LA-UR

97-4181

Approved for public release;
distribution is unlimited

Title:

Two Examples of the Impact of Partitioning with
Chaco and Metis on the Convergence of Additive
Schwarz-Preconditioned FGMRES

Author(s):

Mike DeLong

RECEIVED

FFR 0 2 1998

OSTI

Submitted to:

For External Distribution

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

Los Alamos
National Laboratory

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. The Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

Two examples of the impact of partitioning with Chaco and Metis on the convergence of Additive Schwarz-preconditioned FGMRES

Mike DeLong*

September 29, 1997

Abstract

Algebraic graph partitioning packages that require no geometric or physical information are often used to partition sparse linear systems for solution on parallel computers using Krylov-space iterative methods with domain decomposition-type preconditioning methods. We show through a couple of examples that this can have a substantial impact on the convergence of the iterative methods.

1 Introduction

In the context of developing solvers for Accelerated Strategic Computing Initiative (ASCI) applications, we need to be able to read problems from a single monolithic file extracted from an ASCI code and partition them for solution on a parallel computer. We will be using Krylov-space iterative solvers with domain decomposition-type preconditioners to solve these problems. An overview of domain-decomposition methods can be found in the review article by Chan and Mathews [1] or the book by Smith et. al. [11]. There are several packages available that can be used to partition matrices for solution in this way, using only algebraic information (i.e. the structure and entries of the matrix), as opposed to geometric or physical information from the engineering problem from which the linear system is derived. Two popular packages are Chaco, written by Hendrickson and Leland [6] and Metis, written by Karypis and Kumar [7]. In this paper we will give results showing that these packages should be used with great care, as different algorithms can have a substantial impact on the rate of convergence of the preconditioned iterative method.

2 Two test problems

We used two test problems for our experiments. One is a simple convection-diffusion problem which is easily understood. The other is from a

*Group CIC-19, Los Alamos National Labs, Los Alamos, NM 87545

collection of matrices taken from an ASCI radiation transport package written at Los Alamos National Labs (LANL).

2.1 STRONGX

Our first test problem arises from a finite-difference discretization of a constant-coefficient convection-diffusion problem

$$\nabla^2 u - \sigma u_x - \tau u_y = f, \quad \sigma \gg \tau \quad (1)$$

on the interior of the unit square with Dirichlet boundary conditions ($u = 0$ on the boundary). By holding τ fixed and small and taking σ large it is possible to make the problem strongly convected in the x direction. This problem is adapted from one having the same name, found in the paper by D'Azevedo et. al. [3]. There the authors show that the convergence of an iterative method is highly dependent on the ordering of the unknowns, when the iterative method is preconditioned using a (global) incomplete LU (ILU) factorization: the rate of the decay of the entries in the full LU factors is highly dependent on the ordering of the unknowns, and in turn the quality of the ILU factors. We use uniform grid spacing with $h = 1/101$ and use centered differences for the first-derivative terms. This gives a coefficient matrix with 10,000 equations and 49,600 nonzeros.

2.2 Kershaw 60×60

The second test problem is from Augustus, a code package developed by Mike Hall at LANL. Augustus solves the P_1 equation set

$$\alpha_{td} \frac{\partial \Phi}{\partial t} + \vec{\nabla} \cdot \vec{F} + \sigma_r \Phi = S \quad (2)$$

$$\vec{F} = -D \vec{\nabla} \Phi + \vec{J}_{src} \quad (3)$$

over an arbitrary domain using a cell-based finite-element discretization scheme. Our test problem uses a Kershaw mesh, which discretizes a square domain using nonuniform grid spacing. It has 60 cells in each direction, and the coefficient matrix has 10,561 equations and 56,257 nonzeros. The equations from the cell centers are ordered first, followed by the equations from the cell faces. This results in a matrix with a profile similar to the one shown in Figure 1 for a 19×19 problem. Unfortunately we do not at present have a way to visualize the grid for this problem, so we do not have any pictures of its partitionings. This particular test problem is available on the World Wide Web¹.

3 Additive Schwarz preconditioning

Additive Schwarz (AS) preconditioning is essentially block Jacobi preconditioning, with or without overlapping blocks, with each processor holding one block. The preconditioning is done by solving a local problem, and in

¹See <http://www.c3.lanl.gov/~delong/augustus/>.

Figure 1: Profile of the matrix for a 19×19 Kershaw mesh

our case we solve this local problem using ILUT(10^{-4} , 25)-GMRES(30) with a convergence criterion for the local residual of $\|\mathbf{r}_{k,p}\|_2 \leq 10^{-3}$.

For our experiments we use AS code from the P-SPARSLIB code developed by Saad and Lo (see [9], [10]). It supports AS with subdomains overlapped one equation (by adding the neighbors of perimeter nodes to the local subdomain for the purpose of preconditioning) or with no overlap. We used one equation of overlap for our experiments. The P-SPARSLIB code native partitioning subroutines implement recursive dissection partitioning based on distributed site expansion (see e.g. [5]). However, simple modifications make it possible to use any partitioning, and we have done this in order to be able to use Chaco and Metis to do the partitioning offline.

