INTERNATIONAL JOURNAL OF NUMERICAL MODELLING : ELECTRONIC NETWORKS, DEVICES AND FIELDS,
Vol. 9, 335–344 (1996)

OPTIMIZATION ISSUES IN FINITE ELEMENT CODES FOR SOLVING OPEN DOMAIN 3D ELECTROMAGNETIC SCATTERING AND CONFORMAL ANTENNA PROBLEMS

A. CHATTERJEE, J. L. VOLAKIS AND L. C. KEMPEL

Radiation Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor MI 48109-2122, U.S.A.

SUMMARY

The first part of the paper presents the implementation and performance of a new absorbing boundary condition (ABC) for truncating finite element meshes. This ABC can be applied conformally to the surface of the structure for scattering and antenna radiation calculations. Consequently, the computational domain is reduced dramatically, thus allowing the simulation of much larger structures, and results are presented for three-dimensional bodies. The latter part of the paper discusses optimization issues relating to the solver's CPU speed on parallel and vector processors. It is shown that a jagged diagonal storage scheme leads to a four-fold increase in the FLOP rate of the code, and a standard matrix profile reduction algorithm substantially reduces the inter-processor communication.

1. INTRODUCTION

The finite element method (FEM) is attractive for modelling three-dimensional problems because of its $O(N)$ memory requirement and its flexibility in geometry design and modification. The $O(N)$ memory feature provides favourable scaling properties as the problem size increases. Its geometrical adaptability provides the versatility required for designing complex systems. Thus, FEM has become the method of choice for electromagnetics CAD software, and its applications continue to increase.

In spite of the obvious attractions of FEM for general purpose 3D electromagnetic field solvers, it has a few drawbacks. Since the method was initially used for solving bounded problems, its extension to open problems is not easy. In open problems we are interested in the behaviour of the fields infinitely far away from the structure of interest. However, it is impractical to extend the finite element mesh very far from the scattering or radiating structure. The normal practice is to extend the mesh a few element lengths from the body and apply boundary conditions on the mesh termination surface. These boundary conditions, which are local to the element and hence preserve system sparsity, are called absorbing boundary conditions (ABCs). Although, numerous ABCs exist for 2D problems,[1,2] ABCs for 3D vector problems are comparatively few. Peterson[3] derived vector ABCs for spherical mesh truncations; however, in most practical cases the sphere is the least economical shape of mesh truncation in terms of computer resources. In an earlier paper[4] we derived ABCs which can be enforced on surfaces conformal to the structure of interest (Figure 1), thus optimizing computational cost. Since that time we have implemented these ABCs in a general purpose, 3D finite element solver with success.[5] In this paper we present some results which demonstrate that these ABCs indeed optimize the usage of computational resources without significant degradation in accuracy.

Besides the optimization of the mesh truncation strategy, we carried out optimizations on the numerical aspects of the code. Since a finite element code involves operations on sparse matrices, indirect addressing is a necessary part of the programming task. This feature, combined with very short vector lengths for the sparse matrix, result in poor vectorization and parallellization. Essentially this is the price paid for $O(N)$ storage and improved scalability of the technique. In an earlier paper[6] we had detailed our efforts in parallellizing such a code on various distributed memory, multiprocessor architectures. The parallelization strategies that we had used were extremely successful; the code, however, ran very slowly on vector machines. In this paper we

CCC 0894–3370/96/050335–10

Received 22 April 1995
Revised 19 October 1995

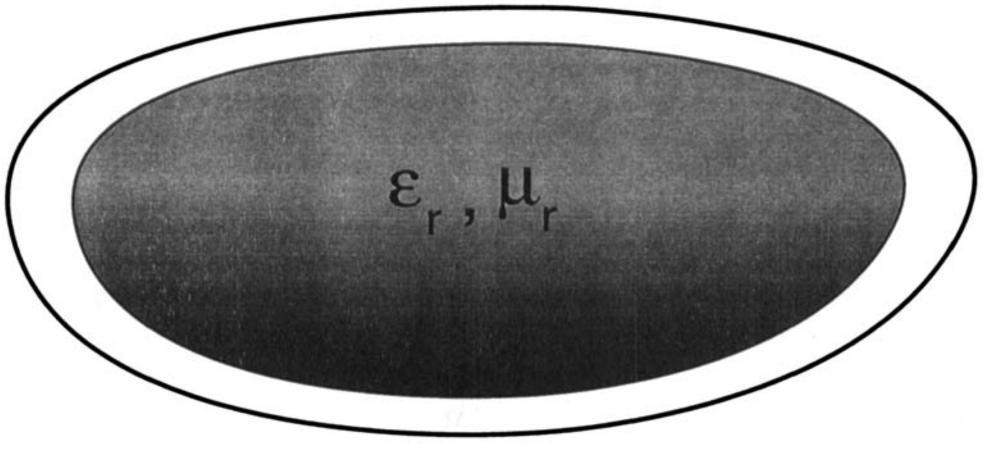


Figure 1. Scatterer enclosed in conformal mesh termination boundary

employ a novel data storage scheme for speeding up the computation on vector architectures and present a strategy for reducing inter-processor communication on multiprocessor machines.

