Efficient domain decomposition methods for electromagnetic modeling of scattering from forest environments

Ines Fenni

Université Pierre et Marie Curie

École doctorale: SMAER

Sciences Mécaniques, Acoustique, Électronique et de Robotique de Paris

Laboratoire d'Électronique et Électromagnétisme (L2E)

Thèse de Doctorat en Électronique

Efficient Domain Decomposition Methods for Electromagnetic Modeling of Scattering from Forest Environments

---

Présentée par

Ines Fenni

Soutenue publiquement le 23 Octobre 2014
devant le jury composé de:

M. Christophe BOURLIER	*Directeur de recherche, CNRS*	Rapporteur
M. Lionel PICHON	*Directeur de recherche, CNRS*	Rapporteur
M. Philippe POULIGUEN	*Responsable scientifique, DGA*	Examinateur
M. Zhuoxiang REN	*Professeur, UPMC*	Examinateur
M. Xavier JUVIGNY	*Chercheur, ONERA*	Examinateur
Mme. Hélène ROUSSEL	*Professeur, UPMC*	Directrice de thèse
M. Raj MITTRA	*Professeur, PennState Univ.*	Co-Directeur de thèse
Mme. Muriel DARCES	*Maître de conférences, UPMC*	Co-Encadrante de thèse

To my beloved parents, husband, sisters and brother ...

“

Light thinks it travels faster than anything but it is wrong.No matter how fast light travels, it finds the darkness has always got there first, and is waiting for it.

”

Terry Pratchett

Acknowledgements

I would like to express my absolute and sincere gratitude toward my supervisors, Prof. Hélène Roussel (L2E, University of Pierre and Marie Curie, France), Dr. Muriel Darces (L2E laboratory, University of Pierre and Marie Curie, France) and Prof. Raj Mittra (EMC laboratory, Pennsylvania State University, USA), for their valuable advise and their continued and significant support during the entire period of this work.

I particularly wish to thank Hélène and Muriel for the confidence and trust they have placed in me, three years ago, when they have decided to offer me the great opportunity to work under their supervision. I am sincerely thankful for the quality of guidance provided by both of them. They always believed in me, and they strongly encouraged and supported me even in the difficult times of my PhD. I more than acknowledge that I see them as two amazing persons from both the human and professional points of view, and two exemplary female role models who are highly committed to scientific research and education. I will always be grateful to Hélène and Muriel for this excellent accompanying of my first attempt at scientific research and particularly computational electromagnetics.

Once again, my sincere thanks to Prof. Mittra for his ongoing support, for his time and for his expert CBFM advice. It has been a great honour for me and a pleasure to be working under his guidance. I would like particularly to thank him warmly for inviting me to the EMC laboratory. I thoroughly enjoyed this great experience, and was really pleased to have met the great people of the EMC lab, and I would like, at this point, to thank Chiara, Giacomo, Mohamed, Ginni and Shawn for the warm welcome I received from them during my stay in State College.

I am absolutely thankful for the education and training offered by the brilliant research teams during my research master's degree at the University of Western Brittany (UBO). The high interest I have today in the electromagnetic propagation domain began in Brest thanks to the motivation, enthusiasm and passion transmitted to me by the researchers of the UBO and Telecom Bretagne. I acknowledge that this master's degree was undoubtedly the very first step in my scientific research career.

Warm regards and thanks go to Dr. Xavier Juvigny and Prof. François Xavier Roux for the interest they have shown in my PhD research work, and for the sterling work and the cooperation that we have enjoyed in relation with the comparison between the CBFM and the ACA. I am looking forward to continuing this collaboration and to materializing it through a joint journal publication.

So many people have been of significant help throughout my PhD at the L2E lab of the University Pierre and Marie Curie (UPMC). Without their valuable support, carrying this work through to a successful conclusion would not have been possible. My sincere gratitude to Prof. Pascal Joly (Laboratory Jacques Louis Lion, UPMC), Mr. Michel Krawczyk (Information Systems Department, UPMC) for the invaluable help and for the amount of time they offered to me to get over the multiple Fortran, OpenMP and MPI perils I have met during my PhD. Warm thanks to Mrs. Isabelle Dupays and the other engineers of the institute of scientific and computing resources (IDRIS) for the enriching MPI training they provided to me and for continuing technical support.

My sincere gratitude is extended to the jury members : Prof. Pichon, Prof. Bourlier, Mr Pouliguen, Prof. Ren and Dr. Juvigny, for accepting to be a part of my PhD thesis and for carefully reading my dissertation.

I acknowledge that this work is supported by the French Agence Nationale de la Recherche (ANR) and the French Direction Générale de l'Armement (DGA) under the Project MOBILE.

My heartfelt thanks to my colleagues and all members of the laboratory of Electronic and Electromagnetics (L2E) where I spent a very pleasant three years thanks to a friendly and supportive work environment. Warm thanks to Sana, Zhi, Solofo, the two Nicolas, Takfarinas, Shehab, Chawki, Ourouk, Lydia for their friendship, their support and their encouragements in recent weeks. Many thanks to Chloé, Annie, Yves and all the support technical team of the L2E lab. I am sincerely grateful to the entire staff of the L2E laboratory for the amazing time I spent with them, for the impassioned and lively discussions I had with each of them and, above all, for the family spirit that exists among all of them.

Last but not least, I particularly wish to thank my parents, my husband, my sisters, my brother and my friends for their invaluable support and their encouragements, especially during the most difficult times. I am very grateful to my dear husband Amine for his constant support and his steadfast patience. Finally, I acknowledge that whatever I am today, it is due to the blessings of my father and my mother.

Contents

Chapter 1

Introduction

This first chapter sets out the context of the research work conducted in this thesis, which is electromagnetic modeling of scattering from forest and indicates the main scientific and technical challenges to be met. We present here the existing 3D full-wave model, developed previously in our lab (L2E : Laboratory of Electronic and Electromagnetism) in order to pinpoint its weaknesses and explain its limitations. Then, we justify the choices made to improve the performances of this previous model, by a quick comparison between the direct and iterative methods. Next we address the novel elements and the main contributions of this new research work. We conclude this chapter with the thesis outline.

1.1 Context of Application

Forests represent a large part of the earth vegetation coverage. They play a major role in the climate changes and in the global carbon cycle. Indeed, the evolution of terrestrial biosphere is the least understood component of the carbon cycle. The greatest uncertainties concern the location and temporal variation in carbon pools. The forest biomass (mass of the above ground portion of live trees per unit area) is identified as a crucial information to acquire, since forests are the greatest contributor to the terrestrial biomass. The contribution of each forest depends mainly on its size and on the number and species of trees that compose it.

In order to identify the contribution of each forest to the global carbon cycle, researchers have been deeply interested into the electromagnetic analysis of forest scattering. Hence, several studies have been conducted to establish a link between the backscattered radar signals and the biomass contained in the forest under observation [1–6]. They showed that the estimation of the biomass is highly dependent on the frequency used to observe the forest, and the key conclusion drawn was that lower frequencies are best suited to this purpose (particularly P and VHF bands). Moreover, they found out that, for these bands of frequency, the main factors contributing to forest backscatter are the trunks and primary branches. The contribution of leaves, secondary branches and needles to the measured signal can be neglected because of the fact that their dimensions are small compared to the incident wavelength.

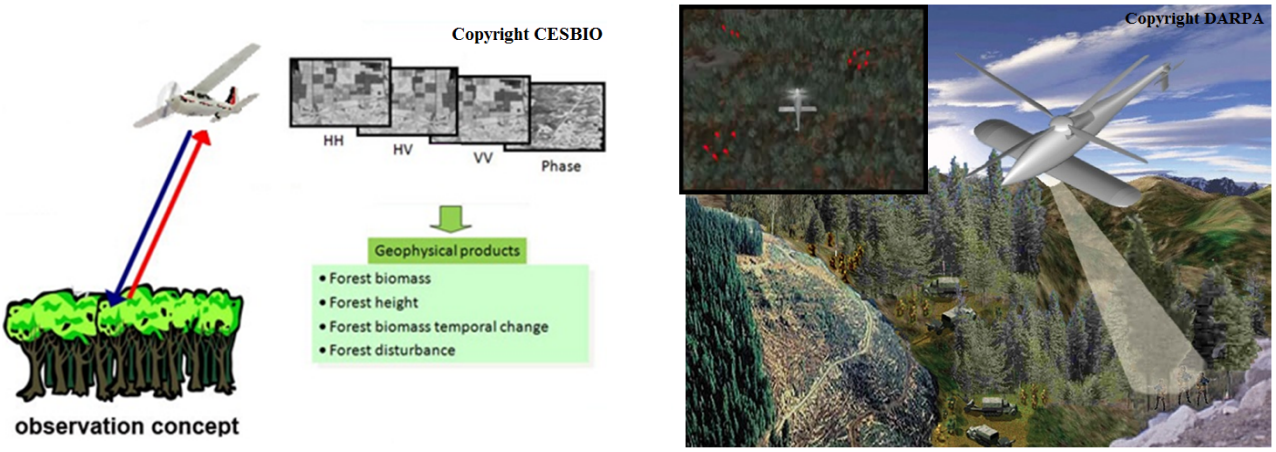
The polarization is also recognized as an important factor for the evaluation of the forest biomass, since the HV channel is more sensitive to the variations of this biomass. These studies are based on the interpretation of experimental data (space/airborne) collected for forest areas whose descriptive parameters, both geometric and electromagnetic, are not always known. The techniques developed for this interpretation are often based on polarimetric and interferometric treatments combined with scattering models based on a simplified representation of the forest (group of cylinders, ellipsoids and spheres). They are also often dedicated to mono-static radar configurations and are not necessarily well adaptable to bi-static ones.

On the other hand, numerous research teams have been interested in the developments of "exact" approaches, or empirical models for the analysis of forest cover. Two types of methods are commonly used for modeling forest radar backscatter : the first approach is based on the radiative transfer theory (RT) and the second approach uses the modified Born approximation (DBA). Models using the radiative transfer theory do not preserve the phase information, since they rely on the transport of energy in the medium. The most famous model, representative of this approach, is MIMICS (Michigan Microwave Canopy Scattering) developed at the University of Michigan [7], while in the other side, [8] is a good reference about the models using the modified Born approximation.

Several studies have also demonstrated that, when we focus on low frequencies (VHF-UHF), we can legitimately represent a complex forest under radar observation, only by the larger elements of the illuminated area (trunks and primary branches), since the smaller ones (secondary branches and foliage) will contribute weakly to the backscattered fields. As example of operational radar in this frequency band, we can mention here is CARABAS developed by the FOA (Swedish Research Establishment) (See Figure 1.1b). [9–12].

The approach proposed by our laboratory L2E (Laboratory of Electronics and Electromagnetism) is quite different in the sense that, initially, the goal was to understand the interactions between an electromagnetic plane wave and the basic elements of the forest (trunk, primary branches and ground). Also, we are not only interested in the geophysical application of forest scattering modeling but also we also want to investigate the military aspect of the forest observation by introducing later a target under the forest cover. The military application is better known under the name of FoPen for Foliage Penetration, and has been the subject of numerous research works [13–18] (see Figure 1.1b).

Hence, a rigorous and coherent approach has been adopted with the previous PhD conducted in the L2E and defended in 2010 by Sami Bellez [19] under the direction of Prof. Roussel. This approach results in a 3D full-wave model based on the integral volumetric representation of the electric fields (EFIE). This model aims to focus on the analysis of bi-static scattering mechanisms by a forest in the VHF and UHF bands. With this 3D full-wave model, no mathematical simplifying approximation was used while solving the volumetric integral representation of the electric field, in order to take into account all possible interactions among the scatterers and between the scatterers and the ground. In the next paragraph, we provide further details about this 3D previous full-wave model.



(A) BIOMASS (B) FORESTER

FIGURE 1.1: Two examples of forest scattering applications : On the left, the project BIOMASS conducted by the European Space Agency (ESA) [20] and on the right FORESTER, the airborne radar system developed by the American Defence Agency DARPA [17, 18].

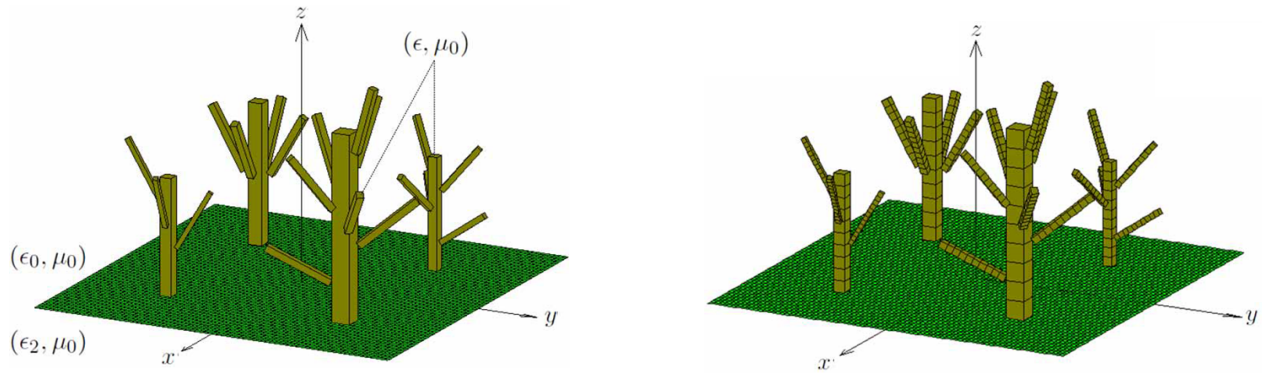
1.2 Existing Forest Scattering Model

The previous research works [19, 21–23] aimed to examine the interactions of an electromagnetic plane wave with a simplified model of a forest medium in a frequency range of 100 to 400 MHz (UHF/VHF). In the current subsection, we recall simply the theoretical description of the model, including the forest representation and the integral representation solved by the MoM.

1.2.1 Forest representation

As shown in Figure 1.2, the trees are described by dielectric vertical and tilted cylinders of square cross section (in a way to discretize it by cubic cells), representing trunks and main branches, respectively. The trees are placed over a horizontal plane separating two semi-infinite homogeneous media, which are the air and the forest ground. The effects of leaves, needles, and the roughness of the soil are ignored in the frequency band under consideration. The orientation of each branch is described by two angles: β (elevation angle) and α (azimuth angle).

As explained earlier, In order to rigorously characterize the interactions of the forest with a plane wave of arbitrary polarization, the 3-D full-wave model is based on the volumetric integral representation of the electric field by using the dyadic Green's function of a two-layered medium [19, 24, 25].



(A) Geometry of the forest (B) Dicretization

FIGURE 1.2: Geometry of the forest before and after discretization. ϵ_1 and ϵ_2 are the free-space and ground permittivities, respectively, and μ_0 is the free-space permeability.

1.2.2 Full-wave model Based on the Integral Representation

The formulation considered previously in [19, 24] is based on the electric-fields integral representation whose kernel is the Green's function of a two layered medium. The total field, $\overline{\mathbf{E}}_t$, at any point $\overrightarrow{r}$ is composed of the reference $\overline{\mathbf{E}}^{\text{ref}}$ and the scattered $\overline{\mathbf{E}}^{\text{s}}$ fields,

$$\overline{\mathbf{E}}_t = \overline{\mathbf{E}}^{\text{ref}} + \overline{\mathbf{E}}^{\text{s}} \tag{1.1}$$

Where $\overline{\mathbf{E}}^{\text{ref}}$ is the field present when the trees are removed and is the coherent summation of the incident field, $\overline{\mathbf{E}}^{\text{i}}$, and the field reflected from the ground, $\overline{\mathbf{E}}^{\text{r}}$,

$$\overline{\mathbf{E}}^{\text{ref}} = \overline{\mathbf{E}}^{\text{i}} + \overline{\mathbf{E}}^{\text{r}} \tag{1.2}$$

The scattered field, $\overline{\mathbf{E}}^{\text{s}}$, is due to the trees occupying a domain Ω made up of the trunks and branches, and is related to the internal field inside Ω, noted $\overrightarrow{E}^t(\overrightarrow{r}')$.

$$\overline{\mathbf{E}}^{\text{s}} = [\nabla\nabla. + k_1^2] \int_{\Omega} \Delta\epsilon(\overrightarrow{r}')\overline{\overline{G}}(\overrightarrow{r}, \overrightarrow{r}')\overrightarrow{E}^t(\overrightarrow{r}')d\overrightarrow{r}' \tag{1.3}$$

Here, $\Delta\epsilon(\overrightarrow{r}')=\frac{\epsilon(\overrightarrow{r}')-\epsilon_0}{\epsilon_0}$ is the permittivity contrast at the location $\overrightarrow{r'} \in \Omega$, k_1 is the wavenumber in air, and $\overline{\overline{G}}(\overrightarrow{r}, \overrightarrow{r}')$ is the dyadic Green's function of the two layers stratified media.
To compute $\overline{\mathbf{E}}_t$, we first have to determine $\overrightarrow{E}^t(\overrightarrow{r}')$ in Ω by solving equation (1.1) when $\overrightarrow{r} \in \Omega$. This equation is solved by means of the method of moments. Once $\overrightarrow{E}^t(\overrightarrow{r}')$ in Ω is determined, $\overline{\mathbf{E}}^{\text{s}}$ at all points $\overrightarrow{r}$ above the ground interface could be calculated.

1.2.3 The Method of Moment

The integral equation given by (1.1) is solved in [19, 24] by using a Method of Moment. The trees are discretized, as shown in Figure 1.2b, into elementary cubic cells small enough to consider that the

field inside is constant. The cell size is equal to or lower than $\lambda_s/10$, where λ_s is the wavelength inside the scatterers (see Appendix A). Then the equation (1.1) reduces to a system of $3N$ linear equations given by:

$$\sum_{q=1}^{3}\sum_{n=1}^{N}(\delta_{mn}\delta_{pq}-\overline{\overline{I}}_{pq}^{mn})\vec{E}_q^t(\vec{r}_n)=\vec{E}_p^{ref}(\vec{r}_m) \tag{1.4}$$

Where

$$m,n=[1...N] \quad ; \quad p,q=[1...3]$$

$$\delta_{mn}=1 \quad if \quad m=n \quad else \quad \delta_{mn}=0$$

$$\delta_{pq}=1 \quad if \quad p=q \quad else \quad \delta_{pq}=0$$

$$\overline{\overline{I}}_{pq}^{mn}=[\nabla\nabla.+k_1^2]\Delta\epsilon(\vec{r}_n)\int_{V_{\text{cell}}}\overline{\overline{G}}_{pq}(\vec{r}_m,\vec{r}'_n)d\vec{r}'_n$$

is the interaction square matrix of size $3N$. q and p are the three components x, y or z of the fields, N is the total number of discretizing cells, m and n are respectively the indices of an observation and source cells, as well as $\vec{r}_m$ and $\vec{r}_n$, are respectively the coordinates of the centers of cells m and n. Once the elements of the interaction square matrix and the reference field vector are calculated, the unknown internal field inside the dielectric cells $\vec{E}_q^t(\vec{r}_n)$ can be found by solving (1.4). The scattered field is obtained by using (1.3) when $\vec{r}_m$ is the location of the observation point (position of the receiving antenna):

$$\begin{aligned}\vec{E}_p^s(\vec{r}_m)=&[\nabla\nabla.+k_1^2]\times\\&\sum_{q=1}^{3}\sum_{n=1}^{N}\Delta\epsilon(\vec{r}_n)\vec{E}_q^t(\vec{r}_n)\int_{V_{\text{cell}}}\overline{\overline{G}}_{pq}(\vec{r}_m,\vec{r}'_n)d\vec{r}'_n\end{aligned} \tag{1.5}$$

At the end of this subsection, it is interesting to mention that this model was validated by comparing the results obtained with those derived by FEKO, and then with measurements done in an anechoic chamber on a scaled model representing the forest [19].

1.3 Challenges and Motivations of the thesis

As mentioned above, the electric-field volumetric integral representation first requires the calculation of the internal field inside the trees. This field is obtained by using the MoM to solve the electric volumetric integral equation and this method requires the discretization of trunks and branches into N elementary cubic cells of side equal to or less than $\lambda_s/10$.

Unfortunately, rigorous numerical methods such as the MoM often become untenable when the problem becomes so large as to be computationally highly expensive, both in terms of CPU time and memory. Hence the application of this model was limited to small areas of forest and to low frequencies.

Indeed, we were not able to carry out simulations with forest simulation scene of $N \geqslant N_{max} \approx 13000$ on our shared-memory workstation of 42 GB of RAM.

Hence, we can say that the major challenge is to :

```
Improve as much as possible the performances of the previous 3D dielectric
full-wave model both in terms of computing time and memory storage required,
in order to to handle electrically larger forest simulation scenes than is
possible with conventional MoM for higher frequencies.
```

In order to respond to this challenge and hence, to improve the performances of the previous model in terms of computation time and memory use, we propose, in this new research work, to solve the equation referred to above by using basis functions adapted to the problem of interest, in the context of the Characteristic Basis Function Method (CBFM) [26, 27]. This method was originally introduced to the computational electromagnetic community to analyze PEC structures such as microstrip circuits or patch antennas. Recent studies have shown that CBFM is also an efficient method for solving large scattering problems involving both PEC and dielectric objects [28–30].

We have chosen this method among others (iterative methods such as FMM [31], MLFMA [32] or AIM [33] and algebraic algorithm such as ACA [34]) since it enables us to solve the electromagnetic problem for multiple excitations efficiently. Indeed, it does not require us to repeat the iterative process anew for each incident angle since. Furthermore, we believe that using a direct solver, as in the case of the CBFM, is more suitable for our 3D dielectric electromagnetic problem than an iterative solver such as the MLFMA or the ACA because of potential convergence problem. In fact, the iterative solver may not converge in some cases, while the direct solver always provides a result [35–37]. That is certainly the major disadvantage of iterative methods such as the ACA or the FMA which prompted us to drop it despite its proven strong performances in terms of CPU time and memory storage [38].

Then we also have the fact that some iterative methods requires restrictive conditions on the EM problem under consideration to ensure the accuracy of the solution. For instance, the ACA, which is considered as one of the most powerful available numerical method, is theoretically only applicable to EM problem whose the integral kernels are asymptotically smooth. Consequently, as affirmed in [34], in a strict sense, the ACA algorithm is not applicable to the electromagnetic wave problems. Nevertheless, as it has proved itself in numerous previous research works, it is considered as a heuristic method for this kind of applications [37].

On the other hand, several previous studies have demonstrated the efficiency of the CBFM when applied to the solution of large electromagnetic scattering problems and have confirmed the advantages of this direct solution method over the existing iterative methods [39–41]. It is also interesting to note that the CBFM was successfully improved by hybridizing it with the FMM and the ACA, in the case of 2D PEC objects [29, 39, 42]. Thus, we benefit at the same time from the advantages of both direct and iterative solvers.

