UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Optimal Coarse Grid Size in Domain Decomposition

Tony F. Chan
Jian-Ping Shao

July 1993

CAM Report 93-24

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

OPTIMAL COARSE GRID SIZE IN DOMAIN DECOMPOSITION

TONY F. CHAN AND JIAN-PING SHAO *

Abstract. In most domain decomposition (DD) methods, a coarse grid solve is employed to provide the global coupling required to produce an *optimal* method. The total cost of a method can depend sensitively on the choice of the coarse grid size H. In this paper, we give a simple analysis of this phenomenon for a model elliptic problem and a variant of Smith's vertex space domain decomposition method [12, 4]. We derive the optimal value H_{opt} which asymptotically minimizes the total cost of method (number of floating point operations in the sequential case and execution time in the parallel case), for subdomain solvers with different complexities. Using the value of H_{opt}, we derive the overall complexity of the DD method, which can be significantly lower than that of the subdomain solver.

Key Words. domain decomposition, Fourier vertex space method, computation complexity, execution time, optimal coarse grid size.

AMS subject classifications: 65N20, 65F10.

1. Introduction .

Domain decomposition (DD) is a class of techniques for solving elliptic boundary value problems in which the solution is obtained by iteratively solving smaller subdomain problems. These methods have received a lot of study in recent years (see [7, 1, 2, 3, 10]). They are attractive because of their inherent parallelism and their *optimal* convergence rates (i.e. independent of the mesh size). The optimality of the convergence rate requires the solution of a coarse grid problem at each iteration. The study of how to incorporate such a coarse grid solve in a DD method has received a lot of study in the literature.

The focus of our paper, however, is on the choice of the *size* H of the coarse grid. It is intuitively obvious that the total cost of a DD method can depend sensitively on this choice, in addition to the obvious dependence on the efficiency of the subdomain solver. A small H generally improves the convergence rate (because the coarse grid problem is a better approximation to the original fine grid problem) at the cost of a more costly coarse grid solve, whereas a large H has the opposite effect. Therefore, an optimal value H_{opt} often exists and indeed has been observed empirically [9, 11]. Surprisingly, there has been almost no systematic study in the literature on this issue. Our approach is to take a simple model elliptic problem and a particular DD method; this allows a simple but complete and easily understood analysis which we think give insights for more general situations.

For concreteness, we focus our analysis on a variant of Smith's vertex space method [12] developed by us earlier [4]. We consider subdomain solvers with different complexity, including banded Gaussian elimination, nested dissection, modified incomplete Cholesky factorization (MIC) and multigrid solvers. For simplicity, we assume the same solver is used for the subdomains and the coarse grid problem, and that these are solved exactly. By expressing the computational complexity as a function of the coarse grid size H and the fine grid size h, we derive the optimal value H_{opt} which *asymptotically* (as h tends to 0) minimizes the total cost of method. Using the value of H_{opt}, we can derive the overall complexity of the DD method as a function of h alone, which can be significantly lower than that of the subdomain

* Department of Mathematics, University of California at Los Angeles, CA 90024. This work was supported in part by the Army Research Office under contract DAAL03-91-G-0150 and subcontract under DAAL03-91-C-0047, by the National Science Foundation under grant ASC 92-01266, and by the Office for Naval Research under contract ONR N00014-92-J-1890.

solver. That is, through the use of DD, a given solver can be made more efficient for solving the original problem, *by using it to solve smaller (but more) sub-problems.* This is a simple consequence of the divide-and-conquer principle. The assumption of the asymptotic limit is not necessary but does allow a close form expression for H_{opt} from which one can see more clearly the general trend.

2. Fourier vertex space domain decomposition method. We consider the following 2nd order self adjoint elliptic problem on $\Omega \subset R^2$:

$$-\nabla \cdot (a(x,y)\nabla u) = f \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega, \tag{1}$$

with $a(x,y) \in R^{2\times 2}$ uniformly positive definite, bounded and piecewise smooth on Ω. We assume that the domain Ω is partitioned into N non-overlapping sub-domains $\Omega_1, \cdots, \Omega_N$ of diameter H,which form the elements of a *quasi-uniform* coarse grid triangulation τ^H. We also assume that the sub-domains Ω_i are refined to produce a fine grid *quasi-uniform* triangulation τ^h having elements of diameter h. Corresponding to the coarse grid and fine grid triangulations, we discretize (1) either by using finite elements, see [6], or by using finite difference methods, see [13], resulting in the symmetric positive definite linear systems $A_h u_h = f_h$ on the fine grid and $A_H u_H = f_H$ on the coarse grid.

