Numerically Efficient Solution of Dense Linear System of Equations Arising in a Class of Electromagnetic Scattering Problems

Jean-René Poirier, Pierre Borderies, R. Mittra, and V. Varadarajan

***Abstract*— In this paper we present an efficient technique for solving dense complex-symmetric linear system of equations arising in the method of moments (MoM) formulation. To illustrate the application of the method, we consider a finite array of scatterers, which gives rise to a large number of unknowns. The solution procedure utilizes preconditioned transpose-free QMR (PTFQMR) iterations and computes the matrix–vector products by employing a compressed impedance matrix. The compression is achieved by reduced-rank representation of the off-diagonal blocks, based on a partial-QR decomposition, which is followed by an iterative refinement. Both the preconditioning and the compression steps take advantage of the block structure of the matrix. The convergence of the iterative procedure is investigated and the performance of the proposed algorithm is compared to that achieved by other schemes. The effectiveness of the preconditioner and the degree of matrix compression are quantified. Finite arrays of variable shape and sizes are considered, and it is demonstrated that the ability to solve large problems using this technique enables one to evaluate the edge effects in the finite array. Such array is basically flat and periodic, but the algorithm is still efficient when variation with strict periodicity or flatness exists.**

***Index Terms*— Electromagnetic scattering, moment methods, numerical analysis.**

I. Introduction

In this paper, we present an efficient algorithm for solving the problem of estimating the edge effects in a finite array of identical scatterers, a problem that requires the solution of a large dense linear system of equations. The numerical algorithm is based on a preconditioned transpose-free quasi-minimal residual (PTFQMR) method, which is combined with a matrix compression scheme involving rank-reducing partial QR decomposition. The latter is followed by an iterative refinement of the solution to obtain the desired accuracy. Both the preconditioning and the compression algorithms are configured such that they can take advantage of the block structure of the matrix. A comparative evaluation of the performance of the present iterative technique is carried out *vis-a-vis* other matrix solution methods—both iterative and direct—with a view to demonstrating the superior computational efficiency of the present solver. We further show that a high matrix compression rate can be achieved by successive refinements without sacrificing the accuracy required. This procedure not only leads to a saving in the memory requirements, and thus enables us to handle large problems which would otherwise be unmanageable, but also contributes to the numerical efficiency of the algorithm.

Next, in Section II, we study the problem of solving the linear system of equations itself. To illustrate the efficiency of the TFQMR algorithm [1], we carry out a comparative study of the performances of several iteration techniques. The important problem of preconditioning is addressed in Section III, both from the points of view of implementation as well as evaluation of the performance of the preconditioner.

The matrix compression process, described in Section IV, includes both the exploitation of the obvious symmetries of the system and the application of the QR compression applied to the off-diagonal blocks. Finally, to improve the accuracy of the result derived with such a highly compressed matrix, we apply the refinement procedure proposed in [2], after generalizing this procedure to an arbitrary order m. We then evaluate the performances of the algorithms in terms of the accuracy of the results obtained for different compression ratios and orders of refinement.

Section V presents some numerical results for truncated arrays. Section VI discusses a number of applications of the iterative technique. Finally, Section VII draws some conclusions on the basis of the results that could not have been derived on the available computer without the use of the compression algorithm.

Manuscript received February 2, 1997; revised December 2, 1997.

J.-R. Poirier and P. Borderies are with ONERA/CERT, Toulouse, 31055 Cédex, France.

R. Mittra is with the Electromagnetic Communication Research Laboratory, Pennsylvania State University, University Park, PA 16802 USA.

V. Varadarajan is with the University of Illinois, Urbana, IL 61801 USA.

Publisher Item Identifier S 0018-926X(98)06099-2.

II. Comparative Study of Iteration Techniques

A. Introduction to Iterative Procedures

A number of iterative procedures are available in the literature and we have tested three of these on a comparative basis by applying them to a representative example.

