# A Parallel Multilevel Fast Multipole Algorithm for GRID Computing Allowing Optical Full-wave Simulations

Jan Fostier [1], [#] Femke Olyslager [1]
[1] *Department of Information Technology, Ghent University*
*Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium, femke.olyslager@intec.ugent.be*

**Abstract**

*We present a parallel multilevel fast multipole algorithm aimed at low cost GRID computer environments connected by a fast switch. The algorithm is a scheduling algorithm where work packets are handled in certain order to ensure minimal idle time of the processors and to avoid simultaneous bursts of communication between the processors. The algorithm is implemented on a method of moment discretisation of a two-dimensional TM electromagnetic scattering problem. Performance results are shown for lens objects 1000 wavelengths in size.*

## 1. INTRODUCTION

The Multilevel Fast Multipole Algorithm (MLFMA) has gained considerable interest during the past decade to solve very large electromagnetic scattering problems using surface integral equation techniques. For an introduction we refer the reader to [1]. The Method of Moments (MoM) technique applied to discretise an integral equation leads to a dense system of linear equations of dimension $N$ by $N$ if the number of discretisations is $N$. An iterative solution of this system where the matrix-vector products are evaluated with MLFMA allows for a solution with a time complexity of $O(N \log N)$ and a similar memory complexity. If the number of iterations can be limited then MLFMA allows for the solution of very large problems with limited computer resources.

The close to linear increase in complexity as a function of the number of unknowns means that the size of the problems that can be handled increases significantly when the available computer resources increase. The GRID technology allows for an easy and affordable construction of a parallel computer system with distributed memory. Parallelisation of electromagnetic simulation techniques has been a subject of considerable research in the past. However, it gained momentum after the introduction of GRID since affordable and very versatile parallel systems became available. For example the book [2] introduces the reader to GRID computing essentially focused on the Finite Difference Time Domain technique. GRID computing facilities come in very different configurations, ranging from a large set of computers widely distributed and connected through the "slow" internet or a cluster of similar computers connected through a "fast" switch. In each case each computer has its own memory. It is obvious that the communication between processors is an important aspect. The less communication is needed the higher the performance of the algorithm, this becomes an essential issue when considering a GRID facility depending on internet communication.

Parallel MLFMA has already gained some attention, we refer e.g. to [3]. In this contribution we will present a parallel MLFMA aimed at GRID computing facilities interconnected by a switch. Hence, communication between processors is fast but not as in a dedicated parallel computer. Our algorithm is based on a heuristic that tries to reduce idle time of processors and that tries to avoid overloading the switch by distributing communication between processors in time. Both the set-up stage as the matrix-vector product stage are parallelised.

Although the algorithm does not make any specific assumptions about the dimensionality of the problem we will apply it to a two-dimensional TM electromagnetic scattering problem. This problem is solved using a boundary integral equation based on Huygens principle that has already been developed in [4]. The far interactions in the resulting MoM matrix are evaluated with MLFMA and the near interactions are evaluated extremely accurate by not only evaluating the self-patch contributions analytically but also the neighbor-patches. Another application of this algorithm, although not parallelised, has been shown in [5] and [6]. Since we aim at simulating very large structures in terms of wavelengths we will use a high-frequency MLFMA. To accelerate the near interactions we will exploit all symmetries in the structure using a special symmetry extracting algorithm based on splay trees [7]. In this paper, with specific optical applications in mind, we will suffice by a simple almost diagonal preconditioner.

To show the performance of our algorithm we will show full-wave optical simulations of lenses with sizes of about 1000 wavelengths and coated with quarter-wavelength anti-reflection coatings. We will also show the perfomance of the algorithm as a function of the number of processors involved.

First the boundary integral equation and its MoM discretisation are introduced. Then we will discuss the MLFMA after which we will explain the heuristic used for the parallelisation. Finally, some examples will demonstrate the performance of the algorithm.