2. CONFORMAL ABCS

2.1. *Theory*

In this Section we present a brief description of the conformal ABCs and examine their performance with respect to spherical ABCs. At first we generalize the Wilcox expansion[7] for a vector field in the Dupin co-ordinate system to read

$$\mathbf{E}(n, t_1, t_2) = \frac{e^{-jk_o n}}{4\pi\sqrt{(R_1R_2)}} \sum_{p=0}^{\infty} \frac{\mathbf{E}_p(t_1, t_2)}{(\sqrt{(R_1R_2)})^p} \tag{1}$$

where $R_i = \rho_i + n$, $i = 1, 2$ and ρ_i is the principal radius of curvature associated with the outgoing wavefront at the target. In this expression, (t_1, t_2) form the orthogonal tangential components and n is the normal component to the ABC surface. The lowest-order term in (1) represents the geometrical optics spread factor for a doubly curved wavefront and reduces to the standard Wilcox expansion[7] for a spherical wave. Next, we apply the $\hat{\mathbf{n}} \times \nabla \times$ operator to the electric field ($\mathbf{E}$) to arrive at the first-order absorbing boundary condition.

$$\hat{\mathbf{n}} \times \nabla \times \mathbf{E} - (jk_o + \kappa_m - \bar{\boldsymbol{\eta}}.)\, \mathbf{E}_t = 0 \tag{2}$$

for a conformal mesh termination boundary. In this expression, k_o is the free space wave number, the subscript t denotes the tangential component of a vector, and:

$$\kappa_m = \frac{\kappa_1 + \kappa_2}{2}$$

$$\bar{\boldsymbol{\eta}} = \kappa_1 \hat{\mathbf{t}}_1 \hat{\mathbf{t}}_1 + \kappa_2 \hat{\mathbf{t}}_2 \hat{\mathbf{t}}_2$$

where $\kappa_{1,2}$ are the two principal curvatures of the ABC surface.

The second-order ABC is obtained by using the $\hat{\mathbf{n}} \times \nabla \times$ operator once more and subsequently simplifying the resulting expression to yield

$$- (D - 2\kappa_m)\, \hat{\mathbf{n}} \times \nabla \times \mathbf{E} + \{4\kappa_m^2 - \kappa_g + D(jk_o - \bar{\boldsymbol{\eta}}\cdot) + (\bar{\boldsymbol{\eta}})^2 \cdot + \kappa_m \Delta\kappa \bar{\boldsymbol{\zeta}}\cdot\}$$

$$\mathbf{E}_t + \nabla \times \{\hat{\mathbf{n}}\, (\nabla \times \mathbf{E})_n\} + \left(jk_o + 3\kappa_m - \frac{\kappa_g}{\kappa_m} - 2\bar{\boldsymbol{\eta}}\cdot\right) \nabla_t E_n = 0 \tag{3}$$

where $\kappa_g = \kappa_1 \, \kappa_2$ is the Gaussian curvature, the subscript n denotes the normal component of a vector, and:

$$D = 2jk_o + 5\kappa_m - \frac{\kappa_g}{\kappa_m}$$
$$\Delta_\kappa = \kappa_1 - \kappa_2$$
$$(\bar{\eta})^2 \cdot \mathbf{E}_t = \kappa_1^2 E_{t_1} \hat{\mathbf{t}}_1 + \kappa_2^2 E_{t_2} \hat{\mathbf{t}}_2$$
$$\bar{\zeta} = \hat{\mathbf{t}}_1\hat{\mathbf{t}}_1 - \hat{\mathbf{t}}_2\hat{\mathbf{t}}_2 \tag{4}$$

The finite element implementation becomes simpler, and in some cases symmetric, if the term $\nabla_t E_n$ can be replaced by a double derivative. Fortunately, on considering the series expansion of the term $\hat{\mathbf{n}} \times \nabla \times \nabla_t E_n$ and simplifying, we have

$$\nabla_t \, (\nabla \cdot \mathbf{E}_t) = jk_o \nabla_t E_n \tag{5}$$

Thus we can rewrite the second-order conformal ABC as

$$(D - 2\kappa_m)\hat{\mathbf{n}} \times \nabla \times \mathbf{E} = \{4\kappa_m^2 - \kappa_g + D\,(jk_o - \bar{\eta}\cdot) + (\bar{\eta})^2 \cdot + \kappa_m \Delta\kappa \bar{\zeta}\cdot\}$$
$$\mathbf{E}_t + \nabla \times \{\hat{\mathbf{n}}(\nabla \times \mathbf{E})_n\} + \frac{1}{jk_o}\left(jk_o + 3\kappa_m - \frac{\kappa_g}{\kappa_m} - 2\bar{\eta}\cdot\right) \nabla_t(\nabla \cdot \mathbf{E}_t) \tag{6}$$

To make (2) and (6) implementable in finite element systems, the results are simplified by taking the dot product of the expression with $\mathbf{E}$, using the divergence condition, the vector wave equation and some vector identities.[5] The first-order conformal ABC in readily implementable form is given by

$$\int_{S_o} \mathbf{E} \cdot P_1(\mathbf{E}) \mathrm{d}S = (jk_o + \kappa_m) \int_{S_o} (E_{t_1}^2 + E_{t_2}^2)\, \mathrm{d}S - \int_{S_o} (\kappa_1 E_{t_1}^2 + \kappa_2 E_{t_2}^2)\, \mathrm{d}S \tag{7}$$

where S_o is the mesh truncation surface and $P_1(\mathbf{E}) = \hat{\mathbf{n}} \times \nabla \times \mathbf{E}$, with the subscript denoting the order of the ABC.

