell University. Additional supporters of this work include the Office of Naval Research under contract No. 0014-88-K-0370 and by the Institute for Scientific Computation at the University of Wyoming though National Science Foundation grant No. RII-8610680.

†Department of Mathematics, Cornell University, Ithaca, NY 14853.
‡Mathematics Department, The University of Wyoming, Laramie, WY 82071.
§Department of Applied Science, Brookhaven National Laboratory, Upton, NY 11973.

We shall be interested in techniques for problems with refinements which are not quite local. As an example, one might consider a front passing through a two dimensional domain. In this case, it might be necessary to refine in the neighborhood of the front.

There are a number of ways of developing preconditioned iterative schemes for the discrete systems resulting from local mesh refinement in the literature. Techniques based on nested multilevel spaces are given in [1],[10],[11]. Techniques based on domain decomposition are given in [3],[14],[15]. The analysis presented there implicitly depends on the shape of the the refinement domain, and hence the resulting algorithms may not be as effective with irregularly shaped refinement regions. These algorithms also require the solution of a subproblem or preconditioner on the refinement regions. This talk will provide alternative preconditioned iterative techniques for these problems based on overlapping domain decomposition. Our algorithms are simpler and possibly more effective when implemented since they often lead to preconditioning subproblems defined on either regular subregions or topologically 'nice' meshes. The refinement region is the union of the subregions.

The proposed mesh refinement strategy is important in that it provides a basic approach for implementing dynamic local grid refinement. An example of a refinement strategy involves starting with a uniform coarse-grid and refining in small subregions associated with a selected set of coarse-grid vertices. These subregions are allowed to overlap and there are no theoretical restrictions on the resulting refinement region (the union of the subregions). Dynamic refinement is achieved by simply dynamically changing the selected set of coarse-grid vertices.

In addition, the technique can be integrated into existing large scale simulators without a complete redesign of the code. This is because most of the computation involves tasks on either the global coarse grid or the refinement grids associated with the refinement subregions. Choosing the coarse and refinement grid structure to be that already used in the code saves considerable development costs. For example, if one uses regularly structured meshes in the coarse and refinement grids, a substantial part of the resulting algorithm only requires operations on regular grids even though the resulting final approximation space is not regular.

The outline of the remainder of the talk is as follows. In Section 2, we define some preliminaries and describe the second-order elliptic problems which will be considered. The overlapping domain decomposition algorithms for grids with partial refinement is defined in Section 3. The theoretical estimates for the resulting preconditioned systems (from [2]) are also given there. Finally, computational aspects and the results of numerical experiments using these preconditioning techniques are discussed in Section 4.

2. THE ELLIPTIC PROBLEM AND PRELIMINARIES

We shall be concerned with the efficient solution of discrete equations resulting from approximation of second-order elliptic boundary value problems in a polygonal domain Ω contained in two dimensional Euclidean space R^2. We consider the problem of approximating the solution u of

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

Here L is given by

$$Lv = -\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} a_{ij} \frac{\partial v}{\partial x_j},$$

and $\{a_{ij}(x)\}$ is a uniformly positive definite, bounded, piecewise smooth coefficient matrix on Ω. The corresponding bilinear form is denoted by $A(\cdot,\cdot)$ and is given by

$$A(v,w) = \sum_{i,j=1}^{2} \int_{\Omega} a_{ij} \frac{\partial v}{\partial x_i} \frac{\partial w}{\partial x_j} \, dx, \tag{2.2}$$

and is defined for functions $v,w \in H^1(\Omega)$. Here $H^1(\Omega)$ is the Sobolev space of order one on Ω. We denote the $L^2(\Omega)$ inner product by $(\cdot,\cdot)$. The weak solution u of (2.1) is the function $u \in H_0^1(\Omega)$ satisfying

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

Here, $H_0^1(\Omega)$ is the subspace of functions in $H^1(\Omega)$ whose traces vanish on $\partial\Omega$.

We consider the above model problem for convenience. Many extensions of the techniques to be presented are possible; for example, one could consider equations with lower-order terms and different boundary conditions.

In this talk, we shall deal with various domains. These domains will always be open.