4 Chaco and Metis

As was mentioned above, the Chaco and Metis graph partitioning packages are described in full in [6] and [7], but we will review the salient points in this section. Both packages take as input a description of an undirected graph $G = (V, E)$ with a vertex $v \in V$ for every equation and an edge $e \in E$ for every nonzero element of A . There may also be a weighting function for the edges, vertices, or both. Both packages produce an integer vector representing a partitioning of G that is balanced in some sense and minimizes some objective function. If the graph is unweighted then the partitioning should have roughly the same number of vertices in each partition and minimize the number of cut-edges: that is, the number of edges e_k connecting two vertices v_i and v_j such that v_i and v_j are in different partitions of V . In general this problem is NP-complete [4], so both packages use a number of heuristics, primarily multilevel methods, to give an approximate solution in polynomial time. Further, if the edges of G are weighted then both packages attempt to produce a balanced partition so that the sum of the weights of the cut-edges is minimized. In addition to a multilevel Kernighan-Lin (*Multilevel-KL*) partitioning method, Chaco includes several simple algorithms. One of these is the “linear” partitioning method that simply divides the lexicographical ordering of the equations into chunks of size $\lceil n/p \rceil$ and assigning one to each of p processors. We

will also use this method in our experiments. Metis includes two multi-level methods: a “k-way” partitioning method which we will call *kmetis* and a recursive bisection method we will call *pmetis*, after the standard way in which they are invoked.

5 Results for unweighted graphs

In this section we give experimental results from runs on an SGI Origin 2000 for small numbers of processors for the test problems given above: STRONGX and Kershaw 60×60 . Both of our test problems are structurally symmetric, so to extract we simply take $G = G(A)$, with an edge for each nonzero entry in A .

Figure 2: STRONGX partitioned into four subdomains

Figures 2(a) and (b) show two partitionings of the grid for the problem STRONGX for four processors: Figure 2(a) gives the partitioning found using *kmetis* and Figure 2(b) gives the partitioning for the Chaco linear partitioning discussed above. The two partitionings are roughly equivalent in terms of load balancing, in that each subdomain contains roughly $n/4$ gridpoints, but the *kmetis* partitioning clearly has smaller edge-cut: for the blue subdomain, for example, the edge-cut for the linear partitioning is roughly twice that of the blue subdomain for the *kmetis* partitioning. This is important for minimizing the communication cost for, say, a matrix-vector multiply, since each cut edge represents a value that must be communicated across a subdomain boundary.

Table 1 gives iteration counts required for FGMRES(20) (see [8]) to satisfy a global convergence criterion of $\|r_k\|_2 \leq 10^{-6}$ using AS preconditioning for the two partitionings shown in Figure 2 and an analogous partitioning for eight processors, as a function of the convection term coefficient σ . Clearly, as STRONGX becomes more strongly convected in the x direction the convergence of AS-FGMRES degrades for the *kmetis* partitioning while it improves (slightly) for the linear partitioning. The cause is intuitively clear: if M is the preconditioning matrix, then the residual

σ	4 processors		8 processors	
	kmetis	linear	kmetis	linear
300	14	12	15	13
1,000	15	8	20	8
3,000	26	6	43	6
10,000	54	5	254	6
30,000	196	4	400+	5
100,000	400+	4	-	5

Table 1: Iteration counts for AS-FGMRES(20)

matrix $R = A - M$ contains more entries that are $O(\sigma)$ than $O(\tau)$ if the nonzeros associated with the cut edges represent more connections in the x directions than in the y direction and vice versa. The linear partitioning cuts more edges in the y direction than in the x direction – in fact, in the case pictured in Figure 2(b) no edges are cut in the x direction. The kmetis partitioning, on the other hand, cuts edges in the x and y directions in roughly equal amounts. As a result as σ increases the ratio of the norm of the residual matrix to the coefficient matrix $\|R\|/\|A\|$ increases for the kmetis partitioning and decreases for the linear partitioning.

Table 2 gives iteration counts for AS-FGMRES for four different partitionings, two from Chaco and two from Metis. As was stated above, the initial ordering of the unknowns (cell centers first, then cell faces) is an entirely “natural” ordering, so we call this ordering the “given” ordering, to distinguish it from the row-oriented natural ordering. We also give results for this problem after reordering the unknowns using the Reverse Cuthill-McKee (RCM) ordering [2], a level-set ordering, which we used on the global problem in an attempt to give the partitioner some clues about the geometry of the problem.

Method	2 processors		4 processors	
	Given	RCM	Given	RCM
Chaco Linear	*	76	*	110
Chaco Multilevel-KL	86	81	97	100
kmetis	27	88	84	111
pmetis	70	92	94	85

Table 2: Iteration counts for Kershaw 60×60

The results here are pretty poor for all cases, with the possible exception of the two-processor case partitioned with kmetis². In a couple of cases preordering with RCM helped slightly, but in some cases it was harmful. The fact that one case converged in 27 iterations while the others took 70 or more suggests that (almost?) all of the partitionings shown are bad for this problem.

