Progress In Electromagnetics Research, Vol. 108, 1–21, 2010

PARALLEL MOM-PO METHOD WITH OUT-OF-CORE TECHNIQUE FOR ANALYSIS OF COMPLEX ARRAYS ON ELECTRICALLY LARGE PLATFORMS

X.-W. Zhao, Y. Zhang, and H.-W. Zhang

School of Electronic Engineering
Xidian University
No. 2 South Taibai Road, Xi'an, Shaanxi 710071, China

D. G. Doñoro

Department of Signal Theory and Communications
University Carlos III of Madrid
Leganés, Madrid 28911, Spain

S.-W. Ting

Department of Electrical and Electronics Engineering
University of Macau
Macao, China

T. K. Sarkar

Department of Electrical Engineering and Computer Science
Syracuse University
Syracuse, NY 13244, USA

C.-H. Liang

School of Electronic Engineering
Xidian University
No. 2 South Taibai Road, Xi'an, Shaanxi 710071, China

Received 21 July 2010, Accepted 23 August 2010, Scheduled 28 August 2010
Corresponding author: X.-W. Zhao (xdzxw@126.com).

Abstract—A Message Passing Interface (MPI) parallel implementation of a hybrid solver that combines the Method of Moments (MoM) with higher order basis functions and Physical Optics (PO) has been successfully used to solve a challenging problem including a 2160-slot waveguide array on an airplane with a maximum dimension larger than 1000 wavelengths. The block-partitioned scheme for the large dense MoM matrix combined with the process-cyclic scheme for the PO discretized triangles is designed to achieve excellent load balance and high parallel efficiency. To break the limitation of physical memory, the parallel out-of-core technique is introduced to tackle large dense systems generated using the MoM formulation. This research provides a solution with reasonable accuracy for solving large on-board antenna problems but has very low memory usage.

1. INTRODUCTION

With the rapid development of computer technology, especially with the advent of multi-core technology in recent years, parallel computation is playing a more and more important role in computational electromagnetics (CEM). The constant development of the computer technology sets objectives more and more ambitious for the electromagnetic solvers, as described by Taboada et al. in [1]. Parallel technology has been used in various CEM methods, such as the Method of Moments (MoM) [2–5], Fast Multipole Method (FMM) [1, 6, 7], Finite Element Method (FEM) [8] and Finite-Difference Time-Domain method (FDTD) [9–10].

On the other hand, there are many challenging realistic problems that urgently need to be overcome. One typical problem is the characteristic analysis of a complex antenna mounted on an electrically large platform [11]. Many different electromagnetic computational approaches have been applied for the solution of this class of problem. One traditional and widely adopted method is the MoM [12]. At this moment, the parallel higher order MoM is capable of solving electrically large problems of several hundred wavelengths in dimensions with up to problems with million level unknowns [4]. However, it is still very difficult nowadays to analyze radiation from on-board antennas by MoM alone when the electrical sizes of the platforms become larger and larger (e.g., thousands of wavelengths) as the operating frequency increases. With this background, parallel hybrid techniques that combine MoM with high-frequency techniques, either ray-based or current-based [13], have a potential to address this class of problem. MoM hybrid with current-based high-frequency asymptotic methods, such as the Physical Optics (PO) [14–17], is a preferable choice since MoM is based on currents as well.

Recently, a parallel in-core hybrid MoM-PO method was proposed for analysis of phased array antennas on electrically large platforms by the authors in [18]. The phased array in the MoM region could not be very large due to the lack of physical memory (RAM), which is a common drawback of the in-core method [2]. Specifically, the array in [18] was a relatively small slotted waveguide array with 10 × 10 elements. In practice, however, an on-board array usually contains thousands of elements.

To deal with very large and complex arrays in the absence of sufficient RAM, this paper presents a parallel out-of-core hybrid MoM-PO method, which is capable of breaking the limitation of RAM. The MoM is hybridized with PO by iterating the voltage matrix of the MoM matrix equation [19]. The out-of-core technique is utilized to tackle large dense systems generated by MoM, and can simulate large and complex arrays by occupying a small amount of RAM in reasonable time. In addition, Higher Order Basis functions (HOBs) [20, 21] with polynomial forms are employed in the MoM region to sharply reduce the number of unknowns compared with the low-order basis functions, in specific, the piecewise Rao-Wilton-Glisson basis functions (RWGs) [22].

The large dense MoM matrix is divided into a number of smaller block matrices that are nearly equal in size and distributed among all participating processes. The strategy of distributing the blocks is designed appropriately according to the parallel lower/upper (LU) decomposition solver to minimize the communication between processes. Note that both the parallel matrix filling and parallel matrix equation solving schemes are combined with the out-of-core technique. For the PO method, the discretized elements on the surface of the PO lit region are partitioned equally through a process-cyclic scheme. For a hybrid MoM-PO method, the corresponding parallelization schemes are integrated to obtain good load balance and high parallel efficiency. The proposed method is validated by utilizing some representative and challenging examples.

The paper is organized as follows. In Section 2, we briefly review the basics of the hybrid MoM-PO method based on the iteration of the MoM voltage matrix. In Section 3, we present the parallelization methodology. Numerical results are presented in Section 4, followed by the conclusion.

2. ITERATIVE VOLTAGE-BASED HIGHER ORDER MOM-PO METHOD

For completeness, the integral equations and the hybrid scheme are briefly reviewed in this section. The readers are referred to [2, 19] for an in-depth discussion of the material.

2.1. Integral Equations

The hybrid MoM-PO method is implemented using the surface integral equations. To construct the hybrid solution, the structure under consideration is decomposed into two parts: a MoM region and a PO region. As the two regions are separated only in the basis-functions space, they can be physically connected or even overlap [15]. The proposed hybrid method is able to deal with the connected cases [19], but it can be more efficient for well separated ones where the two regions are separated by a sufficient distance.

