New Block ILU Preconditioner Scheme for Numerical Analysis of Very Large Electromagnetic Problems

Alex Heldring, Jaun M. Rius, and Leo Ligthart

***Abstract*—Large electromagnetic scattering and radiation problems are tackled by iterative solvers, which require the use of huge preconditioners. Most often, the incomplete LU decomposition (ILU) of the preconditioner is applied to the system matrix at each iteration. However, the preconditioner ILU cannot be done in-core when the size of the preconditioning matrix exceeds the available memory. This paper presents a new preconditioning scheme to do the preconditioner ILU in small blocks that fit in core memory. The resulting approach allows the solution of very large problems in small computers.**

***Index Terms*—Iterative solution of linear systems, Method of Moments, numerical analysis, preconditioning.**

I. INTRODUCTION

INTEGRAL equation methods (IE) [1] are widely used in conjunction with Method of Moments (MoM) discretization [2] for the numerical analysis of electromagnetic radiation and scattering. IE-MoM formulation leads to a full linear system of order N, where N is the number of unknowns of the problem. The operation count for the solution grows with N^3 for direct solution (LU decomposition) or with N^2 (a matrix-vector product) per iteration for an iterative method. The advent of very efficient methods for computing matrix-vector products in the iterative solution of the IE-MoM linear system has made the solution of problems involving full matrices and a very large number of unknowns (more than a hundred thousand) within the reach of present computers. Examples of such efficient algorithms are the conjugate or biconjugate methods using the fast Fourier transform [3], [4], the fast multipole algorithm [5] and its multilevel version (MLFMA) [6], or the Multilevel Matrix Decomposition Algorithm (MLMDA) [7].

However, the MoM matrices arising from large electromagnetic scattering and radiation problems are often poorly conditioned, especially when they are based on the Electric Field Integral Equation (EFIE) [1], as is the case for problems involving open surfaces (infinitely thin structures). This causes iterative methods to converge very slowly or not at all. Therefore, it is crucial to use an efficient preconditioner. A successful preconditioner is obtained by creating a sparse matrix M containing only the largest elements per row of the MoM matrix A and multiplying by M^{-1} both sides of the equation system:

$$A\mathbf{x} = \mathbf{y} \quad \Rightarrow \quad M^{-1}A\mathbf{x} = M^{-1}\mathbf{y}. \tag{1}$$

The linear system matrix, $M^{-1}A$, usually has a very good condition number and convergence can be achieved in very few iterations. The conventional preconditioning scheme computes and stores the Incomplete LU (ILU) decomposition of M [8]. The preconditioner ILU is then applied at each iteration by forward and backward substitution on the working vector $A\mathbf{x}^{(\mathbf{k}}$, where $\mathbf{x}^{(\mathbf{k}}$ is the kth approximation to the unknown $\mathbf{x}$.

The effectiveness of this preconditioning scheme depends strongly on the size of M and on the thresholding value used in the ILU. These two parameters determine the size of the L and U factors (size refers here to the number of nonzero matrix elements). A problem arises when the size of the preconditioner ILU necessary to achieve good convergence is too large for the computer core memory, since the conventional ILU algorithm needs access to the entire matrices L and U as they are built.

This paper proposes a new block ILU algorithm, without loss of efficiency and with limited computational overhead. The blocks can be computed and stored sequentially in core memory, allowing the use of a huge preconditioner ILU that uses much more memory than is available.

II. ALGORITHM

The proposed blocked ILU algorithm is based on the partitioned inverse formulas [9]. If a given matrix A is partitioned into four blocks

$$A = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \tag{2}$$

where P and S are square submatrices, then the inverse can be expressed as

$$A^{-1} = \begin{bmatrix} \tilde{P} & \tilde{Q} \\ \tilde{R} & \tilde{S} \end{bmatrix} \tag{3}$$

with

$$\begin{aligned} \tilde{P} &= (P - QS^{-1}R)^{-1} \\ \tilde{Q} &= -\tilde{P}(QS^{-1}) \\ \tilde{R} &= -(S^{-1}R)\tilde{P} \\ \tilde{S} &= S^{-1} + (S^{-1}R)\tilde{P}(QS^{-1}) \end{aligned} \tag{4}$$

as can be verified by substituting (4) in the product AA^{-1}.

Manuscript received July 5, 2001; revised October 25, 2001. This work was supported by the European Comission and the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT) through Grants FEDER 2FD97-0135 and TIC 98-1037 and by the Departament d'Universitats Recerca i Societat de la Informació (DURSI) of the Generalitat de Catalunya.

