Sparse approximate inverse preconditioners for electromagnetic surface scattering simulations

M. Ganesh[1] S. C. Hawkins[2]

(Received 26 July 2007; revised 31 October 2007)

Abstract

Simulation of electromagnetic waves scattered by a connected three dimensional non-convex obstacle at medium frequencies (where the size of the obstacle is 10 to 100 times the incident wavelength) requires a non-asymptotic approach. Standard boundary element schemes at such frequencies require millions of unknowns. However, recently developed high-order algorithms require only tens of thousands of unknowns at medium frequencies for a class of three dimensional obstacles. At such frequencies we use a sparse approximation to the scattering matrix and so iterative solvers are required. We describe an efficient scheme to solve the associated linear systems using sparse approximate inverse preconditioners. The sparse preconditioners developed in this work facilitate efficient solutions of complex dense linear systems arising in electromagnetic scattering simulations.

See http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/321 for this article, Published November 1, 2007. ISSN 1446-8735

Contents

1 Introduction

Ganesh and Hawkins [9, 10] recently developed high-order surface integral equation Galerkin schemes for simulation of electromagnetic waves and radar cross section computations for a class of three dimensional obstacles. In these algorithms, spectrally accurate approximations of various terms in a surface integral reformulation of the scattering problem, combined with an efficient matrix assembly system, lead to complex dense linear systems with fewer unknowns than are obtained using industrial standard boundary element (method of moments) algorithms such as the Fast Illinois Solver Code [15].

Although the scattering matrices obtained from discretizing surface integral operators are typically dense, due to the choice of orthonormal basis functions, the matrices obtained using the algorithms of Ganesh and Hawkins [9, 10] can be approximated to sufficiently high accuracy using a sparse matrix [11]. Such sparse approximations facilitate reduced storage and hence allow simulations at higher frequencies. Ganesh and Hawkins [9, 10] mainly

solved electromagnetic scattering linear systems using a direct method, but when sparse approximations are used it is efficient to use an iterative solver. For an efficient iterative solution, it is essential to have a good preconditioner.

Ganesh and Hawkins [11] described a preconditioning technique based on an incomplete LU-factorisation (ILU) of a sparse approximation to the scattering matrix. A disadvantage of ILU based methods is that they are challenging to implement in parallel. A parallel implementation is essential for higher frequency problems because, due to the higher number of unknowns involved, the ILU factorisation becomes increasingly expensive. Also, since the number of non-zeros in the ILU factorisation increases due to fill-in, storage difficulties are encountered when the ILU factorisation is performed in serial.

Our focus in this work is on so called medium frequency problems, where the frequency is so high that the *resonance region* techniques (such as boundary element methods) require tremendous computing power, but the frequency is too low for asymptotic techniques to be applied. Asymptotic schemes are yet to be developed for connected non-convex three dimensional obstacle electromagnetic scattering. (For the convex case, the asymptotic ansatz described by Melrose and Taylor [13] is useful.) For medium frequency three dimensional non-convex scattering—which is considered to be the current challenge for computational electromagnetics—we are not able to apply the serial ILU based preconditioning techniques described by Ganesh and Hawkins [11], because the storage required by the preconditioner is more than that available to any single CPU in a standard cluster computing environment, which consists of at most 2 GB to 4 GB memory per processor.

Thus, we are motivated to develop a sparse approximate inverse preconditioner [12, 1, 5, 4] for our problem. The advantages of a sparse approximate inverse preconditioner are that (i) the computation of the sparse approximate inverse can be performed in parallel, and (ii) matrix-vector products with the sparse approximate inverse can be performed in parallel. Sparse approximate inverse preconditioners of various types have been compared

for electromagnetic scattering problems involving linear systems with a few thousand unknowns [1, 5, 4], and for linear systems with about one million unknowns, solved using a fast multipole method [6, 3]. Section 3 describes how we compute a sparse approximate inverse by solving a minimising problem in the Frobenius norm. The crux of the problem is in how to select a sparsity pattern for the sparse approximate inverse. Section 4 describes a technique to select the sparsity pattern that can be applied in parallel without communication, and allows the number of non-zeros in each column of the preconditioner to be specified. Section 5 demonstrates that using this sparse approximate inverse preconditioner we can solve three dimensional non-convex medium frequency electromagnetic scattering problems using a tangential variant of the algorithm of Ganesh and Hawkins [9, 10] with several tens of thousands of unknowns. Such simulations are intractable without preconditioning.

