 cmes, vol.5, no.4, pp.361-372, 2004

Review of Large Scale Computing in Electromagnetics with Fast Integral Equation Solvers

W.C. Chew[1], J.M. Song[1], T.J. Cui[1], S. Velamparambil[1], M.L. Hastriter[1], and B. Hu[1]

Abstract: This paper reviews recent advances in large-scale computational electromagnetics using frequency domain integral equations. It gives a brief history of methods to solve Maxwell's equations, followed by a description of various historical ages in solution technique developments. Then it describes computational electromagnetics followed by a brief description of how fast integral equation solvers such as the multilevel fast multipole algorithm (MLFMA) is constructed using the tree network. Some examples of large scale computing using MLFMA are given. Ray physics used to further accelerate the speed of MLFMA. The parallel implementation of MLFMA in a code called ScaleME is reviewed, and some example calculations and scaling studies are given. Finally, we review the recent development of the fast inhomogeneous plane wave algorithm (FIPWA) for layered media for large scale computing.

1 Introduction

Electromagnetic analysis has always played a central role in electrical engineering due to the importance of Maxwell's theory and related applications. The interaction of electromagnetic fields with complex bodies is an intricate phenomenon which cannot be understood or predicted easily unless solutions of Maxwell's theory are used to enhance its understanding. In the beginning, it was the solutions of Maxwell's equations for simple shapes that were used to understand electromagnetic phenomena. This was the age of simple shapes around early 1900. However, due to the need for solutions of more complex shapes, mathematicians, scientists and engineers sought approximate methods to solve Maxwell's equations. These theories included perturbation theory or asymptotic theory. They greatly expanded the type of problems that could be solved approximately in electromagnetics. It was the age of approximations around 1950s. More recently, the advent of computer technology has called for solutions of complex objects of arbitrary shapes using numerical methods. This is the age of numerical methods, which began around 1960s [Chew, Jin, Michielssen and Song (2001)].

There are two main classes of numerical methods to solve Maxwell's equations, one class involves differential equation solvers, where Maxwell's equations are solved directly using the finite element method, or the finite difference method [Jin (1993), Taflove (1995)]. Another class involves integral equation solvers where Maxwell's equations are first converted to integral equations using Green's function. Then the integral equation can be converted into a matrix equation using a projection method such as the method of moments [Harrington (1982)]. The ensuing matrix equation can then be solved numerically on a computer.

An advantage of differential equation solvers is that they are relatively easy to implement. Moreover, they give rise to a sparse matrix system that is economical to solve and store. However, the unknowns are the fields, and in general, the space around a scatterer has to be discretized as well, yielding a large number of unknowns. Also, grid dispersion error is an important issue, as the fields propagate on a numerical grid. However, there has been much research recently to reduce the grid dispersion error in differential equation solvers [Yang, Gottlieb and Hesthaven (1997), Liu (1998), Forgy and Chew (2002)].

A distinct advantage of integral equation solvers is that they generally entail a smaller number of unknowns since the currents are the unknowns. Since currents occupy a finite support in space, only that part of space needs to be discretized to solve for the current unknowns, which is significantly smaller than the number of unknowns for differential equation solvers. The disadvantage of integral-equation solvers is that they are relatively difficult to implement, and the resultant matrix is dense and expensive to store and solve. However, the recent advent

[1] Center for Computational Electromagnetics and Electromagnetics Laboratory University of Illinois Urbana, IL 61801

of fast integral equation solvers will popularize their usage in electromagnetic analysis [Chew, Jin, Michielssen and Song (2001)]. The fast solvers are matrix free, and they use memory efficiently. Consequently, the resources needed to solve electromagnetic problems of the same size are a lot more economical for integral equation solvers compared to differential equation solvers. When scattering by bulk material occurs, it is efficient to solve the scattering problem with differential equation solvers. In this case, the boundary integral method can be used to truncate the solution domain, and fast integral equation solvers can be used to solve the pertinent integral equation [Chew, Jin, Michielssen and Song (2001), Chap. 13, Liu and Jin (2001)].

2 Solving an Integral Equation

When solving an integral equation, one seeks a set of sources that cooperate with each other to produce a field or a potential that satisfies the boundary condition. Hence, these sources have to know how to interact with each other, and they do so through the integral equation via the Green's function. An integral equation is first converted to a matrix equation in order to seek its solution. One way a solution can be sought for a matrix equation is to perform a series of matrix-vector products. These matrix vector products generate information about the scatterer, and eventually, a solution to the scattering problem can be gotten by using information embedded in these matrix-vector products. The space spanned by the vectors so generated is called the Krylov space [Hestenes and Stiefel (1952)]. We shall illustrate a matrix-vector product with the telephone network analogy.

3 Telephone Network Analogy

A matrix-vector product is physically equivalent to calculating the field at every source point due to all other source points in the scatterer. Hence, every source element "talks" to every other source element. This is like having N telephones, where each telephone is connected to every other telephone by a direct line. Consequently, there are $O\left(N^2\right)$ links needed to connect N telephones together. However, the telephone companies know better. Telephones in close proximity to each other are connected to a hub, and in turn, the hubs are connected to each other (see Figure 1). In this manner, the number of telephone links needed is greatly reduced.

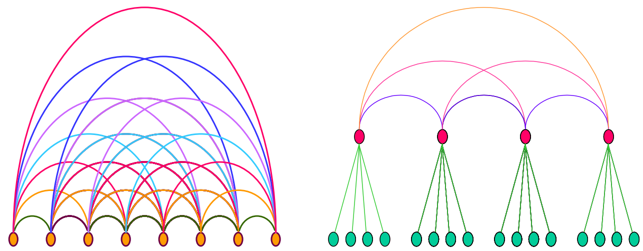

Figure 1 : A telephone network analogy of how the sources talk to each other through the fast multipole method.

In order to facilitate a two-level matrix-vector multiply, it is necessary to factorize elements of the matrix $\overline{\mathbf{A}}$. [Rokhlin (1990)]

$$A_{ij} = \mathbf{V}_{il}^t \cdot \mathbf{T}_{ll'} \cdot \mathbf{V}_{l'j} \tag{1}$$

In the above, the matrix A_{ij} transmits information from current element j to current element i. The factorized form facilitates the three-stage transmission of information from element j to element i, namely, first from element j to hub l', from hub l' to hub l, and then from hub l to element i. The above factorization can be achieved via the use of the translational addition theorem. A logical extension of the idea is to extend the two-level network to a multilevel network, giving rise to a multilevel algorithm as shown in Figure 2. (In computer science, this network is known as an inverted tree or just tree, where the top of the tree is the root, and the bottom part of the tree is the leafy nodes.)