A. Heldring and L. Ligthart are with the International Research Centre for Telecommunications and Radar, Delft University of Technology, 2628 CD Delft, The Netherlands.

J. M. Rius is with the Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: rius@tsc.upc.es).

Publisher Item Identifier S 0018-9464(02)01572-8.

0018-9464/01$17.00 © 2002 IEEE

In order to use (4) recursively on the preconditioning matrix M, it is subdivided into a chosen number of blocks of approximately equal sizes, based on the problem geometry. As a result, M looks like

$$M = \begin{array}{|c|c|c|c|c|}\hline P_n & Q_n^{n-1} & Q_n^{n-2} & \cdots & Q_n^0 \\ \hline R_n^{n-1} & P_{n-1} & & & \vdots \\ \hline R_n^{n-2} & & \ddots & & \\ \hline \vdots & & & P_1 & Q_1^0 \\ \hline R_n^0 & \cdots & & R_1^0 & S_1 \\ \hline \end{array} \tag{5}$$

where $n+1$ is the total number of blocks. Since M contains the largest elements in each row of the linear system matrix A and zeroes elsewhere, with proper column and row ordering the diagonal blocks P_m and S_m are full or almost full, while many of the off-diagonal submatrices Q_m^k and R_m^k are very sparse. In the following, Q_m denotes the entire right-to-diagonal submatrix $[Q_m^{m-1} \cdots Q_m^0]$ and likewise R_m denotes the entire below-diagonal submatrix. Furthermore, $Q_m^{(p)}$ denotes the block $[Q_m^p \cdots Q_m^0]$ and likewise for $R_m^{(p)}$.

First, the ILU factorization of S_1 is computed with a drop tolerance τ_1. The operator $(U^{-1}L^{-1})'$ with $LU = S_1$ is represented by Σ_1', where the prime means that thresholding has been applied to drop small elements. Σ_1' is applied on the right to Q_1 and on the left to R_1, yielding the factors

$$\Theta_1 = Q_1 \Sigma_1' \tag{6}$$

$$\Gamma_1 = \Sigma_1' R_1. \tag{7}$$

All elements below a chosen threshold τ_2 with respect to the largest elements in Θ_1 and Γ_1 are dropped (set to zero). The resulting matrices are denoted with Θ_1' and Γ_1'. Subsequently, the factor

$$\Pi_1 = P_1 - Q_1\Gamma_1' \approx P_1 - Q_1 S_1^{-1} R_1 \tag{8}$$

is computed, and a dropping with threshold τ_2 is applied to Π_1, yielding Π_1'. Then, Π_1' is ILU factorized. The operator $U^{-1}L^{-1}$ with $LU = \Pi_1'$ is represented by $\tilde{P}_1$ The operators

$$\tilde{R}_1 = -\Gamma_1' \tilde{P}_1 \tag{9}$$

and

$$\tilde{Q}_1 = -\tilde{P}_1 \Theta_1' \tag{10}$$

are not explicitly stored, but whenever they are needed as operators on a matrix or vector, (9) and (10) are invoked. This concludes step one.

Now, $\tilde{P}_1$, $\tilde{Q}_1$, and $\tilde{R}_1$ are available as operators, and so is $\Sigma_1' \approx S_1^{-1}$. Subsequently, the operators Γ_m', Θ_m' and $\tilde{P}_m$ are computed and stored sequentially for $m = 2 \cdots n$. Θ_m is defined as $Q_m \Sigma_m$ and can be found with the recursive formula

$$Q_p \Sigma_q' = \left[\left[\Xi_{q-1}^p\right] \quad \left[-\Xi_{q-1}^p \Theta_{q-1}' + Q_p^{(q-2)} \Sigma_{q-1} \right] \right] \tag{11}$$

in which

$$\Xi_{q-1}^p = -\left(Q_p^{q-1} - Q_p^{(q-2)} \Gamma_{q-1}' \right) \tilde{P}_{q-1}. \tag{12}$$

Likewise, Γ_m is defined as $\Sigma_m R_m$ and can be found with the recursive formula

$$\Sigma_q' R_p = \begin{bmatrix} \Lambda_{q-1}^p \\ -\Gamma_{q-1}' \Lambda_{q-1}^p + \Sigma_{q-1} R_p^{(q-2)} \end{bmatrix} \tag{13}$$

in which

$$\Lambda_{q-1}^p = \tilde{P}_{q-1} \left(R_p^{q-1} - \Theta_{q-1}' R_p^{(q-2)} \right). \tag{14}$$

Every operation in (11) and (13) is followed by a dropping with threshold τ_2, to obtain sparser Γ_m' and Θ_m' from Γ_m and Θ_m. At every level, once Γ_m' is known, $\tilde{P}_m$ is computed as $\tilde{P}_m = U^{-1}L^{-1}$ with $LU = \Pi_m'$, where Π_m', obtained from

$$\Pi_m = P_m - Q_m \Gamma_m' \tag{15}$$

and a dropping with threshold τ_2.