In our case, we investigate the suitability of CBFM for the computation of the electromagnetic fields inside and outside 3D dielectric objects modeling tree trunks in a forest environment. When most of previous research work, mentioned above, applied the CBFM to 2D scattering problems using the Rao-Wilton-Glisson (RWG) basis functions, we have applied this new method to 3D dielectric cylinders in the context of a full-wave model based on the electric field volumetric integral equation.

Therefore, a second major challenge is to :

> `Investigate the suitability of the CBFM to the computing of electric fields inside and scattered by dielectric 3D objects and represented by a volumetric integral equation, and investigate the behaviour of this decomposition domain method toward the heterogeneity of a natural forest medium.`

Consequently, the main novelties, concerning the application of the CBFM, consists of the 3D volumetric formulation, the dielectric properties of the cylinders modeling an inhomogeneous forest simulation scene, and finally the application field which is the remote sensing of forest areas.

1.4 Novel elements in the dissertation

In this work, a previous 3D full-wave model for the analysis of forest scattering was enhanced by the application of the CBFM in the context of a conventional MoM. The novel elements which may be mentioned are :

- Application of the CBFM to 3D dielectric object modelling tree trunks and branches, starting with adapting the domain decomposition to the 3D geometry of these trees.

- Comparison between the performances of the CBFM when applied to 3D dielectric objects, with and without extension along the vertical axis $\vec{z}$.

- Generalization of the extension for a multilevel CBFM and its implementation with a large forest simulation scene composed of tree trunks and branches, modelled by dielectric cylinders.

- Application of the Characteristic Basis Function Method (CBFM) on a Non-uniform Mesh, depending on the dielectric properties of the trees.

- Hybridization of the CBFM with the Adaptive Cross Approximation (ACA) algorithm in a context of 3D dielectric scatterers.

- Comparison of the performances of the CBFM to those achieved by the ACA, preceded by a H-matrix partitioning of the initial MoM matrix, in when applied to our 3D dielectric forest scattering model.

- Enhancement of the performances of our 3D model in terms of computing time through the implementation of an Open-MP and MPI parallelized FORTRAN code.

Therefore, the developed 3D forest scattering model makes it possible to deal with a substantial increase in the size of the scattering problems, up to several millions of unknowns, and hence to solve problems involving much larger forest areas at higher frequencies that were not manageable heretofore when using conventional methods, e.g., the MoM.

1.5 Outline of the Dissertation

This dissertation is organized as follows. In chapter 2, we detail the numerical procedure of the CBFM for large scattering problems and we apply its two versions (normal and extended) to our 3D full-wave model. We display some primary numerical results and then we determine the key parameters to be handled in order to ensure the accuracy of the CBFM solution, and to guarantee the best performances in terms of CPU time and memory storage.

Chapter 3 presents the multilevel scheme of the CBFM and stresses the added performance it brings to the 3D forest model in terms of memory use and computing time.

In Chapter 4, we address the different approaches adopted in order to enhance the performances of the CBFM, when applied to our 3D dielectric forest scattering model. These approaches encompass the implementation of the CBFM on a non-uniform mesh depending on the dielectric properties of the trees, the hybridazation of the CBFM with the ACA while generating the final reduced matrix and the acceleration of the CBFM FORTRAN code essentially by its OpenMP then MPI parallelization.

The chapter 5 illustrates a comparison between the CBFM based on a direct solver and the iterative method ACA.

Finally, conclusions and perspectives are drawn in Chapter 6.

Chapter 2

Application of the Characteristic Basis Function Method to 3D forest model

In this chapter, we apply the Characteristic Basis Function Method (CBFM) to compute the electromagnetic field scattered by 3D dielectric objects in the context of forest scattering simulation. We study the effect of some CBFM parameters on the accuracy of the results, and on the performances of the CBFM when compared to the classical MoM. We show that once the CBFM parameters have been appropriately chosen, this new method realizes a significant reduction both in terms of CPU time and memory use, while maintaining a level of accuracy comparable to that of the conventional MoM. Consequently, the CBFM enables us to handle larger forest area simulation scenes than is possible with classical MoM for higher frequencies.

2.1 Overview of the Method

The Characteristic Basis Function Method (CBFM) [26, 28], is designed to solve large-scale electromagnetic problems [29], even with limited computing resources. The CBFM uses a type of macro basis function [27, 43] named the Characteristic Basis Function. The use of these basis functions leads to a significant reduction in the initial number of unknowns, and results in a substantial size reduction of the MoM matrix such that a direct method can be used for its inversion [30, 44].

Even if the object is electrically large, the user can suitably decompose the under consideration geometry into M patches, and then use the CBF Method to reduce the size of the matrix equation. A number of previous studies on the efficiency of the CBFM when solving scattering problems involving dielectric objects have been carried out [27, 29, 30, 44]. In the present work, we investigate the suitability of the CBFM for the computation of the electromagnetic field scattered by 3D dielectric objects representing the trees in a forest environment.

2.1.1 Motivation and advantages

Rather than solving the entire problem in a semi-iterative way, the CBFM creates a single reduced or compressed matrix which can be inverted by using the conventional LU decomposition algorithm.

Using the CBFM, the linear equation arising from the MoM can be compressed into a reduced linear equation with M^2 unknowns, where M is the number of patches that the CBFM creates to partition the object. Thus, in the CBFM, the size of the resultant linear equation to be solved does not directly depend on the dimension of the object. Rather, it depends only on the number of imaginary sub-domains that the CBFM creates to subdivide the object. This property enables the CBFM to solve large-scale electromagnetic problems even when using limited computing resources [27, 43].

Our work is based on the version of the CBFM that has been tailored to solve scattering problems, as opposed to RFIC-types [27] for which the CBFM was originally introduced. The CBFs we introduce serve as a basis set which does not depend on the incident angle. They are computed by using a spectrum of plane-waves incident from several possible directions. Initially we do not include buffered regions as in conventional versions of the CBFM [28]. This first form of application of the CBFM will be referred to as CBFM-N (Normal CBFM). Next, buffered regions are added in order to mitigate the problem of edge effects arising from the boundaries of the CBFM blocks. This version of the CBFM will be referred to as CBFM-E (Extended CBFM).

A comparative study of CBFM-N and CBFM-E is carried out in this chapter by computing the electromagnetic fields inside 3D dielectric objects representing the trunks and the main branches of the trees. The scattered fields are derived by solving the electric field integral equation (EFIE). In common with our previous works [24], the tree trunks and the branches are modeled as dielectric cylinders of square cross-section. The results obtained by using the two CBFM techniques are compared with the solution derived by using a legacy MoM code to validate the CBFM results.

2.1.2 The CBFM formulation

As detailed in the description of the 3D modeling approach presented previously in [24], the first step consists of the dicretization of the tree trunks and branches, modeled as dielectric cylinders, into N elementary cubic cells small enough to assume that the electric field inside is constant. Hence the cell size is taken to be less than or equal to $\frac{\lambda_s}{10}$, where λ_s is the wavelength inside the scatterer. The linear system resulting from the application of a Method of Moments (MoM) can be expressed as:

$$ZE = E_{ref} \tag{2.1}$$

where Z is the $3N \times 3N$ full matrix representing the interactions between the different cells in the cylinders. E_{ref} is the excitation field of size $3N$ and E is the unknown solution vector of size $3N$ that contains information about the total electric field inside the scatterers in the $\vec{x}$, $\vec{y}$ and $\vec{z}$ directions.

The CBFM procedure begins by dividing the 3D geometry of the scatterers into M blocks as shown in Figure 2.1 such that the MoM matrix for each block is manageable in size and therefore could be easily handled by using a conventional direct solver.

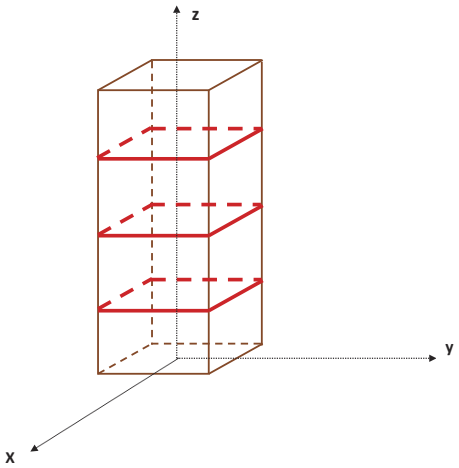


FIGURE 2.1: 3-dimensional rectangular dielectric object divided into M = 4 blocks

A set of Macro-domain Basis Functions [44] is defined on each block which make up the dielectric object. To derive a set of macro basis functions which is invariant to the direction of incidence of the electric field, each block is illuminated by a sufficient number of plane waves impinging upon the object at different angles of incidence, as shown in Figure 2.2, to generate the macro basis functions for all blocks.

Because of the wide number of plane waves used, the MBFs would in general have a certain level of redundancy. To mitigate this problem, we apply Singular Value Decomposition (SVD) algorithm to the set of MBFs to down-select the number of basis functions and remove the redundancy in the process [27, 43].

A threshold is set for the normalized singular values of each block and only a small set of dominant macro basis functions is retained to represent the unknown field and are used as the CBFs for the individual blocks. These new CBFs enable us to construct a linear set of equations, which is reduced as compared to the initial one generated by the EFIE using traditional low-level basis functions.

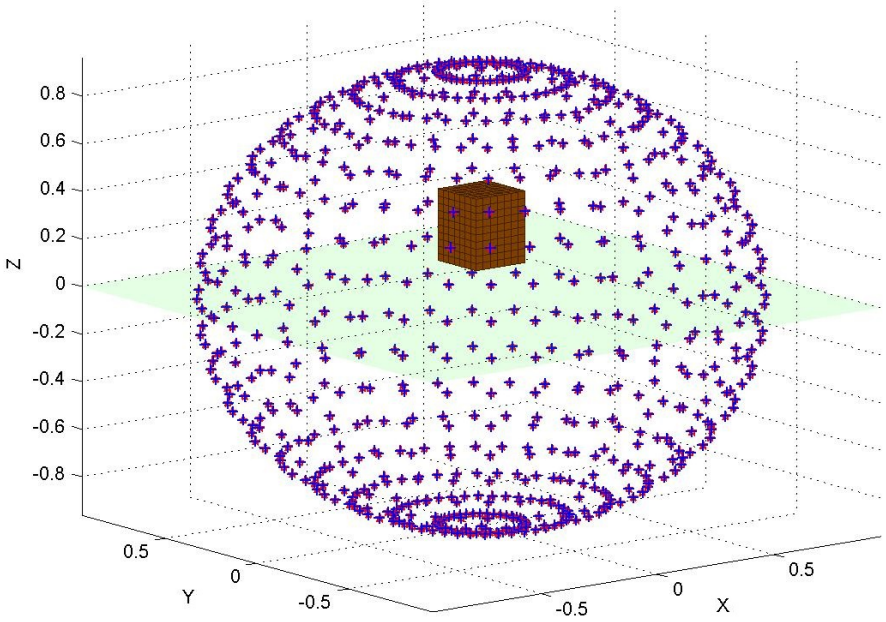


FIGURE 2.2: A 3-dimensional rectangular dielectric individual block illuminated by $N_\theta * N_\varphi = 861$ plane waves at equally spaced discrete angles around the object $d\theta = d\varphi = 9°$.

2.1.3 The CBFM numerical procedure

To illustrate the method, let us consider a vertical dielectric cylinder of square cross section comprised of N cells. We can rewrite the matrix equation (2.1) as :

$$\sum_{\mathbf{n=1}}^{\mathbf{N}} \sum_{\mathbf{q=1}} \mathbf{Z}_{\mathbf{pq}}^{\mathbf{mn}} \mathbf{E}_{\mathbf{n}}^{\mathbf{q}} = \mathbf{E}_{\mathbf{p}}^{\mathbf{ref,m}} \text{ where } p, q = x, y, z \text{ and } m, n = [1, ..., N] \tag{2.2}$$

Instead of inverting the $3N \times 3N$ matrix Z to solve the equation (2.1), which arises from an application of the Method of Moments (MoM) to the EFIE, we apply the Characteristic Basis Function Method, initially without including extensions, or buffered regions.

We divide the scatterer geometry in $M = 4$ blocks as shown in Figure 2.3. The coefficient matrix Z is then divided into $M^2 = 16$ blocks grouping MoM matrix elements belonging to the 4 blocks.

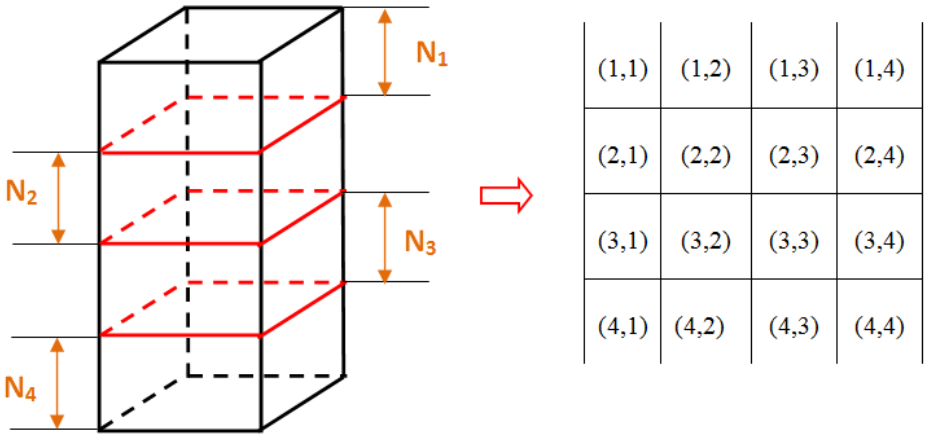


FIGURE 2.3: The coefficient matrix Z is divided in 16 blocks grouping MoM matrix elements belonging to four blocks.

2.1.3.1 Creation of Macro Basis Functions (MBFs)

To create the macro basis functions for each block, we illuminate the dielectric cylinder by $N_{IPWs} = N_\theta \times N_\varphi = 91$ incident plane waves of frequency f, for $d\theta$ and $d\varphi$ equally spaced incident angles (here $d\theta = d\varphi = 30^\circ$ so $N_\theta = 7$ and $N_\varphi = 13$.) The MoM matrix elements that belong to each block can be easily computed and stored as a $3N_{block} \times 3N_{block}$ matrix named Z_{block} or Z_{ii}, where N_{block} is the number of elementary cubic cells per block. Next, the N_{IPWs} macro basis functions are determined for each block by solving the following linear system of equations:

$$Z_{ii} E_{ii}^{MBFs} = E_{ii}^{ref,IPWs} \tag{2.3}$$

where $E_{ii}^{ref,IPWs}$ is a $3N_{block} \times N_{IPWs}$ matrix representing the plane waves excitations and E_{ii}^{MBFs} is a $3N_{block} \times N_{IPWs}$ matrix representing the macro-domain basis functions (MBFs).

To better understand the computing procedure of the N_{IPWs} $(N_\theta \times N_\varphi)$ MBFs, the matrix equation (2.3) can also be written as follows :

$$Z_{ii} E_i^{(\theta,\varphi)} = E_i^{ref,(\theta,\varphi)} \text{ for } \theta = 1, 2, ..., N_\theta \text{ and } \varphi = 1, 2, ..., N_\varphi \tag{2.4}$$

Block matrix equations in (2.4) are used for computing $N_\theta \times N_\varphi$ MBFs for each block by inverting this same equation as follows:

$$E_i^{(\theta,\varphi)} = Z_{ii}^{-1} E_i^{ref,(\theta,\varphi)} \text{ for } \theta = 1, 2, ..., N_\theta \text{ and } \varphi = 1, 2, ..., N_\varphi \tag{2.5}$$

The macro basis functions $E_i^{(\theta,\varphi)}$ for each block i for all angles of incidence of $E_i^{(ref,(\theta,\varphi))}$ are computed. If there are N_1, N_2, ...N_M cells in each block, inverting equation (2.4) would yield M sets of macro basis functions. We arrange the $N_{IPWs} = N_\theta \times N_\varphi$ macro basis functions of the i^{th} block in a

matrix named E_i or E_{block}.

$$E_i = \begin{pmatrix} E_1^{i,\theta_1,\varphi_1} & E_1^{i,\theta_2,\varphi_1} & \cdots & E_1^{i,\theta_{N_\theta},\varphi_1} & E_1^{i,\theta_1,\varphi_2} & \cdots & E_1^{i,\theta_{N_\theta},\varphi_{N_\varphi}} \\ E_2^{i,\theta_1,\varphi_1} & E_2^{i,\theta_2,\varphi_1} & \cdots & E_2^{i,\theta_{N_\theta},\varphi_1} & E_2^{i,\theta_1,\varphi_2} & \cdots & E_2^{i,\theta_{N_\theta},\varphi_{N_\varphi}} \\ E_3^{i,\theta_1,\varphi_1} & E_3^{i,\theta_2,\varphi_1} & \cdots & E_3^{i,\theta_{N_\theta},\varphi_1} & E_3^{i,\theta_1,\varphi_2} & \cdots & E_3^{i,\theta_{N_\theta},\varphi_{N_\varphi}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E_{N_i}^{i,\theta_1,\varphi_1} & E_{N_i}^{i,\theta_2,\varphi_1} & \cdots & E_{N_i}^{i,\theta_{N_\theta},\varphi_1} & E_{N_i}^{i,\theta_1,\varphi_2} & \cdots & E_{N_i}^{i,\theta_{N_\theta},\varphi_{N_\varphi}} \end{pmatrix} \tag{2.6}$$

We have to mention that this procedure, due to the presence of the inversion operation on the matrix Z_{ii}, turns out to be computationally expensive though it is expected to consume much less computation time and memory than needed in the Method of Moments. This problem may be solved via the use of the sparse representation, which consists of approximating the inversion with a vector-vector simple division. Thus instead of inverting the matrix given by (2.4), the macro basis functions for each block are expressed as :

$$\widetilde{E}_i^{(\theta,\varphi)} = \frac{E_i^{ref,(\theta,\varphi)}}{Z_{ii}^i} \text{ for } \theta = 1, 2, ..., N_\theta \text{ and } \varphi = 1, 2, ..., N_\varphi \tag{2.7}$$

Previous works [27, 43] have shown that using the sparse representation leads to results that are not always sufficiently accurate. Hence, in our work, we will initially construct macro-basis function by inverting (2.4), rather than resorting the sparse approximations.

2.1.3.2 Generation of Characteristic Basis Functions (CBFs)

The number of plane waves used to generate the macro basis functions generally exceeds the degrees of freedom associated with each block. Consequently, we need to remove the redundancy in the macro basis functions to improve the condition number of the final reduced matrix. Toward this end, we compute a new set of characteristic basis functions that are linear combinations of the original macro basis functions by retaining the dominant ones using the Single Value Decomposition (SVD) algorithm [27]. Applying the SVD to the macro basis functions in (2.6) leads to:

$$\begin{aligned} E_i &= U^{(i)} S^{(i)} V^{(i)H} \\ &\text{where} \\ U^{(i)} &= (u_1, u_2, \cdots, u_n) \in \mathbb{C}^{N_i \times N_i} \\ V^{(i)} &= (v_1, v_2, \cdots, v_n) \in \mathbb{C}^{N_{IPWs} \times N_{IPWs}} \\ S^{(i)} &= diag(\sigma_1, \sigma_2, \cdots, \sigma_p) \in \mathbb{R}^{N_i \times N_{IPWs}} \text{ and } p = min(N_i, N_{IPWs}) \end{aligned} \tag{2.8}$$

where $U^{(i)}$ and $V^{(i)H}$ are rectangular orthogonal matrices. $S^{(i)}$ is a diagonal matrix containing singular values of $E^{(i)}$ such as :

$$\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3 \cdots \geqslant \sigma_p$$

Next, the singular values are normalized by dividing them by the maximum value σ_1, to obtain the p normalized values of σ :

$$\sigma_j^{Normalized} = \frac{\sigma_j}{\sigma_1} \text{ for } j = 1, 2, \cdots, p \tag{2.9}$$

We then apply a threshold to the normalized singular values and discard the values (set them equal to zero) that fall below the threshold, which is typically chosen to be 10^{-3} or 10^{-4}, depending on the level of accuracy we desire. This filtering process of eliminating the post-SVD CBFs enables us to further reduce their redundancy and, consequently, improve the condition number of the reduced matrix. If we retain S_i normalized singular values for the i^{th} block, then the first S_i columns of $U^{(i)}$, denoted by C^i in (2.10) are used as characteristic basis functions for the block i. Following the above detailed procedure, we construct K primary basis functions (where $K = \sum_{i=1}^{M} S_i$).

$$C^{(i)} = \begin{pmatrix} C_{1,1}^{(i)} & C_{1,2}^{(i)} & \cdots & C_{1,S_i}^{(i)} \\ C_{2,1}^{(i)} & C_{2,2}^{(i)} & \cdots & C_{2,S_i}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N_i,1}^{(i)} & C_{N_i,2}^{(i)} & \cdots & C_{N_i,S_i}^{(i)} \end{pmatrix} \tag{2.10}$$

In the next paragraph, we will use these CBFs to create, using the Galerkin method, the final reduced matrix of size $K \times K$. For our example $M = 4$, the size of the generated reduced matrix is $(K = S_1 + S_2 + S_3 + S_4)^2$. The generation of the reduced matrix will be discussed bellow.

2.1.3.3 Generation of the reduced matrix equation

As explained above, we generate our K characteristic basis functions for the M initial blocks (where S_i is the number of CBFs for the i^{th} block and $K = S_1 + S_2 + ... + S_M$). Then the next step consists of generating the reduced $K \times K$ matrix equation for the unknown complex coefficients $\alpha_k^{(i)}$ by using the Galerkin method. This procedure results in a reduced matrix equation given by :

$$Z^c \alpha = V^c \text{ ; where } \alpha = \sum_{i=1}^{M} \alpha_k^{(i)} \text{ is a vector of dimension } K \tag{2.11}$$

To generate this matrix, referred to herein as Z^c, we need to combine the original MoM matrix and the K characteristic basis functions by using the Galerkin method as follow:

$$Z^c = \begin{pmatrix} C^{(1)t} Z_{11} C^{(1)} & C^{(1)t} Z_{12} C^{(2)} & \cdots & C^{(1)t} Z_{1M} C^{(M)} \\ C^{(2)t} Z_{21} C^{(1)} & C^{(2)t} Z_{22} C^{(2)} & \cdots & C^{(2)t} Z_{2M} C^{(M)} \\ C^{(3)t} Z_{31} C^{(1)} & C^{(3)t} Z_{32} C^{(2)} & \cdots & C^{(3)t} Z_{3M} C^{(M)} \\ \vdots & \vdots & \ddots & \vdots \\ C^{(M)t} Z_{M1} C^{(1)} & C^{(M)t} Z_{M2} C^{(2)} & \cdots & C^{(M)t} Z_{MM} C^{(M)} \end{pmatrix} \tag{2.12}$$

Here $C^{(i)}$ is a $3N_i \times S_i$ matrix which contains the S_i characteristic basis functions corresponding to the N_i cells of block i. Z_{ij} where $(i \neq j)$ is the coupling matrix linking blocks i and j, and Z_{ii} is the self-coupling matrix of block i. The product $C^{(i)t} Z_{ij} C^{(j)}$ is then an $S_i \times S_j$ matrix. The initial matrix is reduced to a $K \times K$ final matrix $(K = S_1 + S_2 + \cdots + S_M)$. Note also that Z^c is diagonally symmetric and the use of this property saves time during its generation but also memory space needed to store it. Following this, we generate V_c, the right hand-side of the equation (2.11), of size $K \times 1$. We do this by combining the original E^{ref} of equation (2.2) and the K characteristic basis functions using the Galerkin method. We should distinguish between E^{ref} of equation (2.2) representing the plane waves excitations for which we calculate the total electric field inside the object in each direction ($\vec{x}$, $\vec{y}$), and $\vec{z}$, and $E_{ii}^{ref,IPWs}$ of equation (2.3) representing the N_{IPWs} ($N_{IPW} = N_\theta \times N_\varphi$) plane waves excitations used to compute the characteristic basis functions (CBFs).

$$V^C = \begin{pmatrix} C^{(1)t} V_1 & C^{(2)t} V_2 & \cdots & C^{(M)t} V_M \end{pmatrix}^T \tag{2.13}$$

The next step is to solve the reduced matrix equation given by (2.11). We compute the $K \times 1$ complex coefficient vector α (S_i coefficients for each block i). For our example with $M = 3$, we compute so $K = S_1 + S_2 + S_3$ complex coefficients α_k.