In the MoM region, both magnetic and electric currents are used for electromagnetic modeling of multiple materials. The integral equation employed is the Poggio-Miller-Chang-Harrington-Wu (PMCHW) formulation [2]. Flexible geometric modeling is achieved by using truncated cones for wires (feeding pins) and bilinear patches to characterize surfaces in the MoM region. Efficient approximation for the unknown currents is obtained by using the Higher Order Basis functions (HOBs) consisting of combinations of polynomials [2].

The PO region is generally an electrically large and smooth metallic structure, such as an airplane. For a metallic surface, only electric current is applied. The Magnetic Field Integral Equation (MFIE) is used in the PO region [14]. Note that the mutual interaction [14] and the edge diffraction effects in the PO region are not taken into account in the proposed MoM-PO method. The surface of the PO region is discretized into planar triangular facets. This is different from the quadrilateral discretization of the MoM surface. However, the hybridization and parallelization schemes in the following sections are independent of the discretization, which enable one to utilize triangular or quadrilateral facets in the MoM region and/or PO region.

2.2. Iterative Voltage-based Hybrid MoM-PO Method

The commonly used approach to hybridize MoM and PO is based on the generation of the complete hybrid MoM-PO system matrix equation, which includes the self-impedance matrices of the MoM and PO regions, and the mutual impedance matrices between the MoM and PO regions. The self-impedance matrix of the PO region can be very sparse since the mutual interactions between the PO currents are neglected in the proposed MoM-PO approach to maximally reduce the storage and computational cost of the hybrid method as compared with the pure MoM [15]. However, when the numbers of unknowns in both the MoM and PO regions are very large, the self-impedance matrix of the MoM, the mutual impedance matrices and the relevant matrix-vector multiplications still incur huge memory requirement and computational complexity [14].

To make the hybrid MoM-PO method more efficient, the method of iterating the voltage matrix of the MoM equation with the PO fields [19] rather than the direct solution of the complete matrix equation [14, 15] is implemented in this paper. When the MoM and PO regions are separated by a sufficient distance so that they are loosely coupled, the iteration procedure can be more efficient than solution of the complete matrix equation [14].

The general procedure of the iterative hybridization scheme includes three steps, which are similar to that in [19]. The distinct difference between the proposed hybrid method and the method in [19] is that the former introduces magnetic currents for including dielectric structures in the MoM region, while the latter can only simulate metallic objects.

The iteration of the MoM voltage matrix is terminated when the following criterion

$$\left\|\mathbf{I}_{\mathrm{iter}+1}^{\mathrm{MoM}} - \mathbf{I}_{\mathrm{iter}}^{\mathrm{MoM}}\right\| / \left\|\mathbf{I}_{\mathrm{iter}}^{\mathrm{MoM}}\right\| < \varepsilon \tag{1}$$

is satisfied, where iter denotes the iteration count, $\mathbf{I}_{\mathrm{iter}}^{\mathrm{MoM}}$ is the expansion coefficient vector of the MoM basis functions, ε is the user-specified convergence parameter, and $\|\cdot\|$ denotes an appropriate vector norm. For a loosely coupled case, the change in current on a given iteration is very small, and hence ε should be set to a small number.

The current distributions in the MoM and PO regions can be obtained once the iterative process is completed. Then the total fields radiated by the currents can be found.

3. PARALLEL SCHEMES FOR OUT-OF-CORE MOM-PO METHOD

The parallel hybrid MoM-PO method mainly consists of two parts, parallelization of MoM and parallelization of PO. The time consuming parts in MoM lie in the matrix filling and its solution, whereas those of PO lie in the PO integrals over all discretized elements. Consequently, we focus on these procedures in the parallelization methodology.

3.1. Parallelization of MoM with Out-of-core Technique

The parallelization of the MoM solution involves two steps. The first step is the matrix filling and the second step is the solution of the matrix equation. Both of these must be handled efficiently. Furthermore, efficient parallel matrix filling for MoM with HOBs introduces new challenges and is quite different from the procedure used in a MoM formulation using the traditional subdomain basis functions, e.g., RWGs [5].

To parallelize the solution of the large dense matrix in a MoM problem, typically one needs to divide the matrix between processes in such a way that two important conditions are fulfilled: each process should store approximately the same amount of data, and the computational load should be equally distributed among the processes that run on different nodes.

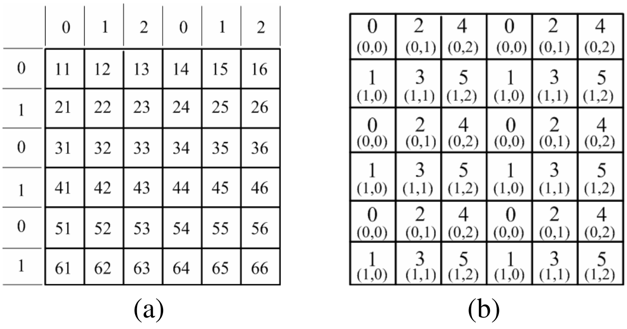
1) *Block Matrices:* The parallel LU decomposition based on the ScaLAPACK [3] library package is employed as the solver and the block storage scheme is designed accordingly.

For explanation purposes, we rewrite the MoM matrix equation in a general form as

$$\mathbf{AX} = \mathbf{B}, \tag{2}$$

where $\mathbf{A}$ denotes the complex dense matrix, $\mathbf{X}$ is the unknown vector to be determined and $\mathbf{B}$ denotes the given source vector.

Assume that the matrix $\mathbf{A}$ is divided into 6 × 6 blocks, which are distributed to 6 processes in a 2 × 3 process grid, as illustrated in Figure 1(a). Figure 1(b) shows to which process the blocks of $\mathbf{A}$ are distributed using ScaLAPACK's distribution methodology.