A close examination of (11) and (13) shows that the operators Σ_m' never have to be computed explicitly: for the computation of Γ_m', Θ_m' and $\tilde{P}_m$ only Γ_k', Θ_k' and $\tilde{P}_k$ with $k = m-1, m-2 \cdots 1$ are needed. At each level m, all the previous factors are loaded and used recursively, down to level one. For symmetrical matrices like the EFIE impedance matrix, the factorization workload and storage is reduced by a factor of about one half because $\Gamma_m = \Theta_m^T$. The choice of the threshold values τ_1 and τ_2 is a tradeoff between the effectiveness and the size of the preconditioner.

Once the preconditioner factors are computed, the preconditioner has to be applied at each iteration step to a working vector $\mathbf{x}$

$$\mathbf{y} \approx M^{-1}\mathbf{x}. \tag{16}$$

The vector $\mathbf{x}$ is subdivided into $n+1$ blocks corresponding to the blocks of M. Let $\mathbf{x}_i$ denote the subset of elements corresponding to block i and $\mathbf{x}_{(\mathbf{i})}$ the subset of elements corresponding to blocks 1 to i. The same notation applies to the vector $\mathbf{y}$. For block one

$$\mathbf{y}_1 = \mathbf{\Sigma}_1' \mathbf{x}_1. \tag{17}$$

The following blocks are computed recursively for $m = 2 \cdots n+1$ with

$$\mathbf{y}_{(\mathbf{m})} = \begin{bmatrix} \nu_{\mathbf{m}} \\ -\Gamma_{m-1}' \nu_{\mathbf{m}} + \mathbf{y}_{(\mathbf{m-1})} \end{bmatrix} \tag{18}$$

where

$$\nu_{\mathbf{m}} = \tilde{P}_{\mathbf{m-1}} \left(\mathbf{x}_{\mathbf{m}} - \mathbf{\Theta}_{\mathbf{m-1}}' \mathbf{x}_{(\mathbf{m-1})} \right). \tag{19}$$

III. Results

All the computations in the examples presented here were done on a PC compatible computer with a 1-GHz AMD Athlon processor and 768 MB of RAM. The programming language was MATLAB 5 with time-critical routines coded in C.

TABLE I
COMPARISON OF CONVENTIONAL ILU WITH BLOCK-ILU PRECONDITIONER FOR THE PEC SPHERE EXAMPLE

	1 block ILU	32 block ILU
ILU threshold	0.01	0.01
matrix dropping threshold	-	0.02
ILU factorisation time	44 m. 3 s.	55 m. 14 s.
ILU total size	170 MB	246 MB
ILU size in core	170 MB	7.7 MB
GMRES step time	1 m. 10 s.	1 m. 13 s.
convergence (1% error)	134 steps	98 steps
total computation time	3 h. 20 m.	2 h. 54 m.

TABLE II
MoM-EFIE SOLUTION FOR THE UNSHIELDED TARA REFLECTOR USING A 20-BLOCK ILU PRECONDITIONER

size of M	220 MB
ILU threshold	0.001
matrix dropping threshold	0.002
ILU total size	1.3 GB
ILU size in core	66 MB
ILU factorisation time	4 hours 26 min.
GMRES step time	5 min. 20 sec.
convergence (1% error)	40 steps
total computation time	7 hours 59 min.

A. Perfectly Conducting Sphere

As a first example, the induced surface current and bistatic RCS of a Perfectly Conducting (PEC) sphere have been calculated. A relatively small sphere was chosen (8λ diameter) in order to compare the conventional ILU preconditioner with the new block ILU and test the computational overhead of the new approach. The sphere was discretized into 32 768 triangular patches, with an average edge length of approximately $\lambda/10$, leading to 49 152 RWG basis functions [10]. The MLFMA near interactions matrix was used as the preconditioner matrix M, as suggested in [7]. The size of M equaled 480 MB and was computed in 400 s. The resulting linear system was solved iteratively using GMRES [8], both using a conventional (one block) ILU decomposition of M and using a 32-block preconditioner. The stopping criterion for the iteration was a residual relative error of less than 1%.