2.1.3.4 Computing of the total electric field inside the object

In the final step, we compute the total electric field inside the object in each direction $\vec{x}$, $\vec{y}$ and $\vec{z}$. This field can be expressed as a linear combination of the K CBFs weighted by the K complex coefficients α obtained previously. It is given by :

$$E = \begin{pmatrix} \sum_{k=1}^{S_1} \alpha_k^{(1)} C_k^{(1)} \\ \sum_{k=1}^{S_2} \alpha_k^{(2)} C_k^{(2)} \\ \vdots \\ \sum_{k=1}^{S_M} \alpha_k^{(M)} C_k^{(M)} \end{pmatrix} \tag{2.14}$$

Here $\sum_{k=1}^{S_i} \alpha_k^{(i)} C_k^{(i)}$ is a $3N_i \times 1$ vector representing the total electric field inside the block i corresponding to a given plane wave excitation. We hence compute the total electric field inside the object in the 3 directions $\vec{x}$, $\vec{y}$ and $\vec{z}$ using the simple or “normal" version of the CBFM termed CBFM-N.

2.1.4 The extended CBFM : improved version of the CBFM-N

In order to improve the above CBFM procedure, namely CBFM-N, each block i is extended, along the vertical axis $\vec{z}$, by including a buffered region to form an extended block represented by the $N_{i,e} \times N_{i,e}$ matrix Z_e (see Figure 2.4). Once again, we follow the same procedure as the one applied in the CBFM-N. In this case, the matrix $C^{(i)}$ (2.10) for block i is a $3N_{i,e} \times S_i$ matrix. We return to the original block

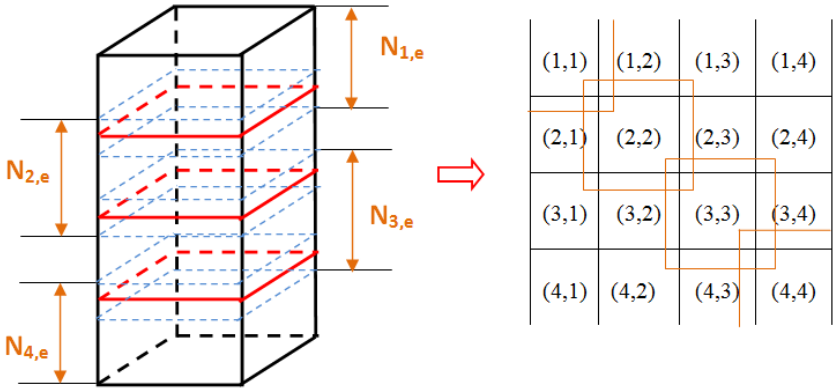


FIGURE 2.4: The coefficient matrix Z is divided in 16 blocks. Note the extension of each block to include a buffer region to form the extended block Z_e.

size N_i when generating the reduced $K \times K$ matrix of equations (2.11) and (2.12). This new procedure is termed CBFM-E, i.e., extended CBFM. As indicated in [28] (where the extension is applied to a 2D structure), the extension of the CBFM block enables us to mitigate the problem of singular behavior of the current distribution introduced by the truncation that creates fictitious edges.

2.1.5 Primary validation of the CBFM compared to a classical MoM

In order to compare the performances of the CBFM-N and CBFM-E, we apply them to a simple scattering example composed of one tree trunk, for which we use the scattering model of trunks above a ground. For this model which has been previously employed in the L2E laboratory, the tree trunks and branches are represented by finite-length dielectric cylinders of square cross-section and the ground is modeled by a horizontal plane separating two semi-infinite homogeneous media. Realistic natural conditions can be described by this model while using appropriate settings of its parameters. Some of these parameters are the dielectric permittivities associated with the cylinders and with the plane interface. Previous studies on forest terrain modeling [13, 14], have shown that these relative permittivities depend on the humidity of the forest environment, which increases with the rainfall rate. Thus, based on these studies, we assume that trees trunks and moist soil can be correctly simulated in the VHF/UHF band by using the relative dielectric parameters $\epsilon_s = 9.6 + 0.01\text{j}$ and $\epsilon_g = 5 + 3.6\text{j}$, respectively.

We first apply the CBFM-N and CBFM-E for this example of a homogeneous dielectric cylinder of square section, whose dimensions are $0.09 \times 0.09 \times 2.7$ m^3, and which is shown in Figure 2.5. The frequency of the incident plane wave is equal to 300 MHz. To compute the total electric field inside the object by using the CBFM procedure, the cylinder representing the tree trunk is discretized into 810 elementary cubic cells, and then divided along the vertical axis into three blocks. Each block is illuminated by $N_{IPWs} = N_\theta \times N_\varphi = 380$ incident plane waves with an angular step of $d\theta = d\varphi = 20^\circ$ ($0 \leqslant \theta \leqslant 180^\circ$ and $0 \leqslant \varphi \leqslant 360^\circ$). The field calculation is carried out for both vertical and horizontal

polarizations. The simulations have been carried out on a shared memory workstation equipped with an Intel Core (TM) i7-2640M (4 processors) running at 2.80 GHz, and using 8 GB of RAM. The simulation conditions are listed in Table 2.1. Note that the size of the buffered region, for the CBFM-E, is named here $Nber_{ext,floors}$ for number of extended floors along the vertical axis $\vec{z}$.

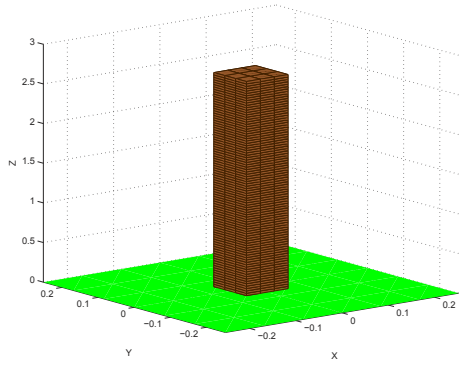
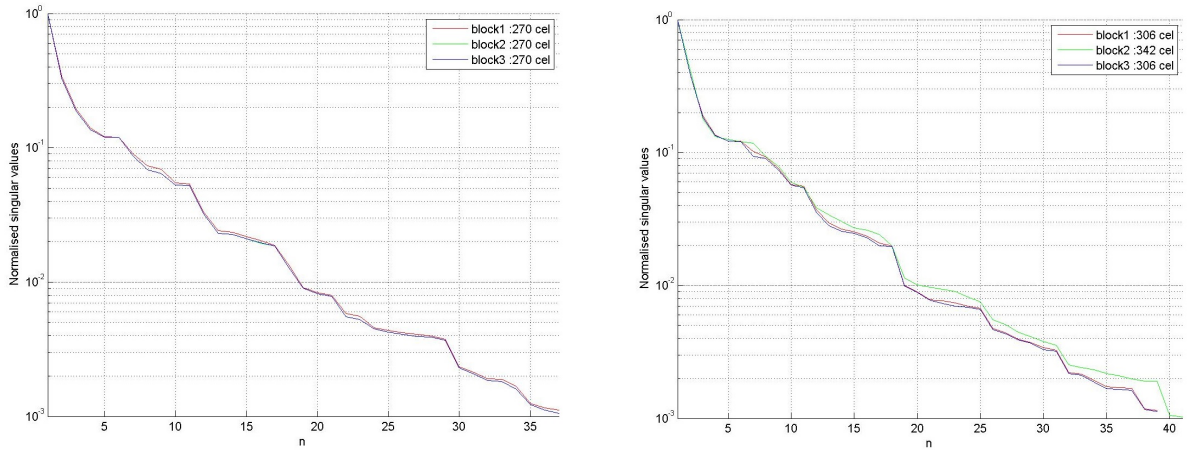


FIGURE 2.5: A vertical cylinder measuring $0.09 \times 0.09 \times 2.7$ m^3 placed over a real ground plane of relative permittivity $\epsilon_g = 5 + 3.6$ j.

TABLE 2.1: Simulation conditions applied to a vertical cylinder placed above a real ground

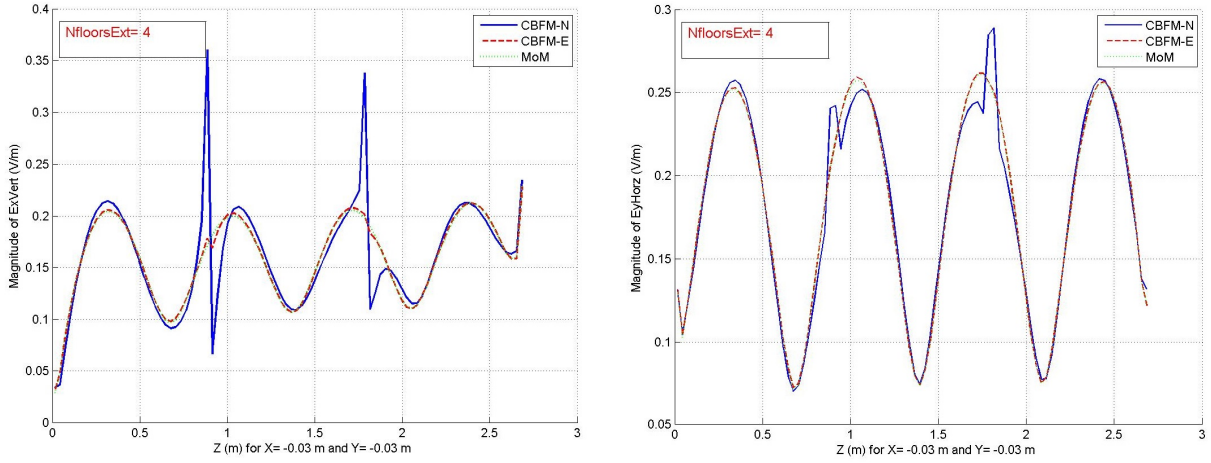
Cylinder dimensions	$0.09 \times 0.09 \times 2.7$ m^3
Cylinder permittivity	$\epsilon_s = 9.6 + 0.01$j
Ground permittivity	$\epsilon_g = 5 + 3.6$j
λ scatterer	$\lambda_s = 0.048$ m
Cell size	$T_c = 0.03$ m
Number of Cells	$Nb_c = 810$ cells
Number of blocks	$N_B = 3$ (from bottom to top)
Number of extended floors	$Nber_{ext,floors} = 4$
Number of plane waves	$N_{IPWs} = N_\theta \times N_\varphi = 380$

The normalised singular values retained after singular value decomposition (SVD) and normalization (using a threshold $\sigma = 10^{-3}$) with the CBFM-N and CBFM-E are represented in Figure 2.6. Hence, the application of the Galerkin procedure (equation (2.12)) results in a reduced matrix $Z^c_{111\times111}$ for the CBFM-N, and $Z^c_{119\times119}$ for the CBFM-E. The total electric fields inside the dielectric cylinder obtained for the incidence direction ($\theta_i = 45°$; $\varphi_i = 40°$) by the CBFM-N and CBFM-E procedures, are compared with the MoM results, in Fig. 2.7, as functions of the vertical axis z. Figure 2.7 shows that the total electric field computed with the CBFM matches relatively well with the MoM solution with some oscillations localized around the boundaries of the blocks, especially when using the CBFM-N ($z = 0.9$ m and $z = 1.8$ m). However, we note that the insertion of buffered regions in the CBFM-E procedure results in a significant reduction of these oscillations; consequently, the results from the CBFM-E are much closer to those derived by using MoM.

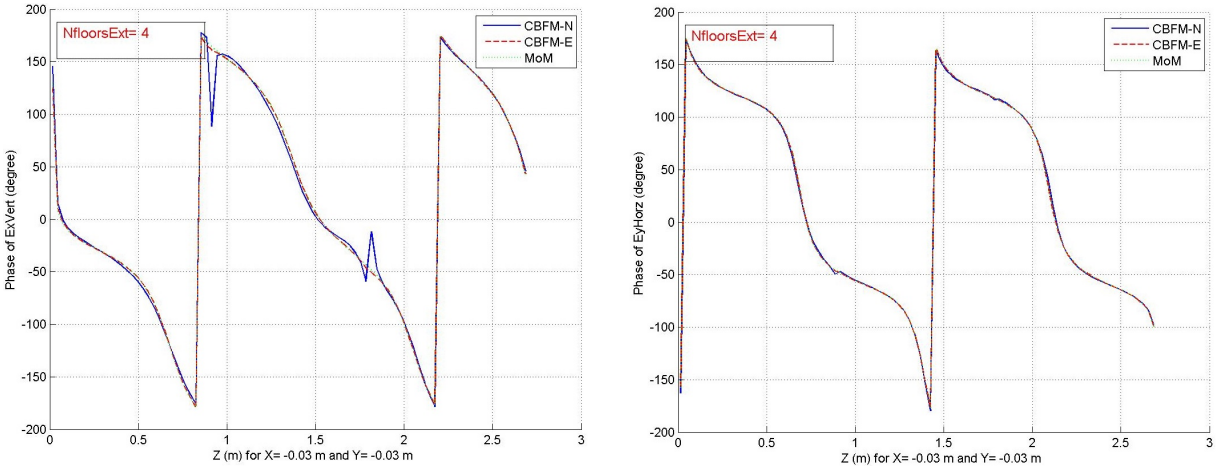


(A) CBFM-N (B) CBFM-E

FIGURE 2.6: Using the CBFM-N (a) and the CBFM-E (b), we represent, on the log y-axis, the normalized singular values for the 3 blocks ($N_{IPWs} = 380$ and $Nber_{ext,floors} = 4$ for the CBFM-E.



(A) E_x^V (B) E_y^H



(C) ϕ_x^V (D) ϕ_y^H

FIGURE 2.7: Variations of the magnitudes and phases of the electric field inside the cylinder for $x = y = -0.03$ m computed for $(\theta_i = 45°; \varphi_i = 40°)$ with the CBFM-N, CBFM-E and MoM.

Table 2.2 presents a comparison of the two CBFM schemes with the MoM. For this example, we have employed a combination of 6400 transmitters-receivers ($1^\circ \leqslant \theta_i \leqslant 80^\circ$ and $1^\circ \leqslant \varphi_i \leqslant 80^\circ$), as shown in Fig. 2.8. The distance separating the transmitters-receivers from the target, namely $R_{t,r}$, is equal to 2000 m. It has the same value for all the simulations performed in this work.

TABLE 2.2: Performance comparison of CBFM and MoM in terms of final matrix size, computing time and accuracy

	Z size	Computation time	$E_{r,max}$
CBFM-N	111	5 min	3.09 %
CBFM-E	119	6 min	0.44 %
MoM	2430	1 hour 3 min	–

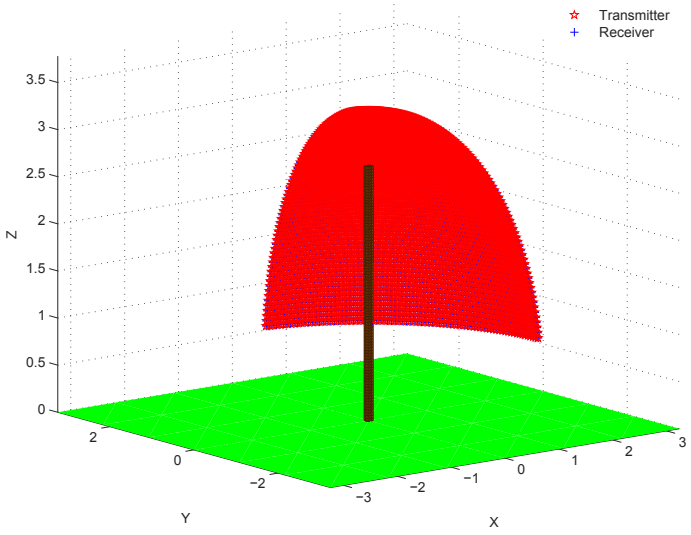


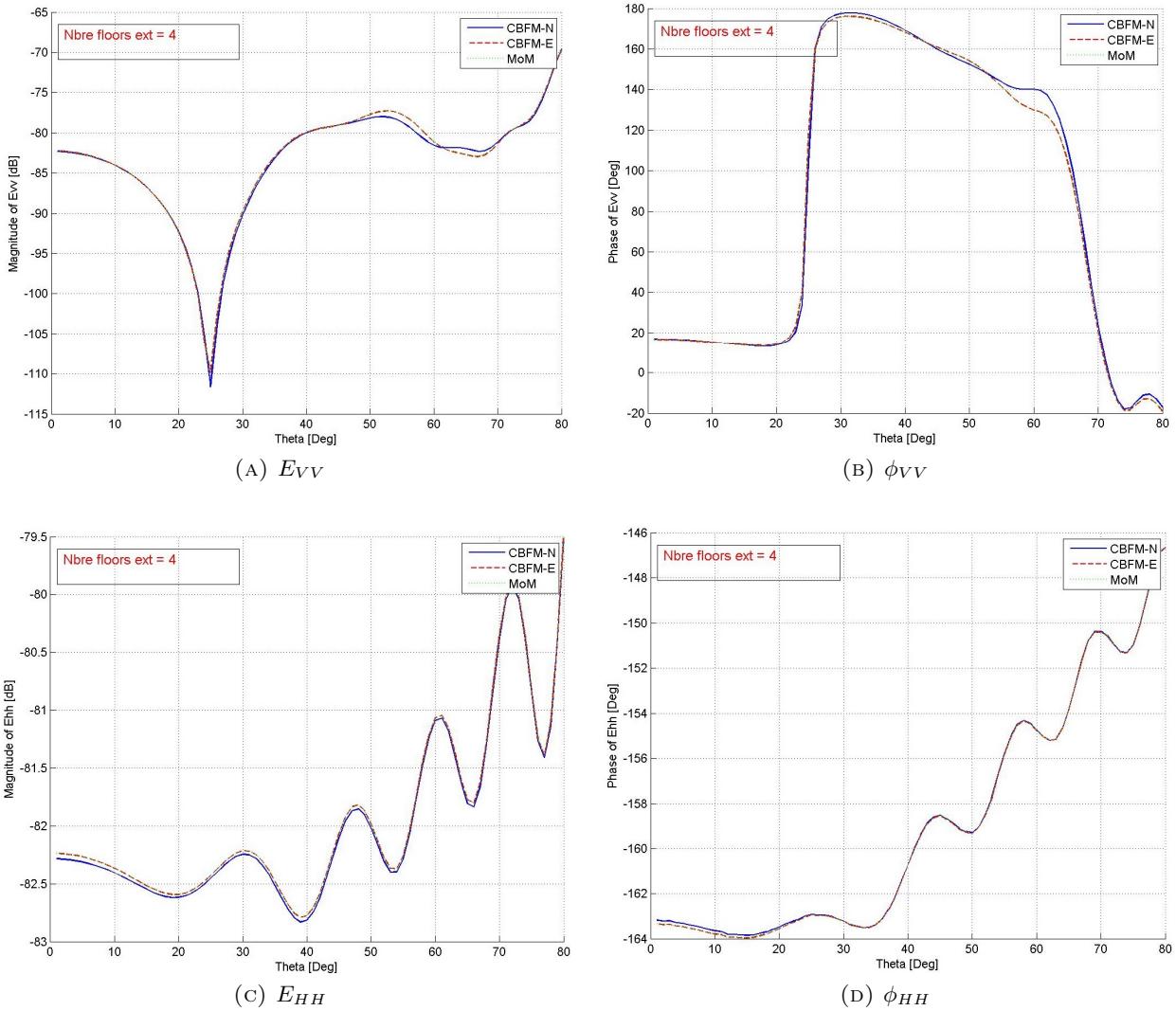
FIGURE 2.8: The electric fields inside the dielectric cylinder are computed for 6400 transmitters-receivers.

We compute E_r, the relative error introduced by the CBFM-N and the CBFM-E, for the three components (x, y and z) of the electric field inside the scatterer, for both polarizations (V and H), for a given incident direction. The relative error with that of the MoM is defined as follows:

$$E_r = \frac{1}{N} \sum_{i=1}^{N_p} \frac{|E^i_{CBFM} - E^i_{MoM}|}{|E^{max}_{CBFM,MoM}|} \tag{2.15}$$

where E^i_{CBFM} and E^i_{MoM} are the complex electric fields inside the object computed, respectively, by using the CBFM-N or CBFM-E, and MoM. $E^{max}_{CBFM,MoM}$ is the maximal value of the magnitude of the electric field and N_p is the number of calculation points (number of $\vec{z}$ positions for which the field inside the scatterer is computed).

Hence, $E_{r,max}$ is the maximal value of E_r obtained by considering each component (E_x, E_y, E_z) and both polarizations (VV and HH)of the electric field. $E_{r,max}$ displayed in Table 2.2 corresponds to the incident direction ($\theta_i = 45°$; $\varphi_i = 40°$).



(A) E_{VV} (B) ϕ_{VV}

(C) E_{HH} (D) ϕ_{HH}

FIGURE 2.9: Variations of the magnitude and phase of the backscattered field computed with the CBFM-N, the CBFM-E and the conventional MoM for an azimuth angle $\varphi_i = 40°$ for $f = 300$ MHz and $R_{t,r} = 2000$ m.

We note that, in addition to yielding accurate results, the CBFM enables us to achieve a significant gain in terms of computation time. In fact, the more complex the problem is, the more significant is the gain realized by the CBFM, in terms of the CPU time over that of the MoM. The CBFM gains the advantage over the MoM, since the latter requires the inversion of a large matrix whose size increases with the number of cells as well as the number of incident angles.