	0	1	2	0	1	2
0	11	12	13	14	15	16
1	21	22	23	24	25	26
0	31	32	33	34	35	36
1	41	42	43	44	45	46
0	51	52	53	54	55	56
1	61	62	63	64	65	66

(a)

0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)
0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)
0 (0,0)	2 (0,1)	4 (0,2)	0 (0,0)	2 (0,1)	4 (0,2)
1 (1,0)	3 (1,1)	5 (1,2)	1 (1,0)	3 (1,1)	5 (1,2)

(b)

Figure 1. Block-cyclic distribution of a matrix as performed by ScaLAPACK: (a) a matrix consisting of 6 × 6 blocks and (b) rank and coordinates of each process owning the corresponding blocks in (a).

In Figure 1(a), the outermost numbers denote the row and column indices of the process coordinates. The top and bottom numbers in any block of Figure 1(b) denote the process rank and the process coordinate of a certain process, respectively, corresponding to the block of the matrix shown in Figure 1(a). By varying the dimensions of the blocks of **A** and those of the process grid, different mappings can be obtained. This scheme can be referred to as a block-cyclic distribution procedure.

For ScaLAPACK, one needs to distribute the matrix over a two-dimensional process grid, but it can be done in different ways and one can, more or less, influence the way in which the data related to the matrix are distributed.

Note that the storage required for the vectors **X** and **B** is negligible compared with that of the large dense matrix **A**. Therefore, the entire vectors can be stored in each process.

2) *Parallel In-core Matrix Filling:* Parallel in-core matrix filling scheme [18] is reviewed to support the parallel out-of-core matrix filling scheme.

The impedance matrix is constructed by looping over the number of geometric elements (number of wires and plates) and performing the calculation of the elements of the impedance matrix. The parallel filling of the matrix could be the same regardless of the choice of the basis functions. However, the matrix filling scheme will be most efficient if the characteristics of the basis functions are taken into account. An efficient matrix filling scheme for a MoM using RWGs was presented in [5]. Different steps than described in [5] need to be taken to deal with HOBs.

There is an advantage in using the polynomial basis functions because the intermediate results obtained in evaluating the elements of the impedance matrix for lower order can be used in the computation of the elements when using higher order polynomials [2]. This advantage improves the efficiency of the matrix filling for HOBs when employed over wires and quadrilaterals and can be implemented quite easily in both the serial and parallel codes.

For parallel matrix filling, an additional improvement can be made to further increase the efficiency of the code. The objective is to eliminate redundant calculations for each process. For the most efficient code, this concept can be applied regardless of the choice of the basis functions. However, the specific details for implementing this are quite different for different basis functions. In the parallel scheme using HOBs, redundant calculations related to the evaluation of the potential functions can be avoided within each process by using a flag, set true or false, for each order of the polynomial on a geometric element. The corresponding pseudocode was given in [3].

When dealing with the right-hand side of the matrix equation in MoM, i.e., **B** in (2), the parallel matrix filling algorithm is much easier to design as compared to the parallel filling algorithm for the impedance matrix [2].

Load balancing is critical to obtain an efficient operation of a parallel code. The parallel matrix filling scheme is able to achieve the good load balancing. Little communication between processes is necessary during the matrix filling, and parallel speedup can be carefully tracked to ensure proper implementation.

3) *Parallel Out-of-core Matrix Filling:* The reason for developing an out-of-core matrix filling algorithm is to enable one to solve large matrix equations, where the impedance matrix may be too large to be stored in the main memory (RAM) of the system. Compared with the in-core matrix filling algorithm, where the matrix is filled once and kept in the RAM, the main idea of designing an out-of-core algorithm is to fill a portion of the matrix at a time and then write this portion to the hard disk rather than keeping it in the RAM. Note that the proposed out-of-core algorithm is different from the virtual memory; the efficiency of the former can achieve 90% of the in-core algorithm [2], while the efficiency of the latter is usually below 60%.

At the beginning of the out-of-core matrix filling algorithm, a matrix of size $N \times N$ is partitioned into a number of slabs, as shown in Figure 2. The number of slabs N_s and the width of each slab S_i are determined by the nature of the application and the particular system architecture of the computing platform on which the job is executed [2].

Before loops over geometric elements, each process goes through a loop of slabs or blocks, from 1 to N_s. Each process calculates the

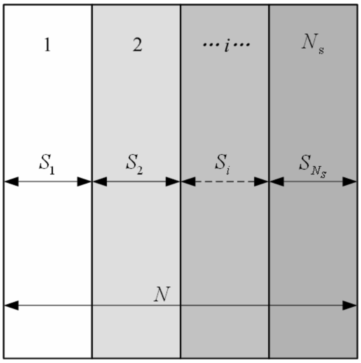


Figure 2. Data decomposition for storing an out-of-core matrix. An $N \times N$ matrix is partitioned into N_s slabs and the width of the ith slab is S_i.

elements for the S_i, which represents the ith out-of-core slab and it sets the global upper and lower bound. For example, for the first slab, the global lower bound is 1 and the upper bound is S_1. For the second slab, the global lower bound is $S_1 + 1$ and the upper bound is $S_1 + S_2$, and so on. Each process fills a portion of the matrix in the same way as the in-core filling algorithm. However, each process pays no attention to the columns that fall outside the bound. After every process has completed the desired action of filling the appropriate portion of the matrix, a synchronous write is called to write the portion of the matrix into a file. Then, each process enters the loop corresponding to the next slab. This procedure avoids the calculation of most of the redundant integrals related to the potential functions, which are used to calculate the elements of the impedance matrix. This is accomplished by using a flag for each order of the geometric element, as described previously for the in-core matrix filling scheme.