2 Problem formulation

The time-harmonic electromagnetic field induced in free space by an incident wave (with frequency ω and wavelength λ so that $\omega\lambda$ is the speed of light) impinging on the surface ∂D of a three dimensional perfect conductor D satisfies the Maxwell equations exterior to the obstacle and the Silver–Müller radiation condition. We reformulate this exterior problem so that the major part of the computation is reduced to approximation of the vector surface potential $\boldsymbol{w}$ that solves the second kind integral equation [8, Theorem 4.19, p. 126]

$$\boldsymbol{w}(\boldsymbol{x}) + \mathcal{M}\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial D, \tag{1}$$

where $\mathcal{M}$ is the magnetic dipole operator, and $\boldsymbol{f}$ is derived from the incident wave through the boundary condition.

Discretizing the surface integral equation (1) using a tangential variant of the algorithms by Ganesh and Hawkins [9, 10] with a carefully chosen tan-

gential spectrally accurate basis, and sparsifying, leads to the linear system

$$Ax = b\,, \tag{2}$$

where A is an $n \times n$ sparse matrix obtained by discretizing $I + \mathcal{M}$, and x and b are discretizations of $\boldsymbol{w}$ and $\boldsymbol{f}$ respectively.

For our medium frequency scattering applications, the linear system (2) may have several tens of thousands of unknowns. Thus it is necessary to work in parallel, and distribute the storage of A across several computer nodes because of memory limitations on a single computer node. To take advantage of the distributed storage and the sparsity of A, it is convenient to use an iterative solver such as the Generalized Minimal Residual (GMRES) algorithm [14]. Although (1) is a second kind integral equation, poor conditioning arises due to complexity in the surface of the scatterer and due to the frequency of the incident wave. Thus it is necessary to use a preconditioner to accelerate the convergence of the iterative solver. Here we consider the right-preconditioned linear system

$$AMy = b\,, \quad x = My\,, \tag{3}$$

where M is the preconditioner.

We solve the preconditioned linear system (3) using GMRES [14] and we compute the preconditioner M using a sparse approximate inverse technique, because the sparse approximate inverse can be computed in parallel and stored using distributed storage.

3 A sparse approximate inverse preconditioner

The sparse approximate inverse preconditioner M is computed as the minimiser of

$$\|AM - I\|_F \tag{4}$$

amongst all matrices with a particular a priori selected sparsity pattern. The cost of computing the preconditioner is controlled by regulating the sparsity of M. If $\|AM - I\|_F$ is small, then one expects the conditioning of AM to be good, leading to fast convergence of the GMRES iterations.

Expanding the Frobenius norm in (4) we have

$$\|AM - I\|_F^2 = \sum_{i=1}^{n} \|Am_i - e_i\|_2^2 ,$$

where $M = [m_1, \ldots, m_n]$ and e_i is the ith Euclidean vector. Thus minimising (4) decouples into n independent least squares problems:

$$\text{minimise} \quad \|Am_i - e_i\|_2 \tag{5}$$

subject to the prescribed sparsity pattern for m_i. These n least squares problems are independent of each other and hence facilitate a naturally parallel implementation (without any communication) to compute the preconditioner M. Let S_i be a vector indexing the nonzero entries in m_i. Then

$$\|Am_i - e_i\|_2 = \|A(:, S_i)m_i(S_i) - e_i\|_2 .$$

Depending on the sparsity of A, some of the rows in $A(:, S_i)$ may be zero. Let T_i be a vector indexing the nonzero rows in $A(:, S_i)$. Then

$$\|Am_i - e_i\|_2 = \|A(T_i, S_i)m_i(S_i) - e_i(T_i)\|_2 , \tag{6}$$

and we solve the least squares problem (5) by solving the reduced problem (6) with matrix $A(T_i, S_i)$ which has size $|T_i| \times |S_i|$.