Table I shows the timing and the matrix sizes for the two computations. The 32-block preconditioner was about 30% larger than the standard one-block preconditioner, but in the 32-block approach, the ILU operations were performed only on very small blocks that require negligible use of core memory. Since the 32-block version was larger, it was also more effective and less iterations were required. The computational overhead of the 32-block ILU versus the conventional approach was small, and was counterbalanced by the smaller number of iterations required by the new algorithm for a faster overall computation. The resulting RCS was virtually equal for the two methods and agreed well with the analytical (Mie series) solution.

B. Parabolic Reflector

The second example is the parabolic reflector used in the Transportable Atmospheric Radar (TARA) project [11] carried out at the International Research Centre for Telecommunications-Transmission and Radar (IRCTR), Faculty of Information Technology and Systems (ITS) of Delft University of Technology (DUT). The reflector diameter is 3 m and the focal distance is 1.54 m. The reflector is illuminated by a dielectric rod antenna at 3.2975 GHz ($\lambda = 9.1$ cm), radiating a linearly polarized, axisymmetric field. The measured feed radiation pattern was used in the computation. The feed is mounted in a cylindrical metal housing (diameter 17 cm, length 20 cm), supported by four thin cylindrical struts (diameter 2.5 cm).

The complete antenna geometry was discretized into triangular patches. The average edge length of the triangles was 0.5

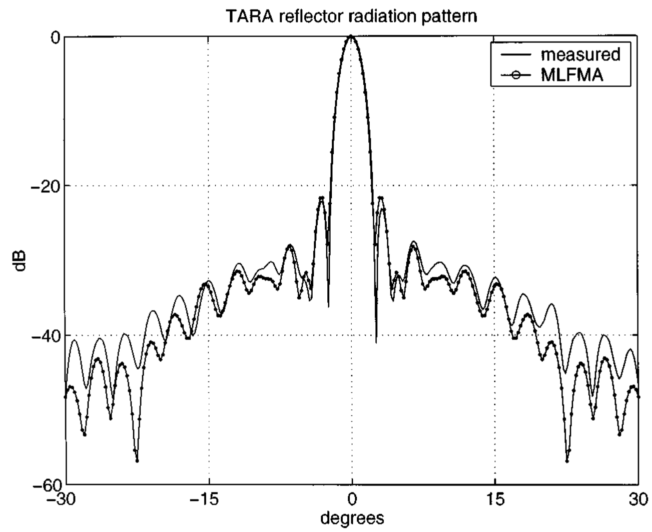

Fig. 1. Radiation pattern of the unshielded TARA parabolic reflector compared with measurements.

cm on the feed housing, the struts, and near the reflector edge, while on the reflector surface far from the edge was 1 cm. This lead to 250 236 triangular patches in total, and to 374 348 RWG basis functions. Making use of the fourfold symmetry of the geometry, the number of unknowns was reduced to 96 620 in order to compress the MLFMA near field interactions matrix [7] and the preconditioner. Unfortunately, the MLFMA far field interactions computation [7] cannot take advantage of symmetries and must deal with the whole uncompressed geometry discretization. The MLFMA near interactions matrix was built in 1000 s and again used as preconditioner. Computation timing and matrix sizes for the GMRES solution of this problem, using a 20-block preconditioner, are given in Table II. This problem was impossible to solve using a standard one-block ILU preconditioner: the largest one-block ILU that fitted in memory needed 480 MB and the residual relative error stagnated at approximately 8% after 173 steps. However, the new block-ILU algorithm allowed the use of 1.3-GB ILU factors and convergence to 1% error was achieved in 40 iterations. Fig. 1 shows the reflector radiation pattern compared with measurements.

The shielded reflector configuration of TARA was also analyzed. The shield size is 2 m and the shield aperture angle 32°. The real feeder pattern was now approximated by $(1+\cos\theta)^{4.75}$. The number of RWG basis functions [10] was 1 483 312. Four-fold symmetry allowed the compression of the

TABLE III
MoM-EFIE SOLUTION FOR THE SHIELDED TARA REFLECTOR USING A 25-BLOCK ILU PRECONDITIONER

size of M	3.28 GB
ILU total size	3.6 GB
ILU size in core	147 MB
ILU factorisation time	33 hours
GMRES step time	24 min. 40 sec.
convergence (1% error)	47 steps
total computation time	52 hours