By comparing with the in-core matrix filling algorithm, it can be found that for each slab, the algorithm is exactly the same. Most of the overhead for filling the out-of-core matrix, excluding that from the in-core matrix calculation for an individual slab, comes from two parts: 1) calculation of the redundant integrals performed on each process, for different elements of the impedance matrix, which belongs to different slabs, and 2) writing the matrix elements to the hard disk.

4) *Parallel Out-of-core Matrix Equation Solving:* During the procedure of finding the solution to the matrix equation, balancing the computational load between all the processes is also important. However, it is less easy to track compared with the parallel matrix filling procedure, since an increase in the number of unknowns or the number of processes executing the solution can increase the amount of communication required between the processes. This increase in communication will decrease the gains of parallel speedup. However, as a rule of thumb, more processes typically means less wall clock time for solving large problems, even though the overall parallel efficiency may sometimes be increased by using fewer processes to solve the same problem.

The solution of the matrix equation is essentially the same regardless of the type of basis functions used for the MoM. The parallel LU decomposition solver based on the ScaLAPACK library package is described in Chapter 2 in [2]. Also, one can refer to Chapter 6 in [2] for a detail discussion about the iterative solvers based on the conjugate gradient (CG) method. The computational complexity of LU decomposition, scales with $O(N^3)$, is much higher than that of CG type methods with $O(N^2)$, where N is the number of unknowns in (2). However, in this paper, the parallel LU decomposition based on the ScaLAPACK is utilized as the parallel equation solver rather than the parallel iterative CG method due to the fact that the iterative CG method may encounter a divergence problem when dealing with complex antennas composed of thin structures and various materials. For the out-of-core matrix, the solution has an extra layer of Input/Output (I/O) operations related to hard disks. The out-of-core solver is designed to be able to manage very large files for each process to write to and read from hard disks, as described in Chapter 2 in [2].

3.2. Parallelization of PO

Distributing to processes the computation of PO currents at all points over the illuminated triangles is the main task to parallelize the PO method. We use the same number of points over each triangle, and hence we can partition triangles rather than points among processes.

What is necessary is to ensure that the triangles in the lit PO region are partitioned equally to each process. This can be realized by distributing the triangles in neighboring regions, either lit or shadowed, to different processes. As depicted in Figure 3, take 120 triangles distributed over 4 processes for example, we distribute triangles $1, 5, 9, \ldots$ to process 0, triangles $2, 6, 10, \ldots$ to process 1, triangles $3, 7, 11, \ldots$ to process 2 and triangles $4, 8, 12, \ldots$ to process 3. This procedure can be realized by the following pseudocode,

$$\text{If } (Mod(tri-1, Np) == this\_ID), tri \in this\ process,$$

where *tri* is the serial number of a given triangle, *Np* is the total number of processes, and *this_ID* is the process rank of *this process*.

When the number of triangles is much greater than the number of processes, this scheme renders the number of triangles in the lit PO region as well as the number of triangles in the shadowed PO region

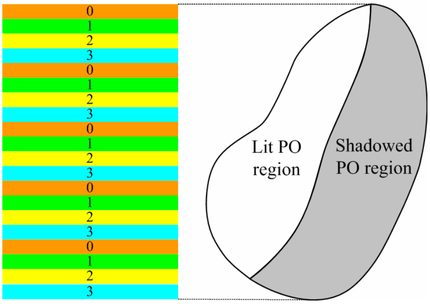


Figure 3. An efficient distribution pattern of PO triangles that are distributed to processes in a process-cyclic manner.

nearly equal in each process and thus ensures good load balancing. This nature of the distribution can also be viewed as a process-cyclic scheme. There is little communication between processes due to the neglect of the mutual interactions within the currents in the lit PO region. Therefore, a near linear speedup can be obtained in the parallel PO part of the parallel hybrid MoM-PO method.

The lit/shadowed determination is carried out through the Z-buffer algorithm [23]. Z-buffer was originally developed to remove hidden surfaces in computer graphics displays. Rius et al. first adopted the usage of the Z-buffer in the GRECO code in the early 1990s to carry out PO calculations [24]. In the algorithm, only the z-depth closest to the observer (that is, the visible surface of the target) is stored in the Z-buffer in pixel form. Note that the Z-buffer code used in this paper is a serial code. However, it is very efficient to determine the lit and shadowed regions for a large number of PO triangles.

4. NUMERICAL RESULTS

4.1. Comparison with the Commercial Software

To validate the accuracy and efficiency of the proposed hybrid MoM-PO methodology, a parabolic reflector fed by a horn is simulated, which is shown in Figure 4. The dimensions of the horn are illustrated in Figure 5. The operating frequency of the antenna is 3.0 GHz.

The simulation is performed on a desktop computer, with a 1.87 GHz CPU and 2 GB of RAM. The radiation patterns obtained using the proposed MoM-PO with $\varepsilon = 10^{-3}$, the coupled MoM-PO in

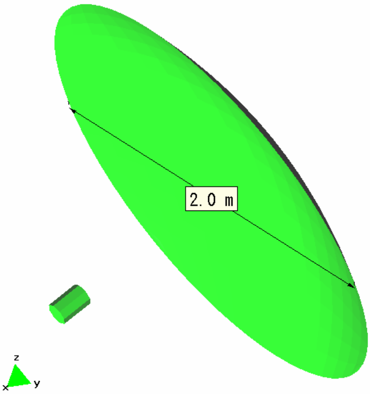


Figure 4. Parabolic reflector fed by a horn.