Next, we compare the magnitude of the total backscattered electric field, for the mono-static case with $\theta_i = [1°; 1°; 80°]$ and $\varphi_i = 40°$. Fig. 2.9 shows that the results obtained with the both CBFM procedures are in good agreement with those derived by the conventional MoM. However, we also

distinguish a noticeable, albeit slight, difference between the MoM and the CBFM-N results, which confirms that the CBFM-E is more accurate than the CBFM-N.

Next, we carry out six simulations for the current example to demonstrate the relationship between the number of points and the computation time while we increase the number of transmitters. The results, shown in Table 2.3, attest to the fact that the gain in computing time achieved in favour of the CBFM, increases with the number of calculation points. This represents an additional advantage of the CBFM when compared to the conventional MoM. Indeed, this new advantage is gained because of the fact that the computation of the fields inside the scatterers, for different angles of incidence, is carried out one at a time, thanks to the CBFM approach to handling the problem at hand (see equation 2.14). This fact not only makes the computation faster, but it also enables us to parallelize the computation of these fields. On contrast to this, with the conventional MoM, the field computation is performed in a single block for all the incident directions via the use of a direct solver for linear system of equations, e.g., the Lapack subroutine GESVX.

N_{RHS}	91	273	546	1001	1911	4641
CBFM-N	9 sec	17 sec	29 sec	52 sec	1 min 32 sec	3 min 45 sec
CBFM-E	13 sec	21 sec	35 sec	58 sec	1 min 56 sec	4 min 1 sec
MoM	55 sec	2 min 38 sec	5 min 31 sec	9 min 38 sec	18 min 44 sec	46 min 35 sec

TABLE 2.3: Computation time while increasing the number of right hand side N_{RHS} (transmitters)

To complete the discussion of this example, and to confirm the observations we have presented earlier in regard to the CBFM-N and the CBFM-E, we present the variations of the magnitude of the backscattered fields by a single cylinder of dimensions $0.004 \times 0.004 \times 0.144$ m^3, at $f = 2$ GHz, for different incident angles ($\theta_i = [0°; 1°; 89°]$ and $\varphi_i = [0°; 1°; 359°]$) for the two polarizations VV and HH (see Figures 2.10, 2.11 and 2.12).

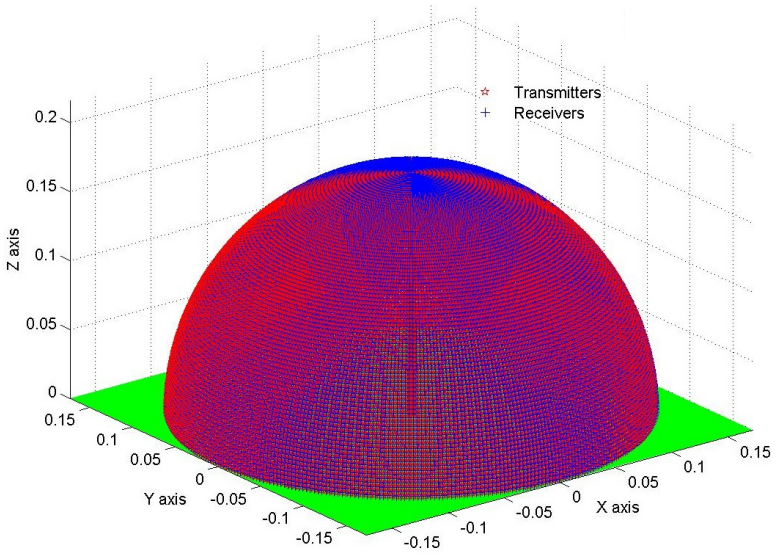
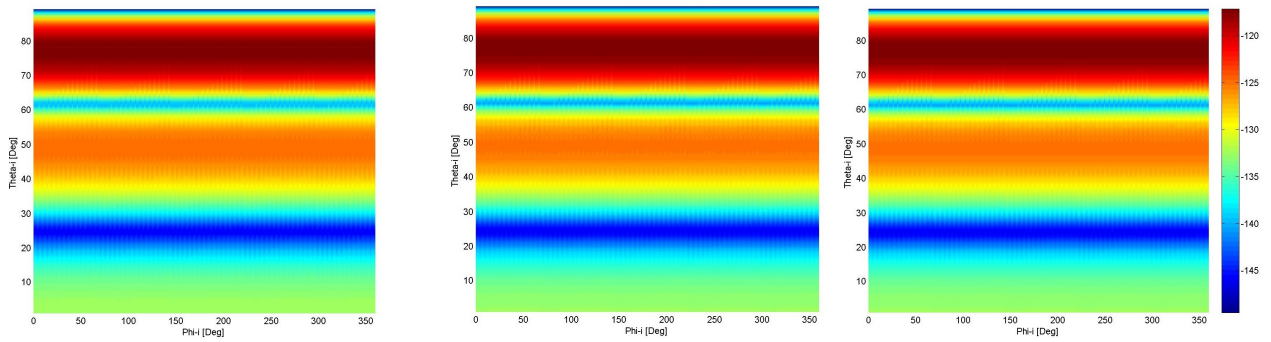
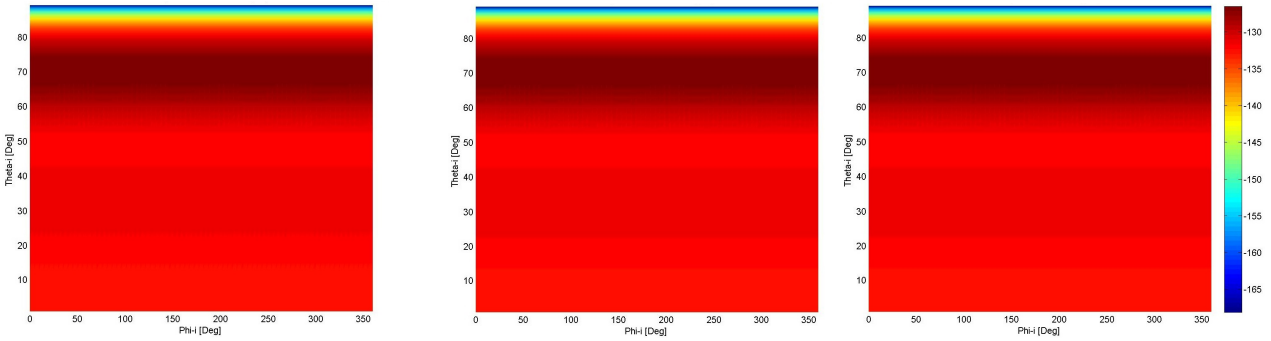


FIGURE 2.10: *The mono-static configuration used to compute the total backscattered electric field* $\theta_i = [0°; 1°; 89°]$ *and* $\varphi_i = [0°; 1°; 359°]$.



(A) CBFM-N (B) CBFM-E (C) MoM

FIGURE 2.11: *Variations of the magnitude of the backscattered fields for the VV polarization, computed with the CBFM-N, CBFM-E and MoM and depending on* (φ_i,θ_i).



(A) CBFM-N (B) CBFM-E (C) MoM

FIGURE 2.12: *Variations of the magnitude of the backscattered fields for the HH polarization, computed with the CBFM-N, CBFM-E and MoM and depending on* (φ_i,θ_i).

In this section, we have introduced and have validated two CBFM-based procedures and compared their performances to that of a conventional Method of Moments. The application of the CBFM to this simple example, provides a good indication of the advantages of the CBFM from the point of view of the computation time.

Furthermore, we expect that this benefit would be even greater when we solve larger and more complex problems. Larger simulation scene will be considered at the end of this section.

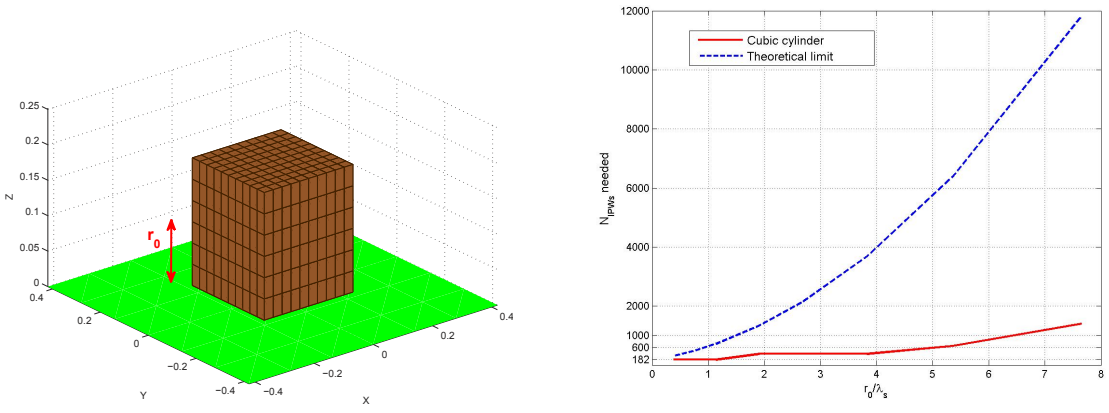
Next, since we need to maintain an admissible level of accuracy of the CBFM comparing to the conventional MoM, next, we discuss how to choose some of the parameters in the CBFM-E procedure to ensure that the desired accuracy is realized. The simulations shown in the next sub-section, have been performed by using a Fortran program on a shared memory workstation equipped with an Intel Xeon x5560 (8 tasks) running at 2 GHz, and using 48 GB of RAM.

2.2 Factors affecting the CBFM performance

In this section, we investigate some CBFM parameters to see how they affect the accuracy of the results, as well as the performance of CBFM when compared with the MoM in terms of computing time and required memory resources.

2.2.1 Choosing the number of plane waves N_{IPWs}:

For both the V- and H- polarizations, the number of plane waves N_{IPWs}, that are used to generate the CBFs for each block, is a crucial factor. This factor has a significant effect on the computation time and the accuracy of the results obtained with the CBFM. Previous studies on sampling criteria for the fields radiated by resonant antennas and scatterers [45] have shown that the fields must be sampled with a minimum of $2(kr_0 + 2\pi)^2$ values for each polarization, where k is the wave number and r_0 the radius of the minimum sphere enclosing the antenna or scatterer under consideration. To verify the validity of this rule for 3D cylindrical objects, we gradually increase the height of a cubic cylinder, or its half-height r_0, viewing it as a single CBFM block. We estimate the number of plane waves needed to generate the CBFs by increasing the number of plane waves for each height and consequently r_0 until the number of CBFs that survive the SVD procedure remains practically unchanged [41]. Then, we plot the progression of the number of plane waves, N_{IPWs} depending on the ratio $\frac{r_0}{\lambda_s}$ as shown in Figure. 2.13. We note that the number of plane waves needed to ensure the accuracy of the method for the cubic cylinder is considerably lower than the theoretical limit mentioned above. The numerical experiment has been performed at the frequency $f = 450$ MHz, by analyzing a single block cylinder placed over a dielectric ground of complex relative permittivity of $\epsilon_g = 5 + 2.1\text{j}$. The cylinder has a complex relative permittivity of $\epsilon_s = 6.5 + 0.24\text{j}$, hence, the wavelength inside the scatterer is $\lambda_s = 0.261$ m.



(A) (B)

FIGURE 2.13: Number of plane waves (N_{IPWs}) needed for the CBFM process versus the ratio $\frac{r_0}{\lambda_s}$ (where r_0 is the half height of the block).

In order to check the validity of these results, we perform two different simulations of the same homogeneous dielectric cylinder whose dimensions are $0.12 \times 0.12 \times 6.2\ m^3$ always with a complex relative permittivity of $\epsilon_s = 6{+}0.24\text{j}$ and placed over a dielectric ground of complex relative permittivity of $\epsilon_g = 5 + 2.1\text{j}$. As for the example of Figure 2.13, the wavelength inside the scatterer is equal to $\lambda_s = 0.261$ m. For the first simulation, the scatterer is divided into 2 blocks of height $h = 3.1$ m each ($r_0 = 1.55$ m), which corresponds to $\frac{r_0}{\lambda_s} \approx 6$. According to the results shown in Figure 2.13, the accuracy of the CBFM process is ensured starting from $N_{IPWs} = 992$ incident plane waves. For the second simulation, the scatterer is divided into 10 blocks each with a height $h = 0.62$ m ($\frac{r_0}{\lambda_s} \approx 1.2$). According to Figure 2.13, for this case, $N_{IPWs} = 182$ incident plane waves should suffice to guarantee results are accurate.

For each simulation, three numerical experiments with three different N_{IPWs} values have been carried out. The N_{IPWs} values chosen refer to the results plotted in Figure 2.13. These experiments enabled us to determine the accuracy obtained with the retained N_{IPWs} values, depending on $\frac{r_0}{\lambda_s}$ and the block height h.

The conditions of the experiments carried out are listed in table 2.4.

		N_{IPWs}
Simulation 1	$N_{Blocks} = 2 \rightarrow \frac{r_0}{\lambda_s} \approx 6 \rightarrow N_{IPWs,needed} = 992$	[650 ; 992 ; 1406]
Simulation 2	$N_{Blocks} = 10 \rightarrow \frac{r_0}{\lambda_s} \approx 1.2 \rightarrow N_{IPWs,needed} = 182$	[182 ; 380 ; 462]

TABLE 2.4: The simulation conditions applied to two vertical cylinder placed over a real ground.

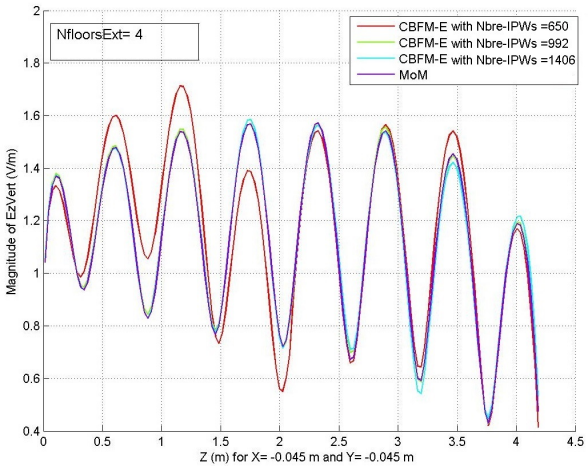
2.2.1.1 Simulation 1

The magnitudes and phases of the total electric field inside the dielectric cylinder have been obtained for the incident angle ($\theta_i = 30°$; $\varphi_i = 0°$) with three different N_{IPWs} values, by using the CBFM-E procedure. A comparison of these results with those derived by using the MoM are plotted in Figure 2.14. Table 2.5 also provides a comparison of the relative error (%) introduced by the CBFM-E with that of the MoM for the 3 numerical experiences that we have performed.

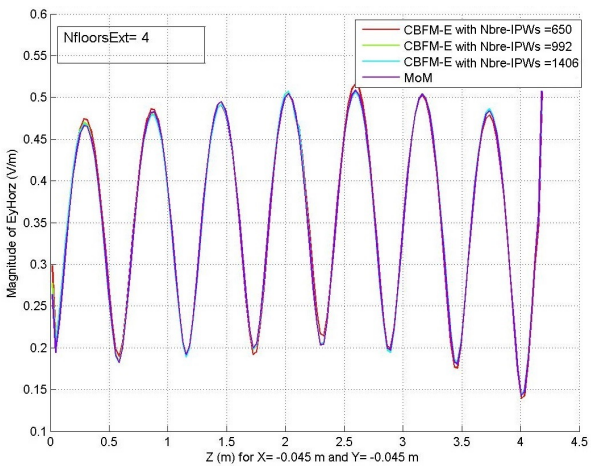
The results plotted in Figures 2.14 confirm the estimates of the N_{IPWs} values, given in Figure 2.13, needed to be retained to obtain the desired accuracy. We note that using $N_{IPWs} = 992$ incident plane waves for the generation of the characteristic basis functions (CBFs) is adequate to ensure the accuracy of the CBFM-E. This conclusion is further strengthened by observing the backscattered field for a mono-static configuration, with $\theta_{i,s} = [0°; 1°; 89°]$ and $\varphi_{i,s} = 0°$. The results plotted in Figure 2.15 show that CBFM-E achieves an excellent level of accuracy with $N_{IPWs} = 992$.

	Relative error with the CBFM-E (%)						CPU time (sec)
N_{IPWs}	Er_x^V	Er_y^V	Er_z^V	Er_x^H	Er_y^H	Er_z^H	–
650	0.88	2.01	8.11	0.2	0.52	3.21	963
992	0.2	0.24	1.16	0.17	0.18	1.59	1315
1406	0.21	0.22	1.56	0.15	0.23	1.64	1996

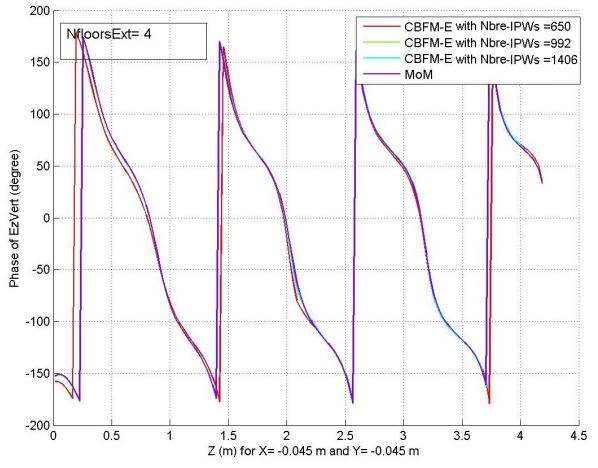
TABLE 2.5: The relative error introduced with the CBFM-E for the three values of N_{IPWs}.



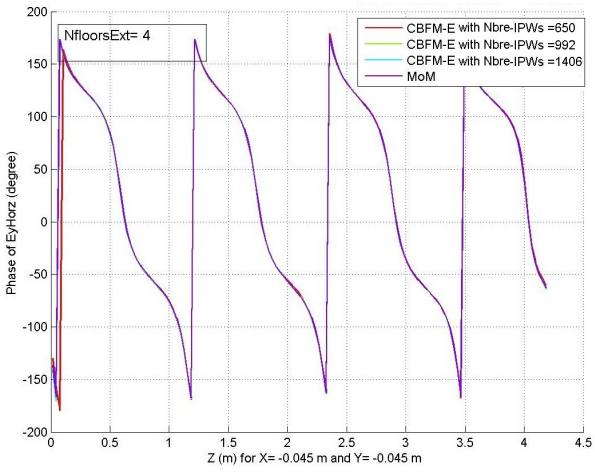
(A) E_z^V



(B) E_y^H

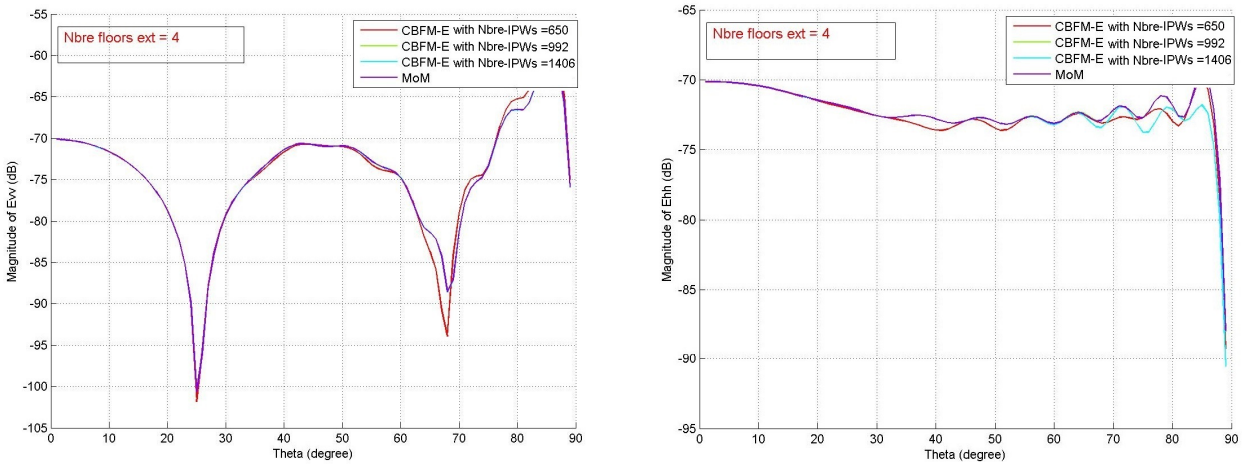


(C) ϕ_z^V



(D) ϕ_y^H

FIGURE 2.14: *A comparison of the magnitudes E_z^V and E_y^H and the phases ϕ_z^V and ϕ_y^H of the electric field inside the scatterer for $x = y = -0.045$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM and with three N_{IPWs} values (650, 992 and 1406).*



(A) E_{VV} (B) E_{HH}

FIGURE 2.15: *The magnitudes of the backscattered field computed for* $\theta_{i,s} = [0°; 1°; 89°]$ *and* $\varphi_{i,s} = 0°$ *by the CBFM-E and with the 3* N_{IPWs} *values (650, 992 and 1406).*

2.2.1.2 Simulation 2 :

For the second simulation, we follow the same lines as in the previous case and derive the magnitude and phase of the total electric field inside the same dielectric cylinder obtained for the incident angle ($\theta_i = 30°$; $\varphi_i = 0°$), for three different N_{IPWs} values. The results computed with the CBFM-E are compared with those obtained from the MoM in Figure 2.16. The relative error (%) introduced by the CBFM-E compared to the MoM for the three numerical experiments is represented in Table 2.6.

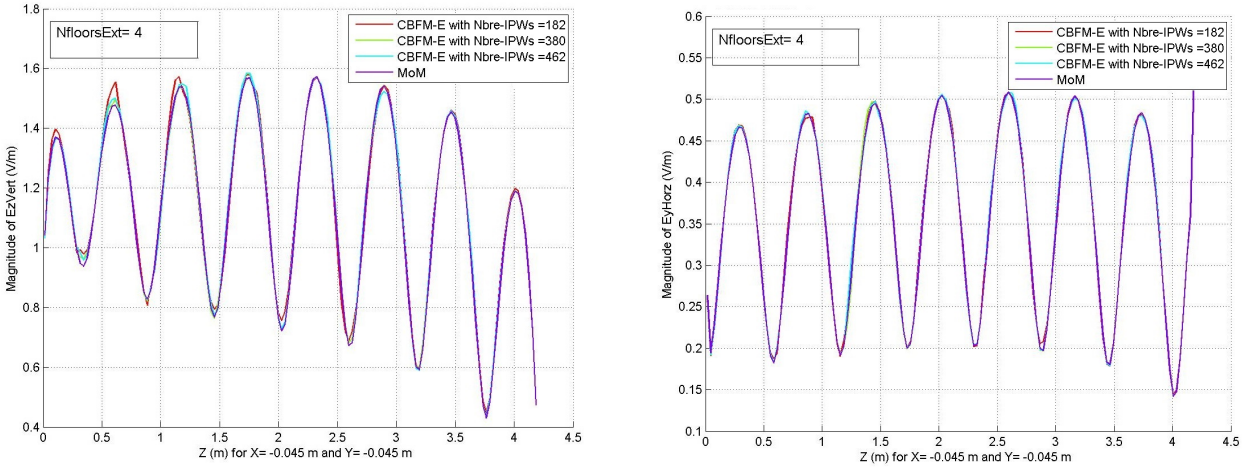
	Relative error with the CBFM-E (%)						CPU time (sec)
N_{IPWs}	E_x^V	E_y^V	E_z^V	E_x^H	E_y^H	E_z^H	–
182	0.81	0.82	1.69	0.29	0.33	0.58	185
380	0.33	0.39	1.03	0.17	0.19	0.47	217
462	0.39	0.34	1.08	0.14	0.2	0.37	218

TABLE 2.6: The relative error introduced with CBFM-E for the 3 values of N_{IPWs}.