The generic one we consider is that of a truncated array of coplanar metallic square patches in free space, illuminated by a normally incident plane wave. The MoM formulation, based on the Rao, Wilton, and Glisson [3] triangular-patching approach, gives rise to a linear system of equations with $n*96$ unknowns, where n is the number of conducting patches in the

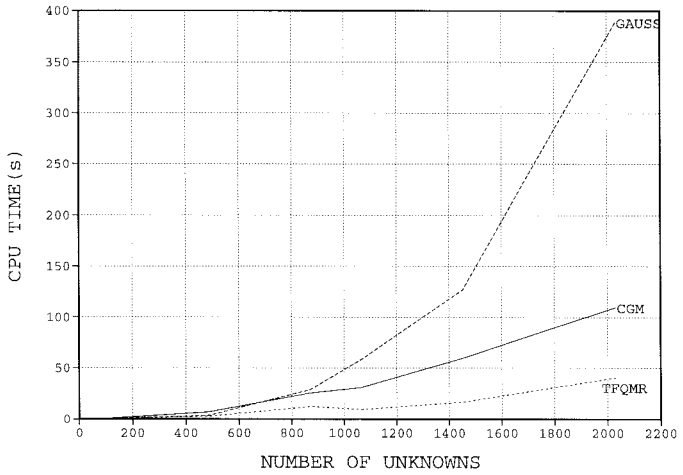

Fig. 1. Computation time as a function of the number of unknowns for TFQMR, CGM, and direct Gauss methods.

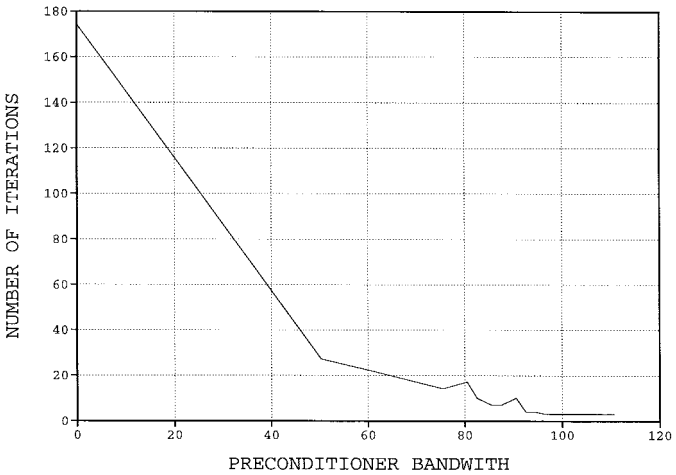

Fig. 2. Efficiency of the band preconditioner: number of iterations versus preconditioner bandwidth.

array. Various configurations of such arrays are considered in this paper to test the algorithms discussed herein.

All of the three iterative techniques used in this work are well known and have been extensively discussed in the literature. They are:

1) Conjugate gradient method (CGM) [4].
2) Generalized minimum residual method (GMRES) [5]–[7].
3) Transpose free quasi minimum residual (TFQMR). The QMR approach is based on Lanczos method described in [6]. The TFQMR approach [1], [2], [8] is an improved version which does not require any extra storage and is also amenable to preconditioning.

B. Numerical Results for Iteration Procedure

One way to judge the convergence rate of an algorithm is to compare the computation time it requires to reduce the residual to a given value, say 10^{-8}. In Fig. 1, we plot the CPU times required to achieve convergence for the CGM and TFQMR algorithms as functions of the number of unknowns, and also compare these results with those for the direct Gauss solver. We omit the corresponding results obtained with the GMRES scheme because of the poor convergence rate it exhibits for the cases studied.

The above results clearly demonstrate the superior efficiency of the TFQMR algorithm. Furthermore, since it is amenable to preconditioning, it is unquestionably the best choice for the cases under study, viz., truncated arrays, which generate diagonally dominant matrix structures. Before closing this section, we point out that the significant gain achieved by using the TFQMR algorithm over the Gaussian elimination scheme is evident from Fig. 1.

III. Preconditioning with TFQMR

A. Implementation of Preconditioning

Preconditioning offers a significant potential for improving the computational efficiency of the matrix solution. Of course, the challenge that confronts us is the derivation of a matrix C such that $\mathrm{C}^{-1}A$ as close as possible to identity matrix I. We then solve an equivalent system $\mathrm{C}^{-1}Ax = \mathrm{C}^{-1}b$ by using an iterative scheme. Since A is both dense and complex in nature, we can only use preconditioners based on an incomplete factorization. In this work, we have investigated the following approaches to generating the preconditioner C.