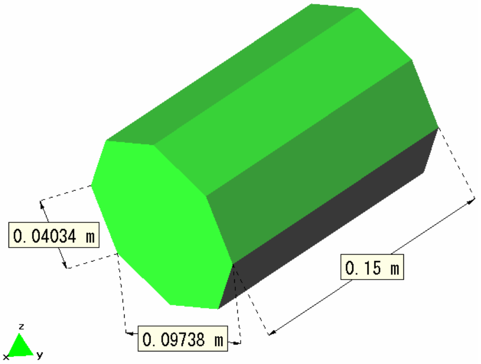
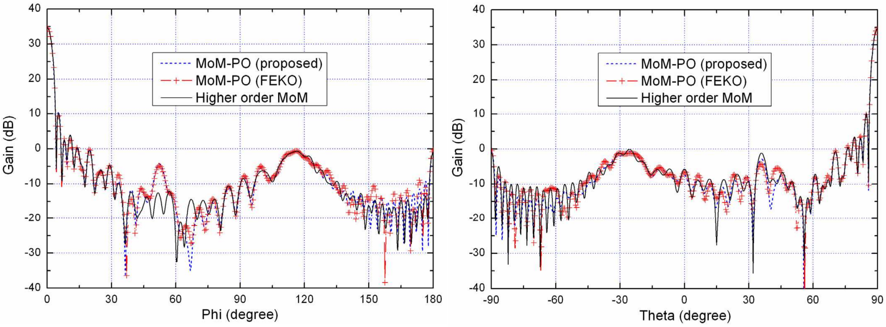


Figure 5. Dimensions of the horn.

FEKO software [25], and the parallel higher-order MoM [3] are plotted together in Figure 6. For the MoM-PO methods, the horn is in the MoM region and the reflector is in the PO region. As shown in Figure 6, the results agree with each other very well in the mainlobe region (i.e., 0° $\sim$ 30° in the *xoy* plane and 60° $\sim$ 90° in the *xoz* plane). The considerable disagreements between them occur in the deep shadow region behind the reflector (i.e., 150° $\sim$ 180° in the *xoy* plane and $-90°$ $\sim$ $-60°$ in the *xoz* plane).

For the PO regions, both the proposed and FEKO's MoM-PO methods have the same number of discretized triangles, i.e., 36,042. While for the MoM regions, the numbers of unknowns are different and listed in Table 1. Note that this example is simulated using only one process for both the MoM-PO methods.

From Table 1, one can see that the HOBs adopted in the MoM region result in less number of unknowns than RWGs. The total computation time of the proposed method is only about 2.8% of the



(a) (b)

Figure 6. Gain of the parabolic reflector antenna: (a) *xoy* plane and (b) *xoz* plane.

Table 1. Comparison between the proposed and FEKO's MoM-PO methods.

Algorithm	Number of unknowns	Matrix filling time (s)	Matrix equation solving time (s)	Voltage iteration time (s)
Proposed MoM-PO	287	0.2	0.05	570.9
FEKO's MoM-PO	1640	20,185.3	1.5	–

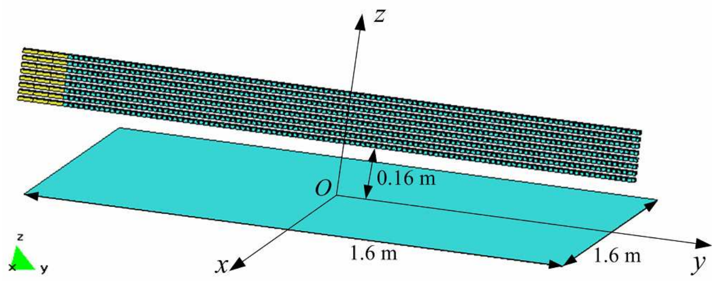


Figure 7. A 108 × 8 slotted waveguide array located above a square metallic plate.

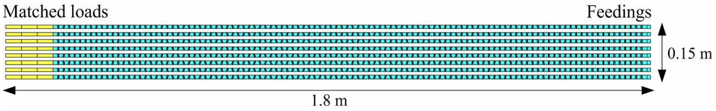


Figure 8. Dimensions of the 108 × 8 narrow-wall slotted waveguide array.

time required by FEKO, which means that the proposed method is about 35.7 times faster than FEKO's MoM-PO method. This benefit not only comes from the less number of unknowns by using HOBs, but also due to the fact that the iterating voltage matrix is utilized in the proposed method instead of modifying impedance matrix, which greatly reduces the computational complexity. In contrast to the proposed method, FEKO uses RWGs in the MoM region and solves the complete MoM-PO matrix equation, as already discussed in Section 2.2.

4.2. Description of the Parallel Computational Platform

The computational platform used in the following two examples is a High-Performance Computing (HPC) cluster with one head node and 24 computing nodes. Each computing node has two quad-core Intel Xeon E5310 1.6 GHz EM64T processors (2 × 4 MB L2 Cache and 1066 MHz FSB), 4 GB RAM, and two 72 GB 15 K rpm SAS hard disks. The nodes are connected with two Infiniband switches. The parallel code is developed using the FORTRAN language based on Message Passing Interface (MPI) and can be applied to both shared memory and distributed memory systems.

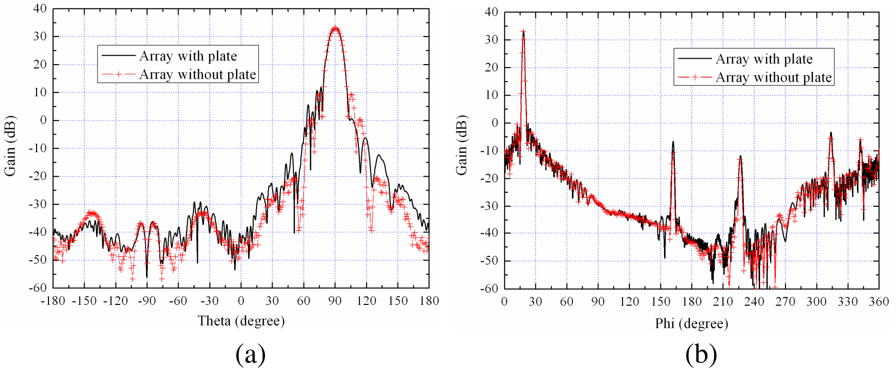


Figure 9. Radiation patterns of the 108 × 8 array with and without the metallic plate: (a) H-plane ($\phi = 18^\circ$) and (b) E-plane ($\theta = 90^\circ$).