Let I denote the union of the interiors of the sub-domains, and let B denote the interface separating the sub-domains, i.e. $I = \cup_i \Omega_i, B \equiv (\cup_i \partial\Omega_i) - \partial\Omega$. According to this partitioning, A_h, x and f can be written in block form:

$$A_h = \begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix} \qquad x = \begin{pmatrix} x_I \\ x_B \end{pmatrix} \qquad f = \begin{pmatrix} f_I \\ f_B \end{pmatrix}.$$

After eliminating x_I, we obtain the reduced Schur complement problem on B :

$$Sx_B \equiv (A_{BB} - A_{BI}A_{II}^{-1}A_{IB})x_B = g \equiv f_B - A_{BI}A_{II}^{-1}f_I. \tag{2}$$

Since computing Sv is much cheaper than calculating and storing S itself, the main idea is now to solve this system by a preconditioned conjugate gradient method (PCG). The Fourier Vertex Space (FVS) method [4] is one such preconditioner.

To describe this, the interface B is further partitioned into a union of edges E_{ij} and cross points $\cup(x_k^H, y_k^H)$, i.e. $B = \cup_{ij} E_{ij} \cup (\cup_k (x_k^H, y_k^H))$, where E_{ij} denotes the edge separating sub-domains Ω_i and Ω_j . Let V_k denote the portion of B within a distance of βH from (x_k^H, y_k^H) for some positive fraction $0 < \beta < 1$ (the vertex region); for details see [12, 4]. We define $R_{E_{ij}}$ and R_{V_k} as the pointwise restrictions of nodal values to E_{ij} and V_k respectively, i.e. $R_{E_{ij}} g = g$ on E_{ij} and $R_{V_k} g = g$ on V_k. Let R_H^T represent linear interpolation from the coarse space V^H to the fine space V^h. We define $S_{E_{ij}} = R_{E_{ij}}^T S R_{E_{ij}}$ and $S_{V_k} = R_{V_k}^T S R_{V_k}$.

The Fourier vertex space preconditioner can now be defined by:

$$M_{FVS}^{-1} \equiv R_H^T A_H^{-1} R_H + \sum_{ij} R_{E_{ij}}^T (\tilde{S}_{E_{ij}})^{-1} R_{E_{ij}} + \sum_k R_{V_k}^T (\tilde{S}_{V_k})^{-1} R_{V_k}. \tag{3}$$

Here $\tilde{S}_{E_{ij}}$ is a standard interface preconditioner on the edge E_{ij}. In [4], a scaled version of the square root of the 1-D Laplacian operator is used and can be inverted efficiently by the Fast Fourier Transform (FFT). $\tilde{S}_{V_k}$ is a dense matrix approximating S_{V_k} and is inverted by a direct solver. It is proved in [12, 4] that M_{FVS} is an optimal preconditioner for S. In theory, the condition that $\beta = O(1)$ implies that the number of nodes of the vertex region has to increase with decreasing h. However, as shown in [12, 4], in practice the size of the vertex region can be chosen to be a small constant (independent of h) without affecting the optimal convergence rate. Therefore, the cost of the edge and vertex space part of the preconditioner is often negligible compared to the cost of the coarse grid solve A_H^{-1}.

3. Optimal Computational Complexity: Sequential Case. We now make the assumption that the cost of the FVS method is dominated by that of solving the subdomain problems (in inverting A_{II} in computing the matrix-vector product Su in PCG) and the coarse grid problem (in inverting A_H in the preconditioner). This is a reasonable assumption if h is small enough and H is neither too small or too large, so that the subdomains have a reasonably large "area-to-perimeter" ratio and the coarse grid is not too small.

Let the complexity of the solver used for both the sub-domain problems and the coarse problem be $O(m^p)$ for the preprocessing phase (e.g. factorization) and $O(m^s)$ for the solution phase on an $m \times m$ grid. For example, for banded Gaussian elimination, MIC and multigrid, $p = 4, 2, 2$ and $s = 3, 2.5, 2$ respectively. For nested dissection, $p = 3$ and the solution phase has complexity $O(m^2 \log m)$. We assume that the iteration number $\aleph$ is bounded and *independent* of the fine and coarse grid sizes, which is supported by the theoretical and numerical results in [12, 4]. For example, numerical experiments in [4] indicate that $\aleph$ is between 9 and 15 for a tolerance of 10^{-5} for a wide range of values for h and H and for widely different coefficients of the elliptic problem. It is then easy to see that the leading order terms of the operation count for the FVS/PCG method are given by:

$$\text{flops}(H) \approx \frac{C}{H^2}(\frac{H}{h})^p + \frac{C}{H^p} + \aleph\{\frac{C}{H^2}(\frac{H}{h})^s + \frac{C}{H^s}\}, \tag{4}$$

where C is a generic constant that depends on the particular solver. The first two terms are the preprocessing cost (e.g. factorization of A_{II} and A_H) and the last two terms are the cost during the PCG iteration. The leading order terms have the form:

$$\text{flops}(H) \approx \frac{C}{H^2}(\frac{H}{h})^\alpha + \frac{C}{H^\alpha}, \tag{5}$$

where $\alpha = \max\{p, s\}$ and the generic constant C may depend on $\aleph$ but is independent of H and h. In other words, the dominant cost consists of solving $1/H^2$ sub-domain problems and one coarse grid problem.