We notice that the relative error stabilizes when $N_{IPWs} = 182$ which confirms, once again, the results given in Figure 2.13. Finally, the magnitude and phase of the backscattered field computed with a mono-static configuration defined by $\theta_{i,s} = [0°; 1°; 89°]$ and $\varphi_{i,s} = 0°$ are plotted in Figure 2.17.

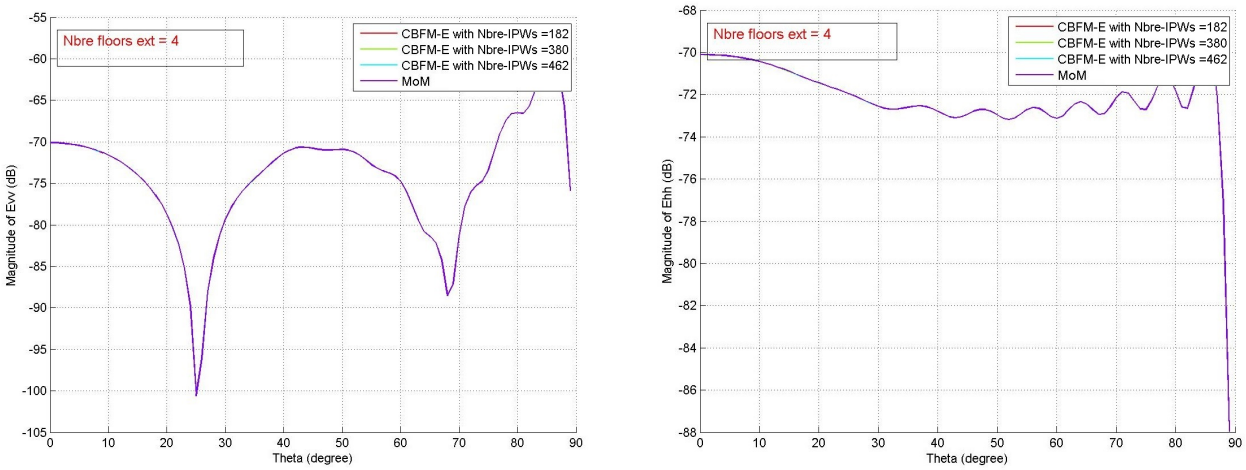
Finally to summarize, the study done in this section confirms, first of all, the crucial effect that the number of incident plane waves used during the generation of the CBFs, N_{IPWs}, has on the performances of the CBFM in terms of computation time and accuracy. Furthermore, since we know that the size of the cells T_c composing the scatterer is always taken around $\frac{\lambda}{10}$, we can retain a constant N_{IPWs} value equal to 182 for CBFM blocks of a height h equal to λ and so equivalent to 10 floors of

cells along the vertical axis. This combination (block size, N_{IPWs}) ensures a good level of accuracy obtained on a reasonable computing time.



(A) E_z^V (B) E_y^H

FIGURE 2.16: *A comparison of the magnitudes E_z^V and E_y^H of the electric field inside the scatterer for $x = y = -0.045$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM and with 3 N_{IPWs} values (182, 380 and 462).*



(A) E_{VV} (B) E_{HH}

FIGURE 2.17: *The magnitudes E_{VV} and E_{HH} of the backscattered field computed for $\theta_{i,s} = [0°; 1°; 89°]$ and $\phi_{i,s} = 0°$ by the CBFM-E and the MoM with the 3 N_{IPWs} values (182, 380 and 462).*

In the next paragraph, we show that reducing the number of incident plane waves (N_{IPWs}) is not the only benefit of reducing the number of unknowns per block. Indeed, the size of the CBFM blocks has also a crucial effect on the performance of the CBFM.

2.2.2 Effect of the blocks size on the CBFM performances :

In this paragraph, we look into the effect of the number and height of the CBFM blocks on the computing time and on the compression rate. In fact, since the number of CBFs retained for each block is much lower than the number of original low-level basis functions for the same block, we define the compression rate CR as the ratio between the number of the original basis functions and the number of post-CBFM unknowns. The inverted Compression rate, namely ICR is defined as

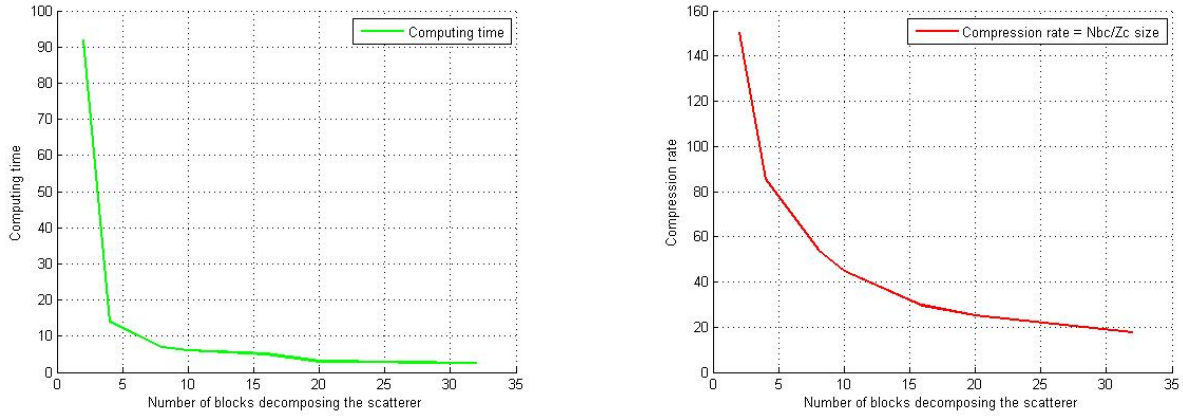
$$ICR\ (\%) = \frac{\text{Size of } Z^c}{\text{Size of } Z^{MoM}} \times 100 = \frac{K}{3Nb_c} \times 100 \tag{2.16}$$

Therefore, we apply the extended version CBFM-E to a homogeneous dielectric cylinder placed over a dielectric ground of complex relative permittivity of $\epsilon_g = 5 + 3.6\text{j}$. The cylinder whose dimensions are $0.18 \times 0.18 \times 4.8\ m$, has a complex relative permittivity of $\epsilon_c = 6 + 0.3\text{j}$ and is illuminated by an incident plane wave, at a frequency $f = 300$ MHz which carry out a wavelength inside the scatterer equal to $\lambda_s = 0.32$ m. The cylinder is discretized with a cell size $T_c \approx \frac{\lambda}{10} = 0.03$ m which results on a total number of cells $Nb_c = 5760$. Hence the total number of original basis functions is equal to $3 \times Nb_c = 17280$.

To study the influence of the parameter N_{Blocks} on the CBFM performances, we make seven successive experiences with the scatterer described bellow, while increasing the number of blocks N_{Blocks} decomposing it. The different simulation conditions and the results are given in table 2.7 and Figure 2.18. h_{Block} refers to the height of the CBFM block, $N_{IPWs,needed}$ refers to the minimum number of plane waves needed to ensure, according to the previous subsection, the accuracy of the results and finally Z^c refers to the final reduced matrix generated thanks to the CBFM process. We note that increasing the total number of blovks, N_{Blocks}, and as a result, reducing the block size, seems very attractive in term of computing time. However, it engender, as well, regrettably a decrease of the compression rate which make the CBFM process expensive in term of memory consumption and thus cancel one of the strength of this method.

N_{Blocks}	2	4	8	10	16	20	32
$h_{Block}\ (m)$	2.4	1.2	0.6	0.48	0.3	0.24	0.15
$\frac{r_0}{\lambda_s}$	3.75	2	< 1	< 1	< 1	< 1	< 1
$N_{IPWs,needed}$	462	380	182	182	182	182	182
$Time$	1 h 32 min	14 min	7 min	6 min	5 min	3 min	3 min
Z^c size	115	202	319	384	585	688	985

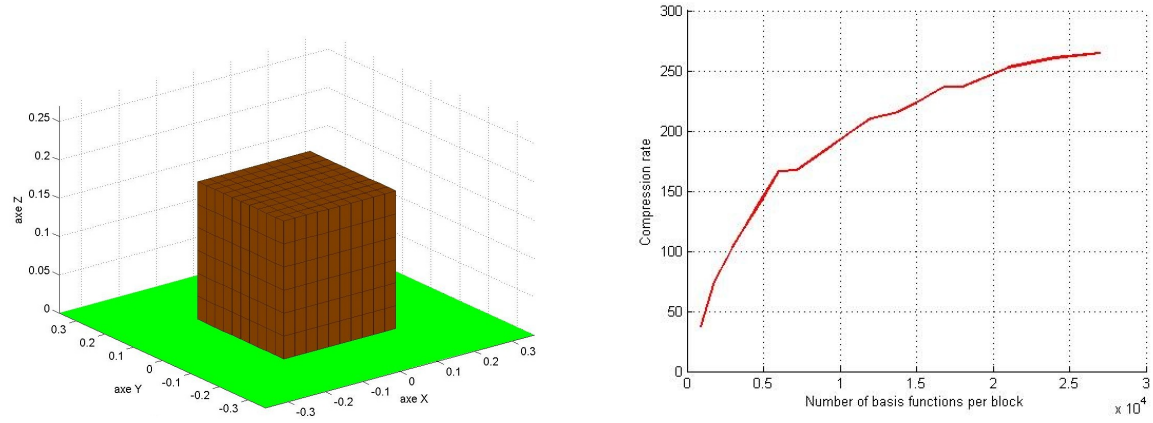
TABLE 2.7: The computing time and the Z^c size depending on the size of the CBFM blocks composing the scatterer.



(A) (B)

FIGURE 2.18: *The effect of N_{Blocks} on the computing time and on the compression rate*

An other simulation is done separately to study the relation between the size of a single block cylinder and the compression rate. We apply the CBFM-E on this cylinder while increasing its size and we note each time the number of generated CBFs. The results plotted in Figure 2.19 corroborate the fact that the compression rate increases with the size of the block.

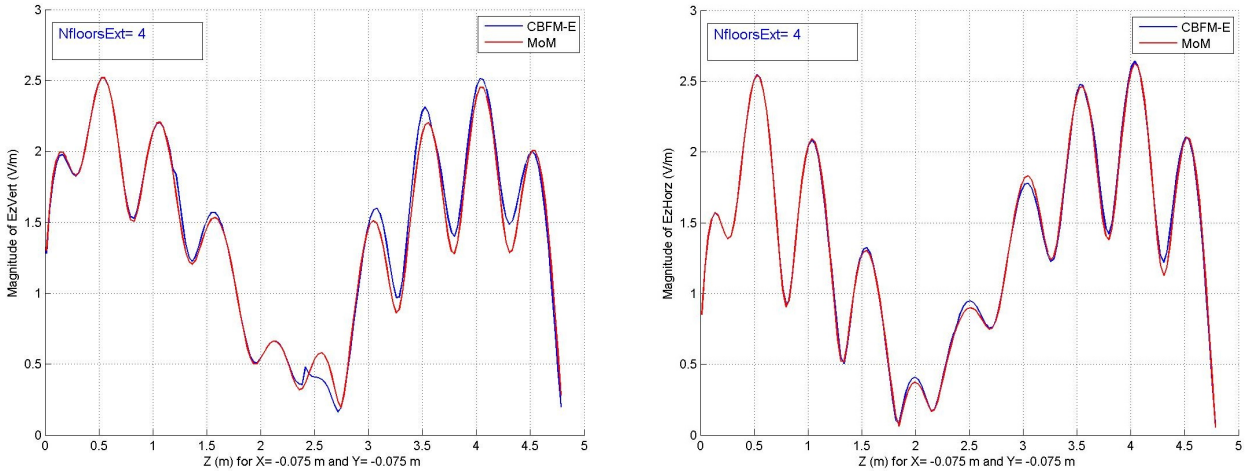


(A) (B)

FIGURE 2.19: *Compression rate versus the number of basis functions per block.*

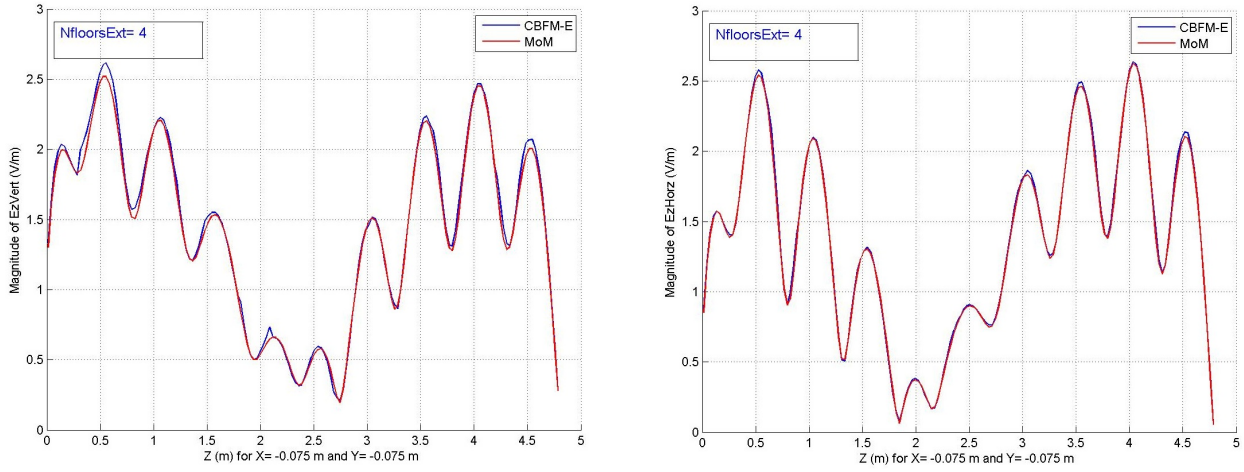
Therefore, it is tempting to increase the blocks sizes as much as possible to achieve a high compression rate. However, as may be seen from table 2.7, the use of large blocks, together in the context of the mono-level CBFM, not only results in a significant increase of the overall CPU time, but also makes it less suitable for an eventual parallelization. To avoid these disadvantages, we prefer to divide, as written in the previous subsection, the scatterer on blocks of around λ of height. We will overcome the engendered compression rate problem thanks to a new multilevel scheme of the CBFM. This scheme named ML-CBFM will be generously detailed on the third chapter of this report.

Carrying on with the N_{Block}, we check now its impact on the accuracy of the CBFM results. The magnitude of the total electric field inside the dielectric cylinder obtained for the incidence direction ($\theta_i = 30°$; $\varphi_i = 0°$), with $N_{Block} = 4$ then $N_{Block} = 16$, computed with the CBFM-E procedures in comparison to the results given by the MoM are plotted respectively in Figure 2.20 and 2.21.



(A) E_z^V (B) E_z^H

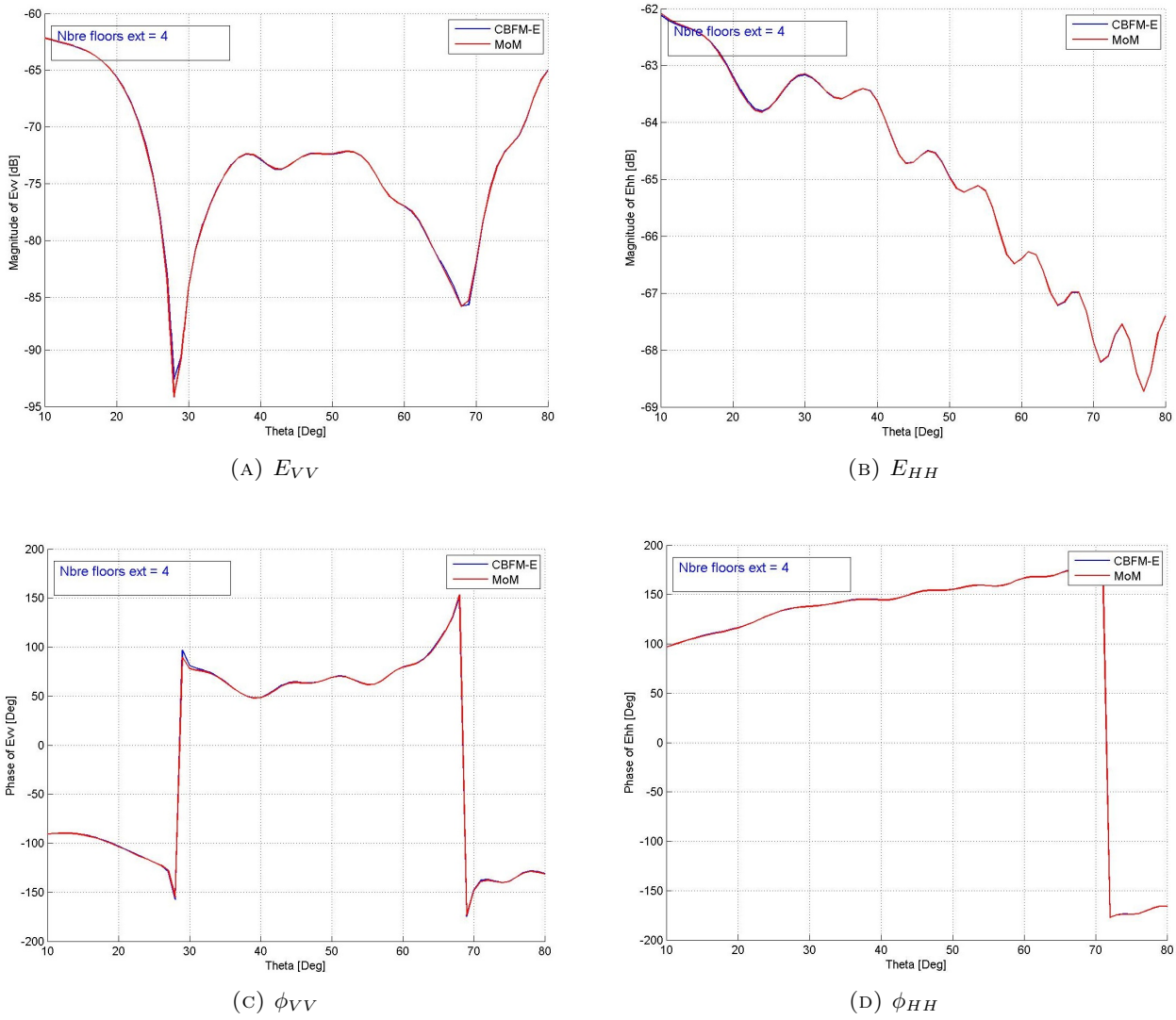
FIGURE 2.20: *The magnitudes E_z^V and E_z^H of the electric field inside the scatterer for $x = y = -0.075$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM. The scatterer is divided on 4 blocks.*



(A) E_z^V (B) E_z^H

FIGURE 2.21: *The magnitudes E_z^V and E_z^H of the electric field inside the scatterer for $x = y = -0.075$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM. The scatterer is divided on 16 blocks.*

We compute the backscattered field for the both cases considering a mono-static configuration defined by $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$. The results are displayed in Figure 2.22 for $N_{Block} = 4$ and Figure 2.23 for $N_{Block} = 16$.



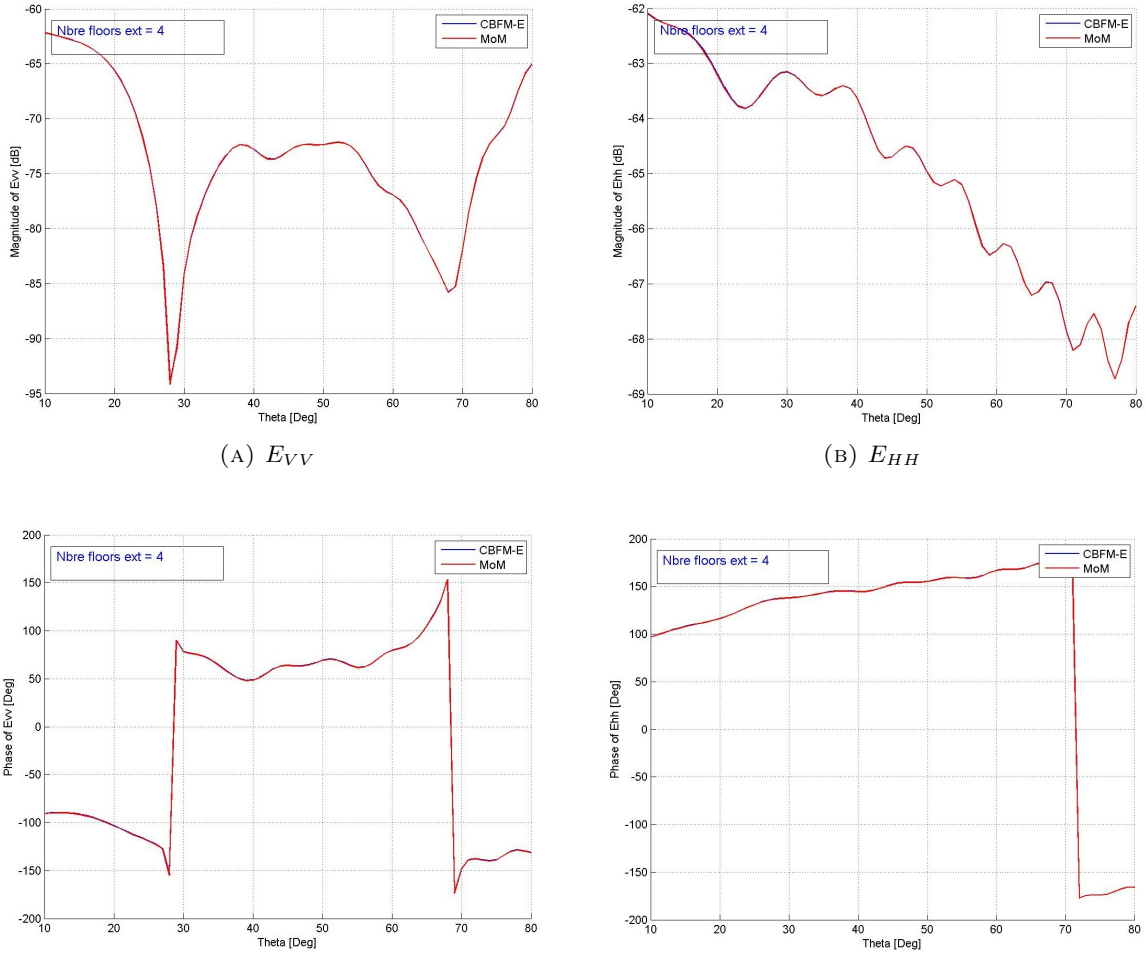
(A) E_{VV} (B) E_{HH}

(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 2.22: *The magnitudes E_{VV} and E_{HH} and the phases ϕ_{VV} and ϕ_{HH} of the backscattered field computed for $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$ by the CBFM-E and the MoM for $N_{Block} = 4$*

We note that, since the number of incident waves N_{IPWs} fit in with the height of the CBFM block h, the latter, and consequently N_{Block}, have not a great influence on the accuracy of the results. To confirm this observation, we get back over the 7 simulations of Table 2.7 and we display the fields while $N_{IPWs} = 462$ (see Figure 2.24 and 2.25).