In order to facilitate the transfer of information in a multilevel tree, a matrix element has to be factorized as:

$$A_{ij} = \mathbf{V}_{il_1}^t \cdot \overline{\beta}_{l_1 l_2} \cdot \overline{\beta}_{l_2 l_3} \cdots \overline{\beta}_{l_{L-1} l_L} \cdot \overline{\beta}_{ll'} \cdot \overline{\beta}_{l_L l_{L-1}} \cdots \overline{\beta}_{l_3 l_2} \cdot \overline{\beta}_{l_2 l_1} \cdot \mathbf{V}_{l'j}$$

In this manner, a multi-stage manner of information transfer is possible. Again, the above factorization is possible via the use of the translational addition theorem.

Therefore in using the multilevel fast multipole algorithm (MLFMA) [Lu and Chew (1994), Song and Chew (1995)] to solve scattering problems, the scatterer is first facetized into elements and enclosed in a cube. The cube is then recursively subdivided into eight smaller cubes until the smallest cube contains about several elements. An abstract communication network is constructed in the computer using an oct-tree structure. The currents induced on each element can then communicate with each other via this abstract communication network. Consequently, a matrix-vector product can be performed in $O(N \log N)$ operations rather than the conventional $O\left(N^2\right)$. When this ability is used in an iterative solver such as the conjugate gradient method, an integral equation with N unknowns can be solved in $cN_{iter}N \log N$ operations. There are more details than what meets the eyes here, and interested readers should consult reference [Chew, Jin, Michielssen and Song (2001)].

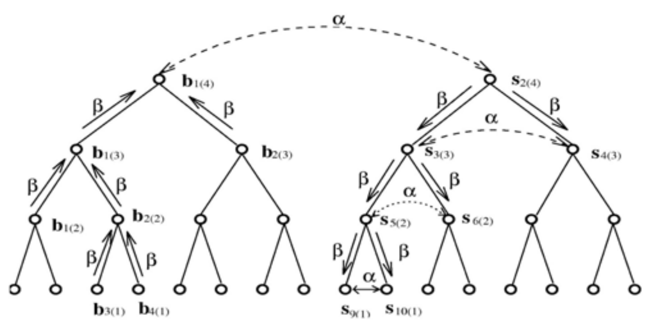

Figure 2 : A multilevel algorithm emulates a multilevel telephone network. When a large number of telephones are involved, the multilevel network greatly reduces the number of links needed to connect all the telephones together.

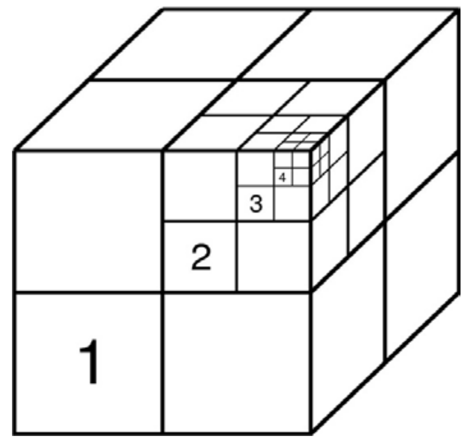

Figure 3 : In using the multilevel fast multipole algorithm (MLFMA) for scattering, an object is first enclosed inside a cube. The cube is then recursively subdivided into eight smaller cubes until the smallest cube contains a few current elements. The current elements then can communicate efficiently with each other via the abstract oct-tree network.

4 Some Examples

As a result of reduced computational complexity, scattering solutions from objects of unprecedented sizes can be solved by MLFMA [Song, Lu, Chew and Lee (1998), Song and Chew (2000)]. The algorithm for fast matrix-vector products, MLFMA, has been incorporated into FISC (Fast Illinois Solver Code), and used to solve large-scale problems. Figure 4 shows the induced currents calculated by FISC on a fictitious aircraft, VFY 218. An advantage of computer simulation is the ability to color-code currents that are not visible even in experiments. Figure 5 shows a simulation of a car where the body is modeled by a perfect electric conductor (PEC), the windshields are modeled by the thin dielectric sheets (TDS), and the tires are modeled by impedance boundary conditions (IBC).

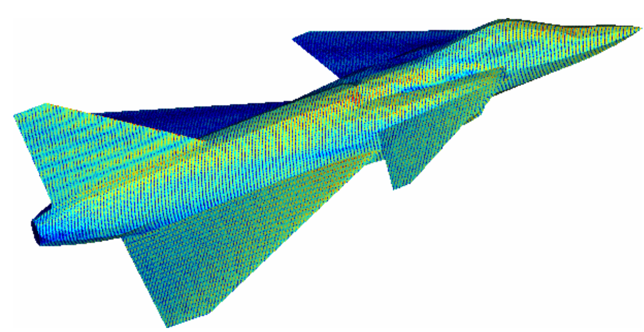

Figure 4 : Induced current on a full-size fictitious aircraft VFY-218 illuminated by a vertically polarized plane wave at 2 GHz. The calculation was done by FISC (Fast Illinois Solver Code), the first to employ MLFMA.

Figures 6 and 7 show some tour-de-force computation with the Fast Illinois Solver Code (FISC) where close

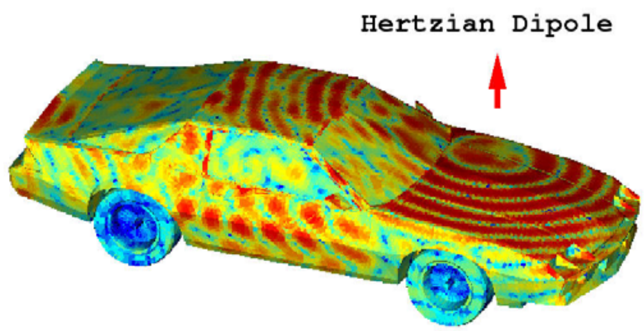

Figure 5 : Irradiation of an '83 Camaro at 1 GHz by a Hertzian dipole. The calculation was done by FISC. The geometry file was derived from VRML (virtual reality modeling language) and an automatic mesh refinement was performed on the geometry file before the calculation.