In the first approach we construct a banded matrix by retaining the off-diagonal terms up to a certain width and factorize the resultant matrix by using the Cholesky-type method applied to the complex-symmetric matrix. The bandwidth is chosen such that it is a good compromise between the increase in the computation time of $\mathrm{C}^{-1}A$ with an increase in the bandwidth, and the gain in the efficiency of iterations as $\mathrm{C}^{-1}A$ tends to approach the identity matrix I.

In the second approach, we factorize a set of diagonal blocks, whose block size may be variable, again by using the method described above. The special choice of block size that correspond to submatrices associated with the individual patches offers considerable advantage, because the diagonal blocks are all identical, and the storage requirements are drastically reduced for this case as well as CPU time. Moreover, since the Green's function naturally decays with the distance, the matrix structure described above is block-diagonally dominant, which is desirable.

B. Numerical Results for Preconditioning

1) The Effect of Preconditioning on the Convergence of the Iteration Procedure: For the example of an array of 15 patches, Fig. 2 shows the number of iterations required to achieve convergence as a function of the bandwidth. We note that the optimum bandwidth coincides with the size of the self-block and that an increase in the bandwidth does not reduce the number of iterations.

Table I presents a comparison of the results derived for several examples by using band and block preconditioners, for a number of different bandwidths and block sizes. These results clearly show that significant improvements can be achieved via preconditioning, confirm the previous conclusions we reached on the choice of the optimum bandwidth, and

TABLE I
EFFECT OF BAND AND BLOCK PRECONDITIONING FOR DIFFERENT BANDWIDTHS NUMBER OF REQUIRED ITERATIONS

Preconditioner	without Prec.	block size 96	block size 288	bandwidth 50	bandwidth 96	bandwidth 110
matrix 864	166	5	5	35	5	5
matrix 1440	173	3	2	27	3	2
matrix 2016	174	4	3	30	3	3

TABLE II
EFFECT OF PRECONDITIONING FOR A LINEAR ARRAY OF 21 PATCHES

Preconditioner	block size 96	bandwith 90	bandwidth 96	bandwidth 250
Number of iterations	4	10	3	1
Total comp. time (sec)	132	145	134	138
Solve time (sec)	2	15	4	8

TABLE III
EFFECT OF PRECONDITIONING FOR A SQUARE ARRAY OF 25 PATCHES

Preconditioner	block size 96	bandwidth 90	bandwidth 96	bandwidth 250
Number of iteration	12	17	8	7
Total comp.Time(sec)	191	198	193	204
Solve time (sec)	7	14	9	20

extend them to the block size as well. The table also shows that efficiencies in terms of required number of iterations are very close for the band and the block approaches, provided that their respective bandwidths are also close. In this event, the block preconditioning becomes the preferred approach.

2) Computation Time: It is interesting to track the effect of preconditioning on the computation time, and we do this below. Tables II and III present the results for a linear array of 21 patches with 2016 unknowns and a square array of 25 patches with 2400 unknowns, respectively.

We observe from the above tables that the preconditioners are so efficient as to make the solve time negligible compared to the I/O time for the matrix fill, the former being approximately one third of the time it takes to compute the matrix elements. We further note that the convergence is most rapid when the size of the block preconditioner is identical to that of the band. These results demonstrate that the PTFQMR algorithm is really a very powerful tool for solving problems of this type. Furthermore, we observe that the availability of an efficient solver is an essential tool for handling large systems that are highly sensitive to numerical error propagation, and for implementing multiple refinement procedures, discussed later in this paper, that require the construction of the solution anew for each step of iteration.

IV. MATRIX COMPRESSION

Matrix compression is not only a useful tool for the reduction of RAM memory requirements, but also is crucial for solving large problems. In this work we carry out the matrix compression in two steps by employing the QR compression scheme as described by a number of authors (see [2], [4], and [9], for instance), after exploiting the presence of redundancies in the matrix elements. This initial compression rate varies from 50% in the general case (up–down symmmetry) to lower rates depending of the geometrical symmetries.