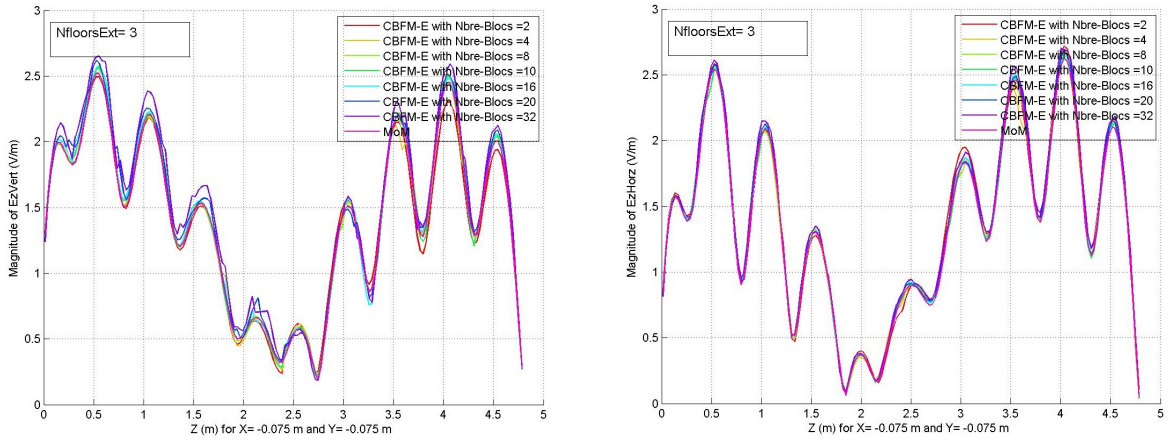
The optimal solution is finally the one that was adopted in the previous subsection. To avoid to overly divide the scatterer but keep in the same time quite a reasonable computation time, we will divide the cylinder on blocks of $h = \lambda$ each and illuminate, consequently, each block, with $N_{IPWs} = 182$ incident plane waves while generating the characteristic basis functions.



(A) E_{VV} (B) E_{HH}

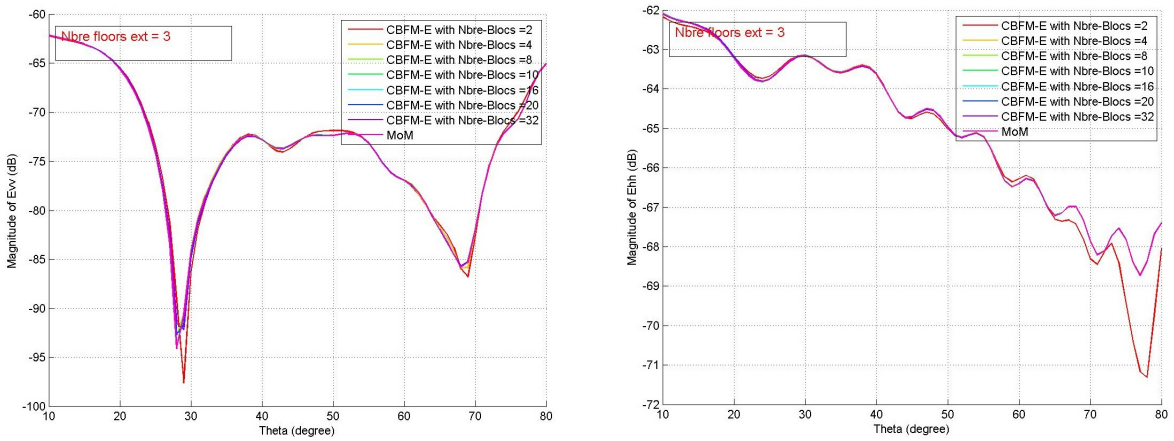
(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 2.23: *The magnitudes E_{VV} and E_{HH} and the phases ϕ_{VV} and ϕ_{HH} of the backscattered field computed for $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$ by the CBFM-E and the MoM for $N_{Block} = 16$*



(A) E_z^V (B) E_z^H

FIGURE 2.24: *The magnitudes E_z^V and E_z^H of the electric field inside the scatterer for $x = y = -0.075$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM while $N_{Block} = [2, 4, 8, 10, 16, 20, 32]$.*



(A) E_{VV} (B) E_{HH}

FIGURE 2.25: *The magnitudes E_{VV} and E_{HH} of the backscattered field computed for $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$ by the CBFM-E and the MoM while $N_{Block} = [2, 4, 8, 10, 16, 20, 32]$*

2.2.3 The effect of the size of the buffered region on the CBFM-E performances :

The performances of the extended version of the CBFM, named the CBFM-E, depends on the number of floors in the buffered region (along the vertical axe z). For the sake of clarity, we define in Figure 2.26 a three block scatterer composed of 17 floors along the vertical axis z distributed as follow [Block 1 = 5 floors; Block 2 = 7 floors; Block 3 = 5 floors]. Applying the CBFM-E, to this example, with two extended floors is equivalent to applying, as shown in Figure 2.26 the CBFM-N to three blocks of respectively 7, 11 and 7 floors along the vertical axis z.

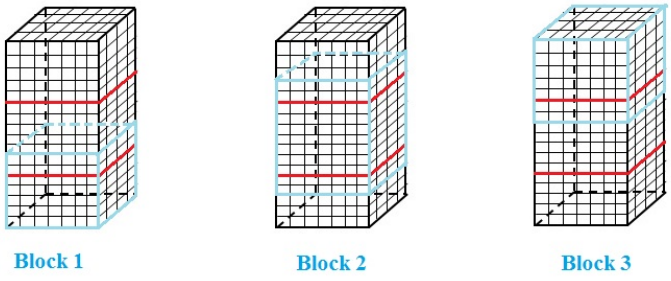


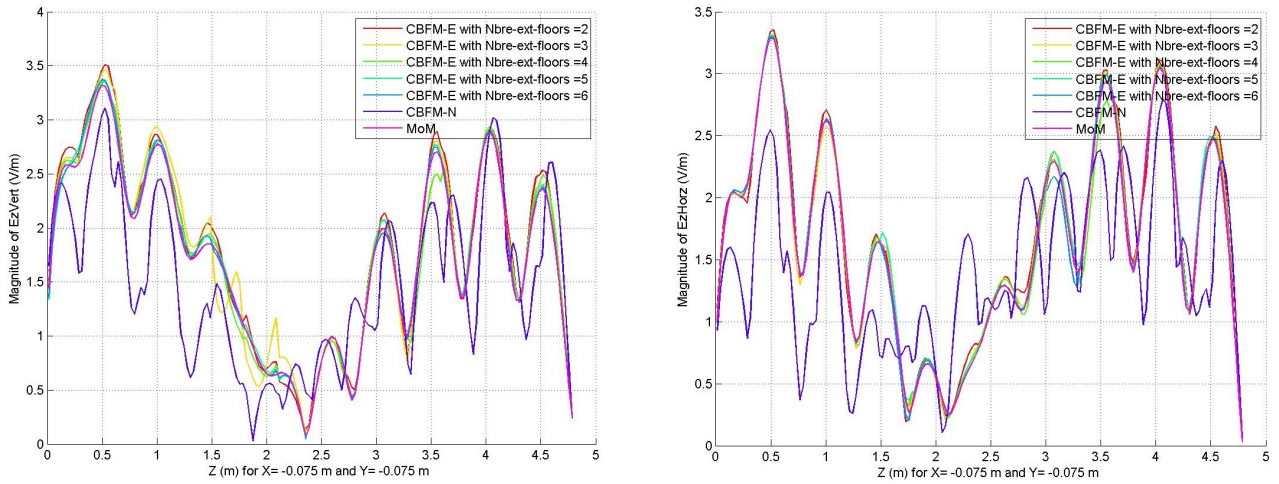
FIGURE 2.26: *Block distribution while applying the CBFM-E with 2 extended floors on a scatterer divided into 3 blocks.*

To study this parameter, named $Nber_{ext,floors}$ for number of extended floors, we apply the CBFM-E on the example of the previous subsection divided in 16 blocks (10 floors by block) illuminated each by 252 incident plane waves with five different values of $Nber_{ext,floors}$. The impact of the variations of this parameter on the computation time and the size of the reduced matrix Z^c is represented in table 2.8. We display also in this table the performances of the CBFM-N and the MoM.

	CBFM-E					CBFM-N	MoM
$Nber_{ext,floors}$	2	3	4	5	6	–	–
CPU time	3 *min* 22 *sec*	4 *min* 42 *sec*	6 *min*	6 *min* 45 *sec*	7 *min*	2 *min*	29 *min*
Z^c size	518	538	558	580	592	485	17280

TABLE 2.8: The computing time and the Z^c size depending on the size of the buffered region.

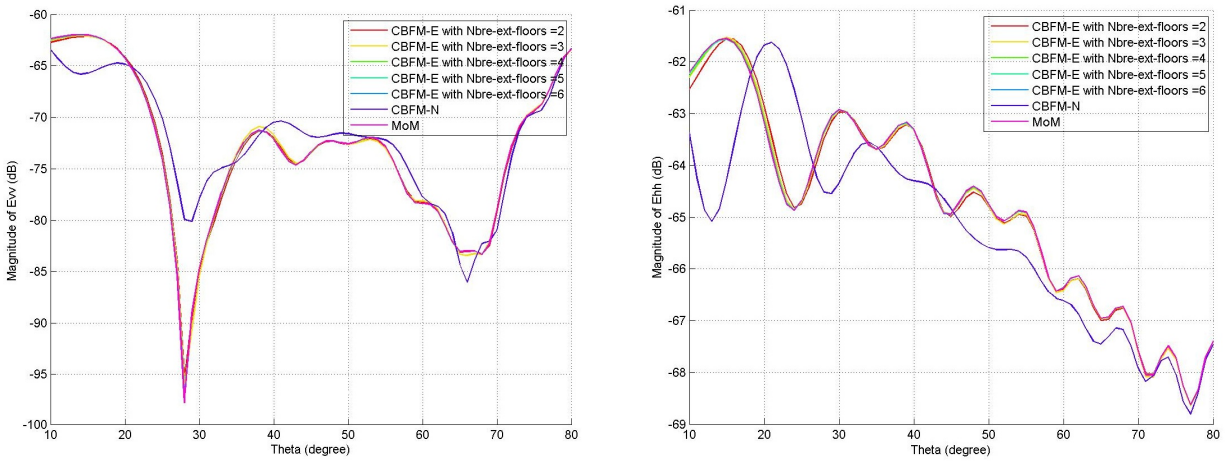
As expected, the addition of a buffered region to each block while generating the CBFs and then its lengthening increase the computation time and the size of the final reduced matrix Z^c. Naturally, this increase is the cost of the improvements brought by the additional floors on the accuracy of the CBFM process. In order to check this fact, we compute the magnitude of the total electric field inside the scatterer obtained for the incidence direction ($\theta_i = 30°$; $\varphi_i = 0°$) and the backscattered field with a mono-static configuration defined by $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$. The calculation is done with $Nber_{ext,floors} = [2, 3, 4, 5, 6]$. We display the results and the relative error introduced by the CBFM in comparison to the results given by the MoM in Figure 2.27, 2.28 and 2.29 and table 2.9.



(A) E_z^V (B) E_z^H

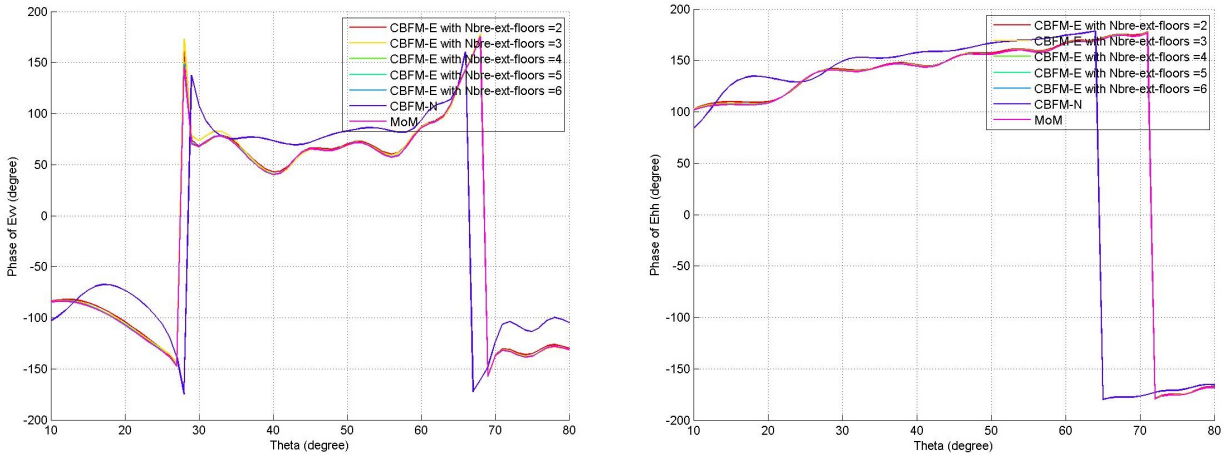
FIGURE 2.27: *The magnitudes E_z^V and E_z^H of the electric field inside the scatterer for $x = y = -0.075$ m computed for ($\theta_i = 30°$; $\varphi_i = 0°$) with the CBFM-E and MoM while $Nber_{ext,floors} = [2, 3, 4, 5, 6]$.*

Indeed, we note that increasing the size of the the buffered region leads to a decrease in the relative error made by the CBFM. Moreover, the CBFM-N, deprived of this region, induces a significant degradation in the quality of the results. We notice that starting from $Nber_{ext,floors} = 4$, the CBFM-E achieves a satisfactory accuracy when compared to the MoM with a relative error introduced on the scattered field lower than 0.5 % both for Er_{VV} and Er_{HH}.



(A) E_{VV} (B) E_{HH}

FIGURE 2.28: *The magnitudes E_{VV} and E_{HH} of the backscattered field computed for $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$ by the CBFM-E , the CBFM-N and the MoM for $Nber_{ext,floors} = [2, 3, 4, 5, 6]$.*



(A) ϕ_{VV} (B) ϕ_{HH}

FIGURE 2.29: *The phases ϕ_{VV} and ϕ_{HH} of the backscattered field computed for $\theta_{i,s} = [10°; 1°; 80°]$ and $\varphi_{i,s} = 0°$ by the CBFM-E , the CBFM-N and the MoM for $Nber_{ext,floors} = [2, 3, 4, 5, 6]$.*

In conclusion to this section, we notice the crucial effect that have some parameters on the CBFM performances in terms of computing time, compression rate and accuracy of the results comparing to a classical MoM. To ensure a high efficiency with the CBFM process, we must take into consideration the effect of these parameters and then operate with the most suitable configuration on block and buffered region size and plane waves number.

For the general case, we adopt the solution described above, we retain a N_{IPWs} equal to 182 with CBFM blocks of 10 floors of cells along the vertical axis and for the CBFM-E, a buffered region of about 4 floors seems enough to ensure the accuracy of the results obtained with the extended version of the CBFM.

	CBFM-N	CBFM-E				
$Nber_{ext,floors}$	0	2	3	4	5	6
Er_x^V	9.66	1.67	1.94	0.92	0.97	0.67
Er_y^V	4.59	1.42	2.07	1	0.53	0.47
Er_z^V	29.99	5.05	4.95	2.98	1.92	1.45
Er_x^H	4.35	1.03	0.96	0.65	0.57	0.44
Er_y^H	9.88	1.71	1.3	0.95	0.71	0.54
Er_z^H	31.96	3.83	2.38	1.9	1.49	1.23
Er_{VV}	16.24	1.68	1	0.38	0.18	0.07
Er_{HH}	16.14	1.77	0.72	0.35	0.14	0.05

TABLE 2.9: The relative error introduced (%) by the CBFM-N and CBFM-E compared to the MoM depending on the number of extended floors $Nber_{ext,floors}$

2.3 Validation of the CBFM on electrically large simulation Scene

While implementing the CBFM-E, we take advantage of the fact that the CBFs, for each block, are computed independently to accelerate the CBFM process. Consequently, we realize that CBFM is highly amenable to parallelization. Towards this end, we implement, with Intel Fortran (64 bits), OpenMP directives in a Fortran program running on a shared memory workstation equipped with an Intel Xeon x5560 (8 tasks) at 2.8 GHz and 48 GB of RAM. The OpenMP directives are employed, not only while we compute the CBFs for the different CBFM blocks, but also while we generate the reduced final linear equation system, which leads to a significant gain in term of the total CPU time. Furthermore, for two identical CBFM blocks (same dimensions and same dielectric properties), the CBFs need to be computed only once and saved for later use to generate the reduced linear equation system, as well as to compute the total electric field inside the scatterer. Copying the CBFs computed previously for similar blocks, instead of computing them anew, enables a significant reduction of the computing time without compromising the accuracy of the CBFM-E solution.

Several examples are shown in this section to illustrate the performance of the CBFM-E while handling cases with electrically larger forest simulation scenes. The simulation conditions are chosen while taking into account all the guidelines discussed previously. We expect a significant reduction in the computation time while maintaining a level of accuracy comparable to that of the conventional MoM. All the simulations have been performed for a frequency $f = 300$ MHz, and the cylinders modeling the tree trunks and branches have been discretized with a cell size $T_c = 0.03$ m. We apply the CBFM to the simulation scenes with a block height h_B almost equal to 0.3 m (10 floors of cells along the vertical axis z) and with an $N_{IPWs} = 380$.

We begin with a moderately large problem with $Nb_c = 11640$ cells. The forest scene, made up of 40 cubic cylinders modeling 8 trees with 4 branches each, is represented in Fig. 2.30. The cylinders

have a complex relative permittivity of $\epsilon_s = 9.6 + 0.01\text{j}$, and they are placed over a dielectric ground of complex relative permittivity of $\epsilon_g = 5 + 3.6\text{j}$. The trees are spaced apart by 0.8 m, and their maximum height is equal to 3.3 m. The variations of the magnitudes and phases of the backscattered fields for the HH polarization with a mono-static configuration defined by $\theta_i = \theta_s = [10°; 1°; 80°]$ and $\varphi_i = \varphi_s = 40°$ are shown in Fig. 2.31. Table 2.10 provides a comparison between the performances of the CBFM-E and the classical MoM in terms of CPU time and memory consumption. We also display in this Table the relative difference (E_r) introduced by the CBFM-E in the backscattered field, for both the VV and HH polarizations when compared with the MoM.

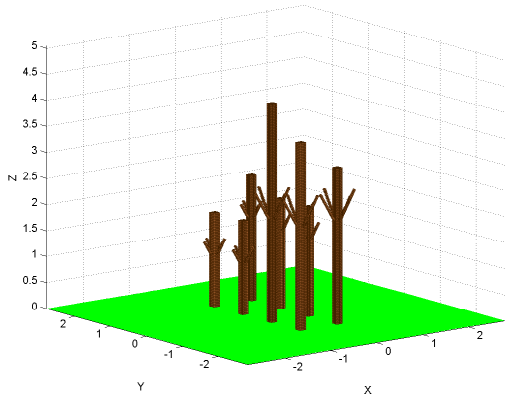
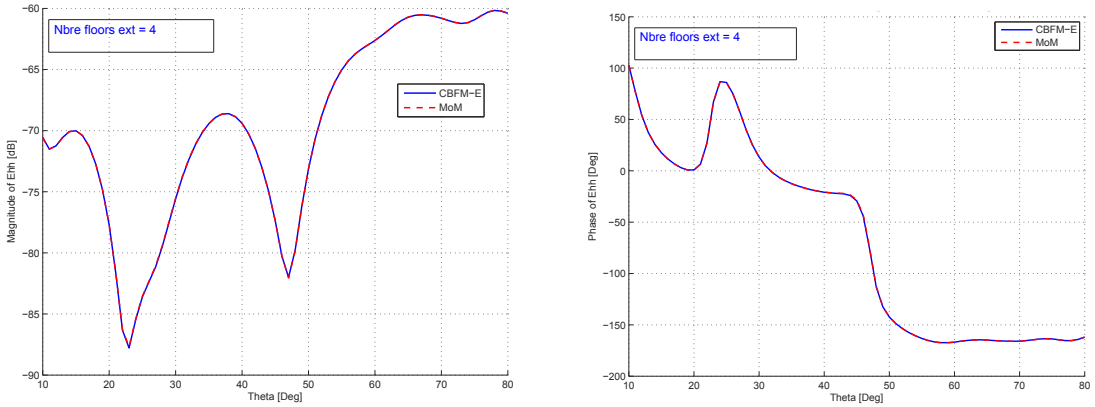


FIGURE 2.30: The simulation scene modeling 8 trees placed over a real ground plane with $\epsilon_g = 5+2.1\text{j}$.

TABLE 2.10: Performances of CBFM-E in terms of reduced final matrix size, computation time and relative error (%) in comparison with the MoM.

	Z_c size	CPU time (min)	Er_{VV}	Er_{HH}
CBFM-E	2480	6	0.45	0.
MoM	34920	146	–	–



(A) E_{HH} (B) ϕ_{HH}

FIGURE 2.31: Variation of the magnitude and phase of the backscattered fields (HH), computed with the CBFM-E and the MoM with a mono-static configuration defined by $\theta_i = \theta_s = [10°; 1°; 80°]$ and $\varphi_i = \varphi_s = 40°$ and with $f = 300$ MHz and $R_{t,r} = 2000$ m.

We observe that the CPU time required by the CBFM-E is 25 times lower than that of the MoM. In addition, the CBFM-E ensures a compression rate of about 14 without actually affecting the accuracy of the results. Therefore, these observations attest to the ability of the CBFM algorithm to overcome the limitations faced by the conventional MoM in terms of computation time and memory use.

For the next example, we consider a much larger case for which $Nb_c = 101800$ cells, corresponding to a maximum height of the trees equal to 10 m. The simulation scene, shown in Fig. 2.32 is composed of 45 cylinders, modeling 9 trees with 4 branches each. The trees are spaced almost 2 m apart. Since the RAM available (48 GB) does not allow us to handle an MoM matrix whose size is larger than about 36000×36000, the conventional MoM becomes untenable starting from $Nb_c \approx 12000$. Hence, we have only used the CBFM-E to handle such a large electromagnetic problem. The variations of the magnitude and phase of the backscattered fields for the VV polarization with a mono-static configuration defined by $\theta_i = \theta_s = [1°; 1°; 80°]$ and $\varphi_i = \varphi_s = 40°$, are shown in Fig. 2.33.