The second-order ABC reduces to

$$\int_{S_o} \mathbf{E} \cdot P_2(\mathbf{E})\, \mathrm{d}S = \int_{S_o} (\alpha_1 E_{t_1}^2 + \alpha_2 E_{t_2}^2)\, \mathrm{d}S + \int_{S_o} \beta(\nabla \times \mathbf{E})_n^2\, \mathrm{d}S$$
$$- \int_{S_o} (\nabla \cdot \mathbf{E}_t)\, \{\nabla \cdot (\gamma \cdot \mathbf{E})_t\}\, \mathrm{d}S \tag{8}$$

where the tensors $\bar{\alpha}$, $\bar{\gamma}$ and the scalar β are given by

$$\bar{\alpha} = \frac{1}{D - 2\kappa_m} [\{4\kappa_m^2 - \kappa_g + D(jk_o - \kappa_1) + \kappa_1^2\}\, \hat{\mathbf{t}}_1\hat{\mathbf{t}}_1$$
$$+ \{4\kappa_m^2 - \kappa_g + D(jk_o - \kappa_2) + \kappa_2^2\}\hat{\mathbf{t}}_2\hat{\mathbf{t}}_2]$$
$$\bar{\gamma} = \frac{1}{jk_o(D - 2\kappa_m)} \left[\left(jk_o + 3\kappa_m - \frac{\kappa_g}{\kappa_m} - 2\kappa_1\right) \hat{\mathbf{t}}_1\hat{\mathbf{t}}_1\right. \tag{9}$$
$$\left. + \,(jk_o + 3\kappa_m - \frac{\kappa_g}{\kappa_m} - 2\kappa_2)\, \hat{\mathbf{t}}_2\hat{\mathbf{t}}_2\right]$$
$$\beta = \frac{1}{D - 2\kappa_m}$$

It should be remarked that the normal component of each surface edge must be made continuous across inter-element boundaries (triangular patches in our case) for the contour integral associated with the third term in (8) to vanish. Moreover, it can be shown that the first-order ABC (7) is always symmetric whereas the second-order ABC (8) is symmetric only when $\kappa_1 = \kappa_2$ on the boundary surface or when the surface is cylindrical and linear edge bases are employed. For a detailed analysis of symmetry considerations, the reader is referred to Reference 5.

2.2. *Results*

In this Section, we present a few results for validating the conformal ABCs derived in the previous Section. A complete description of the numerous geometries that were validated using these ABCs can be found in Reference 5. In Figure 2 we plot the backscatter pattern for a PEC cylindrical inlet having a diameter of 1·24λ and a height of 1·875λ. The mesh termination surface is conformal to the scattering structure and is placed 0·45λ from the target. The resulting problem had only 144,392 unknowns compared to the 191,788 unknowns for a rectangular truncation scheme. A spherical mesh termination would have swelled to about 265,000 unknowns, sampling density and outer boundary distance remaining the same. Thus, use of the conformal ABC reduced the problem size by about 45 per cent and computation time by a similar, if not greater, amount. The savings in computational resources are quite significant even when we compare the rectangular and cylindrical termination schemes – a 25 per cent reduction in problem size and a similar decrease in computation time.

The next issue is about accuracy. We have tested this geometry extensively by increasing the distance of the absorbing boundary, but the far-field results remained unchanged. The discrepancy between the measured data, the BOR and the FE-ABC computation is quite puzzling. However, confidence in the FE-ABC data may be found from the fact that a similar modelling effort by Shankar[10] using FDTD follows our computation very closely. It should also be remarked that the far-field pattern for different mesh truncation surfaces – rectangular box and cylindrical box – were exactly the same; the inaccuracy, therefore, cannot be attributed to the quality of the conformal ABC.