A. Block QR Compression

Let us now discuss the global compression of the impedance matrix A into an approximate matrix A_c. We do this by individually compressing each of the off-diagonal blocks using an incomplete QR factorization, which approximates these blocks with an equivalent matrix of an inferior rank. The QR factorization entails the decomposition of a matrix B in the form $\mathrm{B} = \mathrm{QR}$, where R is an upper triangular matrix and Q is an orthogonal matrix. Using the Householder transformation method we can write

$$\mathrm{B} = (\mathrm{H}_{n-1}\mathrm{H}_{n-2}\cdots\mathrm{H}_1)^{-1}\mathrm{R} = \mathrm{QR}$$

where H_i are Householder matrixes [9]. The step k of the algorithm can be written as

$$\mathrm{B}_k = \mathrm{H}_{k-1}\cdots\mathrm{H}_1\mathrm{B} = \begin{pmatrix} \mathrm{R}_1^{(k)} & \mathrm{R}_2^{(k)} \\ 0 & \mathrm{R}^{(n-k)} \end{pmatrix}.$$

If B is such that $\mathrm{R}^{(n-k)}$ is close to zero, then there exists a matrix $\tilde{\mathrm{B}}$ of rank k which is close to B, and is defined by

$$\tilde{\mathrm{B}} = \tilde{\mathrm{Q}}\tilde{\mathrm{R}} = \mathrm{B} - \mathrm{U},$$
$$\text{with } \mathrm{U} \approx 0,\ \tilde{\mathrm{Q}} = (\mathrm{Q}^{(k)}\ \ 0),\ \tilde{\mathrm{R}} = \begin{pmatrix} \mathrm{R}_1^{(k)} & \mathrm{R}_2^{(k)} \\ 0 & 0 \end{pmatrix}.$$

Next, define a threshold τ that provides us the criterion to determine if $\mathrm{R}^{(n-k)} \approx 0$. Specifically, at each transformation step k, we compare the norms of all the lines of $\mathrm{R}^{(n-k)}$ with the norm of the first line of $\mathrm{R}^{(k)}$ by computing their ratio. If the maximum ratio is weaker than a given threshold τ, then all of the terms of $\mathrm{R}^{(n-k)}$ are set to zero; otherwise, the algorithm is allowed to continue.

B. Refinement of the Solution

Next, to improve the accuracy of the result derived by using a compressed matrix, we apply a refinement procedure proposed in [2] after generalizing it to an arbitrary order m

$$\mathrm{A}I = V \Leftrightarrow [\mathrm{A}_c + (\mathrm{A} - \mathrm{A}_c)].[I_0 + I_1 + \cdots + I_k + \cdots] = V.$$

Next, we note that by solving the following system of equations

$$\mathrm{A}_c I_0 = V$$
$$\cdots\cdots\cdots$$
$$\mathrm{A}_c I_k = -(\mathrm{A} - \mathrm{A}_c) I_{k-1}$$

in a successive manner, we can derive an iterative solution which is as close to the true solution as we desire, provided that the spectral radius of $\mathrm{A}_c^{-1}(\mathrm{A} - \mathrm{A}_C)$ is less than one. This iterative refinement, if limited to the order k, leads to ($\delta = ||\mathrm{A} - \mathrm{A}_c||$):

$$||I_k|| \le \delta^k \left\|\mathrm{A}_c^{-1}\right\|^k ||I_0|| = 0(\delta^k)$$
$$||\mathrm{A}I - V|| \le \delta^{k+1} \left\|\mathrm{A}_c^{-1}\right\|^k ||I_0|| = 0(\delta^{k+1}).$$

In practice, such a second-order refinement is usually found to be sufficient. We note, in passing, that each iteration step

TABLE IV
COMPUTATION TIMES ASSOCIATED WITH DIFFERENT EXAMPLES

	Example-1	Example-2	Example-3
Matrix fill (sec)	131	410	556
QR compression and solution (sec)	44	132	238
Refinement δ^2 (sec)	76	227	381

requires extra input-outputs involving the mass memory of the uncompressed matrix A. However, we note also that A_c is involved only in the solution of the linear systems, while A is used solely to form the right hand side.

V. NUMERICAL RESULTS DERIVED BY USING BLOCK-PRECONDITIONED TFQMR

We have applied the block-preconditioned TFQMR solver to the following three examples involving different number of unknowns.

1) Example-1: truncated, periodic array of 25 patches, configured in a square shape involving 2400 unknowns (initial compression 25%).
2) Example-2: truncated, periodic linear array of 35 patches with 3360 unknowns (initial compression 25%).
3) Example-3: nonperiodic array, same as Example 1 with one element removed (initial compression 50%).