3. The overlapping algorithms

In this section, we shall define iterative methods for problems with partial refinement based on overlapping domain decomposition. We start with a coarse mesh $\cup\tau_H^i$ consisting of triangles of quasi-uniform size H. The associated finite element space M_0 is defined to be the set of continuous piecewise linear functions on the coarse mesh which vanish on $\partial\Omega$. The interior nodes of this mesh will be denoted $\{x_i\}$, for $i = 1,\ldots,N_c$. The mesh refinement is defined in terms of a number of coarse grid subdomains $\{\Omega_i\}$ for $i = 1,\ldots,K$. By convention, Ω_i is defined to be the interior of the union of the closures of the coarse grid triangles. The refinement regions will also be referred to as "the subdomains." We assume that they have limited overlap in that any point of Ω is contained in at most a fixed number (not depending on H) of the subdomains. We define the domain of refinement Ω^r to be the union of the subdomains, $\Omega^r = \cup_{i=1}^{K} \Omega_i$. There are no theoretical restrictions concerning the definition of the refinement subregions except that they are defined in terms of the coarse grid triangles and satisfy the overlap property as described above.

We provide two examples of this construction. For both examples, the subregions are associated with coarse grid nodes. For the first example, we define the region associated with a coarse-grid node x_i as the subdomain Ω_i which contains the coarse-grid triangles having x_i as a vertex. For the second example, we consider a mesh which is topologically equivalent to a regular rectangular mesh (see Figure 3.1). In this case, we define Ω_i to be the four quadrilaterals which share the vertex x_i. Some reasons for such a choice will be explained later. In either case, an index set $I \subseteq [1,\ldots,N_c]$ is selected and the domains $\{\Omega_i\}$ with $i \in I$ are used to define the refinement region. By possibly changing the numbering of the coarse grid nodes, we assume, without loss of generality, that $I = [1,2,\ldots,K]$. There are no additional restrictions concerning this set I and hence rather complex refinement regions are possible.

The composite space is defined in terms of a quasi-uniform mesh $\{\tau_h^i\}$ on Ω of size $h < H$ which satisfies

$$\cup_i \partial\tau_H^i \subseteq \cup_i \partial\tau_h^i.$$

The space of continuous piecewise linear functions with respect to this triangulation (which vanish on $\partial\Omega$) will be denoted by $\tilde{M}$. Note that this space is introduced for the construction and analysis of the composite grid space. It is not used in actual computation since it has too many degrees of freedom in Ω/Ω^r. The subspace M_i associated with the subdomain Ω_i is defined by

$$M_i = \{\phi \in \tilde{M} \mid \text{support } \phi \subseteq \Omega_i\}. \tag{3.1}$$

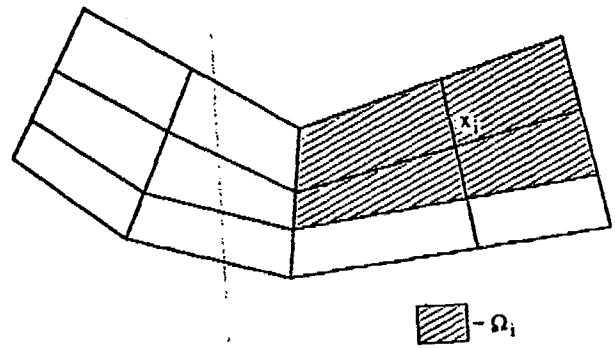

Figure 3.1
A distorted rectangular mesh.

The composite finite element space is then defined to be

$$M = \sum_{i=0}^{K} M_i.$$

Note that the space M provides finer grid approximation in the refinement region Ω^r. An illustrative example of a mesh so generated is given in Figure 3.2. The nodes on the boundary of the refinement region which are not coarse-grid nodes are slave nodes since, by continuity, the values of functions in M on these points are completely determined by their values on neighboring coarse-grid nodes. The operator $A_i : M_i \mapsto M_i$ is defined for $v \in M_i$ by

$$(A_i v, \phi) = A(v, \phi) \qquad \text{for all } \phi \in M_i.$$

Our goal is to efficiently solve the composite grid problem: Given a function $f \in L^2(\Omega)$, find $U \in M$ satisfying

$$A(U, \phi) = (f, \phi) \qquad \text{for all } \phi \in M. \tag{3.2}$$

As above, we define $A : M \mapsto M$ by

$$(Av, \phi) = A(v, \phi) \qquad \text{for all } \phi \in M.$$

Problem (3.2) can then be rewritten as

$$AU = F, \tag{3.3}$$

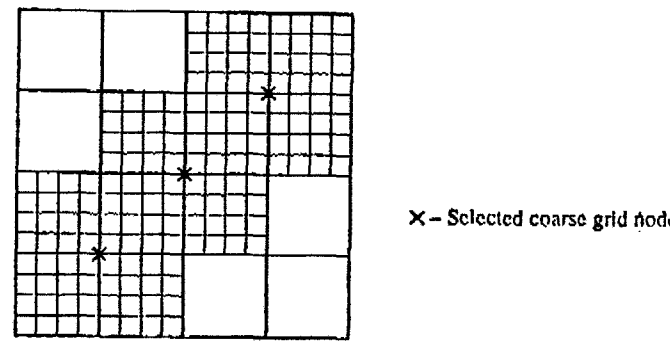

Figure 3.2
A composite grid.

for appropriate $F \in M$. We will develop preconditioners for (3.3) by using overlapping domain decomposition.