The next geometry presented in this paper is unique in its own way. A conesphere is basically a hemisphere attached to a cone. It is a difficult geometry to mesh since a surface singularity exists at the tip of the cone. The singularity can be removed in two ways: (i) by creating a small region near the tip and detaching it from the surface; or (ii) by chopping off a small part near the tip of the cone. The second option inevitably leads to small inaccuracies for backscatter from the conical tip; however, we chose this option since the conical angle in our tested geometry was

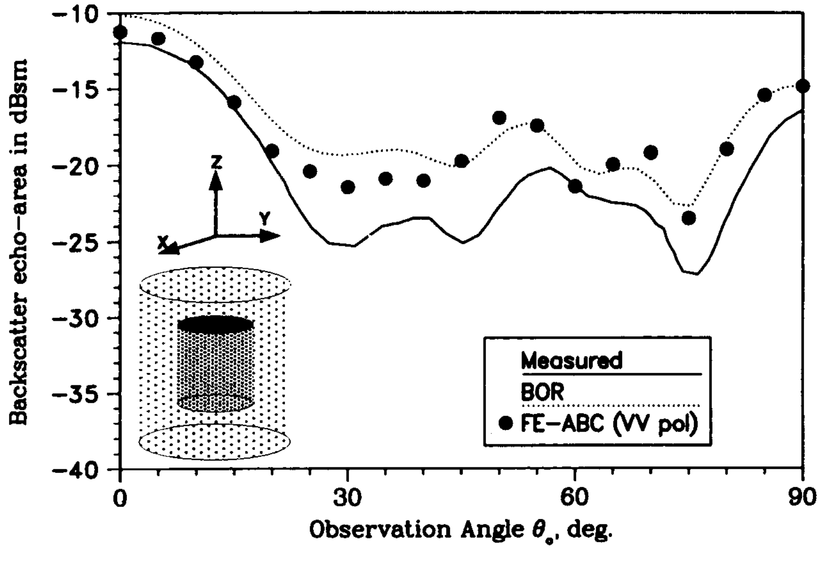


Figure 2. Backscatter pattern of a perfectly conducting cylindrical inlet (diameter = 1·25λ, height = 1·875λ) for VV polarization. Black dots indicate computed values, the solid line is measured data[8] and the dotted line represents reference data from a body of revolution code.[9] Mesh termination surface is a circular cylinder

extremely small (around 7°) and the mesh generator failed to mesh the first case on numerous occasions. In Figure 3 we plot the backscatter patterns of a 4·5λ long conesphere having a radius of 0·5λ for $\theta\theta$ and $\phi\phi$ polarizations. The mesh truncation surface is a rectangular box placed 0·4λ from the surface of the conesphere. The far-field results compare extremely well with computations from a body of revolution code.[11]

The proposed ABCs have also been implemented for truncating finite element meshes in computing the radiation and scattering patterns of printed antennas on conformal platforms as illustrated in Figure 4. Various patch configurations situated on cylindrical platforms were considered for the purpose of examining the performance of ABCs for this application. Among those studied, we present the analysis of a 2 cm × 3 cm patch antenna printed atop a metallic cavity which is filled with a 5 cm × 6 cm × 0·07874 cm substrate having a dielectric constant of ϵ_r = 2·17. This cavity is recessed in a metallic cylinder (whose infinite dimension is along the z axis) with a radius of 15·28 cm, and the scattering and radiation calculations for this patch were carried out at 3 GHz. The second-order vector ABC was placed $\tau\lambda_o$ from the cavity aperture while the lateral walls of the ABC were placed 0·5λ_o from the cavity aperture. The monostatic scattering with observation in the x-y plane due to an H_z-polarized plane wave is shown in Figure 5 for different values of τ. The FE-ABC method recovers the creeping wave interactions even when