A. Computation Time

Table IV lists the computation times associated with the three principal steps involved in the solution process, viz., filling the matrix; solving the system with the compressed matrix; and, second-order refinement. The tolerance value chosen for the examples above is $\tau = 0.01$. We should add the remark that the computation times include the time needed for the input–output operations that are machine dependent and have not been optimized on the Cray computer used to solve the problem.

The following observations may be made about the results presented in Table IV. First, a major part of the matrix fill-time is devoted to the computation of the entries in the matrix. The computational burden for this calculation is proportional to N^2 unless any redundancies are exploited. However, since the uncompressed matrix is stored in the mass memory, this computation may be carried out in advance as a separate process. As mentioned previously, the solve time is relatively small, as is the time required to carry out the compression. For the case of multiple right-hand sides, the compression step has to be performed only once. The bulk of the time in the refinement process is consumed by the input–output operations involving the uncompressed matrix. The CPU time needed for this process is approximately the same as the solve time.

B. Compression Efficiency

We have carried out additional numerical experiments with the above three examples for different values of τ to determine the efficiency of compression. Below we tabulate the compression rate realized for the entire matrix and the rms relative errors $\|I - I_c\|/\|I\|$ in the coefficients of current densities

TABLE V
RELATIVE RMS ERROR IN CURRENT DENSITY COEFFICIENTS FOR (a) EXAMPLE 1, (b) EXAMPLE 2, AND (c) EXAMPLE 3

τ	compression rate	no refinement	1st order ref.	2nd order ref.
.01	12.9 %	11.6 %	1.3 %	.48 %
.05	5.7 %	12.9 %	1.3 %	.46 %
.1	3.7 %	15.5 %	1.7 %	.63 %
.2	2.1 %	33.2 %	10.3 %	4.7 %

(a)

τ	Compression rate	no refinement	1st order ref.	2nd order ref.
.01	7.3 %	6 %	.3 %	.01 %
.05	3 %	6 %	.3 %	.02 %
.1	2 %	6.8 %	.38 %	.035 %
.2	1.5 %	21 %	3.8 %	.72 %

(b)

τ	Compression rate	no refinement	1st order ref.	2nd order ref.
.01	25.1%	6.5 %	.95%	.14%
.05	12.5 %	6.9 %	.85 %	.17 %
.1	8.1 %	9.3 %	2.35 %	.50%
.2	5 .7%	12.2 %	4.4 %	1.4 %

(c)

TABLE VI

	Example-1		Example-2		Example-3	
τ	co-pol	X-pol	co-pol	X-pol	co-pol	X-pol
0 to .1	-19.26	-57.43	-14.53	-53.72	-11.98	-50.17
.2	- 19.27	-57.02	- 14.44	- 49.53	- 11.99	- 49.73

(I) without refinement and, the same error with the first- and second-order refinements, respectively.

The results tabulated above lead to the following conclusions for the problems investigated.

1) It is possible to employ compression rates of 2% without unduly sacrificing the accuracy of the results. For example, for a given RAM size, the use of compression enables us to tackle the problem of a truncated array which is seven or more times larger than can be solved without the use of compression.
2) We have been able to demonstrate the gain in efficiency realized via the use of the refinement procedure that has been proposed earlier in [2], but has not been tested previously to the best of our knowledge. We find that each refinement step enhances the accuracy of the result by about an order of magnitude. Another important feature of the refinement procedure is that it leads to a convergent solution even when the initial solution is considerably different from the final result.
3) Despite the considerable improvement achieved in terms of the required RAM size, the need for the storage of the uncompressed matrix on the disk is not obviated.
4) The achieved accuracy in the results does vary from problem to problem. However, this level is comparable to that realized in the process of discretization in MoM. Hence, for a required accuracy level, one must consider

the tradeoff between the compression and the order of refinement.

5) Relatively good results obtained for the examples considered herein can be attributed to the structure of the associated matrices that are diagonally dominant, and are therefore well-suited for deriving rapidly convergent solutions via iterative methods. We cannot summarily extrapolate these results to the general case of an arbitrary structure.
6) The results achieved for the compression rate are dependent on the τ chosen for these examples. However the degree of accuracy is strongly dependent on the array configuration. For instance, this accuracy is much better for the linear array than it is for the square one. For the latter, the off-diagonal blocks may have relatively significant norms and the compression procedure appears to have a larger impact on the efficiency of the solution.
7) The compression rate is dependent not only on the array configuration, but also on the number of cells as well as the number of unknowns in this cells. When optimal QR compression is reached, most nondiagonal blocks are compressed in a few units. This high degre of numerical degeneracy, as well as the high level of compression rate, can be attributed to the phasing properties of the impedance matrix terms for blocks corresponding to cells sufficiently separated.