4.3. A Slotted Waveguide Array with 108 × 8 Elements over a Metallic Plate

This example is presented to demonstrate the parallel efficiency of the proposed parallel hybrid MoM-PO method. Consider a slotted waveguide array [26–28] with 108 × 8 narrow-wall inclined slots over a square metallic plate with an edge length of 1.6 m. As shown in Figure 7, the plate is in the xoy plane and its center is at the origin of coordinate. The waveguides are oriented parallel to the y-axis and located 0.16 m above the plate. Each waveguide is a WR-90 waveguide (X-band) with dimensions of 22.86 mm × 10.16 mm. The distance between two neighboring waveguides is 20.95 mm and the distance between any two adjacent slots on each waveguide is 15.5 mm. The dimensions of the array are shown in Figure 8. One end of each waveguide is terminated by a matched load [29], and the other end is placed a feeding pin as an excitation in the waveguide. A −20 dB Taylor amplitude distribution is utilized in the array feed.

The operating frequency of the array is 9.375 GHz. In the parallel hybrid MoM-PO analysis, the array is in the MoM region and the plate is in the PO region. Figure 9 plots the radiation patterns of the array over the plate as well as the patterns of the array without the plate. The latter are computed by the parallel out-of-core MoM [3]. It is seen from Figure 9 that the pattern is distorted by more than 10 dB in the H-plane, especially in the region of $0^\circ \sim 75^\circ$ and $105^\circ \sim 180^\circ$ due to the existence of the plate, while the distortion is relatively little in the E-plane, which is less than 5 dB in most angles. Note that the H-plane is the $\phi = 18^\circ$ cut plane, but not the xoz plane for the model in Figure 7.

Table 2. Comparison of the simulation of the 108 × 8 array over a plate with respect to processes.

Number of processes (Process grid)	In-core buffer per process (MB)	Matrix filling time (s)	Matrix equation solving time (s)	Time for one voltage iteration (s)
24 (4 × 6)	340.0	12,672.3	13,703.9	11,887.7
48 (6 × 8)	340.0	6648.8	6589.1	6290.0
96 (8 × 12)	340.0	3745.6	3622.8	3727.7
192 (12 × 16)	340.0	1870.7	2259.8	2430.2

The surface of the slotted waveguide array is discretized into bilinear patches and the plate into planar triangles. The number of unknowns is 75,704 in the MoM region and the number of triangles is 143,692 in the PO region. We use seven Gauss integration points over each triangle in the computation of the induced PO currents and set a small number equal to 10^{-3} as the iteration convergence parameter ε in (1). The iteration converges in 3 steps for this model (the iteration is the voltage-based iteration as discussed in Section 2.2). The in-core buffer and time for the parallel out-of-core hybrid solver by using different processes are listed in Table 2. In cases of different processes, the in-core buffer of the parallel out-of-core solver for each process is the same, i.e., 340.0 MB. For this simulation, the proposed out-of-core hybrid solver requires 11.3 GB RAM (double precision) when using 24 processes, while an in-core hybrid solver would require at least 85.4 GB RAM (double precision) [18]. If we use one process for this simulation, we only need 340.0 MB RAM in theory, but the simulation time could be very long in practice. The out-of-core technique allows us to simulate this model by using a small amount of RAM, and hence it is very favorable for solving problems in cases of lack of RAM.

Take the time for 24 processes as a reference, the parallel efficiencies for this simulation are depicted in Figure 10. In the cases of 48 and 96 processes, the parallel efficiencies for the matrix filling, matrix equation solving and voltage iteration are higher than 80%. In the case of 192 processes, the parallel efficiencies for the matrix

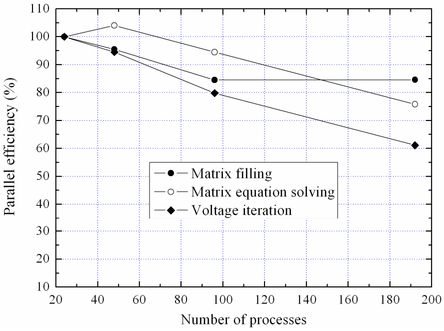


Figure 10. Parallel efficiencies for the simulation of the 108 × 8 array over a metallic plate.

equation solving and voltage iteration decrease to a level below 80%. Increase in the number of processes deteriorates the performance. This is expected because the ratio of the communication volume to computation increases with increase in the number of processes for this problem with small size. However, as a rule of thumb, more processes typically means less time for solving the problem. It should be noted that the parallel efficiency for the matrix equation solving exceeds 100% when using 48 processes. The reason is that process grids have a significant impact on the efficiency, which have been discussed in detail in our previous work [2].

4.4. A Slotted Waveguide Array with 108 × 20 Elements over a Large Airplane

Consider next a realistic problem that a slotted waveguide array of 108 × 20 narrow-wall slots over a large airplane. The waveguide parameters are the same as those of the 108 × 8 array analyzed in Section 4.3 and the array model is shown in Figure 11. The airplane model is 36 m long, 40 m wide and 10.5 m high. The corresponding electrical sizes of the model are 1125λ, 1250λ and 328.1λ, where λ is the free-space wavelength at 9.375 GHz. The array location on the airplane is depicted in Figure 12. The array rotates in azimuth plane around its vertical axis, scans in elevation plane by phase control, and hence radiates and receives signals all around the structure.