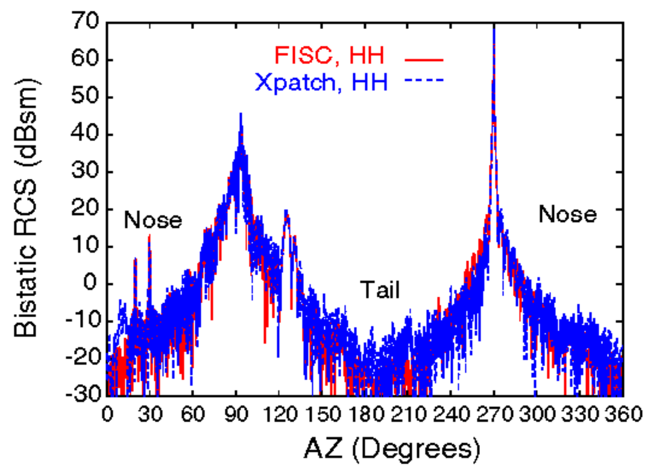

Figure 7 : Bistatic radar cross section (RCS) of the VFY 218 at 8 GHz for horizontal polarization. The number of unknowns is 9,990,918. We used 126 nodes of a 128-node Origin2000, dedicated mode, with 45.5 GB of memory, 0.4 hour for filling matrix, 11.9 hours for 43 iterations in GMRES (15 to reach 0.01 residual error), and 6 minutes for 3,600 RCS points.

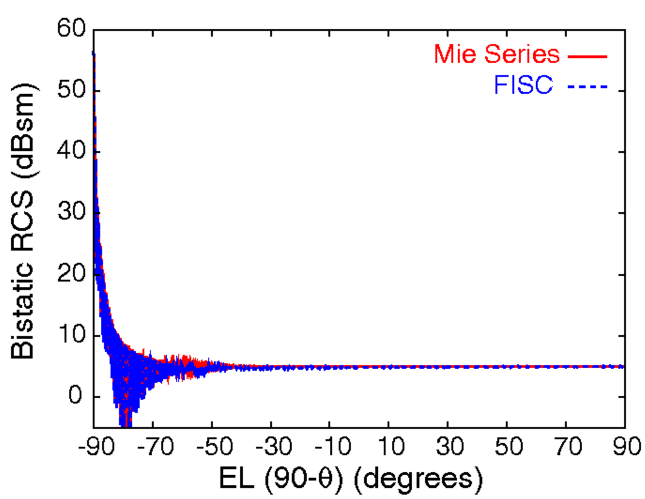

Figure 6 : Scattering solution of a sphere whose diameter is 120 wavelengths. We used 32 nodes of Origin2000, 26.7 GB of memory, 1.5 hours for filling matrix, 13.0 hours for 43 iterations in GMRES (15 to reach 0.01 residual error), and 3 minutes for 1800 points of RCS.

to 10 million unknowns were needed [Song and Chew (2000)]. The size of the VFY 218 being solved in Figure 7 is about 412 wavelengths long, 237 wavelengths wide, and 109 wavelengths tall. Had a differential equation solver such as an FDTD been used in solving the problem, it would have entailed over 40 billions unknowns. Over 400 copies of FISC have been distributed together with XPATCH by SAIC/DEMACO.

FISC can also be used to generate time-domain responses of a complex scatterer such as an aircraft. Figure 8 shows the use of FISC to synthesize the time-domain response from the VFY 218 up to the frequency of 1.2 GHz [Song and Chew (2002)]. In principle, it can be used to solve for scattering solution of a time domain pulse having frequency content up to 8 GHz. Open inlet contributes most responses comparing with the aircraft with closed inlet as shown in Figure 9.

5 Ray Physics for Acceleration

When the frequency is high, and the wavelength short, the behavior of electromagnetic waves is quite different from its low frequency counterpart. Currents start to form beams, and rays account for the interaction between two groups of sources far apart. We have used ray

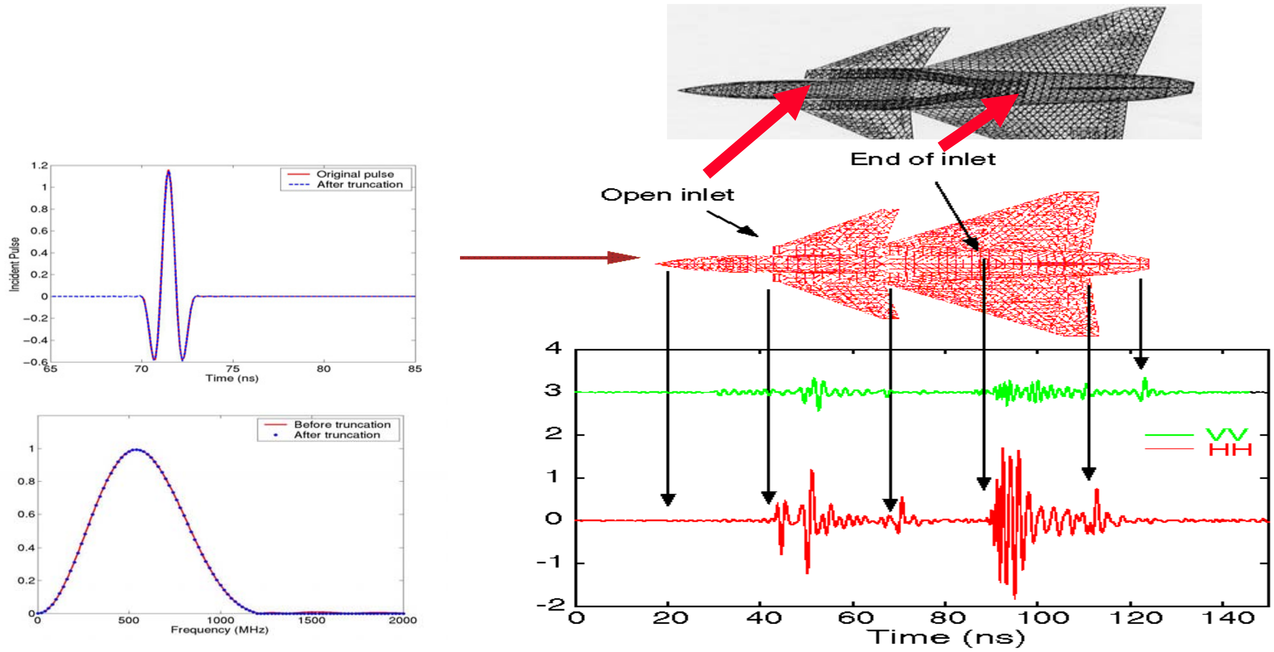

Figure 8 : Time-domain computation with the FISC. The computation was done over many frequency points-294 points from 28 MHz to 1.2 GHz with a center frequency of 530 MHz. A second-order Blackman-Harris pulse is synthesized for the incident wave. The plot for the VV polarization is shifted by 3 units for clarity.