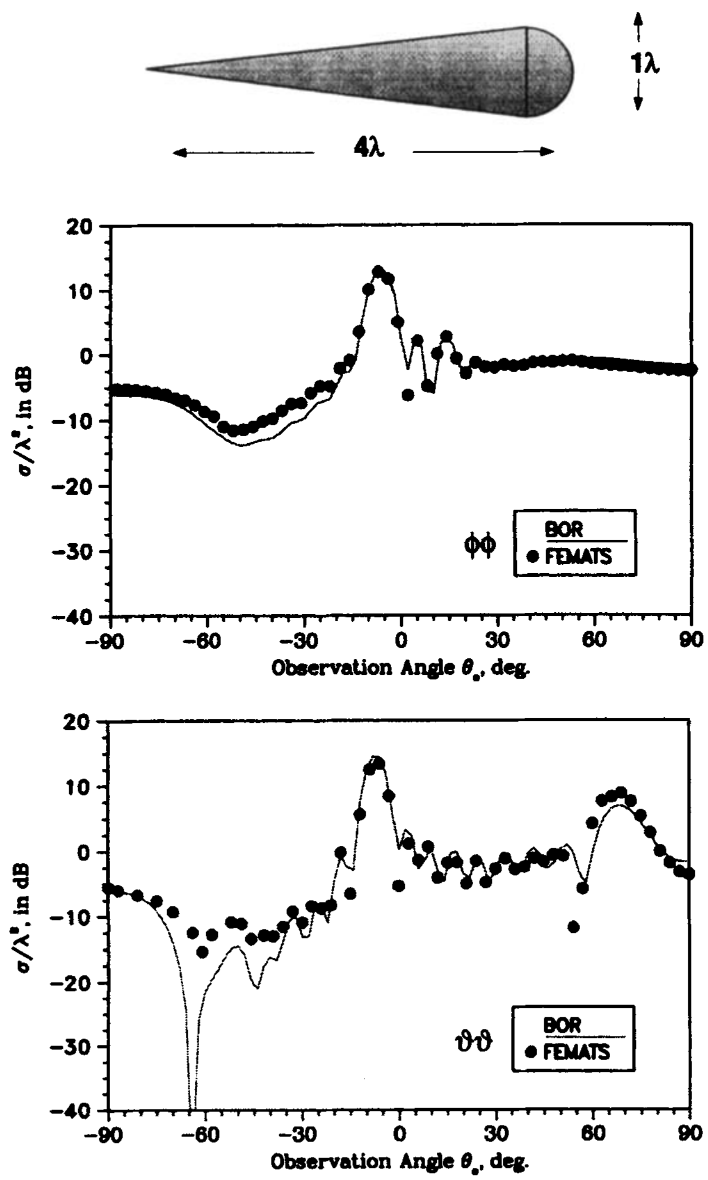


Figure 3. Backscatter pattern of a perfectly conducting conesphere for $\phi\phi$ and $\theta\theta$ polarizations. Black dots indicate computed values using the FE-ABC code (referred to as FEMATS) and the solid line represents data from a body of revolution code.[11] Mesh termination surface is a rectangular box

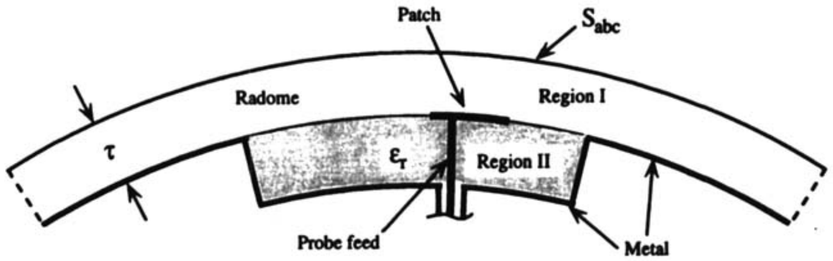


Figure 4. Cavity-backed patch antenna with ABC mesh termination

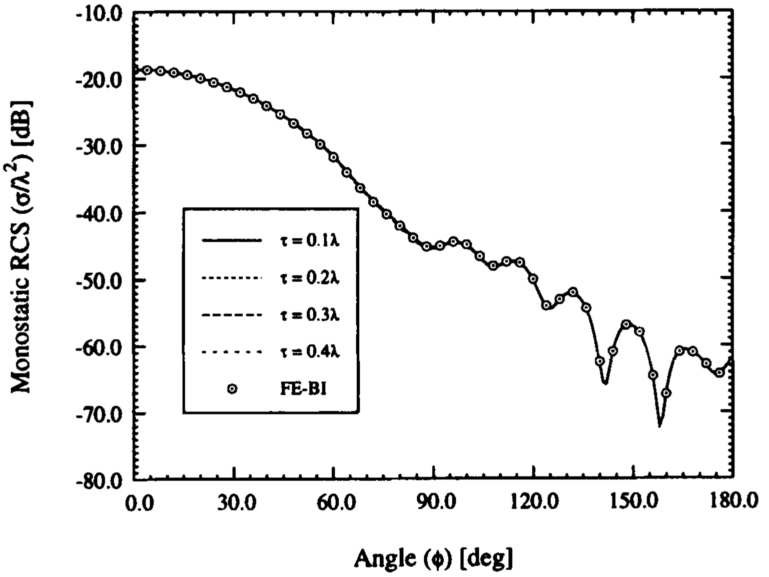


Figure 5. Comparison of FE-ABC and FE-BI methods for computing the H_z-polarized monostatic scattering of a cavity-backed patch. The inset illustrates the convergence with respect to ABC displacement for 0° to 10° of the pattern

the ABC surface is quite close to the cavity aperture. The H-plane antenna pattern for an axially polarized patch is shown in Figure 6, where the probe feed is placed at $\phi_f = 0°$ and $z_f = -0{\cdot}375$ cm (with $\phi_f = 0°$ and $z_f = 0$ corresponding to the centre of the patch). As in the case of scattering, the radiation pattern calculated via the FE-ABC method is seen to be in excellent