There are basically two classes of these preconditioners, the additive and multiplicative. The additive version defines the preconditioner B_a for A of (3.3) by

$$B_a = \sum_{i=0}^{K} R_i Q_i.$$

Here, Q_i denotes the $L^2(\Omega)$ projection operator onto M_i and R_i is a symmetric positive definite operator on M_i. Explicit choices for R_i will be discussed later; however, we note that it suffices to take R_i to be a preconditioner for A_i.

The multiplicative version is defined by applying the R_i consecutively. The multiplicative preconditioner B_m applied to a function $W \in M$ is defined as follows:

(1) Set $Y_0 = 0$.

(2) For $i = 1, \ldots, K+1$, define Y_i by

$$Y_i = Y_{i-1} + R_{i-1} Q_{i-1}(W - AY_{i-1}). \tag{3.4}$$

(3) For $i = K+2, \ldots, 2K+2$, define Y_i by

$$Y_i = Y_{i-1} + R_{2K+2-i} Q_{2K+2-i}(W - AY_{i-1}). \tag{3.5}$$

(4) Set $B_m W = Y_{2K+2}$.

It is not difficult to see that B_m is a symmetric linear operator on M.

The operators B_a and B_m defined above will be effective as preconditioners A if they satisfy the following:

(1) They are relatively inexpensive to evaluate.

(2) They lead to well conditioned linear systems.

The first criterion involves implementation issues and will be discussed later in more detail. The second criterion requires that the condition numbers $K(B_a A)$ and $K(B_m A)$ be small. In the case of the additive algorithms, this is equivalent to the existence of positive constants c_0, c_1 satisfying

$$c_0 A(v,v) \leq A(B_a A v, v) \leq c_1 A(v,v) \qquad \text{for all } v \in M, \tag{3.6}$$

with c_1/c_0 small. A similar statement holds for the product algorithm.

The analysis presented in [2] requires the following hypotheses. It is assumed that there are positive constants C_0 and ω which do not depend on h, H or the subdomains and satisfy

$$C_0 A(w,w) \leq A(R_i A_i w, w) \leq \omega A(w,w) \qquad \text{for all } w \in M_i. \tag{3.7}$$

This means that the operators R_i are spectrally good preconditioners for A_i. For the product algorithm, we also assume that $0 < \omega < 2$. The following theorem is proved in [2].

THEOREM 3.1. *Assume that there are no isolated points on the boundary of Ω^r. Then the condition numbers $K(B_a A)$ and $K(B_m A)$ remain bounded independently of h, H and the choice of subdomains $\{\Omega_i\}$.*

REMARK 3.1: The analysis given in [2] uses techniques from both the theory of overlapping domain decomposition [12],[13] as well as the standard domain decomposition theory [5]-[8] to provide the result for the additive algorithms. The result for the multiplicative version follows from that for the additive and the application of a general theory given in [9].

REMARK 3.2: The hypothesis concerning isolated points on the boundary of Ω^r is included to provide a uniform estimate for the preconditioned systems. If $\partial\Omega^r$ contains isolated points then it is possible to show (cf. Remark 4.2 of [2]) that the condition number grows at most on the order of $\ln^2(H/h)$. This sort of decay is actually seen in the last numerical example in Section 6.

REMARK 3.3: There is very little restriction concerning the way that the domains Ω_i are defined. Note that if only one refinement domain is used, then Theorem 3.1 provides a result for the imbedded space case proposed in [3]. Alternatively, one can consider the case where Ω^r is all of Ω and hence $M = \tilde{M}$. In this case, Theorem 3.1 guarantees uniform bounds for the condition numbers without putting restrictions on the shapes of the subdomains $\{\Omega_i\}$. Thus, for example, the subdomains can be taken to be strips as long as the coarse problem is included.

4. COMPUTATIONAL ASPECTS AND NUMERICAL EXAMPLES

In this section, we discuss some of the computational properties associated with the method. In particular, we shall consider its feasibility for use in dynamic refinement strategies. We shall also see that with this type of method, it is possible to develop highly vectorizable and parallelizable code. Finally, we provide the results of numerical examples illustrating the condition numbers for the preconditioned systems described earlier.