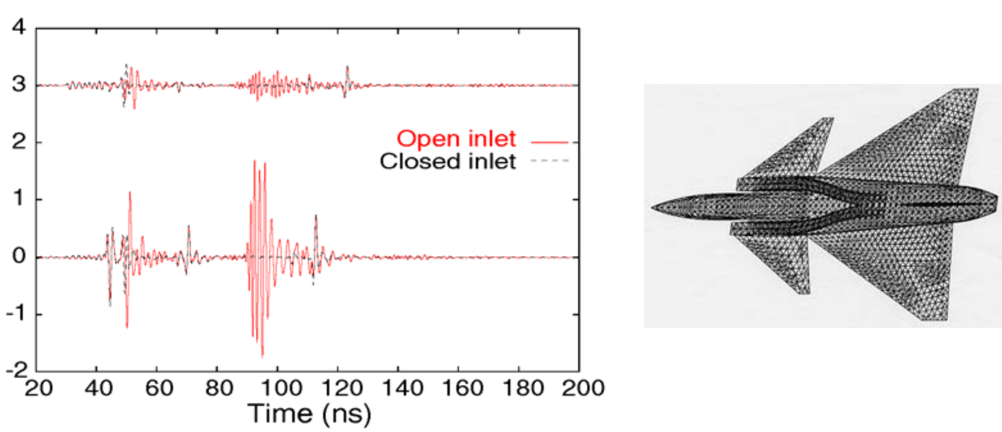

Figure 9 : Time domain scattering from a VFY 218 with opened and closed engine inlets.

physics to speed up the calculation of inter-group interaction in MLFMA [Warnick and Chew (2001), Chew, Cui and Song (2002), Cui, Chew, Chen and Song (2002)].

In MLFMA, in order to send the field (information) from the source group to the destination group, the outgoing far field pattern (radiation pattern) of the source group is first computed. This far field pattern is stored as a row vector. We call the generation of a far field pattern where the field emanates from a point, the aggregation stage. Then the translator converts the outgoing far field pattern to incoming far field pattern (incoming pattern) at the destination group, which is termed the translation stage. The incoming pattern is again stored as a row vector. A final disaggregation stage is used to generate the field at the destination source points. Since the translator is a diagonal matrix, it is represented by a column vector, and it is a function of angle as shown in Figure 10.

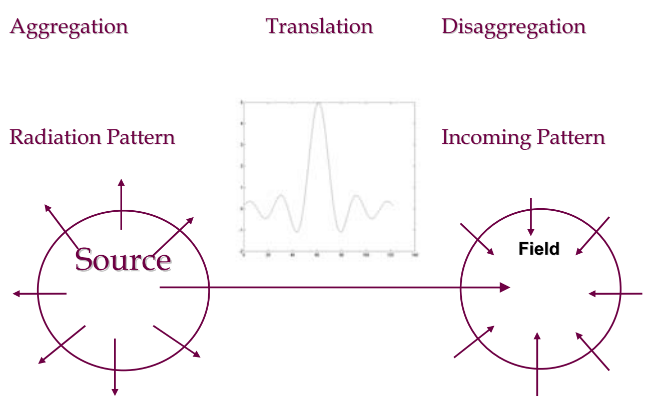

Figure 10 : The aggregation stage, translation stage, and the disaggregation stage in the fast multipole algorithm.

When ray physics becomes dominant, the translation function (or translator in short) becomes a sharp peak function reminiscent of a ray that dominates the interaction between two groups. Hence, by proper design, this ray physics behavior can be enhanced in the translator. We have found that by proper windowing of the series that generates the translation function, a sharp peak translator will result. The equation for the translator in 3D scattering is given by

$$T_{mn}\left(\hat{k}\cdot\hat{r}_{mn}\right) = \sum_{l=0}^{L} i^l (2l+1)\, h_l^{(1)}(kr_{mn})\, P_l\left(\hat{k}\cdot\hat{r}_{mn}\right) w_{mn} \tag{2}$$

where $h_l^{(1)}(kr_{mn})$ is a spherical Hankel function of the first kind, and $\hat{r}_{mn}$ is a unit vector that points from the center for group m to group n, and $\hat{k}$ is the direction of the plane wave that emanates from the source group. The above summation is divergent for increasing L, and the translator should be regarded as a distribution. If w_{mn} is chosen to be unity, the translator exhibits Gibbs phenomenon typical of an abruptly truncated Fourier series. This is shown in Figure 11 corresponding to the unwindowed function. When a proper window is assumed, a sharper translator ensues.

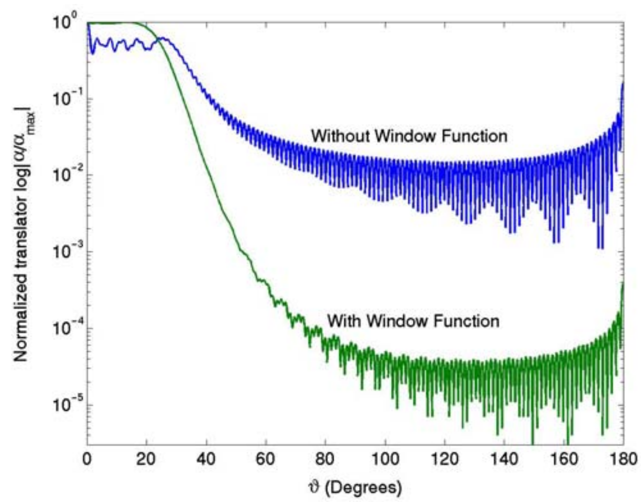

Figure 11 : The plot of the translator as a function of ϑ, the angle between $\hat{k}$ and $\hat{r}_{mn}$. A sharply peaked translator results when the function is windowed in the spectral space. Here, r_{mn}, the distance between the two group centers, is 20λ

Figure 12 shows the windowed and un-windowed translators for different group separations. The effect of the ray physics becomes more pronounced when the group separation becomes larger. This physics can be used to reduce the computational load in MLFMA. We have hybridized MLFMA with the fast far field approximation (FAFFA) where FAFFA is used to calculate the interaction when two groups are very far apart. When the group separations are moderately large, we use the ray propagation fast multipole algorithm (RPFMA) to account for the interaction. The performance of the hybrid algorithm (MLFMA-RPFMA-FAFFA) is shown in Table 1. As much as 40.3% saving in CPU time is possible, along with 12.8% saving in memory requirements.