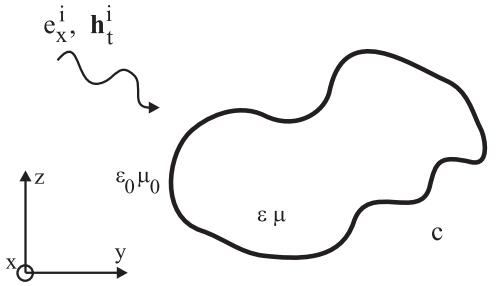


**Fig. 1:** Object illuminated by an incoming field.

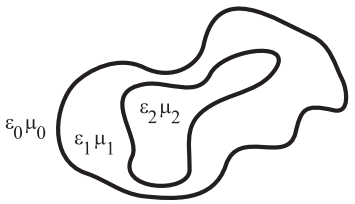


**Fig. 2:** Object in an object.

## 2. METHOD OF MOMENTS

Consider a cylindrical body (see Fig. 1), parallel to the $x$-axis, with material parameters $\epsilon$ and $\mu$ that is illuminated with a TM incoming field $e_x^i$, $\mathbf{h}_t^i$, where the subscript '$t$' indicates vectors in the $yz$-plane. We assume an $e^{j\omega t}$ time dependence. The boundary curve is denoted by $C$. The boundary integral equation can then be written as [4]

$$\lim_{\mathbf{r}\to C^-} \oint_C [e_x \frac{\partial G}{\partial n'} - \frac{jk^2}{\omega\epsilon} G h_t] dc'$$

$$= e_x^i - \lim_{\mathbf{r}\to C^+} \oint_C [e_x \frac{\partial G_0}{\partial n'} - \frac{jk_0^2}{\omega\epsilon_0} G_0 h_t] dc', \qquad (1)$$

$$\lim_{\mathbf{r}\to C^-} \oint_C [-\frac{j\omega\epsilon}{k^2} e_x \frac{\partial^2 G}{\partial n \partial n'} - \frac{\partial G}{\partial n} h_t] dc'$$

$$= h_t^i - \lim_{\mathbf{r}\to C^+} \oint_C [-\frac{j\omega\epsilon}{k_0^2} e_x \frac{\partial^2 G_0}{\partial n \partial n'} - \frac{\partial G_0}{\partial n} h_t] dc', \qquad (2)$$

with $k^2 = \omega^2\epsilon\mu$ and the Green function

$$G(\mathbf{r}|\mathbf{r}') = \frac{j}{4} H_0^{(2)}(k|\mathbf{r} - \mathbf{r}'|), \qquad (3)$$

and similarly for $k_0^2$ and $G_0$. $C^-$ and $C^+$ denote that the contour $C$ is approached from the inside and outside respectively. The unknowns are the tangential electric field $e_x$ and the tangential magnetic field $h_t$ to the contour $C$. These unknowns are equivalent to the equivalent magnetic and electric surface current densities. For simplicity we assumed a single object, the extension to multiple objects is straightforward. In particular we also consider objects embedded in other objects as is shown in Fig. 2.

The contour $C$ is divided into a number of segments on which pulse basis functions and overlapping triangular basis functions are defined. The pulse basis functions are used to expand $h_t$ and as test functions for (2). Conversely the triangular basis functions are used to expand $e_x$ and as test functions for (1). In this way a consistent Galerkin MoM is obtained (see also [4]).

In order to obtain a high accuracy the logarithmic singular part is extracted from the Green function in some cases. The basis and test function integrations for this logaritmic part are evaluated analytically for self-patch integrations (i.e. the basis and test functions are defined over the same segment) and for neighbor-patch integrations (i.e. the basis and test functions are defined over adjacent segments). Especially the last ones require tedious analytical calculations. For more details we refer to [6]. The other basis and test function integrations are evaluated using Gaussian quadrature rules.

Finally, the MoM results in a linear system of unknowns

$$Z \cdot X = B, \qquad (4)$$

with $X$ a vector containing the unknown expansion coefficients of $e_x$ and $h_t$ in triangular and pulse basis functions respectively. $Z$ is the interaction matrix and $B$ is a vector representing the tested incoming fields. This system is solved iteratively using the TFQMR algorithm [8].

## 3. NEAR INTERACTIONS - SPLAY TREES

Often large structures contain symmetries where two pairs of interacting segments are geometrically equal. This obviously means that the corresponding two elements in the interaction matrix $Z$ are equal. Computing time can be saved if these symmetries are recognized, the corresponding interactions only need to be calculated once. If one has $N$ segments then there are $N^2$ interactions. A brute force method comparing all these interactions to each other would lead to $O(N^4)$ computing time, which obviously would jeopardise the whole algorithm. Even if one restricts symmetry extraction to the near interactions this would still lead to $O(N^2)$ computing time overwhelming the $O(N \log N)$ complexity of MLFMA. To avoid this we extract symmetry for the near interactions using a splay tree [7]. The splay tree will require somewhat more memory but yields a drastic reduction in the set-up time [6].