C. Influence on the Radar Cross Section

We now present some results that show the errors in the co- and crosspolarization results for the RCS, and how these errors relate to the computed current densities with second order refinements. The results obtained for τ values between 0 and 0.1 are the same and only a very slight difference appears in the computed cross-pol results for $\tau = 0.2$.

VI. Applications

In this section we illustrate the use of the iteration algorithms, developed previously, to compute the current density distribution in a finite periodic array. Periodic arrays are typically analyzed under the assumption that the array is infinite, and that it is sufficient to deal with a single cell of the array, which requires relatively few unknowns. Our objective is to evaluate, accurately, the effects of truncation of the array on the induced current density distribution. We begin by defining the error parameter $E = ||J - Jref||/||Jref||$ where $Jref$ is the reference current density on the center element, which is far removed from the edges.

The elementary cell is shown in Fig. 3, together with its mesh and the coordinate axes. We assume that the illuminating plane wave is normally incident, at the resonant frequency, and with a polarization parallel to Ox. We first consider a linear array configuration of these elementary cells, and follow this up with the more general case of a rectangular array configuration.

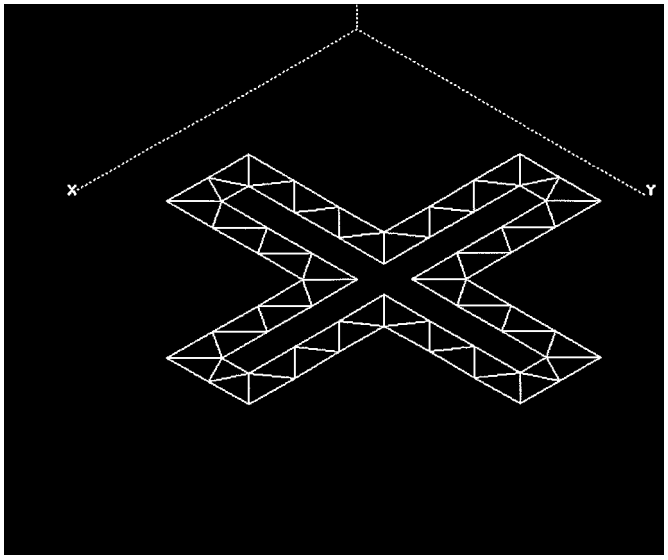

Fig. 3. Elementary cell description.

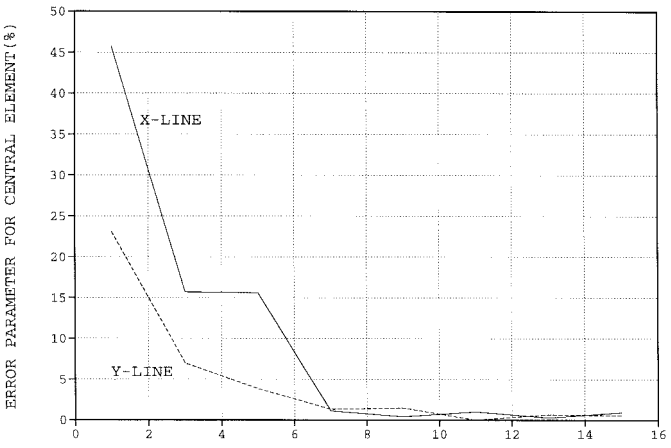

Fig. 4. Linear array: evolution of the currents on the central cell as a function of the number of neighboring cells.

A. Linear Array

We have investigated linear arrays of $1, 3, \cdots, 21$ cells, that are orientated either along Ox or Oy. The reference cell is taken to be the eleventh of the longest arrays. Fig. 4 shows the progressive evolution of the error E as defined above for the different size arrays, beginning with the smallest case. We observe that the error E becomes less than 1% once the central cell is surrounded by at least three cells on each side, regardless of whether the orientation is along Ox or Oy, and that this error can be very significant for smaller arrays. We note, further, that the edge effects are more pronounced when the polarization of the incident field is parallel to the array axis.