We solve the least squares problems (6) in parallel using a dense QR-factorisation of $A(T_i, S_i)$, which has complexity $O(|T_i||S_i|^2)$. Thus the complexity of the least squares solve will be reduced if A is replaced by an approximation to A that is sparser than A, because $|T_i|$ typically depends on the sparsity of A.

4 Selecting sparsity pattern

The challenge in sparse approximate inverse computations is selecting a suitable sparsity pattern for M, because in general one does not know the sparsity or structure of A^{-1}. The sparsity pattern of A is often used to suggest a sparsity pattern for M [7].

In the problems we consider, the sparsity pattern of A has too many non-zeros to be used as a sparsity pattern for M. Thus we obtain a sparsity pattern for M from a sparse approximation to A [1, 12].

Simple algebraic strategies to obtain a sparse approximation to A include thresholding [12] and selection of the largest entries in each column [1]. Other sparsification strategies developed for matrices that arise from boundary element methods, based on topological information or geometric information [5], are not applicable for matrices arising from our mesh free, high order algorithm.

Ganesh and Hawkins [11] used a thresholding technique that can be implemented in parallel without communication. This technique often leads to expensive sparse approximate inverse computations because one cannot specify the number of non-zeros a priori. We want to strictly limit the number of non-zeros so that the cost of computing the sparse approximate inverse is limited. Therefore, we obtain a sparse approximation $D = (d_{ij})$ to $A = (a_{ij})$ by combining some of the ideas above in a two-part process:

$$d_{ij} = \begin{cases} a_{ij}\,, & \text{if } |a_{ij}| > \tau \max_{\widehat{i}} |a_{\widehat{i}j}| \text{ and } i \in V_j, \\ 0\,, & \text{otherwise,} \end{cases} \tag{7}$$

where V_j indexes the k largest elements in column j of A. Thus in each column we retain the k largest elements, provided they are above the threshold. The threshold is useful from a computational point of view even when the column contains more than k elements larger than the threshold, because it limits the number of elements that must be sorted in order to select the

k largest. This substantially reduces the time needed to compute the preconditioner. The thresholding part of the strategy (7), which follows Ganesh and Hawkins [11], can be performed in parallel without communication when A and D are stored by columns.

Combining the ingredients above, we construct our preconditioner M as follows.

1. Compute $B \approx A$ using thresholding (7) with tolerance ϵ_A and k_A elements per column.
2. Compute $P \approx A$ using thresholding (7) with tolerance ϵ_M and k_M elements per column.
3. Compute sparse approximate inverse M of B using (4)–(6) where the sparsity pattern for M is chosen to be the sparsity pattern of P.

In our code we overwrite P with M so that no extra storage is used for P.

5 Numerical results

We demonstrate our preconditioner with scattering simulations for a perfectly conducting scatterer that was used to demonstrate the accuracy of high-order methods [9, 10]. This fountain shaped scatterer is illustrated in Figure 1. This object is considered to be challenging for scattering simulations because it possesses concave regions. Since the exact scattered field and far field are not known for plane wave scattering by the fountain shaped obstacle, we give results for scattering of an incident wave generated by a magnetic dipole contained inside the obstacle. For this problem we are able to give exact errors in the computed far field, and thus verify the accuracy of our computed solution to (3). In our tables we give the relative error in the

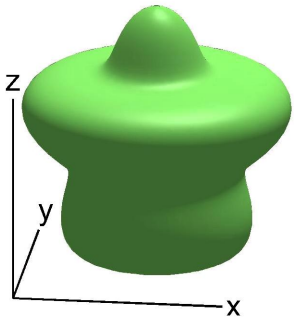

FIGURE 1: Fountain shaped obstacle.

far field, measured in the infinity norm. In practice the infinity norm is computed at more than 1300 points in the far field.

