Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, April 1996.

Parallel Iterative Solvers and Preconditioners Using Approximate Hierarchical Methods (An Extended Abstract) *

Ananth Grama, Vipin Kumar, Ahmed Sameh
Department of Computer Science,
University of Minnesota
Minneapolis, MN 55455
{ananth, kumar, sameh}@cs.umn.edu

Abstract

In this paper, we report results of the performance, convergence, and accuracy of a parallel GMRES solver for Boundary Element Methods. The solver uses a hierarchical approximate matrix-vector product based on a hybrid Barnes-Hut / Fast Multipole Method. We study the impact of various accuracy parameters on the convergence and show that with minimal loss in accuracy, our solver yields significant speedups. We demonstrate the excellent parallel efficiency and scalability of our solver. The combined speedups from approximation and parallelism represent an improvement of several orders in solution time. We also develop fast and paralellizable preconditioners for this problem. We report on the performance of an inner-outer scheme and a preconditioner based on truncated Green's function. Experimental results on a 256 processor Cray T3D are presented.

1 Introduction

Boundary Element Methods (BEM) are often used to solve integral equations using potential theory. These methods discretize the boundary of the domain into panels. Using the associated Green's function, the potential at each panel is represented as a sum of contributions of every other panel. Applying the Dirichlet boundary condition yields a large scale linear system of equations.

For an n panel boundary discretization, the $n \times n$ linear system arising from this approach is dense. Solving the system using direct factorization methods has a computational

*This work was supported by IST/BMDO through Army Research Office contract DA/DAAH04-95-1-0538, and by Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to computing facilities were provided by Minnesota Supercomputer Institute, Cray Research Inc. and by the Pittsburgh Supercomputing Center. Related papers are available via WWW at URL: http://www.cs.umn.edu/ kumar.

complexity of $\Theta(n^3)$. This limits the values of n to a few thousands. Therefore iterative methods are used to solve these systems. The complexity of the underlying matrix-vector product (mat-vec) is $\Theta(n^2)$. For large values of n, this becomes extremely expensive. The physical properties of the underlying domain allow us to use approximate hierarchical techniques that can compute this mat-vec in $\Theta(n)$ or $\Theta(n \log n)$ time.

Parallel formulations of hierarchical methods have been explored in the context of particle dynamics for regular distributions [1, 12, 6, 8, 5] and irregular distributions [3, 4, 2, 9, 11, 10]. These techniques have been explored to solve integral equations by Nabors et al. [7]. In [3], we presented a highly scalable parallel formulation of a dense matrix-vector product for unstructured and adaptive discretizations based on hierarchical methods. Here, we present the performance and accuracy of a GMRES solver based on this parallel mat-vec. We also present techniques for preconditioning these systems and study their performance.

2 Parallel Formulation of Hierarchical Matrix-Vector Product

Hierarchical methods rely on aggregating the effect of distant panels into a series. The combined effect of these panels is then applied using the series rather than individual interactions. Algorithms such as Barnes-Hut and Fast Multipole FMM) are traditionally used for such computations. We use a hybrid of the Barnes-Hut and FMM codes for computing mat-vecs. The hierarchical representation of the domain is identical to FMM. The interactions are computed using the multipole acceptance criteria of the Barnes-Hut method.

Parallel formulations comprise two major steps: tree construction (the hierarchical representation of the domain) and tree traversal. Starting from a distribution of the panels to processors, each processor constructs its local tree. Processors communicate the top nodes in the tree to form a globally consistent image of the tree.

Each processor now proceeds to compute the potential at the panels assigned to it by traversing the tree. On encountering a node that is not locally available, there are two possible scenarios: the panel coordinates can be communicated to the remote processor that evaluates the interaction; or the node can be communicated to the requesting processor. We refer to the former as function shipping and the latter as data shipping. Our parallel formulations are based on the function shipping paradigm. We discuss the advantages of function shipping in [3]. Table 1 presents the runtimes, efficiency and computation rates for up to 266 processors of a Cray T3D for a variety of highly unstructured problems.