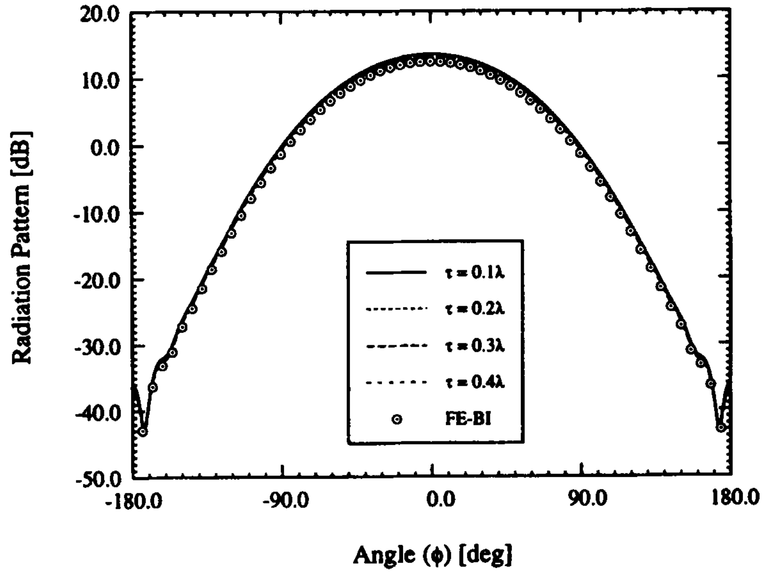


Figure 6. Convergence of FE-ABC method for computing H-plane radiation pattern of a cavity-backed axially polarized patch. The reference data are provided by a rigorous FE-BI formulation for the same cavity-backed antenna

agreement with the pattern computed by the more rigorous finite element-boundary integral (FE-BI)[12] approach even when the mesh is terminated only 0·3λ from the aperture. We remark, though, that the FE-ABC method is considerably more flexible than the FE-BI approach and may be used for coated and doubly curved platforms.

3. VECTORIZATION/PARALLELLIZATION STRATEGIES

Since our focus is on solving large problems, the code must be optimized to run fast on vector and parallel architectures. In Reference 6 we detailed our parallellization strategy and presented results on speed-up and inter-processor communication. However, the performance of the code on vector processors was not very encouraging. In the subsequent Sections we outline our optimization scheme for vector computers and present techniques for reducing inter-processor communication on distributed memory architectures.

3.1. *Vector optimization*

Since a sparse matrix has a very small number of non-zeros per row by definition and only the innermost loops are vectorizable, it is difficult to obtain good vector performance from such codes. Further, indirect addressing is an inherent part of sparse data structures – a feature which allows us to exploit the $O(N)$ storage characteristic but reduces speed on vector machines. Therefore, there are two main problems which limit the vectorizability of a sparse matrix code – short vector lengths and indirect addressing. The latter problem cannot be corrected but the first bottleneck can be removed. This is done by storing the matrix in a different format such that the vector lengths are approximately equal to the order of the system being solved. In the traditional storage system – compressed sparse row (CSR) format – the non-zeros of the matrix and their corresponding column numbers are stored in a long complex and integer vector, respectively, with another short integer vector to store the number of non-zeros per row. However, this does not permit vectorizability since the average vector length is very small – 16 in our case. The ITPACK format[13] alleviates the short vector length problem by storing the entire matrix in a rectangular block. In this block, the number of rows equals the row count of the original matrix and the number of columns equals the maximum number of non-zeros in a row of the matrix; rows containing fewer non-zero elements are padded with zeros. This scheme works very well for matrices where the average non-zeros per row are approximately the same. In our case, this storage technique is not very beneficial since approximately 30 per cent of the space is lost in zero padding.

The storage format that works best for our type of matrix is called the *jagged diagonal* format.[14] The rows are ordered in decreasing order of the number of non-zeros per row. The rows containing the maximum number of non-zero entries are thus placed at the top of the matrix and the rows with the minimum non-zero entries are shuffled to the bottom. In the actual storage scheme, the leftmost elements of each row are stored as a dense vector with an additional vector indicating the column numbers of each element. The matrix is thus stored as a collection of vectors of decreasing length. The inner loop of the matrix–vector multiplication routine traverses the entire length of a jagged diagonal, which can be of the order of the system being solved. This feature greatly enhances vectorization. The storage requirement of the above format can be made to be the same as the previously mentioned CSR format through careful programming. The altered matrix–vector multiplication routine then runs at around 275 MFLOPS on a Cray C-90, whereas the older code with CSR storage peaked at 60 MFLOPS. The dot product reaches speeds of 550 MFLOPS and the vector updates execute at 600 MFLOPS. It must be mentioned that the CRAY C-90 is a substantially faster machine than the Cray YMP but the CSR formatted matrix–vector multiplication routine runs about four times slower on the C-90. Therefore, we can reliably state that the method of *jagged diagonals* is the best sparse matrix storage scheme in terms of computer storage and vectorizability. The still slower execution speeds of the matrix–vector multiply compared with the vector update is due to the indirect addressing in the inner loop which causes memory contention.

3.2. *Reduction of processor communication*

In the previous Section we discussed optimization from the viewpoint of a vector processor. In this Section we propose a scheme for reducing interprocessor communication on multiprocessor architectures. In Reference 6 we outlined our success in parallellizing the computationally intensive portions of a finite element code on distributed memory architectures. It was also pointed out that further speed-ups could be achieved only through reducing data communication among the various processors.

The majority of processor time in a finite element code is spent in the equation solver. In our code, we employ an iterative solver since it preserves the sparsity of the finite element matrix, making minimal demands on computer storage. In the biconjugate-gradient algorithm, there are principally two operations in which the most intensive communication takes place. The first is the sparse matrix–vector multiplication and the second is the search vector update at the end of each iteration.