This model is analyzed using the parallel hybrid MoM-PO method with the array in the MoM region and the airplane in the PO region. Intuitively, the airplane may have a great impact on the radiation pattern of the array, especially for the case when the array rotates to point the mainlobe towards the tail of the airplane. In this simulation,

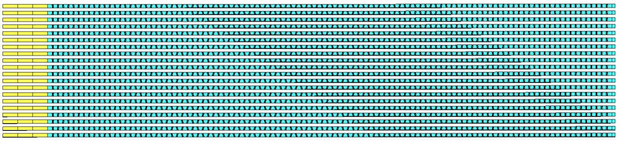


Figure 11. A large slotted waveguide array with 108 × 20 elements.

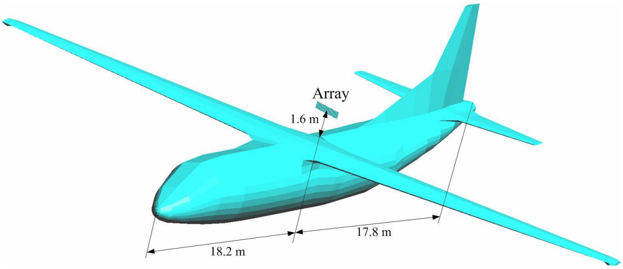
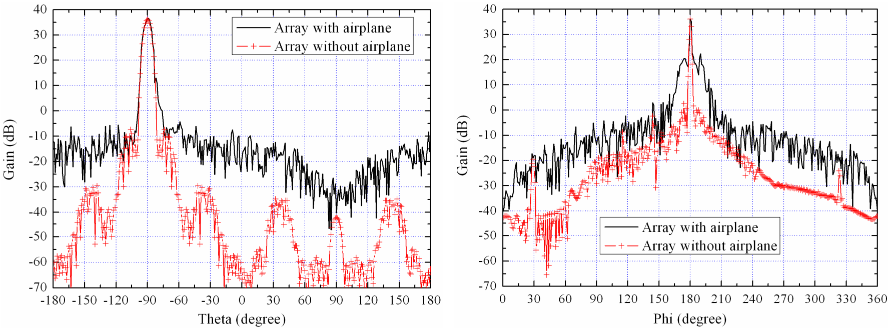


Figure 12. The 108×20 slotted waveguide array over a large airplane.



(a) (b)

Figure 13. Radiation patterns of the 108×20 array with and without the airplane: (a) H-plane ($\phi = 0°$) and (b) E-plane ($\theta = 90°$).

a -40 dB Taylor amplitude distribution is utilized in the array feed and the mainlobe direction is directed towards the tail. The radiation patterns of the array over the airplane are shown in Figure 13, in which the corresponding results of the array alone are also given. From the comparison, it is clearly seen that the sidelobe levels of the array over the airplane significantly increase in both the H-plane and E-plane, especially in the E-plane. In specific, the first sidelobe level increases from -35 dB to -15 dB in the E-plane, which is mainly due to the effect of the airplane tail. The performance of the array is degraded too much that we consider that this case does not meet the

Table 3. Comparison of the simulation of the 108 × 20 array over an airplane with respect to processes.

Number of processes (Process grid)	In-core buffer per process (MB)	Matrix filling time (s)	Matrix equation solving time (s)	Time for one voltage iteration (s)
96 (8 × 12)	340.0	56,857.0	50,259.7	185,465.4
192 (12 × 16)	340.0	29,008.7	29,564.5	114,062.4

electromagnetic compatibility (EMC) requirements. In the situations when the mainlobe is radiating along the nose and wing of the airplane, the airplane may have less impact on the radiation properties of the array.

The number of unknowns in the MoM region is 189,260 and the number of triangles in the lit PO region is 2,685,168. In the parallel simulation, seven Gauss points on each triangle are employed in the computation of the induced PO currents. In this example, the mainlobe of the array is rotated in azimuth to point towards the tail. The iteration converges to the criterion of 10^{-3} with 1 step for this model (the iteration is the voltage-based iteration as discussed in Section 2.2). The in-core buffer and time for the simulation using 96 and 192 processes are listed in Table 3. Take the time for 96 processes as a reference, we can find the parallel efficiencies are 98.0%, 85.0% and 82.3% for the matrix filling, matrix equation solving and voltage iteration with the usage of 192 processes, respectively. For this simulation, the proposed out-of-core hybrid solver requires 63.8 GB RAM (double precision) when using 192 processes, while an in-core hybrid solver would require at least 533.7 GB RAM (double precision) [18]. Therefore, the proposed parallel hybrid MoM-PO method provides a very efficient solution to realistic problems of complex antennas on large platforms but requires a small amount of RAM.

5. CONCLUSION

An efficient parallel hybrid MoM-PO method with the out-of-core technique is developed to solve extremely large on-board antenna problems. The combination of the block matrix scheme for parallelizing

MoM and the process-cyclic scheme for assigning PO workload ensures good load balance. The proposed out-of-core technique allows us to solve challenging problems with very low memory usage. Numerical results confirm the accuracy and efficiency of the proposed parallel technique and its applicability to the analysis of radiation from on-board antenna systems including complex antennas and large platforms.

ACKNOWLEDGMENT

This work is supported in part by the Basic Science Research Fund in Xidian University of China (JY10000902002).