²For the starred cases (*) the P-SPARSLIB setup code crashed.

6 Results for weighted graphs

Chaco and Metis will also partition graphs with weights on the edges, vertices, or both. In this section we give results for the two test problems with weighted edges, where the goal is minimize the “size” of the residual matrix $R = A - M$ in some sense by keeping the partitioning software from breaking connections that have large coefficients.

We need an undirected weighted graph, so we take the weighting function to be $w_{ij} = (|A| + |A^T|)_{ij}$. That is, the weight of the edge connecting vertices i and j is the entry of the matrix $|A| + |A^T|$. This gives a symmetric matrix (and thus an undirected graph), while preserving the magnitude of the connection between nodes i and j . For STRONGX for the case $\sigma = 30,000$ this gives weights of 990 in the x direction and 2 in the y direction.

(a) kmetis

(b) pmetis

(c) M-KL, Bisection

(d) Quadrissection

Figure 3: STRONGX partitioned using weighted edges

Figure 3 gives four partitioning of the STRONGX grid using graphs with weighted edges: kmetis, pmetis, and two Chaco multilevel-KL (M-KL) partitionings: one using bisection at the coarsest level, the other

using four-way partitioning (Quadrissection)³. Note that three of the four cases give at least two subdomains that are virtually indistinguishable from those of the linear partitioning of Figure 2. However, none of the cases exactly replicate the linear partitioning, and as a result we would expect that the convergence results would not be identical to those in Table 1.

Method	STRONGX	Kershaw 60 × 60
M-KL Bisection	5	98
M-KL Quadrissection	63	82
K-Metis	400+	82
P-Metis	263	78

Table 3: Results for weighted graphs

Table 3 gives results for STRONGX with $\sigma = 30,000$ and Kershaw 60×60 using weighted graphs with the weighting function described above for four processors. Note the wide variety of results for STRONGX: the best case takes only one more iteration than the linear case (Table 1), but the other three cases take more iterations than the unweighted kmetis case (also in Table 1). For the Kershaw 60×60 problem the results the best case is slightly better than the kmetis unweighted case (78 iterations vs. 84), but the worst case is worse than the pmetis unweighted (98 iterations vs. 94). On average it is not clear that anything has been gained by using weighted edges for the Kershaw problem.

7 Conclusions

The predictive power of a study of this size is limited, but the following items seem clear:

1. Edge-cut (communication volume) is not the most important characteristic of a distributed problem to be considered when partitioning for use with AS preconditioners.
2. When using AS preconditioning it is important to use a partitioner that will preserve strong connections and thus keep the residual matrix small in norm.
3. Partitioning using only an unweighted graph information is insufficient to preserve these strong connections.
4. Using the magnitudes of the entries of the coefficient matrix A to weight the edges of the graph is not sufficient to make every partitioner yield an acceptable partitioning.

Finally, as a result, it is clear that in the context of AS preconditioning, algebraic partitioning packages such as Chaco or Metis should be used with great care.

³In both cases there were 100 vertices at the coarsest level.

References

- [1] T. Chan and T. Mathews. Domain decomposition algorithms. *Acta Numerica*, pages 61–143, 1994.
- [2] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings 24th National Conference of the Association of Computing Machinery*, pages 157–172, New Jersey, 1969. Brandon Press.
- [3] E. D’Azevedo, P. Forsythe, and W. Tang. Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems. *SIAM J. Matrix Anal. Appl.*, 13(3):944–961, 1992.
- [4] M. Garey, D. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
- [5] T. Goehring and Y. Saad. Heuristic algorithms for automatic graph partitioning. Technical Report UMSI 94-29, University of Minnesota Supercomputer Institute, Minneapolis, MN, 1994.
- [6] B. Hendrickson and R. Leland. The chaco user’s guide version 2.0. Technical Report SAND95-2344, Sandia National Laboratories, 1995.
- [7] G. Karypis and V. Kumar. Metis: Unstructured graph partitioning and sparse matrix ordering. Technical report, University of Minnesota, 1995.
- [8] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Stat. Comp.*, 14:461–469, 1993.
- [9] Y. Saad and G. Lo. PPARSLIB users manual: A portable library of parallel sparse linear solvers. Technical Report UMSI 95-180, University of Minnesota, 1996.
- [10] Y. Saad and A. Malevsky. PPARSLIB: A portable library of distributed memory sparse iterative solvers. In V. E. Malyshev et al., editor, *Proceedings of Parallel Computing Technologies (PaCT-95), 3-rd international conference, St. Petersburg, Russia, Sept. 1995*. Springer Verlag, Lecture Notes in Computer Science series, 1995.
- [11] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition*. Cambridge University Press, 1996.