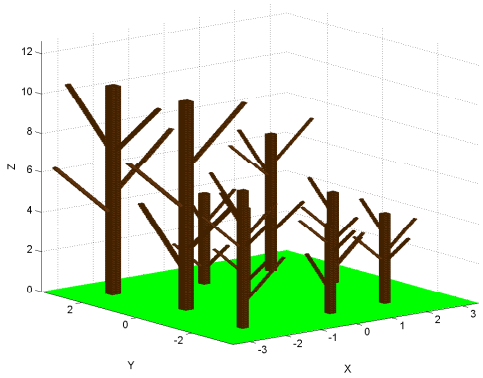
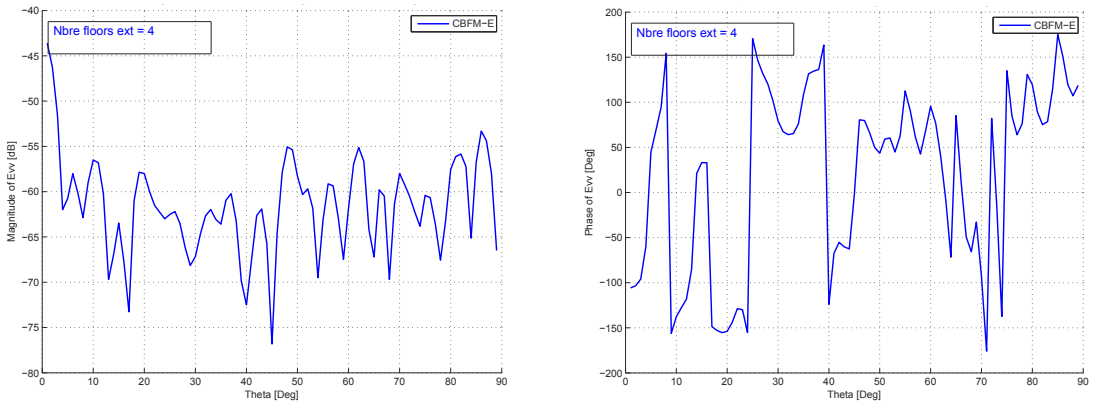


FIGURE 2.32: The simulation scene modeling 9 trees placed over a real ground plane with $\epsilon_g = 5+2.1\text{j}$.



(A) E_{VV} (B) ϕ_{VV}

FIGURE 2.33: Variation of the magnitude and phase of the backscattered fields (VV), computed with the CBFM-E for a mono-static configuration defined by $\theta_i = \theta_s = [1°; 1°; 89°]$ and $\varphi_i = \varphi_s = 40°$.

While the classical MoM is totally unable to handle this large example because of the limitation of memory resources, the CBFM-E enables us to compute the backscattered field in 104 minutes, thanks to the compression rate $CR = 21.93$.

2.4 Conclusion

In this chapter, an extended version of the characteristic basis function method (CBFM-E) has been applied, to 3D dielectric scatterers representing a forest scene. The CBFM-E exhibits a significantly improved performance for this 3D forest simulation, both in terms of the computing time and the memory usage, when compared to the conventional MoM.

To ensure that the CBFM-E achieves a good level of accuracy while computing the scattered field without using excessive memory resources while using the least amount of CPU time, we must stay within certain range of values when setting CBFM parameters, such as the block height h_{Block} and the number of incident plane waves (N_{IPW}) used to generate the macro-basis functions (MBFs). Once properly set, the CBFM-E is so efficient that it is able to treat, in just a few minutes, electromagnetic problem sizes totally intractable with the classical MoM.

The obtained results will certainly help stimulate increased interest in solving problems involving even larger structures, including forest areas, and performing simulations at higher frequencies.

Chapter 3

The Multilevel Characteristic Basis Function Method (ML-CBFM)

Even if the conventional CBFM, particularly with its extended version, provides a significant improvement in terms of CPU time and memory consumption comparing to the classical MoM, it is obvious that the CBFM process will face a memory insufficiency problem when the size of the generated reduced matrix becomes large. Therefore, in order to increase the compression rate and consequently to reduce the size of this final matrix, a multilevel CBFM scheme has been developed. This new process consists of recursive application of the mono-level CBFM and it leads to a better compression of the associated matrix and thus, extends the CBFM range of applicability.

Hence, in this chapter, we present an overview of the multilevel characteristic basis function method (ML-CBFM). We detail its numerical procedure and we give several primary results to demonstrate the ability of the ML-CBFM procedure to increase the compression rate and, thus, to handle electrically large problems.

3.1 Motivations

As explained previously in paragraph 2.2.2, the height of the block namely h_{Block}, chosen to decompose the simulation scene at the beginning of the CBFM procedure, has a crucial effect on the computing time and the compression rate. In fact, increasing h_{Block} results in a significant increase of the compression rate CR but unfortunately, at the same time, engenders a dramatic increase of the corresponding CPU time (as shown in Figure 2.18). This is due to the increase of the number of cells per block which considerably slows down the CBFM procedure during the resolution of the linear equation system specific to each block (see Equation (2.4)) and during the generation of the final reduced matrix ((see Equation (2.13))). In addition, as detailed in paragraph 2.2.1, increasing h_{Block} results in an increase of the number of incident plane waves N_{IPWs} needed to ensure the accuracy of the CBFM

solution which creates further computational burden for the resolution of the equation (2.4). And finally, greater CBFM blocks would mean a smaller number of blocks which reduces the potential of the parallelization of the CBFM code especially when the number of available CPUs is quite important which will engender a situation of under-exploited computer resources. Let us take the example of a large forest simulation scene discretized into $Nb_c = 262040$ cells. Two simulations were executed, using a shared memory workstation equipped with 8 processors, with two different values of CBFM block height, respectively $h_{Block,Sim1} = \lambda$ and $h_{Block,Sim2} = 4\lambda$. The forest simulation scene is represented in Figure 3.1 and the conditions and results corresponding to these two simulations are shown in Table 3.1.

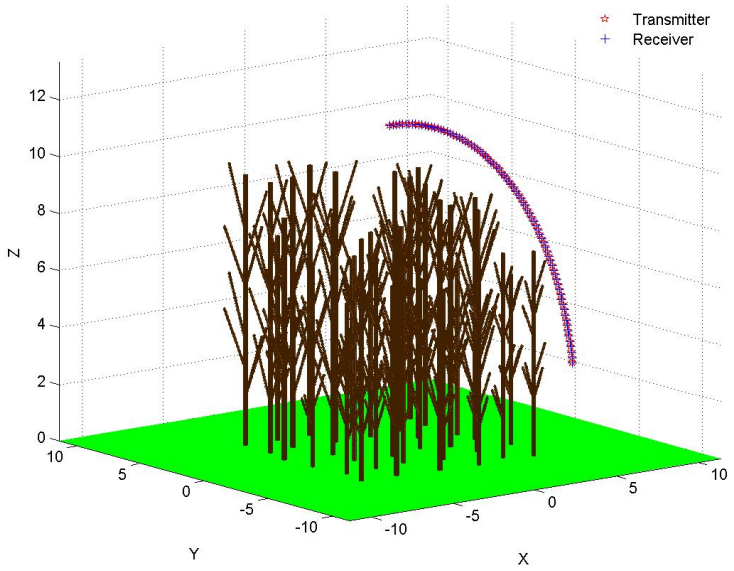


FIGURE 3.1: A numerically large simulation scene modeling 36 trees placed over a real ground plane with $\epsilon_g = 5 + 2.1\text{j}$.

TABLE 3.1: Performances of CBFM-E in terms of compression rate and computation time depending on the height of the CBFM blocks namely h_{Block}.

	Compression Rate					Computing time (min)	
h_{Block}	$Nbc_{Block,max}$	Z^c size	RAM (GB)	CR	ICR (%)	CBFs	Z^c
λ	250	83080	220	9.46	10.57	6	219
4λ	1000	34507	38	22.78	4.38	56	356

Thanks to these two simulations, we observe the impact of increasing h_{Block} on the performances of the CBFM-E both in CPU time and memory resources needed to achieve the computing task. In this case, it is interesting to mention that the first simulation failed to product final results, since the available 42 GB of memory does not allow us to store and handle the corresponding final matrix.

Hence, the use of large blocks in the context of the mono-level CBFM increases enormously the need in terms of memory and the overall CPU time making, thus, untenable and inefficient the CBFM

process. To overcome these difficulties, we propose the multilevel scheme of the CBFM which will enable us the use of large blocks resulting in an important compression rate without great time consuming.

3.2 Overview of the Multilevel Scheme of the CBFM

In this paragraph, we detail the numerical procedure of the multilevel CBFM-E (MLCBFM-E) and we discuss the influence of some factors, such as the number of incident plane waves (N_{IPWs}) and the CBFM block extension, on the accuracy of the MLCBFM-E solution.

3.2.1 Numerical Procedure

The concept of the ML-CBFM is based on an iterative application of the conventional CBFM process, in which the generated CBFs are progressively grouped to form the upper-level blocks. Figure 3.2 illustrates the block decomposition with a two-level ML-CBFM.

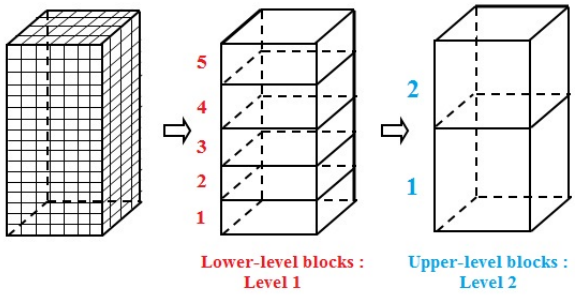


FIGURE 3.2: *A two-level decomposition of a cubic cylinder.*

In this scheme, we construct the upper-level CBFs by defining them in terms of their lower-level counterparts [41]. To clarify the process, we define a sequence of blocks $\{seq\} = \{ i_1,i_2, \ldots, i_L \}$. We also define a concatenation operation between this sequence and a new lower level block $\{seq, b\} = \{ i_1,i_2, \ldots, i_L, b \}$.

Thus, the n^{th} basis function, pointed by a sequence (seq) of blocks arranged in descending order from the higher to the lower one, is expressed in equation 3.1 :

$$\vec{F}_{\{seq\},n} = \sum_{b}^{B_{\{seq\}}} \sum_{m}^{N_{\{seq,b\}}} I^{(n)}_{\{seq,b\},m} \vec{F}_{\{seq,b\},n} \tag{3.1}$$

where B_{seq} is the number of blocks in the great block pointed by $\{seq\}$; $N_{\{seq,b\}}$ is the number of CBFs in the block pointed by $\{seq\}$; and, $I^{(n)}_{\{seq,b\},m}$, is the weight for the m^{th} CBF forming the next lower level [41].

At the end of each level, we generate, as done for a mono-level CBFM (detailed in paragraph 2.1.3), a reduced matrix Z^c_{level} representing the interactions between the CBFs formed during this level. Hence,

the elements of the final reduced matrix correspond to the reaction terms between the CBFs belonging to the highest level. The next step consists of solving the final reduced linear equation. So we compute the K_L complex coefficients $\alpha_{k,L}$ where L designate the highest level.

In order to compute the K_1 complex coefficients $\alpha_{k,1}$ corresponding to the first level, we take the opposite way, and construct the low-level α by defining them in terms of this low-level CBFs and the next upper-level α as shown in equation 3.2.

$$\begin{pmatrix} \alpha^{(l-1)}_{\{seq,1\}} \\ \alpha^{(l-1)}_{\{seq,2\}} \\ \vdots \\ \alpha^{(l-1)}_{\{seq,B_{seq}\}} \end{pmatrix} = J_{\{seq\}} \alpha^{(l)}_{\{seq\}} \tag{3.2}$$

Finally we compute the total electric field inside the scatterers as a linear combination of the K_1 CBFs computed previously for the first level weighted by the K_1 complex coefficients $\alpha_{k,1}$ as done with the mono-level CBFM.

For a sake of clarity, let us consider the example of a simple forest simulation scene comprised of 4 trees trunks of height between 3 and 4.5 m. With a simulation frequency equal to 300 MHz and a uniform dielectric constant $\epsilon_s = 6.2 + 0.2\text{j}$, the wavelength inside the scatterers is equal everywhere to $\lambda_s = 0.401$ m.

We start by dividing the electromagnetic scene into 25 CBFM blocks. Then, in order to apply a two-Level CBFM-E, we chose to divide these 25 initial blocks in 4 blocks, as shown in Figure 3.3, to be considered while computing the CBFs for the second level. In fact, each cylinder representing a tree trunk is considered as a 2^{nd}-level block gathering together six or seven 1^{st}-level CBFM blocks.

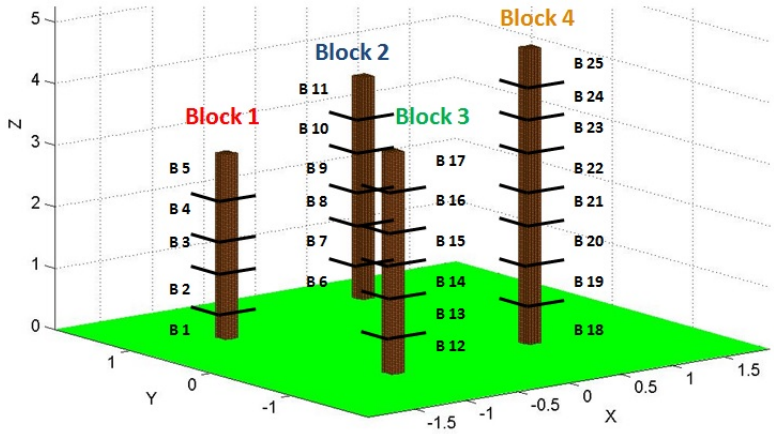
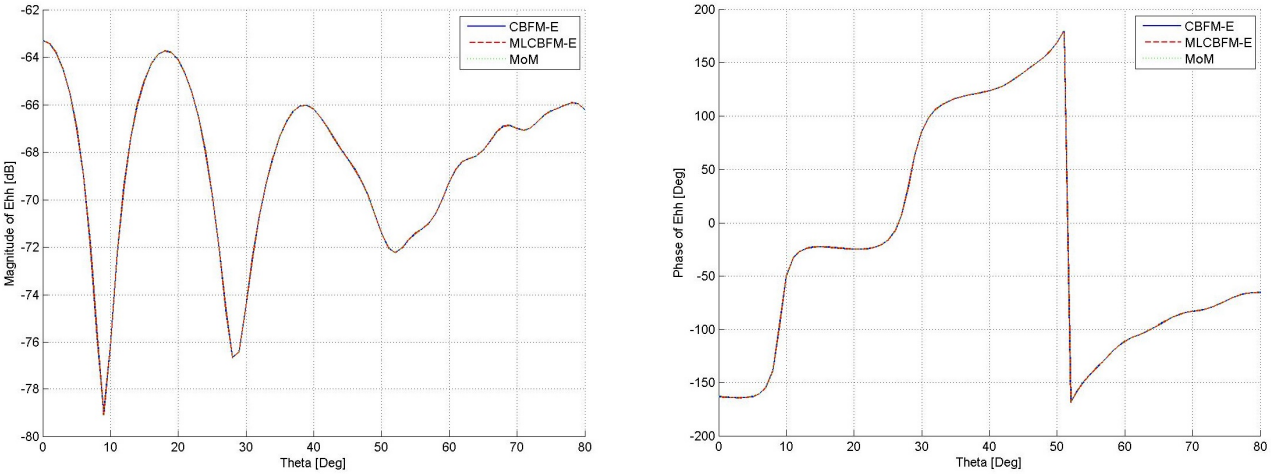


FIGURE 3.3: *The blocks dividing a 4 trees simulation scene in order to apply a 2-Level CBFM-E.*

Table 3.2 shows the conditions corresponding to the application of the two-level CBFM-E to the example of Figure 3.3 and a first comparison between the results obtained with a classical mono-level CBFM-E and those obtained with the two-level CBFM-E is illustrated in Figure 3.4.

TABLE 3.2: Conditions of the application of the CBFM-E for the 2 levels of the MLCBFM-E.

	size of $Z_{level}^{initial}$	N_{Blocks}	$h_{Block,max}$ (m)	$Nbc_{Block,max}$	Size of Z_{level}^{c}
$1^{st}level$	24000	25	$2\lambda_s = 0.6$	320	1143
$2^{nd}level$	1143	4	$12\lambda_s = 4.8$	448	405



(A) E_{HH} (B) ϕ_{HH}

FIGURE 3.4: Variation of the magnitude and phase of the backscattered fields (HH), computed with the CBFM-E, the MLCBFM-E and the MoM with a mono-static configuration defined by $\theta_i = \theta_s = [0°; 1°; 80°]$ and $\varphi_i = \varphi_s = 0°$ and with $f = 300$ MHz and $R_{t,r} = 2000$ m.

Since the multilevel scheme of the CBFM-E is based on the use of larger CBFM blocks, we need, in order to ensure the accuracy of the MLCBFM-E, to check the condition on the number of incident plane waves (N_{IPWs}) used while generating the CBFM. In fact, the question arises as to whether the value of N_{IPWs} ensuring the accuracy of the mono-level CBFM-E is satisfactory to equally ensure the accuracy of the MLCBFM-E, or we need to take into account the higher CBFM block of the higher level while evaluating the N_{IPWs} value. This issue is discussed in the next paragraph.

3.2.2 Number of Incident Plane Waves (N_{IPWs}) for the MLCBFM-E

As detailed in paragraph 2.2.1, the number of plane waves, N_{IPWs}, used to generate the CBFs for each CBFM block, has a significant impact in the accuracy of the results obtained with the CBFM-E.

For the multilevel scheme of the CBFM-E, the number N_{IPWs} is set in the beginning of the first level and used to compute the matrix $E_{ii,1}^{ref,IPWs}$ in order to generate the CBFs for the block ii by solving the following linear equation system :

$$Z_{ii,1} E_{ii,1}^{MBFs} = E_{ii,1}^{ref,IPWs} \tag{3.3}$$

where ii is one of the blocks dividing the simulation scene for the first level of the MLCBFM-E, $E_{ii,1}^{ref,IPWs}$ is a $3Nbc_{ii} \times N_{IPW}$ matrix representing the plane waves excitations for the first level and $E_{ii,1}^{MBFs}$ is an $3Nbc_{ii} \times N_{IPW}$ matrix representing the macro-domain basis functions (MBFs) associated to the block ii.

For a higher level (l), the matrix $E_{B,l}^{ref,IPWs}$ associated to the block B, is computed, as shown in equation (3.4), by combining the matrix corresponding to the previous level $(l-1)$ and the characteristic basis functions (CBFs) computed for the different block ii composing the electromagnetic simulation scene for the level (l).

$$E_{B,l}^{ref,IPWs} = \left(\sum_{b=1}^{N_b} C^{(b)t} E_{b,l-1}^{ref,IPWs}\right)^T = \left(C^{(1)t}E_{1,l-1}^{ref,IPWs} \quad C^{(2)t}E_{2,l-1}^{ref,IPWs} \quad \cdots \quad C^{(N_b)t}E_{N_b,l-1}^{ref,IPWs}\right)^T \tag{3.4}$$

where N_b is the number of blocks associated to the level $l-1$ and composing the current block B associated to the current level l and $C^{(b)}$ represents the CBFs of the block b while $1 < b < N_b$. Hence, here a crucial concern arises in connection with the choice of N_{IPWs} at the beginning of the MLCBFM-E procedure. Since the value chosen is indirectly applied to the highest level, the question to be addressed therefore is whether it is necessary, to ensure the accuracy of the MLCBFM-E, to take into account the highest height of the CBFM blocks associated to the highest level, while choosing the value of N_{IPWs}.

For this purpose, we go back to the previous simple example illustrated in Figure 3.1, and we compare the results obtained with a conventional MoM and a mono-level CBFM-E with those obtained with a two-level CBFM-E for different values of N_{IPWs}. With the decomposition of the simulation scene into $N = 25$ blocks of height $h_{Block} = 2\lambda_s$ $(r_0/\lambda = 1)$, according to the paragraph 2.2.1, a number $N_{IPWs} = 380$ of incident plane waves is satisfactory to guaranty the accuracy of the mono-level CBFM.

As shown in table 3.2, the multilevel scheme of the CBFM-E leads us to consider higher CBFM block (for the 2^{nd} level) of height $h_{Block} = 12\lambda_s$ $(r_0/\lambda = 6)$ corresponding, thus, to $N_{IPWs,needed} \approx 992$. Hence, to determine the value needed to ensure the accuracy of the MLCBFM-E solution in comparison with the mono-level CBFM, we compute the backscattered electric field while increasing $N_{IPWs,MLCBFM}$ from 380 to 1722.

The performances achieved by the MLCBFM-E are represented in table 3.3 and the computed backscattered field is illustrated in Figure 3.5 depending on the chosen value of $N_{IPWs,MLCBFM}$.

The obtained results provides an answer to our above-mentioned question and confirm that we have to take into account the highest value of h_{Block} corresponding to the highest level of the MLCBFM-E procedure in order to ensure its accuracy.

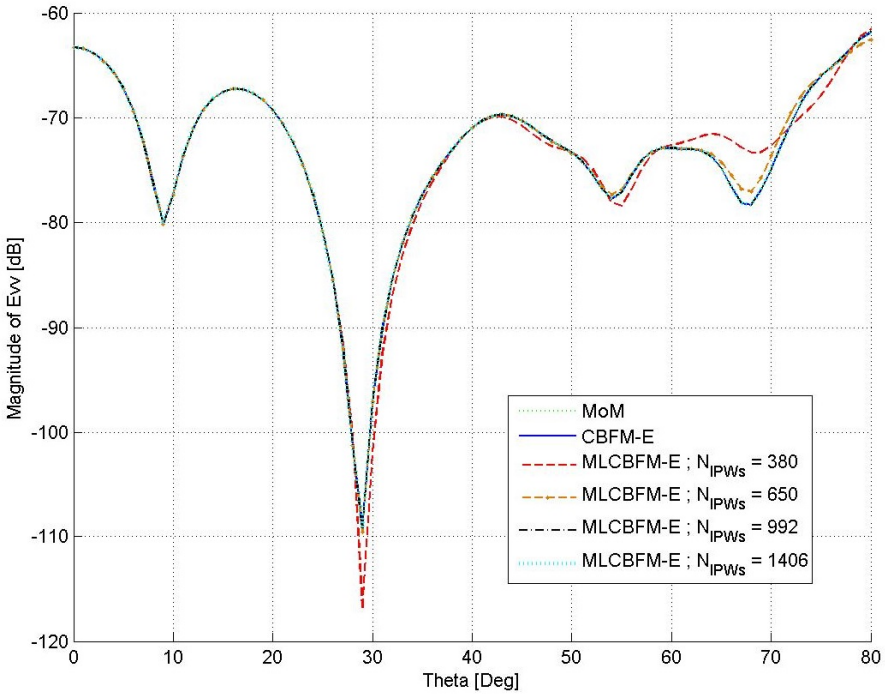


FIGURE 3.5: *The magnitude of the backscattered field (VV) computed for* $\theta_{i,s} = [0°; 1°; 80°]$ *and* $\varphi_{i,s} = 0°$ *by the MLCBFM-E and with the 4* N_{IPWs} *values (380, 650, 992 and 1406).*

	Performances of the MLCBFM-E				
N_{IPW}	Size of final Z^c	CR	ICR (%)	CPU time (sec)	Er_{VV} (%)
380	362	66.3	1.5	61	3.41
462	405	59.26	1.7	73	4.45
650	456	52.6	1.9	80	1.23
992	484	49.6	2.02	119	0.
1406	490	48.98	2.04	153	0.
1722	490	48.98	2.04	180	0.