## 4. FAR INTERACTIONS - MLFMA

The number of unknowns in a scattering problem can be high for two reasons. First the structure can be large measured in wavelengths and at least 10 to 30 unknowns per wavelength are needed depending on the required accuracy. Second the geometry could contain small geometrical details requiring a fine discretisation to resolve. The high-frequency MLFMA, based on plane wave expansions, is especially suited to tackle structures that are many wavelengths in size. On the other hand a multipole based low-frequency MLFMA is suited for small but complex structures with many unknowns per wavelength. Both techniques can be combined. However, in our

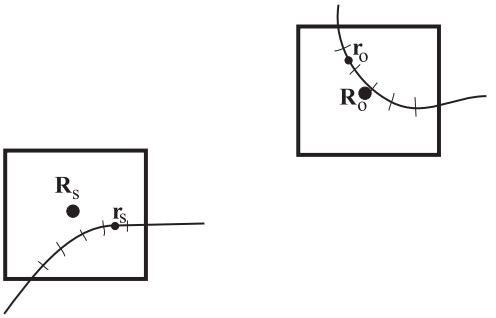


**Fig. 3:** Two interacting groups in MLFMA.

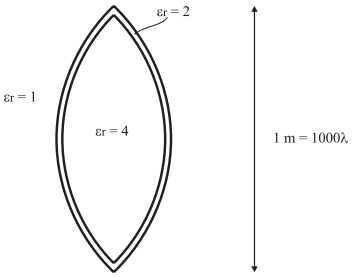


**Fig. 4:** Coated lens.

approach we mainly focus on problems that are (extremely) large compared to wavelength motivating the high-frequency MLFMA.

The MLFMA is a fast technique to evaluate the fields in a set of $N$ points due to $N$ sources located in those same points. We will not go into details of this technique and restrict ourselves to the main characteristics (for a detailed account see e.g. [1]). The segments are grouped into a number of groups and instead of calculating the individual interactions between each two segments the interactions are calculated groupwise. This is illustrated in Fig. 3 and mathematically expressed through the following expansion of the Green function

$$H_0^{(2)}(k|\mathbf{r}_o - \mathbf{r}_s|)$$

$$\approx \sum_{n=-Q}^{n=+Q} [e^{j\mathbf{k}(\phi_n)\cdot(\mathbf{r}_s-\mathbf{R}_s)} T_n(\mathbf{R}_o - \mathbf{R}_s) e^{-j\mathbf{k}(\phi_n)\cdot(\mathbf{r}_o-\mathbf{R}_o)}], \quad (5)$$

where $\mathbf{R}_s$ and $\mathbf{R}_o$ respectively are the centers of the source and the observation group and where $\mathbf{r}_s$ and $\mathbf{r}_o$ are two arbitrary points on the segments in the source and observation groups. $\mathbf{k}(\phi_n)$ are wavevectors along equidistant directions $\phi_n$, $n = -Q, ..., 0, ..., Q$ and $T_n$ is the translation operator given by

$$T_n(\mathbf{R}) = \frac{1}{2Q+1} \sum_{n'=-Q}^{n'=+Q} H_{n'}^{(2)}(k|\mathbf{R}|) e^{jn'(\Phi-\phi_n-\pi/2)}, \quad (6)$$

with $\Phi$ the angle between $\mathbf{R}$ and the $y$-axis.

The first part (the desaggregation) in (5) depends only on the segments in the source group and is an expansion of plane waves along directions $\phi_n$. The second part depends only on the centers of the observation and the source group and represents the translation of plane waves. The third part (the aggregation) depends only on the segments of the observation group and is an aggregation of incoming plane waves.

In the MLFMA this is extrapolated hierarchically by grouping groups into groups and so on. In that way all the interactions, normally requiring a computational complexity of $O(N^2)$ can be reduced to $O(N \log(N))$.

## 5. Parallelisation Heuristic

The MLFMA divides the whole structure in a grid of hierarchic squares. Using a space filling curve these squares are allocated to the different processors such that near squares are as much as possible allocated to the same processor. Of course, the geometries under consideration will encompass many empty squares that are eliminated. The load over the processors is balanced. This is a rather straightforward procedure.