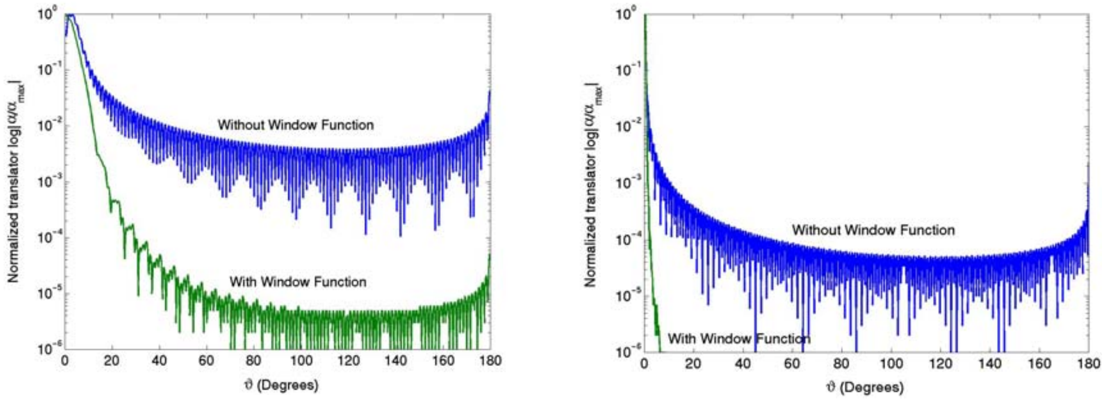

Figure 12 : The translator function windowed and un-windowed for two different group separations. (on the left) $\widehat{r}_{mn} = 100\lambda$, (on the right) $\widehat{r}_{mn} = 100\lambda$.

Method	Top Level	g	Memory (MB)	CPU time (s)
Conventional MLFMA	2	100	4436.55	33821.85
MLFMA-RPFMA-FAFFA	3	1.5	3977.20 (10.4%)	29344.25 (13.2%)
MLFMA-RPFMA –FAFFA	4	2.0	3910.10 (11.9%)	21214.76 (37.3%)
MLFMA-RPFMA-FAFFA	5	2.0	3867.30 (12.8%)	20185.36 (40.3%)
MLFMA-RPFMA-FAFFA	5	3.0	3867.30 (12.8%)	22751.87 (32.7%)

Table 1 : Comparison of the CPU and memory requirements of the hybrid algorithm (MLFMA-RPFMA-FAFFA) compared to conventional MLFMA. The parameter g controls the degree of switching over to the ray-physics algorithm (RPFMA-FAFFA). When g is large, conventional MLFMA is used, but the smaller g is, the more the ray-physics algorithm is used. In this simulation, the frequency is 2 GHz, with 2,067,798 unknowns involved, and 9-level algorithms are used.

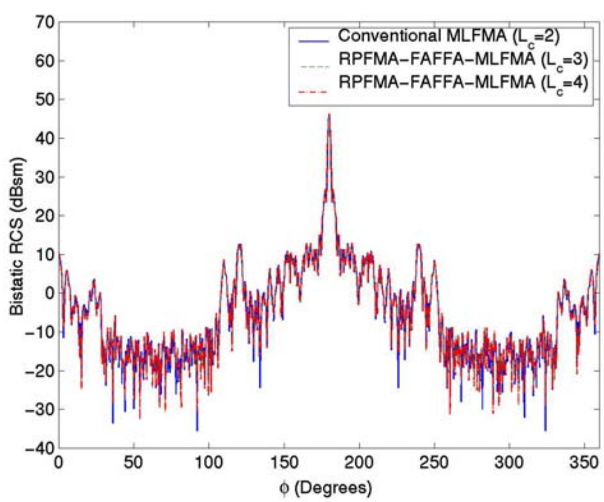

Figure 13 : The bistatic RCS of the VFY 218 for the sample calculation in Table 1. No noticeable degradation of the RCS occurs with the use of RPFMA-FAFFA-MLFMA.

6 Parallelization

The ability to create an efficient parallel computation algorithm is extremely important for large-scale computing. Parallel algorithms allow us to harness the power of multiprocessor supercomputers, and if done efficiently, will permit us to solve problems of unprecedented sizes. Due to the advent of MLFMA, a matrix-vector product can be performed in $O(N \log N)$ operations, making the solution of scattering problems via integral equation solvers highly attractive compared to differential equation solvers. However, differential equation solvers can be easily parallelized via the domain-decomposition method, and hence, the size of the problem that can be solved is limited only by the availability of computational resource.

Due to the complexity of MLFMA, some researchers thought that it is not parallelizable due to the increased communication cost. If this had been the case, it would have spelled the end of truly large-scale computing for MLFMA. Fortunately enough, we have parallelized MLFMA even for very large scale problems through a code called ScaleME [Velamparambil, Song and Chew (1999), Velamparambil, Chew and Hastriter (2002)].

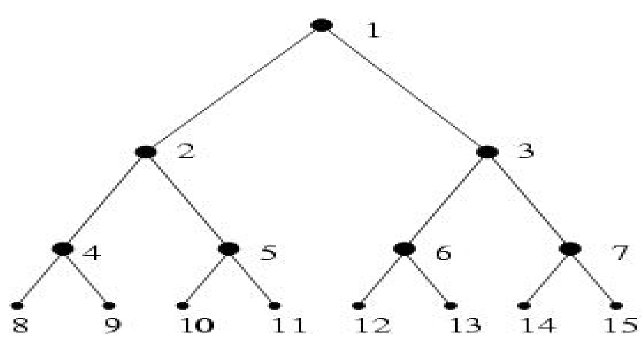

Figure 14 : The swelling of information occurs at the top of the tree. Hence if the tree is large, huge amount of information can flow through the "links" at the top of the tree.