In our experiments we use a parallel version of our algorithm written in Fortran. Distributed storage is used for the matrices A and M. The least squares solve (6) is performed using ZGEQRF and other routines from the LAPACK library [2]. In all experiments we use GMRES [14] restarted after 1000 iterations. Due to the high order nature of the discretization of (1), we expect several digits accuracy in our computed far field. Therefore we terminate the iterations when the residual norm has been reduced by a factor of 10^{-8}, and we correspondingly choose the thresholding parameters for the preconditioner to be $\epsilon_M = \epsilon_A = 10^{-8}$.

Due to the high CPU time required for medium frequency electromagnetic scattering simulations, we first demonstrate the parameter dependencies of the preconditioner for a relatively small problem with 16 560 unknowns simulating scattering by a fountain shaped obstacle with diameter 16 times the incident wavelength. The number of GMRES iterations and CPU time using $4 \times 2\,\text{GHz}$ single core Opteron CPUs are displayed for several values of k_A and k_M in Table 1. We see that the quality of the preconditioner, as indicated by the number of GMRES iterations and the CPU time required for the solve,

TABLE 1: Investigation of parameters for fountain obstacle with diameter 16 wavelengths using 16 560 unknowns. Sparse approximate inverse preconditioner computed in parallel using 4×2 GHz single core Opteron CPUs.

preconditioner		CPU time			
k_M	k_A	assembly	solve	iterations	rel. error
50	50	197.1 s	76.6 s	604	5.55e-6
50	100	326.1 s	80.8 s	575	5.55e-6
50	200	663.3 s	81.9 s	589	5.55e-5
100	100	339.7 s	11.5 s	128	5.56e-6
100	200	643.5 s	10.5 s	114	5.55e-6
200	200	693.7 s	1.2 s	1	5.55e-6

TABLE 2: Convergence of GMRES for fountain obstacle at medium frequencies with diameter between 30 and 40 times the wavelength. Sparse approximate inverse preconditioner computed in parallel using 16×2 GHz dual core Opteron CPUs.

		preconditioner	CPU time			
size	unknowns	$k_M = k_A$	assembly	solve	iterations	rel. error
30λ	51840	100	1.4 h	0.9 h	2798	3.75e-8
40λ	72960	200	4.8 h	1.0 h	2150	6.03e-5

increases with k_M and has little dependence on k_A. In contrast, the cost of constructing the preconditioner increases approximately linearly with k_A but in these experiments has little dependence on k_M. Thus the best results are obtained with $k_A = k_M$.

The results in Table 1 show an inverse relationship between the cost of assembling the preconditioner and the number of GMRES iterations required to reduce the residual norm by a fixed amount. Minimising the total time to obtain a solution is therefore a trade off between the cost of assembling the preconditioner and the cost of performing the GMRES iterations. In the case

of multiple linear systems with one matrix but many right hand sides, the preconditioner can be computed and stored, whilst GMRES iterations must be performed for each right hand side. This consideration is a key factor in the trade off. Such multiple linear systems arise, for example, in monostatic radar cross section computations.

Next we demonstrate the preconditioner for challenging problems simulating scattering by the fountain shaped obstacle at medium frequencies where the diameter is between 30 and 40 times the incident wavelength, with up to 72 960 unknowns. The number of GMRES iterations and CPU time using 16×2 GHz dual core Opteron CPUs are displayed in Table 2. In these experiments, based on the results for the smaller 16 wavelength problem, we use $k_A = k_M$. For the 30 wavelength problem $k_A = k_M = 100$ is sufficient for fast convergence of GMRES. For the higher frequency 40 wavelength problem a better choice is $k_A = k_M = 200$, for which the higher assembly time is more than offset by the reduced solve time.