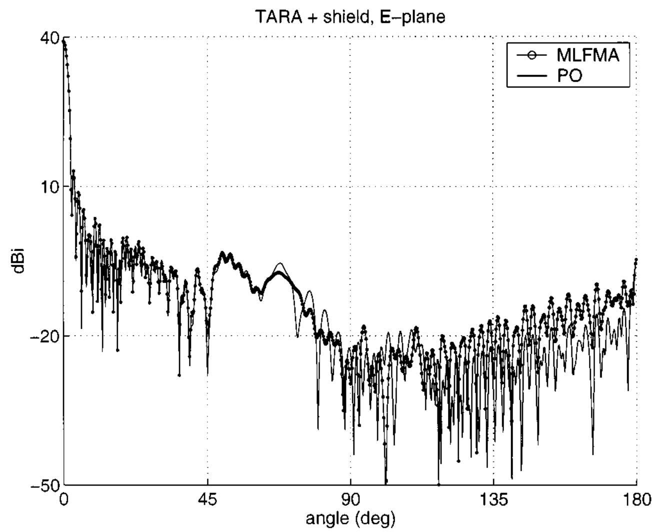

Fig. 2. E-plane radiation pattern of the shielded TARA parabolic reflector compared to PO approximation.

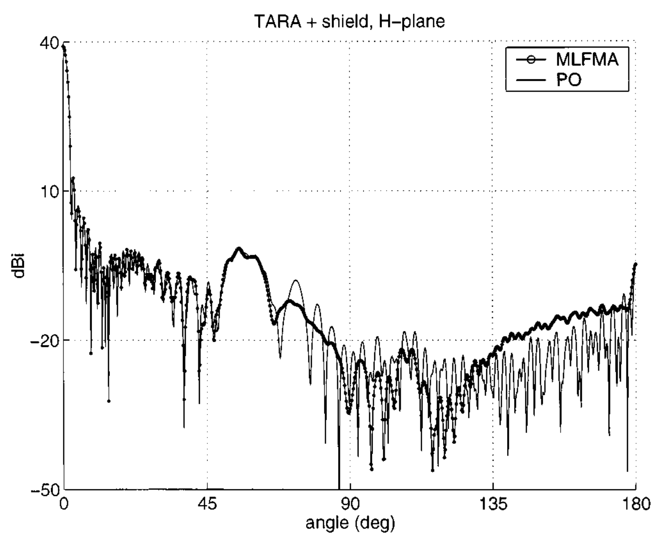

Fig. 3. H-plane radiation pattern of the shielded TARA parabolic reflector compared to PO approximation.

near field matrix and the preconditioner to 372 022 unknowns. Table III shows the computational requirements and Figs. 2 and 3 the radiation pattern computation compared to the Physical Optics (PO) approximation.

IV. CONCLUSION

A new block-ILU preconditioning algorithm is presented that allows the use of large preconditioners on small computers, as it overcomes the requirement for a conventional ILU of not exceeding the core memory size. Larger preconditioners significantly improve the convergence speed of an iterative solution method like GMRES. The new block ILU has been applied to the MoM solution of the EFIE with small memory requirements to perform the ILU decomposition of the preconditioner blocks. The results show a small computational overhead compared to the conventional one-block ILU for relatively small problems, and excellent convergence for large problems that cannot be tackled by the conventional approach. The choice of the number of blocks is not critical; it must not be too small to allow the block-ILU decomposition in core memory, and not so large as to make the computational overhead significant.

REFERENCES

[1] N. Morita, N. Kumagai and, and J. R. Mautz, *Integral Equation Methods for Electromagnetics*. Norwood, MA: Artech, 1990.

[2] R. F. Harrington, *Field Computation by Moment Methods*. New York: MacMillan, 1968.

[3] T. Sarkar, Ed., *Application of Conjugate Gradient Method to Electromagnetic and Signal Analysis*. Amsterdam, The Netherlands: PIER-5, Elsevier, 1991.

[4] C. F. Wang, F. Ling, and J. M. Jin, "A fast full-wave analysis of scattering and radiation from large finite arrays of microstrip antennas," *IEEE Trans. Antennas Propagat.*, vol. 46, pp. 409–418, Oct. 1998.

[5] R. Coifman, V. Rohklin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian description," *IEEE Antennas Propag. Mag.*, vol. 35, pp. 7–12, 1993.

[6] J. Song, C. C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 1488–1493, Oct. 1997.

[7] J. M. Rius, J. Parrón, E. Úbeda, and J. R. Mosig, "Multilevel matrix decomposition algorithm for analysis of electrically large electromagnetic problems in 3-D," *Microwave Opt. Technol. Lett.*, vol. 22, no. 3, pp. 177–182, Aug. 1999.

[8] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS, 1996.

[9] W. H. Press e.a, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[10] M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shapes," *IEEE Trans. Antennas Propag.*, vol. 30, pp. 409–418, May 1982.

[11] S. H. Heijnen and L. P. Ligthart, "TARA: Transportable Atmospheric RAdar," in *Proc. 28th Eur. Microwave Conf.*, vol. 1, 1998, pp. 61–66.