The workload is divided in small packets, involving the calculation of near interactions, aggregations, translations or desaggregations These packets are stacked in a certain order. The main principle is to handle first those packets that might result in data that another processor might need. If for a certain packet information from another processor is needed that is not yet available then it will be handled later. As soon as information is available for another processor that information will be made available to be sent to the other processor and will be sent as soon as the other processor is ready to accept it. Near interactions usually require no communication between processors and are postponed as long as possible to a moment where the processor has nothing else to do, hence near interactions get lowest priority.

This heuristic has the advantage that communication does not come into bursts which would overload the switch and which would result in waiting cycles in the processors. Of course defining and sceduling all the work packets is a complex task.

Also the set-up stage is parallelised. The splay tree is divided over the processors, each having its own tree of a smaller size than the total tree.

## 6. Numerical Examples

All simulations are performed on a system consisting of eight (four times two cores) 64 bit AMD Opteron 270 processors running at 2 GHz with 2 GByte of RAM for each processor. An 1 Gbit/s switch interconnects the four machines with two processors. Possible fast communication between two cores in the same machine was not exploited. Parallelisation and communication was implemented using the Message Passing Interface (MPI) [9].

The basic geometry that we will consider is a lens as shown in Fig. 4. We use a frequency of 299.792 THz corresponding to a wavelength of 1 mm. The lens has a diameter of 1 m, hence 1000 wavelengths. The relative permittivity of the lens is 4. The lens consists of circular cylindrical parts with a curvature radius of 1 m. We will also consider a lens coated with a

quarter wavelength transformer. The coating has a relative permittivity of 2 and a thickness of $0.25/\sqrt{2}$ mm. The lens is illuminated by a Gaussian beam with a waist of 30 cm diameter at the center of the lens.

The number of unknowns is 84000 for the uncoated lens and 144000 for the coated lens. The number of unknonws does not double because the discretisation of the external coating surface can be somewhat larger than the discretisation of the internal coating surface. The reason for this is that the wavelength in the coating is a factor $\sqrt{2}$ larger than the wavelength in the lens.

The set-up time for the uncoated lens is 5.12 s whereas the set-up time for the coated lens is 61.24 s. Due to the circular symmetric nature of the lens the number of equal interactions is high. However, we note a considerably higher set-up time for the coated lens because there are many more near interactions that differ from each other. Almost all near interactions between a segment on the external coating surface and a segment on the internal coating surface are different from each other due to the different discretisation density used on both surfaces.

Using the almost diagonal preconditioner the number of iterations is 300 for the uncoated lens and 336 for the coated lens where the precission of the TFQMR algorithm was set to $1e^{-5}$. The total solve time was respectively 101.56 s and 195.52 s mainly explained due to the larger number of unknowns. The low number of iterations in both cases is a manifestation of the simplicity of the physics in the problem, there are but few interference and resonance phenomena making the lens problem a very well-conditioned problem. Also the type of integral equation used leads to a well-conditioned problem. Without preconditioner the number of iterations is respectively 1519 and 751 which requires a CPU-time of 523.47s and 444.47s respectively. This is an interesting observation. The uncoated lens contains more interference which manifests itself in more iterations. These interferences are of a local nature which means that the preconditioner can capture this part of the physics. We also note that the accuracy of the evaluation of the near interactions and in particular of the self-patch and neighbor-patch interactions is decremental to limit the number of iterations.

In Fig. 5 a density plot of the electric field amplitude is shown for the uncoated lens. The area shown is 2 m by 2 m in size and contains 1000000 pixels that were calculated in 84.93 s using the parallel MLFMA algorithm. Fig. 6 shows the same region for the coated lens requiring 85.97 s. Note the higher field density in the focal point in Fig. 6 compared to Fig. 5. In Fig. 7 and Fig. 8 for both cases a close up is shown for a region of 1 cm by 1 cm just at the surface of the uncoated and coated lens respectively. Again 1000000 pixels were calculated now requiring 1184.05 s and 1087.47 s. The reason of this significantly increased CPU-time is because the number of near pixels is much higher yielding less benefit from MLFMA. The higher standing wave ratio in front of the lens is clearly visible in the uncoated lens compared to the coated lens. This is even more profound if we look at the electric field along a line through the axis of the lens as shown in Fig. 9. Note that the standing wave ratio inside the lens also is lower in the coated lens compared to the uncoated lens. The high accuracy of the simulations is also clearly visible by considering the continuity of the field at the interfaces.