B. Square Array

Next we go on to discuss the numerical results we have obtained for square arrays of $1, 3 * 3, 5 * 5, \cdots, 13 * 13$ elements. Notice that in this case the number of unknowns is 10 140 whereas for our computer RAM size available was 3000. Once again, the reference cell is taken to be the central element of the largest array. Fig. 5 shows the evolution of the

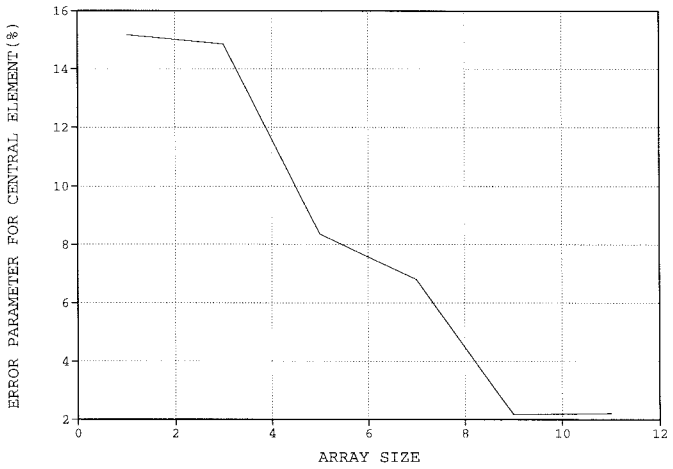

Fig. 5. Square array: evolution of the currents on the central cell versus number of neighbors.

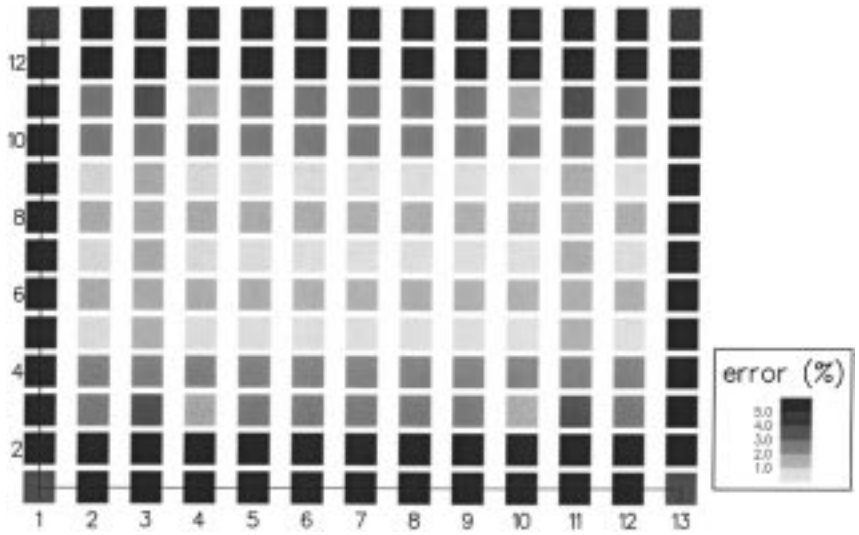

Fig. 6. Evolution of the currents on the cells on the $13 * 13$ array; reference is the central cell of the array.

error in the central cell for the different arrays. We note that the edge effects are more significant now than they were for the linear array, since it is necessary to have at least a $9 * 9$ size array for the center element to stabilize.

We could observe that a small array ($5 * 5$) behaves considerably differently from the infinite array, a result that is not totally unexpected. Next, we saw that we need to go to a zone in the center which is sizable, for instance to $11 * 5$ elements in the $13 * 13$ array for which the error is plotted in function of the cell location. We further observe that the behavior of E is different in the x and y directions; for instance, the edge effect is seen to be significant only over the edge cell in the x-direction, whereas it spreads over four cells in the orthogonal direction. Thus the edge effects can be more significant in this case than they are for the linear array.

These plots show the regions where the infinite array approximation is valid, as well as the region where the edge effects should be taken into account and corrected. We should point out that the previous results are dependent not only on the polarization of the incident field but also on the geometry of the elementary cell. Thus it is necessary, in general, to investigate the particular geometry by using an algorithm, such as the compression code, that enables one to handle a large problem.

VII. Conclusions

In this work we have successfully illustrated the application of a technique for the efficient solution of large, dense, linear systems arising in the problem of truncated arrays. The method consists of four steps, viz., use of iterative procedure; preconditioning; matrix compression; and solution refinement.