6 Conclusions

Sparse linear systems derived from dense linear systems arising in electromagnetic scattering simulations with several tens of thousands of unknowns are solved iteratively using a sparse approximate inverse preconditioner implemented in parallel, with sparsity pattern chosen a priori using the two part process described in this work. We have demonstrated the preconditioner by simulating electromagnetic scattering by a non-convex object with diameter forty times the wavelength using multiple processors for preconditioner assembly and for matrix vector products. Iterative solution of such linear systems is intractable without preconditioning, and are hard to precondition using factorisation-based preconditioners, due to the difficulty in computing these preconditioners in parallel.

This work is the first stage of a project to solve medium frequency scat-

tering problems using sparsification combined with a high order spectral Galerkin method. Efficient implementation relies on the ability to determine a priori the sparsity pattern of the matrix A. Such pattern selection is well established for wavelet based methods but the corresponding theory for the spectrally accurate method is a future goal of the project.

Acknowledgements Support of the Australian Research Council is gratefully acknowledged.

References

[1] G. Alleon, M. Benzi, and L. Giraud. Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics. *Numerical Algorithms*, 16:1–15, 1997. doi:10.1023/A:1019170609950. C157, C158, C161

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. C163

[3] B. Carpentieri. Fast iterative solution methods in electromagnetic scattering. Technical Report 17/2006, University of Graz, 2006. http://www.uni-graz.at/~carpenti/papers/IMA17-06.pdf. C158

[4] B. Carpentieri, I. S. Duff, and L. Giraud. Robust preconditioning of dense problems from electromagnetics. In *Numer. Anal. and App. Lecture Notes in Computer Science 1988*, pages 170–178. Springer, 2000. http://www.uni-graz.at/~carpenti/papers/rousse.pdf. C157, C158

[5] B. Carpentieri, I. S. Duff, and L. Giraud. Sparse pattern selection strategies for robust frobenius-norm minimization preconditioners in electromagnetism. *Numer. Linear Algebra Appl.*, 7:667–685, 2000. doi:10.1002/1099-1506(200010/12)7:7/8¡667::AID-NLA218¿3.0.CO;2-X. C157, C158, C161

[6] B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations. *SIAM J. Sci. Comput.*, 27:774–792, 2005. doi:10.1137/040603917. C158

[7] E. Chow. A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.*, 21:1804–1822, 2000. doi:10.1137/S106482759833913X. C161

[8] D. Colton and R. Kress. *Integral Equation Methods in Scattering Theory*. Wiley, 1983. C158

[9] M. Ganesh and S. C. Hawkins. A spectrally accurate algorithm for electromagnetic scattering in three dimensions. *Numerical Algorithms*, 43:25–60, 2006. doi:10.1007/s11075-006-9033-7. C156, C158, C162

[10] M. Ganesh and S. C. Hawkins. A hybrid high-order algorithm for radar cross section computations. *SIAM J. Sci. Comput.*, 29:1217–1243, 2007. doi:10.1137/060664859. C156, C158, C162

[11] M. Ganesh and S. C. Hawkins. Sparse preconditioners for dense complex linear systems arising in some radar cross section computations. *ANZIAM J.*, 48:C233–C248, 2007. http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/61. C156, C157, C161, C162

[12] L. Y. Kolotolina. Explicit preconditioning of systems of linear algebraic equations. *J. Sov. Math.*, 43:2566–2573, 1988. C157, C161

[13] R. B. Melrose and M. E. Taylor. Near peak scattering and the corrected kirchhoff approximation for a convex obstacle. *Adv. in Math.*, 55:242–315, 1985. doi:10.1016/0001-8708(85)90093-3. C157

[14] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986. doi:10.1137/0907058. C159, C163

[15] J. M. Song, C. C. Lu, W. C. Chew, and S. W. Lee. Fast Illinois solver code (FISC). *IEEE Antennas Propag. Mag.*, 40:27–34, 1998. doi:10.1109/74.706067. C156

Author addresses

1. **M. Ganesh**, Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA.
mailto:mganesh@mines.edu

2. **S. C. Hawkins**, School of Mathematics and Statistics, University of New South Wales, Sydney, NSW 2052, Australia.
mailto:stuart.hawkins@unsw.edu.au