REFERENCES

1. Taboada, J. M., M. G. Araujo, J. M. Bertolo, L. Landesa, F. Obelleiro, and J. L. Rodriguez, "MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetics," *Progress In Electromagnetics Research*, Vol. 105, 15–30, 2010.
2. Zhang, Y. and T. K. Sarkar, *Parallel Solution of Integral Equation Based EM Problems in the Frequency Domain*, Wiley, New Jersey, 2009.
3. Zhang, Y., M. Taylor, T. K. Sarkar, H. Moon, and M.-T. Yuan, "Solving large complex problems using a higher-order basis: Parallel in-core and out-of-core integral-equation solvers," *IEEE Antennas and Propag. Mag.*, Vol. 50, No. 4, 13–30, Aug. 2008.
4. Zhang, Y., T. K. Sarkar, M. Taylor, and H. Moon, "Solving MoM problems with million level unknowns using a parallel out-of-core solver on a high performance cluster," *IEEE Antennas Propag. Soc. Int. Symp.*, 1–4, June 1–5, 2009.
5. Zhang, Y., M. Taylor, T. K. Sarkar, A. De, M.-T. Yuan, H. Moon, and C.-H. Liang, "Parallel in-core and out-of-core solution of electrically large problems using the RWG basis functions," *IEEE Antennas and Propag. Mag.*, Vol. 50, No. 5, 84–94, Oct. 2008.
6. Yang, M.-L. and X.-Q. Sheng, "Parallel high-order FE-BI-MLFMA for scattering by large and deep coated cavities loaded with obstacles," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 13, 1813–1823, 2009.
7. Yuan, J., Y. Qiu, and Q. Z. Liu, "Fast analysis of multiple antennae coupling on very electrical large objects via parallel technique," *Journal of Electromagnetic Waves and Applications*, Vol. 22, No. 8–9, 1232–1241, 2008.

8. Hennigan, G. and S. Castillo, "Open region, electromagnetic finite-element scattering calculations in anisotropic media on parallel computers," *IEEE Antennas Propag. Soc. Int. Symp.*, 484–487, Jul. 11–16, 1999.
9. Lei, J.-Z., C.-H. Liang, W. Ding, and Y. Zhang, "EMC analysis of antennas mounted on electrically large platforms with parallel FDTD method," *Progress In Electromagnetics Research*, Vol. 84, 205–220, 2008.
10. Liu, Y., Z. Liang, and Z. Yang, "A novel FDTD approach featuring two-level parallelization on PC cluster," *Progress In Electromagnetics Research*, Vol. 80, 393–408, 2008.
11. Zhang, Y. J., S. X. Gong, and Y. X. Xu, "Radiation pattern synthesis for arrays of conformal antennas mounted on an irregular curved surface using modified genetic algorithms," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 10, 1255–1264, 2009.
12. Harrington, R. F., *Field Computation by Moment Methods*, IEEE Press, New York, 1993.
13. Bouche, D. P., F. A. Molinet, and R. Mittra, "Asymptotic and hybrid techniques for electromagnetic scattering," *Proc. IEEE*, Vol. 81, No. 12, 1658–1684, Dec. 1993.
14. Hodges, R. E. and Y. Rahmat-Sammi, "An iterative current-based hybrid method for complex structures," *IEEE Trans. Antennas Propag.*, Vol. 45, No. 2, 265–276, Feb. 1997.
15. Djordjevic, M. and B. M. Notaros, "Higher order hybrid method of moments–physical optics modeling technique for radiation and scattering from large perfectly conducting surfaces," *IEEE Trans. Antennas Propag.*, Vol. 53, No. 2, 800–813, Feb. 2005.
16. Chen, X. J., X. W. Shi, and L. Xu, "A hybrid method used to analysis the double reflection between two curved surfaces," *Journal of Electromagnetic Waves and Applications*, Vol. 22, No. 8–9, 1191–1198, 2008.
17. Hu, B., X.-W. Xu, M. He, and Y. Zheng, "More accurate hybrid PO-MoM analysis for an electrically large antenna-radome structure," *Progress In Electromagnetics Research*, Vol. 92, 255–265, 2009.
18. Zhang, Y., X.-W. Zhao, D. G. Doñoro, S.-W. Ting, and T. K. Sarkar, "Parallelized hybrid method with higher-order MoM and PO for analysis of phased array antennas on electrically large platforms," *IEEE Trans. Antennas Propag.*, to be published.
19. Zhang, Y., X.-W. Zhao, M. Chen, and C.-H. Liang, "An efficient

MPI virtual topology based parallel, iterative MoM-PO hybrid method on PC clusters," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 5, 661–676, 2006.

20. Kolundzija, B. M. and A. R. Djordjevic, *Electromagnetic Modeling of Composite Metallic and Dielectric Structures*, Artech House, Norwood, 2002.
21. Yuan, H. B., N. Wang, and C. H. Liang, "Fast algorithm to extract the singularity of higher order moment method," *Journal of Electromagnetic Waves and Applications*, Vol. 22, No. 8–9, 1250–1257, 2008.
22. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, Vol. 30, No. 3, 409–418, May 1982.
23. Zha, F.-T., S.-X. Gong, Y.-X. Xu, Y. Guan, and W. Jiang, "Fast shadowing technique for electrically large targets using Z-buffer," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 2–3, 341–349, 2009.
24. Rius, J. M., M. Ferrando, and L. Jofre, "GRECO: Graphical electromagnetic computing for RCS prediction in real time," *IEEE Antennas and Propag. Mag.*, Vol. 35, No. 2, 7–17, Apr. 1993.
25. http://www.feko.info/
26. Ebadi, S. and K. Forooraghi, "Comparative study of annular and rectangular waveguides for application in annular waveguide slot antennas," *Journal of Electromagnetic Waves and Applications*, Vol. 22, No. 16, 2217–2230, 2008.
27. Hua, Y. and J. Li, "Analysis of longitudinal shunt waveguide slots using FEBI," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 14–15, 2041–2046, 2009.
28. Yuan, H.-W., S.-X. Gong, Y. Guan, and D.-Y. Su, "Scattering analysis of the large array antennas using the synthetic basis function method," *Journal of Electromagnetic Waves and Applications*, Vol. 23 No. 2–3, 309–320, 2009.
29. Zhao, X.-W., Y. Zhang, T. K. Sarkar, S.-W. Ting, and C.-H. Liang, "Analysis of a traveling-wave waveguide array with narrow-wall slots using higher order basis functions in method of moments," *IEEE Antennas Wireless Propag. Lett.*, Vol. 8, 1390–1393, 2009.