The optimal coarse grid size H_{opt} is obtained by setting the first derivative of function in (5) (with respect to H) to zero, giving:

$$H_{opt}(\alpha) = (\frac{\alpha}{\alpha - 2})^{\frac{1}{2\alpha-2}} h^{\frac{\alpha}{2\alpha-2}} \qquad \text{for } \alpha > 2. \tag{6}$$

Using this value of H_{opt}, we obtain for the asymptotic complexity:

$$\min_H \text{ flops}(H) \approx \text{ flops}(H_{opt}) \approx C\{(\frac{\alpha}{\alpha-2})^{\frac{\alpha-2}{2\alpha-2}} + (\frac{\alpha}{\alpha-2})^{\frac{-\alpha}{2\alpha-2}}\}h^{\frac{-\alpha^2}{2\alpha-2}}. \tag{7}$$

When $\alpha = 2$, i.e. an optimal solver such as a multigrid method,

$$\text{flops}(H) \approx \frac{C}{h^2} + \frac{C}{H^2}, \tag{8}$$

TABLE 1
The sequential complexity of solvers on an $n \times n$ grid; coarse grid size n_H.

Basic Solver	Complexity	Optimal n_H	Complexity of DD Solver using optimal n_H
Multigrid	$O(n^2)$	1	$O(n^2)$
MIC	$O(n^{2.5})$	$0.58n^{5/6}$	$O(n^{2.08})$
Nested Dissection	$O(n^3)$	$0.76n^{3/4}$	$O(n^{2.25})$
Band-Cholesky	$O(n^4)$	$0.89n^{2/3}$	$O(n^{2.67})$
	$O(n^\alpha), \alpha \to \infty$	$n^{1/2}$	$O(n^{\alpha/2})$

which indicates that H should be chosen as large as possible ($O(1)$ in our model problem.)

Note that H_{opt} is independent of the constant C (i.e. the solver). Clearly, H_{opt} depends non-monotonically on the complexity exponent α. For $\alpha = 2.5, 3, 4$, $H_{opt} = 5^{1/3}h^{5/6}, 3^{1/4}h^{3/4}, 2^{1/6}h^{2/3}$ respectively. As $\alpha \to \infty$, $H_{opt} \to h^{1/2}$.

The complexity of the FVS algorithm, using H_{opt}, is given by:

$$\text{flops}(H_{opt}) \approx O((\frac{1}{h})^{\gamma(\alpha)}),$$

where $\gamma(\alpha) = \frac{\alpha^2}{2\alpha-2}$. For $\alpha = 2, 2.5, 3, 4$, $\gamma = 2, 2.08, 2.25, 2.67$ respectively. As $\alpha \to \infty$, $\gamma(\alpha) \to \alpha/2$. Thus, using a domain decomposition approach results in a substantial reduction in the asymptotic complexity of the solver. The reduction is greater the higher the complexity of the solver is.

We summarize these complexity results in Table 1, where we present the results in terms of an $n \times n$ fine grid ($n = 1/h$) and an $n_H \times n_H$ coarse grid ($n_H = 1/H$).

4. Optimal Computational Complexity: Parallel Case. In the parallel case, the operation count model has to be replaced by a true timing model, taking into account both the arithmetic cost and the communication cost. However, in the spirit of the asymptotic analysis used in the last section, we can make some simplifying assumptions which allow us to extract useful information from our model. The most important assumption we shall make is that the communication cost is not dominant over the arithmetic cost, which is valid if the number of unknowns in the interior of a subdomain is not too small compared to those on the boundary (i.e. a small perimeter-to-area ratio) and is consistent with our assumption in Sec. 3. The full treatment with communication cost can be found in [5].

We shall also assume that there are enough processors so that the subdomain problems are solved completely in parallel. A crucial issue is how to solve the coarse grid problem in a parallel environment. According to Gropp [8], one of the best methods is to collect the necessary data on one processor, solve it there and then broadcast the result. Finally, we can do the coarse grid solve either (a) sequentially, after the subdomain solves, or (b) in parallel to the subdomain solves.

Making these assumptions, it is easy to see that the leading order terms of the parallel time of the FVS method is:

$$\text{time}(H) \approx \begin{cases} C(H/h)^\alpha + C(1/H^\alpha) & \text{case (a)} \\ \max\{C(H/h)^\alpha, C(1/H^\alpha)\} & \text{case (b),} \end{cases}$$

where C is a generic constant modeling the time per arithmetic operation. In both

TABLE 2
The sequential complexity of solvers on an $n \times n \times n$ grid; coarse grid size n_H.