We consider the earlier discussed examples where the domain of refinement is defined by simply selecting coarse-grid nodes and a rule for defining the refinement region associated with a coarse node. Specifically, we consider the example where the coarse mesh is defined from quadrilaterals and the refinement region associated with a coarse-grid vertex is defined to be the four quadrilaterals which share the vertex. An easy way to implement this refinement involves using vectors of unknowns with some redundancy. Associated with each quadrilateral, we associate a vector which contains the fine-grid unknowns in the quadrilateral and its boundary. The program is designed to operate on a data structure which contains a coarse-grid vector and a list of fine-grid vectors corresponding to the quadrilaterals appearing in the refinement regions. This process is controlled by a list of pointers which connect the location of quadrilateral fine-grid vectors in this data structure

to the coarse grid node refinement regions in which they appear. A simple control structure is also developed to handle the redundancy in the data vectors. These control structures can be easily derived from the list of coarse-grid refinement nodes and the coarse mesh geometry. Thus, a dynamic change in the refinement region only requires a simple (and of negligible cost) computation of some control pointers associated with the coarse grid.

An advantage of the proposed approach is that it can be used to invoke refinement without the use of the general data structures associated with meshes which are not regular. One assigns a regular mesh topology to the coarse mesh and to the meshes in the refinement subregions. This means that even though the composite mesh is highly irregular, all of the problems (on M_i, $i \in I_0$) which need to be solved or preconditioned will be on regular rectangular meshes. Similarly, it is possible to decompose the evaluation of the composite grid operator into pieces which involve operator evaluation on the topologically rectangular mesh parts. For these topologically rectangular meshes, highly efficient modules for preconditioning and operator evaluation are available for both vector and parallel computing architectures.

We shall consider the model problem

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{4.1}$$

where Δ denote the Laplacian and Ω is the unit square $[0,1] \times [0,1]$. To define the coarse mesh, the domain Ω is first partitioned into $m \times m$ square subdomains of side length $H = 1/m$. Each smaller square is then divided into two triangles by one of the diagonals (e.g. the diagonal which goes from the bottom left to the upper right hand corner of the square). The coarse-grid approximation space M_0 is defined to be the set of functions which are continuous on Ω, are piecewise linear with respect to the triangulation, and vanish on $\partial\Omega$. The space $\tilde{M}$ is defined from a similar finer mesh of size $h = H/l$ for some integer $l > 1$.

For our first two examples, we consider an application where it is required to refine along the diagonal connecting the origin with the point $(1,1)$. Such a refinement might be necessary if the function f has large gradients near this diagonal but is well behaved in the remainder of Ω. Accordingly, we select the coarse-grid nodes on the diagonal for refinement. We define the refinement region associated with a refinement node to be the four coarse mesh squares which have that node as a corner. Note that the refinement region is highly irregular even though the coarse problem and the refinement subproblems involve regular rectangular meshes.

We will illustrate the rate of convergence of preconditioned algorithms for solving (3.2) where $A(\cdot,\cdot)$ is given by the Dirichlet form. To do this, we shall numerically compute the largest and smallest eigenvalue (λ_1 and λ_0 respectively) of the preconditioned operator $B_a A$. As is well known, the rate of convergence of the resulting preconditioned algorithms can be bounded in terms of the condition number $K(B_a A) = \lambda_1/\lambda_0$. We shall not report results for preconditioning with the product operator B_m, although our previous experience [9] suggests that the product version will converge somewhat faster than the additive.

Table 4.1 gives the largest and smallest eigenvalue and the condition number of the system $B_a A$ as a function of h. In this example, we took $R_i = A_i^{-1}$; i.e., we solved exactly on the subspaces $\{M_i\}$. For Table 4.1, $m = 4$ and there are three refinement subdomains $(0,1/2) \times (0,1/2)$, $(1/4,3/4) \times (1/4,3/4)$, and $(1/2,1) \times (1/2,1)$. Note that both the upper and lower eigenvalues appear to be tending to a limit as the ratio $h/H \mapsto 0$. Similar behavior is seen in Table 4.2, which corresponds to $m = 8$ and uses seven smaller refinement subregions.