Our parallel formulation of the hierarchical mat-vec yields excellent performance for a wide range of processors. The speedup scales almost linearly up to 256 processors even for very small problems (4elt.xyz and g_28060). For the range of problem sizes where hierarchical methods are really useful (pscan and g_108196), the formulation yields a relative speedup of over 3.5 on going from 64 to 256 processors. Clearly, for these problem instances, the overhead due to communication and load imbalance is low. The per-processor performance of the code is about 20 MFLOPS which is respectable considering the code is highly unstructured in nature.

Problem	$p = 64$			$p = 256$		
	Runtime	Eff.	MFLOPS	Runtime	Eff.	MFLOPS
4elt.xyz	0.44	0.84	1220	0.15	0.61	3545
pscan	3.74	0.93	1352	1.00	0.87	5056
g_28060	0.53	0.89	1293	0.16	0.75	4357
g_108196	2.14	0.85	1235	0.61	0.75	4358

Table 1: Runtimes (in seconds), efficiency, and computation rates of the T3D for different problems for $p = 64$ and 256.

3 GMRES Based on Hierarchical Matrix-Vector Product

We implement a parallel restart-GMRES around the hierarchical matrix-vector product. The accuracy of the mat-vec is impacted by a number of parameters such as the degree of the multipole expansion and the multipole acceptance criteria. We study the impact of inaccuracies in the matrix-vector product on the convergence of the solver. We demonstrate that it is possible to get accurate convergence with significant savings in computation time using hierarchical methods. Table 2 presents the Log of relative residual norm for GMRES with various degrees of approximation executed on a 64 processor T3D. The parallel runtime indicates that hierarchical methods are capable of yielding significant savings in time at the expense of slight loss of accuracy.

Table 2: Convergence (Log_{10} of Relative Error Norm) and runtime (in seconds) of the GMRES solver on a 64 processor Cray T3D. The problem consists of 28060 panels.

Iter	Accurate	$\alpha = 0.5$		$\alpha = 0.667$		$\alpha = 1.0$	
degree $\rightarrow$		4	8	4	8	4	8
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	-3.015280	-3.015172	-3.015274	-3.015179	-3.015274	-3.018524	-3.015268
20	-4.066225	-4.067801	-4.069889	-4.062450	-4.068376	-4.066583	-4.073438
30	-4.964690	-4.972114	-4.976746	-4.961868	-4.973845	-4.968101	-4.986162
40	-5.722380	-5.828296	-5.836648	-5.816061	-5.832452	-5.824179	-5.858259
50	-5.895243	-6.527266	-6.527266	-6.548455	-6.570732	-6.594213	-6.619036
60		-6.673394	-6.673394	-6.770304	-6.770304	-6.895243	-6.895243
Time		53.93	64.91	35.245	41.34	21.64	25.10

Figure 1 plots the reduction in residual norm with iterations for the accurate and the worst case (most inaccurate mat-vec). It can be seen that even for the worst case accuracy, the residual norms are in near agreement until a relative residual norm of 10^{-6}. Furthermore, the approximate iterative method runs significantly faster than the accurate one.

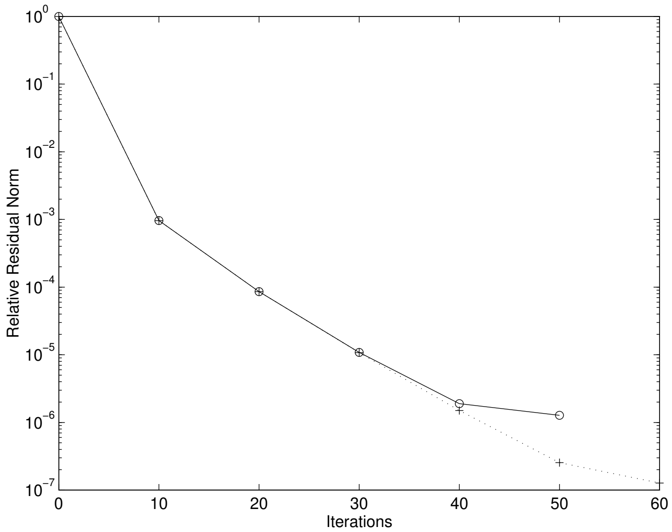

Figure 1: Relative residual norm of accurate and approximate iterative schemes.