Basic Solver	Complexity	Optimal n_H	Complexity of DD Solver using optimal n_H
Multigrid	$O(n^3)$	1	$O(n^3)$
MIC	$O(n^{3.5})$	$0.61n^{7/8}$	$O(n^{3.06})$
Nested Dissection	$O(n^6)$	$0.93n^{2/3}$	$O(n^4)$
Band-Cholesky	$O(n^7)$	$0.95n^{7/11}$	$O(n^{4.45})$
	$O(n^\alpha), \alpha \to \infty$	$n^{1/2}$	$O(n^{\alpha/2})$

cases, the optimal value of H can be easily seen to be:

$$H_{opt} = \sqrt{h},$$

independent of α (i.e. the solver). We note that this optimal choice of H_{opt} implies that *the size of each subdomain problem is equal to the size of the coarse problem.* It also implies that the optimal number of processors is n $(= (1/\sqrt{h})^2)$.

The parallel time of the FVS method using H_{opt} is:

$$\text{time}(H_{opt}) = O(n^{\alpha/2}),$$

and the speed-up is:

$$\text{Speed-up} = O(n^\alpha)/O(n^{\alpha/2}) = O(n^{\alpha/2}).$$

Note that the speed up is greater than $O(n)$ (the number of processors) if $\alpha > 2$. This "superlinear" speed-up is possible because we are not parallelizing a "fixed" algorithm — the FVS algorithm with the optimal coarse grid has different sequential complexity for different n.

5. Higher Dimensional Problems. A similar analysis can also be extended to a d-dimensional problem. For a solver of complexity $O(m^\alpha)$ on an m^d grid, the results in the sequential case are:

$$H_{opt} = (\frac{\alpha}{\alpha - d})^{\frac{1}{\alpha - d}} h^{\frac{\alpha}{2\alpha - d}}, \qquad \text{flops}(H_{opt}) = O(h^{-\frac{\alpha^2}{2\alpha - d}}).$$

The $d = 3$ case is summarized in Table 2. The results in Sec. 4 for the parallel case are independent of d, except that the optimal number of processors is $n^{d/2}$.

6. Concluding Remarks. The results obtained above should also apply to other *optimal* domain decomposition methods, such as other substructuring methods and overlapping Schwarz methods. The optimal coarse grid size is obtained as a simple balance between the cost of the subdomain solves and the cost of the coarse grid solve. Therefore, the conclusions are valid for any DD method, as long as these costs dominate the overall cost and the convergence rate is independent of H and h.

REFERENCES

[1] T. CHAN, R. GLOWINSKI, , J. PÉRIAUX, AND O. WIDLUND, eds., *Domain Decomposition Methods*, Philadelphia, 1989, SIAM. Proceedings of the Second International Symposium on Domain Decomposition Methods , Los Angeles, California , January 14 - 16, 1988.

[2] ———, eds., *Domain Decomposition Methods*, Philadelphia, 1990, SIAM. Proceedings of the Third International Symposium on Domain Decomposition Methods , Hoston, Texas, 1989.

[3] ———, eds., *Domain Decomposition Methods*, Philadelphia, 1991, SIAM. Proceedings of the Fourth International Symposium on Domain Decomposition Methods , Moscow, USSR, 1990.

[4] T. F. Chan, T. P. Mathew, and J. P. Shao, *Efficient variants of the vertex space domain decomposition algorithm*, Tech. Rep. CAM 92-07, Department of Mathematics, UCLA, 1992. To appear in SIAM J. Sci. Comp.

[5] T. F. Chan and J. Shao, *The optimal parallel complexity of domain decomposition*, tech. rep., 1993. In preparation.

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

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

[8] W. D. Gropp, *Parallel computing and domain decomposition*, in Domain Decomposition Methods, T. Chan, D. E. Keyes, G. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, 1992, SIAM.

[9] W. D. Gropp and D. E. Keyes, *Domain decomposition with local mesh refinement*, Tech. Rep. RR-726, Yale University, Dept. of Comp. Sci., August 1989. Accepted for SIAM J. Sci. Stat. Comp.

[10] D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and R. G. Voigt, eds., *Domain Decomposition Methods*, Philadelphia, 1992, SIAM. Proceedings of the Fifth International Symposium on Domain Decomposition Methods, Norfolk, VA, 1991.

[11] B. F. Smith, *Domain decomposition algorithms for partial differential equations of linear elasticity*, Tech. Rep. 517, Department of Computer Science, Courant Institute, 1990 Sept. Ph.D thesis.

[12] ———, *An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 364–378.

[13] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, 1962.