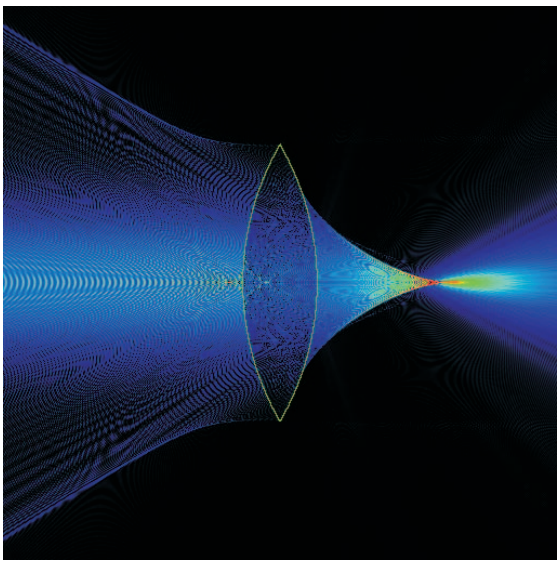

**Fig. 5:** Field density in the uncoated lens.

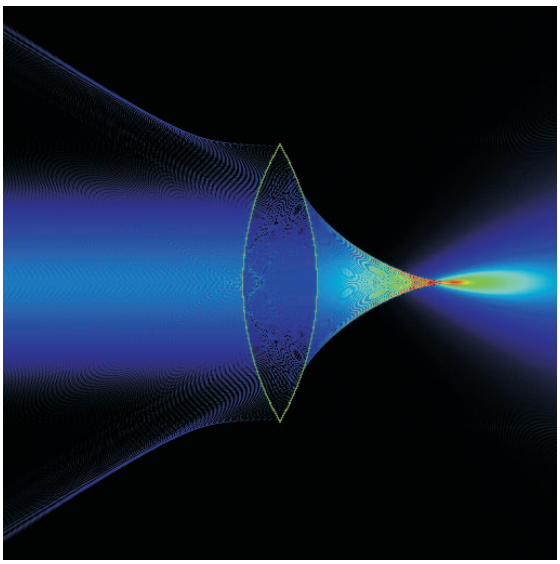

**Fig. 6:** Field density in the coated lens.

Finally we consider an efficiency study of the algorithm as a function of the number of processors. We consider the coated lens case. The uncoated lens gives similar results. Fig. 10 shows the efficiency as a function of the number of processors used. The matrix-vector multiplication efficiency slowly decreases as the number of processors increases but it is still 83% for eight processors which is very reasonable.