The exorbitant communication cost in MLFMA comes about because of the physical nature of Helmholtz field [Chew (2002)]. A matrix-vector product transports information in a Helmholtz field through long distances if the scatterer is large. But the information content in a Helmholtz field does not diminish with distance. Hence, swelling of information occurs at the top of the inverted tree in Figure 14, very much like the case of a telephone trunk line with the links near the top of the tree carrying more information than the links near the bottom of the tree. A simple way to parallelize MLFMA is to split the workload according to the nodes of the tree [Rankine and Board (1994)]. For example, the workload of nodes 2, 4, 5, 8, 9, 10, 11 can be assigned to one processor, while the rest can be assigned to another processor. While this method works for Laplacian fields, it does not work well for Helmholtz fields. A large communcation cost arises because the information from node 2 to node 1 has to be sent across processors and this information can be rather large. Instead, we have adopted another strategy—a two-pronged approach as shown in Figure 15. Near the bottom of the tree, we split the workload according to the workload on each node. But near the top of the tree, the tree is replicated in each processor. However, each processor has only part of the radiation pattern of the nodes. Since each processor has all the nodes near the top of the tree, node-to-node information transfer occurs without any communication cost. Moreover, each processor is responsible for part of the information transfer process, amortizing the workload over different processors. By so doing, the communication cost is greatly reduced. We call the levels near the top part of the tree the shared levels, while those near the bottom of the tree, are called distributed levels. Figure 16 shows the parallel efficiency of ScaleME versus the number of processors when the scattering solution for a pencil-shaped target is being computed.

7 Fast Algorithm for Layered Media

The scattering of an object laying on top of a layered medium or embedded in a layered medium is a problem of great interest. The integral equation for scattering can be formulated in terms of the layered medium Green's function. However, an analytic closed form layered medium Green's function does not exist, and the solution is in terms of Sommerfeld integrals, which are numerically expensive to evaluate. Despite this complexity, we have successfully developed a fast algorithm called the fast inhomogeneous plane wave algorithm (FIPWA) to solve for the scattering solution of an object on top of a layered medium as well as below a layered medium [Hu and Chew (2000), Hu and Chew (2001)].

Figure 19 shows the current distribution on a tank when the tank is hanging in free space, and when it is sitting on

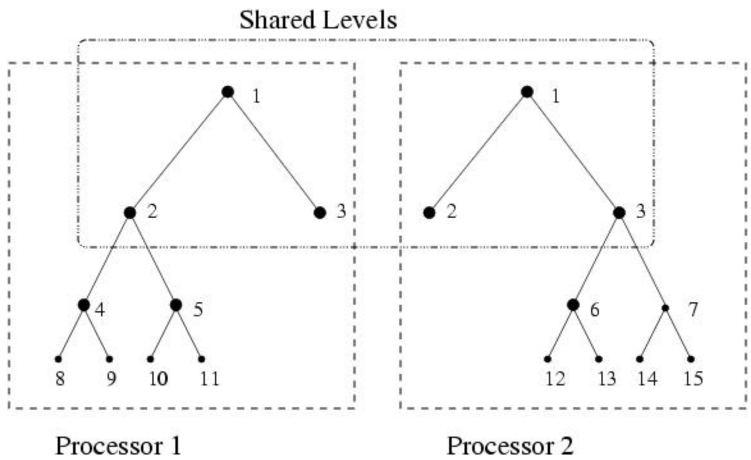

Figure 15 : The splitting of a tree into shared levels and distributed levels. The shared level portion of the tree is replicated in each processor, while the distributed levels are split according to the workload at each node. In this manner, the swelling of information near the top of the tree does not cause exorbitant communication cost.

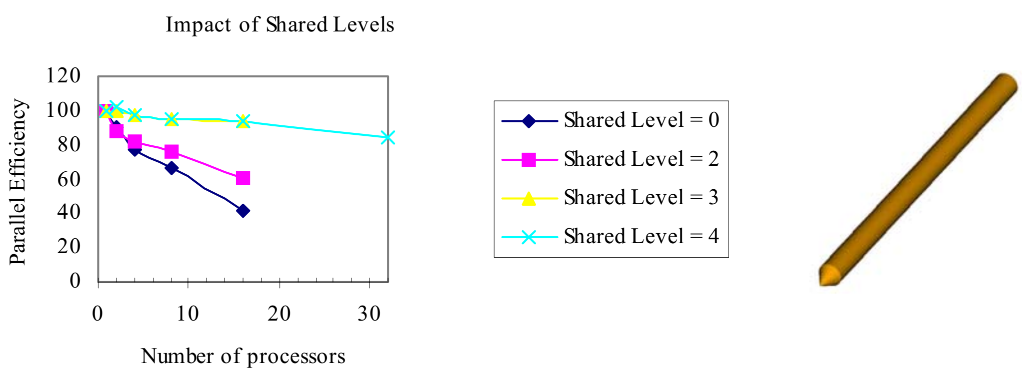

Figure 16 : A 3 m long pencil-shape target solved with ScaleME with different shared levels. The number of unknowns is 291,774, and the number of MLFMA levels is 9.

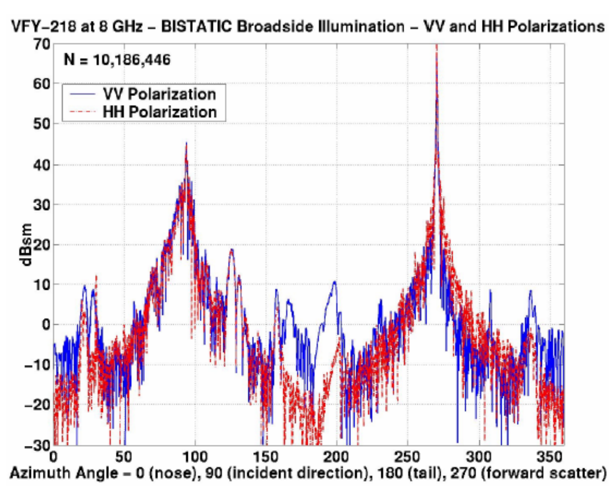

Figure 17 : A tour-de-force computation with ScaleME on a VFY 218 at 8 GHz. The scaleable code computes a matrix-vector product about 7 times faster than FISC for this scattering problem.