In the matrix–vector multiply, each processor computes a block of the result vector by multiplying the corresponding block of rows of the sparse matrix with the operand vector. Since the operand vector is distributed among the processors, data communication is required. Each processor does two things: (i) it sends out a request for those matrix entries it does not own but needs for performing the multiplication; and (ii) sends out some of the matrix entries it owns on request from other processors. The communication pattern is determined by the sparsity structure of the matrix, which in our case is derived from an unstructured mesh. Therefore the communication pattern is unstructured and irregular. However, on reordering the matrix using a standard profile reduction algorithm (part (b) of Figure 7), the coefficient matrix becomes banded. As a result of the banded structure of the matrix, communication should occur only between adjacent processors. In fact, by storing a few extra matrix entries in each processor, inter-processor communication can be removed altogether in the matrix–vector multiplication phase. This should result in substantial speed-up of the algorithm on multiprocessor machines. However, the time taken for communication due to the vector update in the last stage of the iterative algorithm still remains the same.

4. CONCLUSION

In this paper we have talked about optimization strategies that were employed to improve our finite element code from the algorithmic and the numerical point of view. We have achieved notable success on both fronts.

The problem size was reduced by a significant amount owing to the use of conformal ABCs,

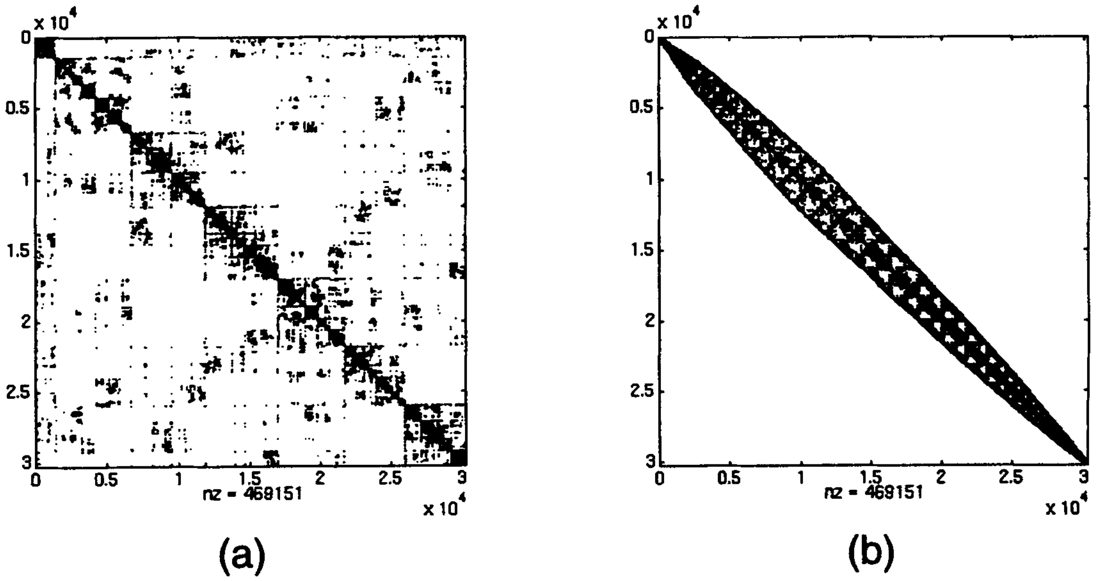


Figure 7. (a) Original sparse matrix; (b) re-ordered sparse matrix using a profile reduction algorithm)

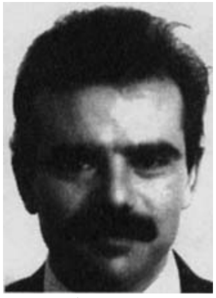
and the savings are only going to increase as the problem size gets larger. Higher-order ABCs may enable us to bring the mesh termination surface even closer to the target, enabling us to do larger problems with the available computer resources. The numerical aspect is also important for addressing the utility and the feasibility issues for solving practical 3D problems in a reasonable amount of time. As processor speeds increase and parallel architectures mature, the speed and performance of the code will vastly improve. Therefore, issues concerning the performance of the finite element–ABC technique and its implementation on large-scale computing architectures will continue to be important in the years to come.