4 Preconditioning Techniques

We implement two preconditioning strategies to accelerate the convergence of the GMRES solver:

The hierarchical representation of the domain provides us with a convenient way of approximating the system matrix. Increasing the accuracy of the mat-vec increases the number of direct interactions (and thus the runtime). Conversely, reducing the accuracy (by increasing the α parameter of the Barnes-Hut method, or by decreasing the degree of the multipole series) reduces the runtime. It is therefore possible to visualize a two level scheme in which the inner scheme runs a low accuracy mat-vec. The solve from the inner iteration can be used to accelerate the convergence of the outer iteration. The accuracy of the inner iteration can be increased as the system converges. We have implemented this scheme and preliminary results indicate that for certain accuracies of the inner iteration, this scheme works well.

A primary drawback of the two level scheme implemented above is that the inner iteration is still poorly conditioned. We have also implemented a single level scheme based on truncated Green's functions. In this scheme, the system matrix is approximated by assuming that a panel's influence on another can be neglected if it is sufficiently small. The scheme identifies the k nearest neighbors of each panel. The corresponding matrices are constructed for each panel and factorized. Since we assume that the meshes are not adaptive, the factors need to be computed just once. These factors can be used to precondition the original system. We are currently in the process of evaluating the performance of this scheme. Preliminary results suggest good convergence properties for this scheme.

We expect to present a complete evaluation of both of these schemes in the full-length version of this paper.

References

[1] J. A. Board, J. W. Causey, J. F. Leathrum, A. Windemuth, and K. Schulten. Accelerated molecular dynamics with the fast multipole algorithm. *Chem. Phys. Let.*, 198:89, 1992.

[2] Ananth Grama, Vipin Kumar, and Ahmad Sameh. Scalable parallel formulations of the barnes-hut method for n-body simulations. In *Supercomputing '94 Proceedings*, 1994.

[3] Ananth Grama, Vipin Kumar, and Ahmad Sameh. Parallel matrix-vector product using hierarchical methods. In *Proceedings of Supercomputing '95*, San Diego, CA, 1995.

[4] Ananth Grama, Vipin Kumar, and Ahmed Sameh. On n-body simulations using message passing parallel computers. In *Proceedings of the SIAM Conference on Parallel Processing*, San Francisco, 1995.

[5] L. Greengard and W. Gropp. A parallel version of the fast multipole method. *Parallel Processing for Scientific Computing*, pages 213–222, 1987.

[6] J. F. Leathrum and J. A. Board. Mapping the adaptive fast multipole algorithm into mimd systems. In P. Mehrotra and J. Saltz, editors, *Unstructured Scientific Computation on Scalable Multiprocessors*. MIT Press, Cambridge, MA, 1992.

[7] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Multipole accelerated preconditioned iterative methods for three-dimensional potential integral equations of the first kind. *J. on Sci. and Stat. Comp.*, 15(3):713–735, May 1994.

[8] K. E. Schmidt and M. A. Lee. Implementing the fast multipole method in three dimensions. *J. Stat. Phys.*, 63:1120, 1991.

[9] J. Singh, C. Holt, T. Totsuka, A. Gupta, and J. Hennessy. Load balancing and data locality in hierarchical n-body methods. *Journal of Parallel and Distributed Computing*, 1994 (to appear).

[10] M. Warren and J. Salmon. Astrophysical n-body simulations using hierarchical tree data structures. In *Proceedings of Supercomputing Conference*, 1992.

[11] M. Warren and J. Salmon. A parallel hashed oct tree n-body algorithm. In *Proceedings of Supercomputing Conference*, 1993.

[12] F. Zhao and S. L. Johnsson. The parallel multipole method on the connection machine. *SIAM J. of Sci. Stat. Comp.*, 12:1420–1437, 1991.