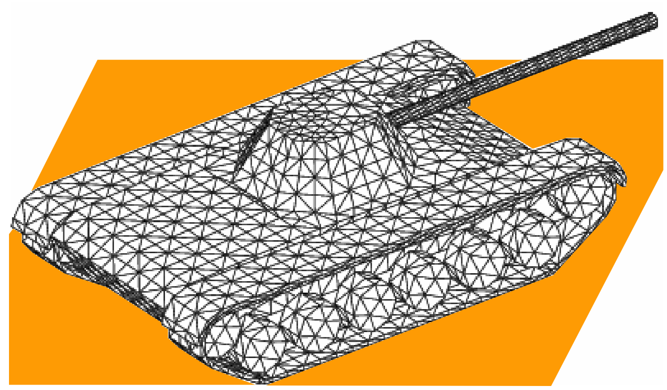

Figure 18 : Scattering of a large object on top of a layered medium. This is a relatively more difficult problem because the Green's function is expressed in terms of Sommerfeld integrals.

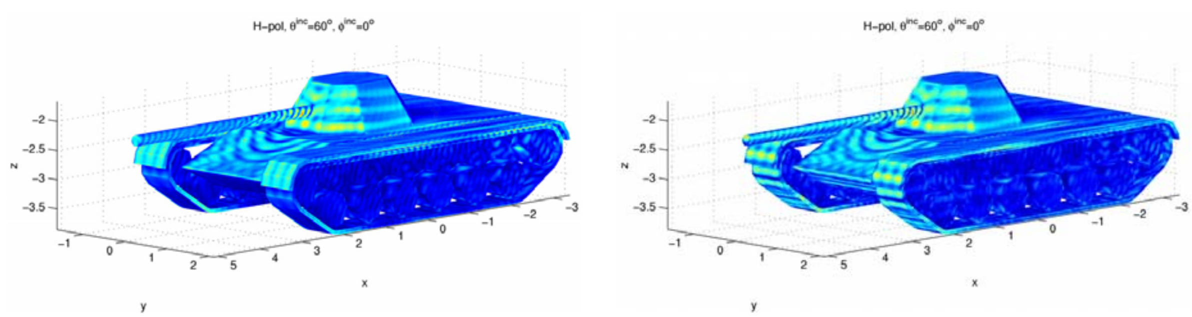

Figure 19 : Scattering solution of a full-size tank with 1,210,458 unknowns at 1.2 GHz using the fast inhomogeneous plane wave algorithm (FIPWA). Eight levels were used in this multilevel algorithm. The figure shows the current distribution on the tank when it is unaffected by the ground (left), and when it is sitting on the ground (right).

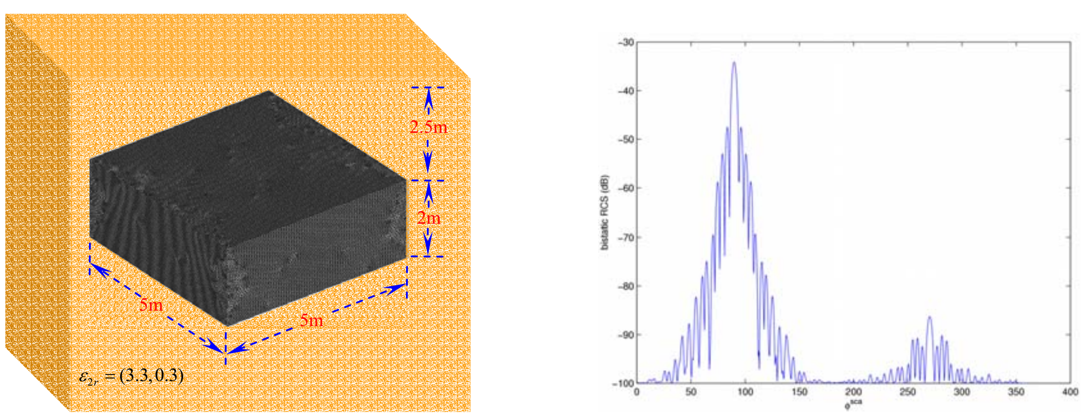

Figure 20 : The bistatic RCS of an underground bunker for vertical polarization. The incident angles are $\left(\theta^{inc},\varphi^{inc}\right) = (60^\circ. -90^\circ)$. The first strong peak corresponds to specular reflection, while the second weaker peak is back scattering. The frequency is 900 MHz, with 1,074,588 unknowns. Eight level FIPWA has been used.

	Solution time	Memory
FIPWA	18 hours (SGI Origin 2000)	3.94 GB
Full matrix (est.)	~11 years	~9.3TB

Table 2 : The resource for solving the buried bunker problem compared to traditional method for solving integral equation.

a ground. There is a noticeable difference in the current distribution on the caterpillar wheels of the tank for these two cases.

Figure 20 shows the application of FIPWA for the scattering solution of a buried bunker simulated with over 1 million unknowns. Table 2 shows the resources required by FIPWA compared to traditional method. It is clear that without fast integral equation solvers, simulation of problems of this size is not possible. Moreover, FIPWA for layered media is quite efficient. It consumes about 13 percent more memory and 10 percent more CPU time for the layered medium compared to free-space calculations.

In addition to the aforementioned works, it is to be noted that much work is ongoing in using global basis function to accelerate integral equation solution [Mittra and Prakash (2004)]. In addition, some activities are in the area of scaleable large scale computing [Namburu, Mark, and Clarke (2004)], as well as applications of fast solvers to other arena [Volakis, Sertel, Jørgensen, and Kindt (2004)].

8 Conclusions

Computational electromagnetics is itself a science that is a mélange of electromagnetics, mathematics, and computer science. Since electromagnetics is very central to electrical engineering and many of its associated technologies, computers will replace pencils and papers as the new age analysis tools. This is especially true due to the rapid progress in computer hardware and computational electromagnetics algorithms. Though solving a ten-million-unknown problem is in the realm of supercomputing presently, it will be a routine practice in the future, as we know that the supercomputers of today will become the desktop computers of tomorrow.