REFERENCES

1. A. Bayliss and E. Turkel, 'Radiation boundary conditions for wave-like equations', *Commun. Pure Appl. Math.*, **33**, 707–725 (1980).
2. B. Engquist and A. Majda, 'Absorbing boundary conditions for the numerical simulation of waves', *Math. Comput.*, **31**, 629–651 (1977).
3. A. F. Peterson, 'Absorbing boundary conditions for the vector wave equation', *Microw. Opt. Technol. Lett.*, **1**, 62–64 (1988).
4. A. Chatterjee and J. L. Volakis, 'Conformal absorbing boundary conditions for the vector wave equation', *Microw. Opt. Technol. Lett.*, **6**, (16), 886–889 (1993).
5. A. Chatterjee and J. L. Volakis, 'Conformal absorbing boundary conditions for 3-d problems: Derivation and applications', *IEEE Trans. Antennas Propag.*, **43**, 860–866 (1994).
6. A. Chatterjee, J. L. Volakis and D. Windheiser, 'Parallel computation of 3D electromagnetic scattering using finite elements', *Int. J. Numer. Model.* **7**, 329–342 (1994).
7. C. H. Wilcox, 'An expansion theorem for electromagnetic fields', *Commun. Pure Appl. Math.*, **9**, 115–134 (1956).
8. M. Schuh, A. Woo, M. Sanders and H. T. G. Wang, 'Radar cross-section measurement data of four small cavities', Technical Report 108782, NASA Ames, November 1993.
9. A. Glisson and D. R. Wilton, 'Simple and efficient numerical techniques for treating bodies of revolution', Technical Report 105, University of Mississippi, 1982.
10. V. Shankar, W. F. Hall, A. Mohammedian and C. Rowell, 'Development of a finite-volume, time-domain solver for Maxwell's equations', *Rockwell Technical Report* prepared for NASA/NDC under contract N62269-90-C-0257, May 1993.
11. J. M. Putnam and L. N. Medgyesi-Mitschang, 'Combined field integral equation formulation for axially inhomogeneous bodies of revolution', Technical Report MDC QA003, McDonnel Douglas Research Labs, December 1987.
12. L. C. Kempel and J. L. Volakis, 'Scattering by cavity-backed antennas on a circular cylinder', *IEEE Trans. Antennas Propag.*, **42**, 1268–1279 (1994).
13. D. R. Kincaid and T. C. Oppe, 'ITPACK on supercomputers', *Numer. Methods, Lect. Notes Math.*, **1005**, 151–161 (1982).
14. E. Anderson and Y. Saad, 'Solving sparse triangular linear systems on parallel computers', *Int. J. High Speed Comput.*, **1**, *(1), 73–95 (1989).*

Authors' biographies

Arindam Chatterjee received the B.Tech. degree in electronics engineering from the Indian Institute of Technology, Kharagpur, in 1989. He graduated with the M.S. and Ph.D. in electrical engineering from the University of Michigan, Ann Arbor, in 1991 and 1994, respectively. He was a graduate research assistant at the Radiation Laboratory of the University of Michigan from July 1989 to August 1994. From September to December 1994 he worked in the same institution as a post-doctoral research fellow. Since January 1995, he has been employed at Compact Software. His current interests include modelling of microwave circuits in packaged and open environments, and volumetric mesh generation techniques.



John L. Volakis was born on 13 May 1956 in Chios, Greece, where he also attended the Gymnasium of Males. He obtained his B.E. degree, summa cum laude, in 1978 from Youngstown State University, Youngstown, OH, the M.Sc. in 1979 from the Ohio State University, Columbus, Ohio, and the Ph.D. degree in 1982, also from the Ohio State University. He has been with the University of Michigan, Ann Arbor, MI, since 1984 where he is now a Professor in the Department of Electrical Engineering and Computer Science (EECS). From 1982 to 1984 he was with Rockwell International, Aircraft Division, and during 1978–1982 he was a graduate research associate at the Ohio State University ElectroScience Laboratory. His primary research deals with the development and application of analytical and numerical techniques to large-scale scattering, printed antennas and bioelectromagnetics. His work has emphasized both numerical and diffraction methods, and along with his students, he has made pioneering contributions to these areas. Dr Volakis had published over 100 refereed journal articles, more than 100 conference papers, several book chapters on numerical methods and co-authored the book *Approximate Boundary Conditions in Electromagnetics* (IEEE Press, 1995). In collaboration with his students, he has also written a number of scattering and antenna analysis software for the industry and government services. In 1993 he received the University of Michigan EECS Department Research Excellence Award. Dr Volakis has served on various posts of the local IEEE AP/MTT/ED Southeastern Michigan Chapter from 1985 to 1988, as an associate editor of the *IEEE Transactions on Antennas and Propagation* from 1988 to 1992, chaired the 1993 IEEE Antennas and Propagation Society Symposium and Radio Science Meeting, and is a current member of the AdCom for the

IEEE Antennas and Propagation Society. He now serves as an associate editor for *Radio Science*, the *J. of Electromagn. Waves and Appl.* and the *IEEE Antennas and Propagation Society Magazine*. He is a Fellow of the IEEE and a member of Sigma Xi, Tau Beta Pi, Phi Kappa Phi, and Commission B of URSI.

Leo C. Kempel earned his BSEE (1989) at the University of Cincinnati, and his MSEE (1990) and Ph.D (1994) at the University of Michigan, all in electrical engineering. From August 1989 to August 1994 he was a graduate student research assistant and then postdoctoral fellow at the University of Michigan's Radiation Laboratory, where he performed research in computational and theoretical electromagnetics. Dr Kempel is now a senior electrical engineer in the Electromagnetic Applications Group of Mission Research Corporation located in Valparaiso FL. His current research interests include computational electromagnetics, ultra-wideband radar, and high power microwave applications. He is a member of the IEEE, Tau Beta Pi and Eta Kappa NU.