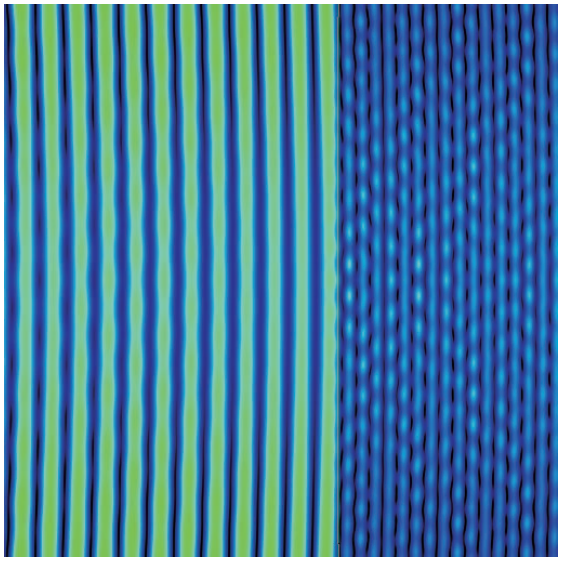

**Fig. 7:** Close up of the field density in the uncoated lens.

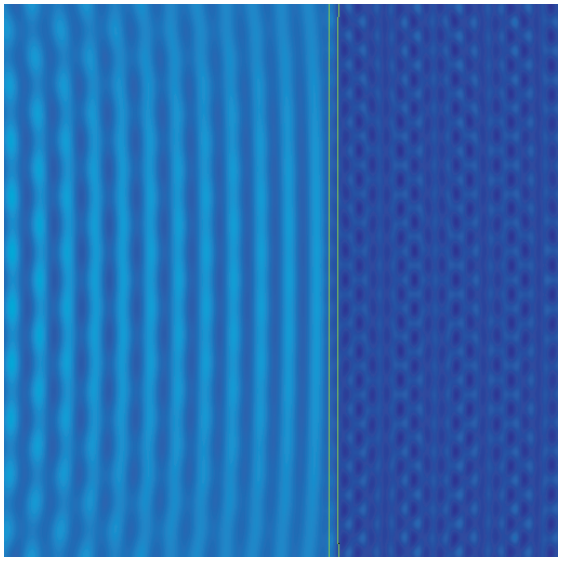

**Fig. 8:** Close up of the field density in the coated lens.

E.g. the best result in [3] had an efficiency of 81% for eight processors but these were obtained on part of one node of an IBM supercomputer p690 model 681 allowing extremely fast communication between processors. The efficiency of the set-up time becomes higher than 100% when four or more processors are used. This is due to the fact that the splay tree becomes smaller in each processor when the number of processors increases making the algorithm more efficient. Obviously this means that our splay tree implementation when used on only a few processors could be more efficient.

The reader might argue that the number of unknowns used in the example is not so high. At the time of the conference

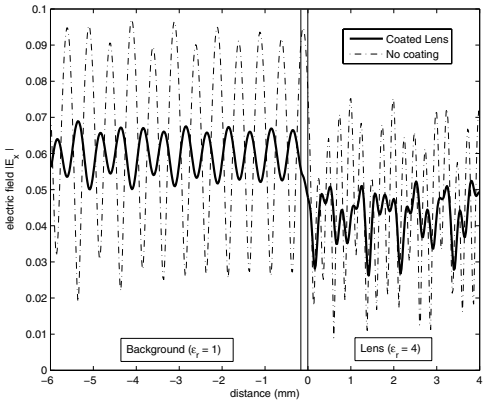


**Fig. 9:** Electric field along a line through the axis of the lens.

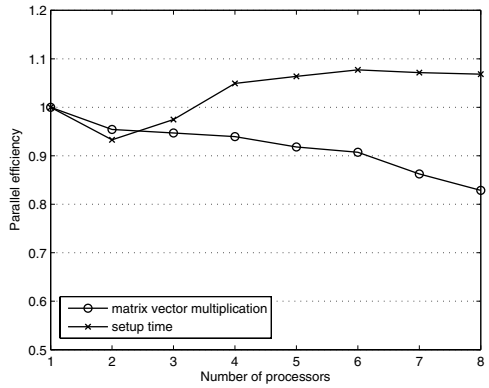


**Fig. 10:** Efficiency as a function of the number of processors.

we will show results containing several million of unknowns. The largest example we have calculated now has close to two million unknowns. We could also have calculated a ten times larger lens but this would not provide new information and it would become impossible to simulate the uncoated lens on a single processor.

## 7. Conclusions

We have presented a parallel MLFMA algorithm allowing very high parallelisation efficiency on a cheap GRID cluster connected by a fast switch. The algorithm was applied to a two-dimensional TM scattering problem allowing the simulation of structures of 1000 wavelengths in size and about 150000 unknowns in less than 5 minutes on eight 2 GHz processors.


## Acknowledgment

J. Fostier has a grant from the IWT, Belgium.


## References


[1] W. C. Chew, J. M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*, Boston: Artech House, 2001.

[2] L. Taricone and A. Esposito, *Grid Computing for Electromagnetics*, Boston: Artech House, 2004.
[3] F. Wu, Y. Zhang, Z. Z. Oo, and E. Li, "Parallel multipole fast multipole method for solving large-scale problems", *IEEE Antennas and Propagation Magazine*, Vol. 47, No. 4, 2005, pp. 110-118.
[4] F. Olyslager, D. De Zutter, and K. Blomme, "Rigourous analysis of the propagation characteristics of general lossless and lossy multiconductor transmission lines in multilayered media," *IEEE Trans. Microwave Theory and Techn.*, Vol. MTT-41, No. 1, 1993, pp 79-88.
[5] F. Olyslager and J. Fostier, "Fast and accurate evaluation of the shielding effectiveness of complex enclosures," In: *17th International Zurich Symposium on Electromagnetic Compatibility*, Singapore, 2006.
[6] J. Fostier and F. Olyslager, "Fast and accurate evaluation of the shielding effectiveness of complex enclosures," Submitted to *IEEE Trans. Electromagnetic Compatibility*.
[7] Sleator and Tarjan, "Self-adjusting binary search trees," *JACM*, Vol. 32, 1985, pp 652-686.
[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1987.
[9] M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra, *MPI: The Complete Reference*, Cambridge, MA, The MIT Press, 1996.