Engineers have insatiable appetites for high-resolution computing. Hence, in the future, we would like to see the development of even faster algorithms. Then advances in computer hardware technology will also amplify our capability to solve larger and more complex problems.

Many computational electromagnetics codes are run to study trends in computer-aided design. Even though high-accuracy computing is very important in certain critical technologies, low-accuracy and fast computing, which is also robust, is very important in the engineering design world. Hence, the role of approximate computing will also become important in the future.

While high-accuracy computing will demand mathematical rigor, approximate computing calls for engineering and physical intuition. Both problems are equally difficult and important.

References

Chew, W. C. (2002): Computational Electromagnetics – The Physics of Smooth Kernel Versus Oscillatory Kernel, and Wavelets Versus Fast Multipole, ACES Digest, (invited plenary lecture).

Chew, W. C.; Cui, T. J.; Song, J. M. (2002): A FAFFA-MLFMA algorithm for electromagnetic scattering, *IEEE Trans. on Antennas and Propagation*, accepted for publication.

Chew, W. C.; Jin, J. M.; Michielssen, E.; Song, J. M. (editors) (2001): Fast and Efficient Algorithms in Computational Electromagnetics, Artech House, Boston, MA. J. M. Jin, The Finite Element Method in Electromagnetics.

Cui, T. J.; Chew, W. C.; Chen, G.; Song, J. M. (2002): A FAFFA-RPFMA-MLFMA algorithm for largescale electromagnetic scattering, *IEEE Trans. Antennas Propagat.*, submitted for publication.

Forgy, E. A.; Chew, W. C. (2002): A time-domain method with isotropic dispersion and increased stability on an overlapped lattice, *IEEE Trans. Antennas Propagat.*, vol. 50, no. 7.

Harrington, R. F. (1982): *Field Computation by Moment Method*, Malabar, FL: Krieger Publ.

Hestenes, M. R.; Stiefel, E. (1952): Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stand*., Sect. B, vol. 49, pp. 409–436.

Hu, B.; Chew, W. C. (2000): Fast inhomogeneous plane wave algorithm for electromagnetic solutions in layered medium structures - 2D case, *Radio Science*, vol. 35, no. 1, pp. 31-43.

Hu, B.; Chew, W. C. (2001) Fast inhomogeneous plane wave algorithm for scattering from objects above the multi-layered medium, accepted by *IEEE Trans. Geosci. Remote Sensing*, vol. 39, no. 5, pp. 1028-1038.

Hu, B.; Chew, W. C. (2001): Fast inhomogeneous plane wave algorithm for scattering from objects above the multi-layered medium, accepted by *IEEE Trans. Geosci. Remote Sensing*, vol. 39, no. 5, pp. 1028-1038.

Jin, J. M. (1993): *The Finite Element Method in Electromagnetics*, New York: John Wiley \& Sons.

Liu, Q. H. (1998): The PSTD algorithm for acoustic waves in inhomogeneous, absorptive media, *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 45, no. 4, pp. 1044 -1055.

Liu, J.; Jin, J. M. (2001): A novel hybridization of higher order finite element and boundary integral methods for electromagnetic scattering and radiation problems, *IEEE Trans. Antennas Propagat.*, vol. 49, no. 11.

Lu, C. C.; Chew, W. C. (1994): A multilevel algorithm for solving boundary-value scattering, *Micro. Opt. Tech. Lett*., vol. 7, no. 10, pp. 466–470.

Mittra, R.; Prakash, V. V. S. (2004): The characteristic basis function method: a new technique for fast solution of radar scattering problems, *CMES: Computer Modeling in Engineering & Sciences*, vol. 5, no. 5, pp. 435-442.

Namburu, R. R.; Mark, E. R.; Clarke, J. A. (2004): Scalable electromagnetic simulation environment, *CMES: Computer Modeling in Engineering & Sciences*, vol. 5, no. 5, pp. 443-454.

Rankine, W. T.; Board, J. A. (1994): A portable distributed implementation of the parallel fast multipole tree algorithm, Tech. Rep. 95-002 , Dept. Elect. Eng., Duke University, Durham, NC.

Rokhlin, V. (1990): Rapid solution of integral equations of scattering theory in two dimensions, *J. Comp. Phys.*, vol. 86, pp. 414-439.

Song, J. M.; Chew, W. C. (1995): Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering, *Micro. Opt. Tech. Lett*., vol. 10, no. 1, pp. 14–19.

Song, J. M.; Chew, W. C. (2000): Large scale computations using FISC, *IEEE Antennas Propag. Soc.Int. Symp.*, Salt Lake City, Utah, vol. 4, pp. 1856–1859, July 16-21.

Song, J. M.; Chew, W. C. (2002): Broadband time-domain calculation using FISC, IEEE AP-S International Symposium, San Antonio, Texas, vol. 3, pp.552-555.

Song, J. M.; Lu, C. C.; Chew, W. C.; Lee, S. W. (1998): Fast Illinois solver code (FISC) *IEEE Antennas Propag. Magazine,* (invited), vol. 40, no. 3, pp. 27-34.

Taflove, A. (1995): *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Norwood, MA: Artech House.

Velamparambil, S. V.; Chew, W. C.; Hastriter, M. L. (2002): Scalable electromagnetic scattering computation, *IEEE Antennas Propagat. Symp*., San Antonio, Volume: 3 , 16-21, pp. 176.

Velamparambil, S. V.; Song, J. M.; Chew, W. C. (1999): A portable parallel multilevel fast multipole solver for scattering from perfectly conducting bodies," *IEEE Antennas Propagat. Symp.*, vol. 1, pp. 648-651.

Volakis, J. L.; Sertel, K.; Jørgensen, E.; Kindt, R. W. (2004): Hybrid finite element and volume integral methods for scattering using parametric geometry, *CMES: Computer Modeling in Engineering & Sciences*, vol. 5, no. 5, pp. 463-476.

Warnick, K.; Chew, W. C. (2001): High Frequency Asymptotic Representation of the Fast Multiple Method Translation Operator, *IEEE Antennas & Propagation International Symposium and USNC/URSI Meeting*, Boston, MA, p. 330, July 8-13, 200.

Yang, B.; Gottlieb, D.; Hesthaven, J. S. (1997): Spectral simulations of electromagnetic wave scattering, *J. Comp. Phys*., vol. 134, no. 2, pp. 216-230.