A comparative study of various iterative methods using Krylov subspaces has been carried out. It has been found that a rapid convergence rate is needed for these methods to avoid numerical error propagation in large systems. The TFQMR scheme has been found to provide superior performance over several of the other methods tested. Both the block and band preconditioning schemes of this method have been implemented and tested. They show significant improvements, as evidenced by the reduction in the number of iterations to approximately 2% of the nonpreconditioned case, with only a slight increase in the memory requirement.

The preconditioning has been found to be optimal when the diagonal blocks are identical to the selfblocks associated with the individual cells. This choice is also found to be optimal from the CPU time and memory points of view.

The process of matrix compression has been implemented in two steps. The first one, which exploits the obvious symmetries of the system, with a weak loss of generality, yields a compression ratio of about 25%. This step was followed by the QR compression of the off-diagonal blocks, and this strongly improves the compression ratio, bringing it up to approximately 2%, which represents a significant improvement in one's ability to handle a large problem size. The accuracy of the results obtained with the compressed matrix has been further improved via multiple-order refinement, achieving levels of accuracy better than 1% when compared to the reference solution.

This method should be closely compared with other frequently reported ones, such as in [10] and [11], in terms of efficiently and accuracy. Future extensions of the method to conformal arrays and to antennas mounted on complex structures that are often treated with hybrid methods are contemplated. It is anticipated that further refinements would be required to treat these more general cases, and also to tackle the general case of an arbitrary structure or scatterer.

ACKNOWLEDGMENT

The authors wish to thank Dr. Huard and Dr. Bendali from INSA for their assistance and E. Gimonet from CERT for her informatic support, also they are very grateful to the anonymous reviewers for their helpful comments on the original version of this paper.

REFERENCES

[1] R. W. Freund, "A transpose-free quasiminimal residual algorithm for non-Hermitian linear systems," *SIAM J. Sci. Comput.*, vol. J4, pp. 470–482, 1993.

[2] V. Varadarajan and R. Mittra, "Numerically efficient solution of dense linear system arising in electromagnetic scattering problems," in *Workshop Approx. Numer. Methods Solution Maxwell Equations Proc.*, Oxford, U.K., Mar. 1995.

[3] S. M. Rao, D. R. Wilton, and A. W. Glisson "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 409–418, May 1982.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[5] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming.* New York: Addison-Wesley, 1973.

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

[7] Y. Saad and M. H. Schultz, "EGMRES: A generalized minimal residual algorithm for solving nonsymetric linear systems," *SIAM J. Stat. Comput.*, vol. 7, no. 3, July 1986.

[8] R. W. Freund, "Conjugate gradient-type methods for linear systems with complex symmetric linear systems," *SIAM J. Stat. Comput.*, vol. 13, no. 1, Jan. 1992.

[9] P. G. Ciarlet, "Introduction à l'analyze numérique matricielle et à l'optimization," *Masson*, Oct. 1990.

[10] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian description," *IEEE Antennas Propagat. Mag.*, vol. 35, June 1993.

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

Jean-René Poirier received the Eng. degree from the Institut National des Sciences Appliquées, Option Génie Mathématique et Modélization, France, in 1996. He is currently working toward the Doctorat degree in applied mathematics at the Centre d'Etudes et de Recherches de Toulouse (CERT) in collaboration with the Laboratory Mathématiques pour l'Industrie et la Physique (MIP), Université Paul Sabatier, France.

Pierre Borderies was born in France in 1953. He received the Dipl.Eng. from Ecole Superieure d'Electricite, Paris, France, in 1975.

After a period of teaching in Venezuela, he joined the Centre d'Etudes et de Recherches de Toulouse, part of Office National d'Etudes et de Recherches Aérospatiales (CERT-ONERA) in 1979. Since then, he has been working as a Research Engineer in the Micowaves Department. He has worked in the fields of large reflector antennas, microwave devices, and radar targets imaging, characterization, and identification. From 1990 to 1991 he spent a sabbatical year at New York University, Farmingdale, NY. His current research interests include radiation and scattering of antennas, frequency selective surfaces, ultrawideband scattering, subsurface targets identification, natural targets scattering, and remote sensing.

R. Mittra, photograph and biography not available at the time of publication.

V. Varadarajan, photograph and biography not available at the time of publiation.