Table 4.1
Condition numbers for 3 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/8	2.44	0.50	4.9
1/16	2.50	0.41	6.1
1/32	2.51	0.38	6.6
1/64	2.52	0.36	6.9
1/128	2.52	0.35	7.1

In almost all realistic applications, the direct solution of subproblems is much more expensive than the evaluation of a suitable preconditioner. To illustrate the effect on the convergence rate of the preconditioned iteration, we next consider the previous example but with the direct solves on the subspaces replaced by multigrid preconditioners. Specifically, we employ the V-cycle multigrid algorithm (cf. [4]) using one pre- and post-smoothing Jacobi iteration on each grid level. This leads to a preconditioning operator $R_i : M_i \mapsto M_i$ which satisfies

$$0.4A(v,v) \leq A(R_i A_i v, v) \leq A(v,v) \qquad \text{for all } v \in M_i. \tag{4.2}$$

Table 4.2
Condition numbers for 7 overlapping subregions

h	λ_1	λ_0	$K(B_a A)$
1/16	2.46	0.47	5.2
1/32	2.52	0.39	6.5
1/64	2.54	0.35	7.2
1/128	2.54	0.34	7.5

The constant 0.4 above was computed numerically and holds for all of the subspace problems which are required for this application, including M_0.

Tables 4.3 and 4.4 provide the eigenvalues and condition numbers for the above examples when direct solves were replaced by multigrid preconditioners. Note that in all of the reported runs, the condition number with multigrid preconditioners was at most 5/4 times as large as that corresponding to exact solves. Such an increase in condition number is negligible in a preconditioned iteration. In contrast, the computational time required for the multigrid sweep is considerably less that that needed for a direct solve (especially in more general problems with variable coefficients).

Table 4.3
Preconditioned subproblems, 9 overlapping subregions

h	λ_1	λ_0	$K(B_aA)$
1/8	2.37	0.53	4.5
1/16	2.12	0.33	6.4
1/32	2.07	0.27	7.6
1/64	2.04	0.25	8.2
1/128	2.02	0.24	8.4

Table 4.4
Preconditioned subproblems, 7 overlapping subregions

h	λ_1	λ_0	$K(B_aA)$
1/16	2.36	0.40	5.9
1/32	2.11	0.28	7.5
1/64	2.06	0.24	8.8
1/128	2.03	0.22	9.4

As a final example, we consider a case where the isolated point hypothesis of Theorem 3.1 is not satisfied. Specifically, we consider a coarse mesh of size $H = 1/4$ and select the four nodes with (x, y) values (1/4,1/2), (3/4,1/2), (1/2,1/4), and (1/2,3/4). The refinement region is everything but the subsquares $[0, 1/4] \times [0, 1/4]$, $[0, 1/4] \times [3/4, 1]$, $[3/4, 1] \times [0, 1/4]$, and $[3/4, 1] \times [3/4, 1]$. Note that, to satisfy the hypotheses of the theorem, it would be necessary to include a refinement region centered at the coarse-grid node $(1/2, 1/2)$. Table 4.5 gives the smallest eigenvalue for the operator B_aA as a function of h. The function $(.32 + .36 \log_2(h^{-1}))^{-2}$ is also provided for comparison. These results suggest that smallest eigenvalue λ_0 decays as predicted by the theoretical bound $C/\ln(H/h)^2$ (see Remark 3.2).

Table 4.5
A "bad" example in two dimensions.

h	λ_0	$(.32 + .36 \log_2(h^{-1}))^{-2}$
1/8	.50	.51
1/16	.32	.32
1/32	.22	.22
1/64	.16	.16
1/128	.12	.12

REFERENCES

1. R.E. Bank, T. Dupont, and H. Yserantant, *The hierarchical basis multigrid method*, Num. Math. **52** (1988), 427–458.
2. J.H. Bramble, R.E. Ewing, R.R. Parashkevov, and J.E. Pasciak, *Domain decomposition methods for problems with partial refinement*, Proceedings of the Copper Mountain Meeting on Iterative Methods, April, 1990 (submitted).
3. J.H. Bramble, R.E. Ewing, J.E. Pasciak and A.H. Schatz, *A preconditioning technique for the efficient solution of problems with local grid refinement*, Comp. Meth. Appl. Mech. Eng. **67** (1988), 149–159.
4. J.H. Bramble and J.E. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comp. **49** (1987), 311–329.
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.
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.
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.
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.
9. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition and multigrid*, (preprint).
10. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu, *Multigrid results which do not depend upon elliptic regularity assumptions*, (in preparation).
11. J.H. Bramble, J.E. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., (in print).
12. M. Dryja and O. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Technical Report, Courant Institute of Mathematical Sciences **339** (1987).
13. P.L. Lions, *On the Schwarz alternating method*, In the Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G.H. Golub, G.A. Meurant, and J. Periaux (eds.), 1987.
14. S. McCormick and J. Thomas, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp. **46** (1986), 439–456.
15. O. Widlund, *Optimal iterative refinement methods*, Technical Report, Courant Institute of Mathematical Sciences **391** (1988).