TABLE 3.3: Performances of the MLCBFM-E in terms of reduced final matrix size, computation time and relative error (%) in comparison with the MoM depending on the value of N_{IPWs}.

Table 3.3 allows us also to examine once again the impact of the N_{IPWs} value on the CPU time and memory resources needed to resolve the electromagnetic problem. Hence, the computational burden brought by a great value of N_{IPWs}, for the implementation of the MLCBFM procedure, prompts us to avoid to consider CBFM blocks of height $> 20\lambda_s$, for instance, corresponding to $r_0/\lambda > 10$ and hence, a necessary value of $N_{IPWs} > 2000$. Consequently, imposing a maximum limitation on the height of the blocks of the higher level will lead us to a situation with several 2^{nd} level (and up) blocks in the same tree trunk. That is why we aim, in the next paragraph, to apply the CBFM block extension (inside a tree) for different levels starting from the 2^{nd} one.

3.2.3 Generalization of the CBFM block Extension

Initially, the extension of the CBFM blocks concerned only the first level of the MLCBFM-E. Thereby, while computing the scattered field with the CBFM-E and ML-CBFM-E, we distinguished, a slight difference between the results obtained with these two methods. To overcome this limitation, the extension of the CBFM blocks is generalized and applied for the higher level since the blocks belong to the same scatterer.

Let us apply a two-level CBFM-E to the simple example of a tree trunk. The cylinder modeling the trunk is divided into 9 blocks for the first level and into 2 great blocks $(5;4)$ for the second level. Figure 3.6 illustrates the extension of the 3^{rd} CBFM block for the level 1, then the extension of the 1^{st} block of level 2.

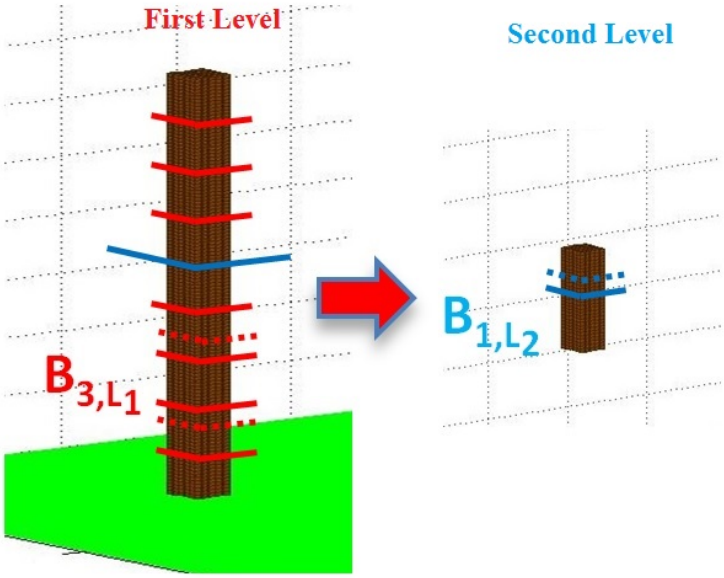


FIGURE 3.6: *A two-level block extension applied to a single tree trunk divided into* 9 *blocks for the* 1^{st} *level, and into* 2 *blocks for the* 2^{nd} *level.*

Unlike the extension of the CBFM blocks corresponding to the 1^{st} level, the size of the buffered region, starting from the 2^{nd} level, is not measured in number of floors along the vertical axis $\vec{z}$ (see paragraph 2.2.3). For a multilevel extension, for each level (l) $(l \geqslant 2)$ presenting a case of multiple blocks in the same scatterer, the size of the buffered region, for the block I, is calculated on the basis of the number of CBFs generated with the previous level $(l-1)$ as follows :

$$\begin{aligned} Ne_{I^{(l)},bottom} &= S_{i_{b,I}^{(l-1)}};\ \text{where}\ i_{b,I}^{(l-1)} = \sum_{N=1}^{I^{(l)}-1} N_{blocks,N}^{(l-1)} \\ Ne_{I^{(l)},upper} &= S_{i_{u,I}^{(l-1)}};\ \text{where}\ i_{u,I}^{(l-1)} = \sum_{N=1}^{I^{(l)}} N_{blocks,N}^{(l-1)} + 1 \end{aligned} \tag{3.5}$$

where $I^{(l)}$ is the number of the block to be extended associated to level (l), $S_{i_{b,I}^{(l-1)}}$ is the number of CBFs generated for the block number $i_{b,I}$ associated to level $(l-1)$ and $S_{i_{u,I}^{(l-1)}}$ is the number of CBFs generated for the block number $i_{u,I}$ associated to level $(l-1)$. $N_{blocks,N}^{(l-1)}$ is the number of blocks associated to level $(l-1)$ composing the block N associated to the level (l). For a sake of clarity, the blocks $i_{b,I}^{(l-1)}$ and $i_{u,I}^{(l-1)}$ corresponding to the block $I^{(l)}$ are illustrated in Figure 3.7.

In order to prove the relevance of the generalized block extension approach, we apply a 2-level extended CBFM-E (named here MLCBFM-E2) to a simple forest simulation scene of 9 cylinders modeling 9 tree trunks discretized into $Nb_c = 35080$ cells, and we compare the results obtained with those produced by a classical 2-level CBFM-E (MLCBFM-E1, where the extension concerns only the 1^{st} level).

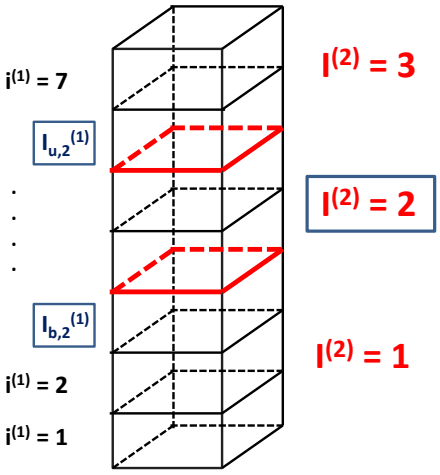


FIGURE 3.7: *Representation of the blocks $i_{b,I}^{(l-1)}$ and $i_{u,I}^{(l-1)}$ (level $(l-1)$) corresponding to the block $I^{(l)}$ (level (l))*

We display, in Figure 3.8, the electric field inside the 1^{st} tree trunk of hight $h = 4.5$ m divided into 15 blocks for the 1^{st} level, then, into 3 blocks for the 2^{nd} level.

We note that a single block extension at the beginning of the 1^{st} level, is not sufficient to avoid the oscillations due to the edge effect, and, hence, the generalized extension approach, namely the MLCBFM-E2, achieves a much better concordance with the mono-level CBFM-E. While representing the backscattered field arising from these two methods, we found that the oscillations, for this example, have not a significant effect on the magnitude and phase of the backscattered field both for the polarizations VV and HH. This is likely due to the small numerical size and the low complexity of the under consideration simulation scene.

In order to confirm it, we applied the two approaches to a numerically larger simulation scene of $Nb_c = 565920$ cells modeling nine trees with four branches each as shown in Figure 3.9 and we display the magnitude and phase of the backscattered fields for both polarizations VV and VH in Figure 3.10.

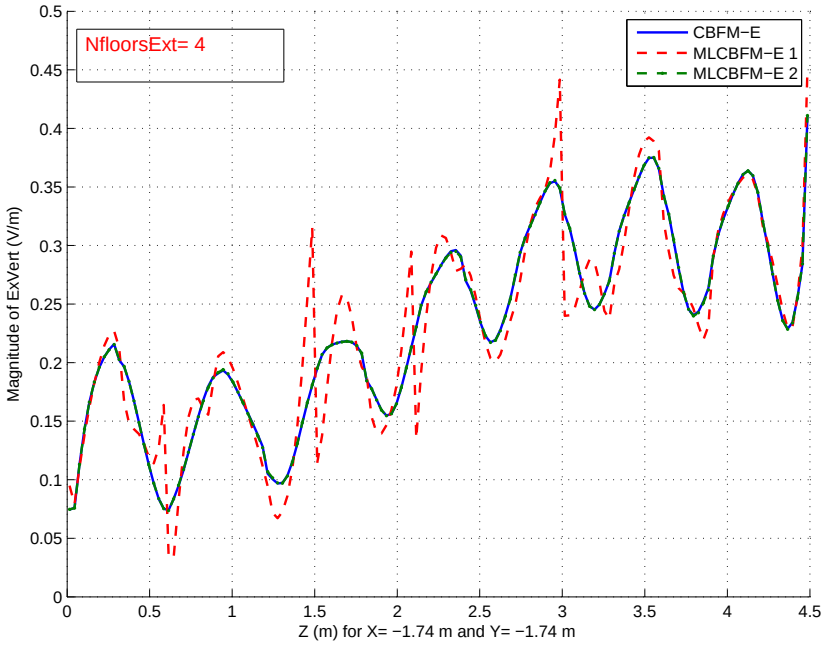


FIGURE 3.8: *Variations of the magnitude E_x^V of the electric field inside the cylinder for $x = y = -1.74$ m computed for ($\theta_i = 0°$; $\varphi_i = 40°$) with the classical MLCBFM-E (MLCBFM-1) and the generally Extended MLCBFM (MLCBFM-2).*

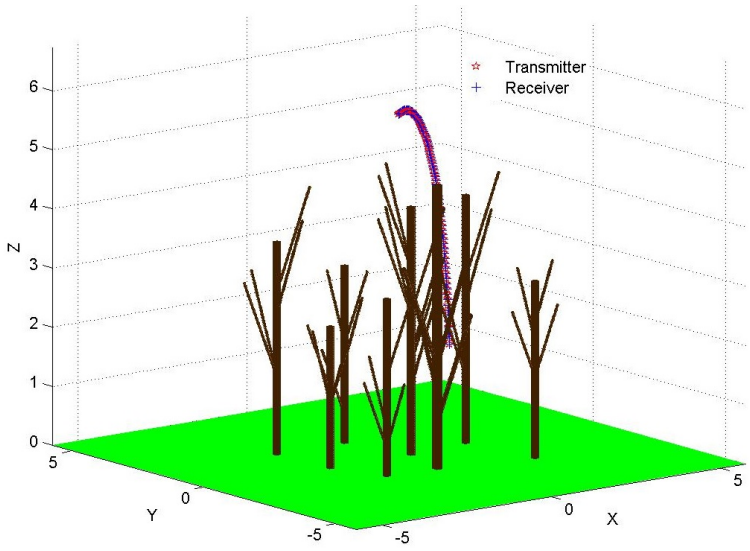
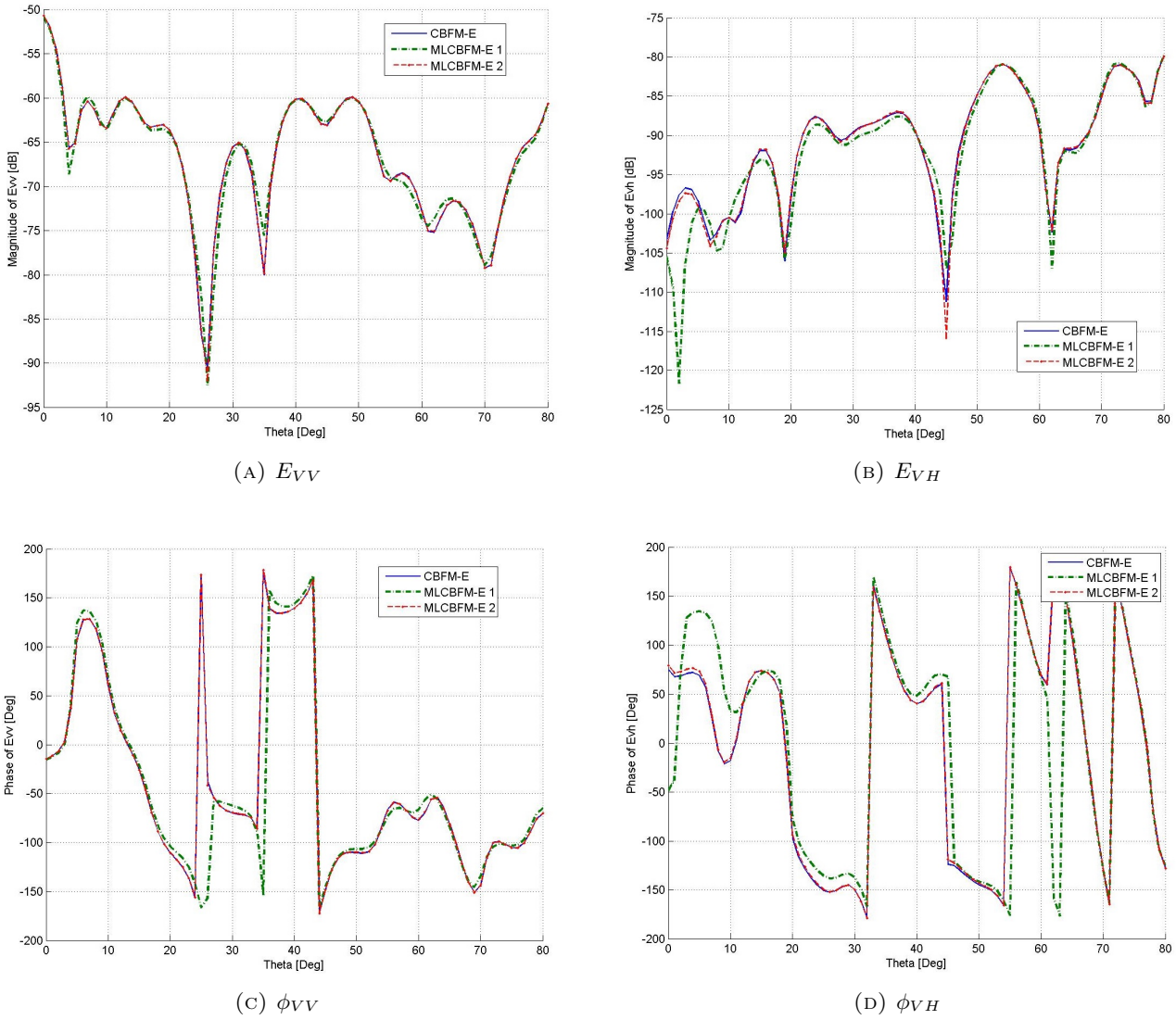


FIGURE 3.9: *A numerically large simulation scene comprised of 9 trees with 4 branches each and discretized into $Nb_c = 565920$.*

It is, thus, clear that we must apply, when needed, the extension of the CBFM blocks for the different levels of the multilevel scheme in order to guarantee, whatever the simulation scene is, the accuracy of the MLCBFM-E. Hence, in the rest of this report, the term MLCBFM-E refers to the generally extended multilevel scheme of the CBFM (noted MLCBFM-E2 in the current paragraph)



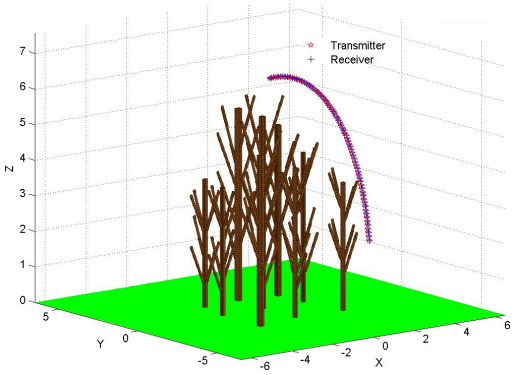
(A) E_{VV} (B) E_{VH}

(C) ϕ_{VV} (D) ϕ_{VH}

FIGURE 3.10: *Variations of the magnitude and phase of the backscattered field (polarization VV and VH) computed for $\theta_{i,s} = [0°; 1°; 80°]$ and $\varphi_{i,s} = 40°$ by the classical MLCBFM-E (MLCBFM-1) and the generally Extended MLCBFM (MLCBFM-2).*

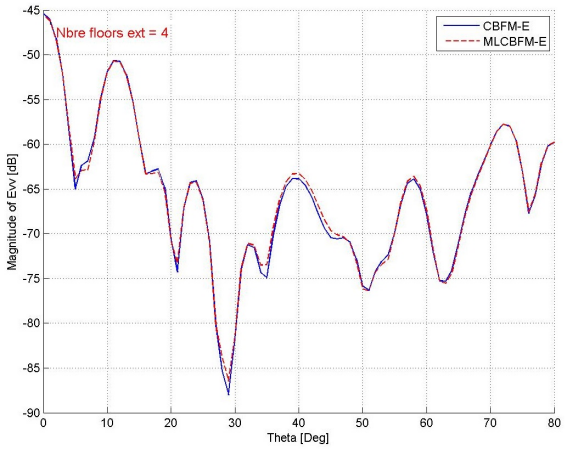
3.3 Numerical Results

We conclude this chapter by comparing the performances of the MLCBFM-E to those achieved by a mono-level CBFM-E, in terms of CPU time and compression rate, when applied to an electrically large forest simulation scenes. We start with a medium large example representing 9 trees with 8 branches each, modelled by 81 cylinders and discretized into $Nb_c = 162000$ cells. The conditions of application of the both mono-level and multilevel CBFM-E and the forest simulation scene are shown in table 3.4. Then, we display, in Figure 3.11, the magnitude and phase of the backscattered electric fields derived by these two methods with $\theta_{i,s} = [0°; 1°; 80°]$ and $\varphi_{i,s} = 15°$ for both polarizations, VV and HH. The performances of the CBFM-E in terms of computing time and compression rate are compared to those achived by the MLCBFM-E in Table 3.5.

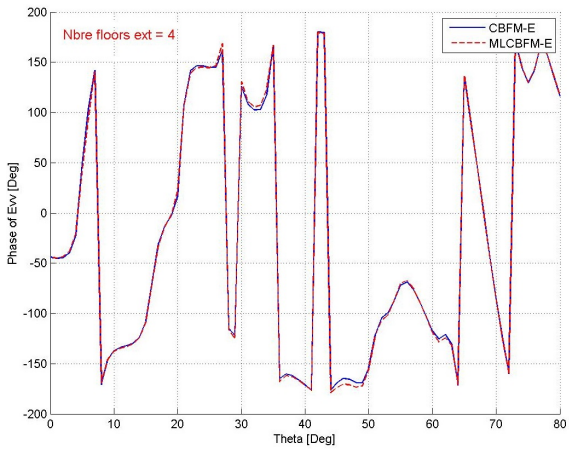


Frequency	f = 300 MHz
Scatterers permittivity ϵ_s	$10.2 + 1.2$ j
Ground permittivity ϵ_g	$5 + 2.1$ j
λ_s scatterer	0.316 m
Cell size	$T_c = 0.02$ m
Number of Cells	$Nb_c = 162000$
Number of blocks	$N_B = 484$
Block height	$h_B = 12T_c \approx \lambda_s$
N_{IPW}	380 ($d\theta = d\varphi = 20^\circ$)
Number of levels (MLCBFM-E)	3
Block repartition (MLCBFM-E)	$[484; 113; 32]$

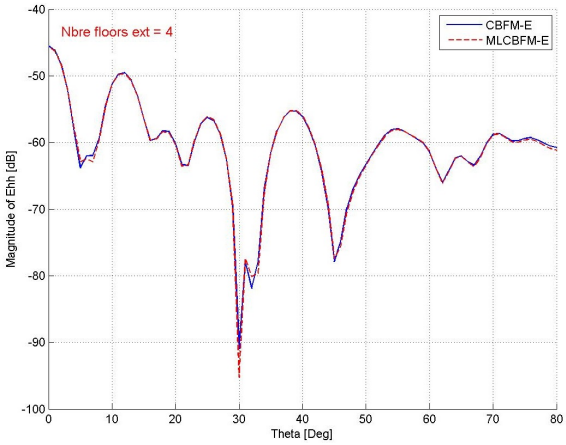
TABLE 3.4: With a 3-Level CBFM-E, the forest simulation scene is divided into 484 blocks for the first level, 113 blocks for the second, and 32 blocks for the third and last level.



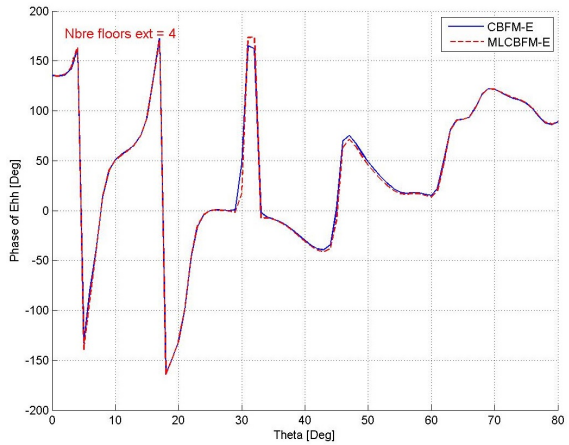
(A) E_{VV}



(B) ϕ_{VV}



(C) E_{HH}



(D) ϕ_{HH}

FIGURE 3.11: Variations of the magnitude and phase of the scattered field (polarization VV and HH) computed for $\theta_{i,s} = [0^\circ; 1^\circ; 80^\circ]$ and $\varphi_{i,s} = 15^\circ$ by the mono-level CBFM-E and a 3-level CBFM-E.

TABLE 3.5: Performances of the mono-level and the 3-level CBFM-E in terms of CPU time and compression rate. $(Z^c)^{-1}$ (sec) refers to the CPU time spent to solve the final system of linear equations associated to Z^c.

	Size of final Z^c	CR	ICR (%)	$(Z^c)^{-1}$ (sec)	Total CPU (sec)
CBFM-E	17152	28.3	3.53	6330	11460
MLCBFM-E	3879	125.3	0.8	348	5482

Figure 3.11 shows the MLCBFM-E yield accurate results comparing to those derived by the mono-level CBFM-E. At the same time, the MLCBFM-E enables us to achieve a higher compression rate, and to reduce significantly the size of the final matrix Z^c. This has a direct impact on the total CPU time, since the smaller is the size of Z^c , the faster is the resolution of the associated system of linear equations. Indeed, as can be seen in Table 3.6, the second and third level make a substantive contribution to the compression rate without adversely impacting on the CPU time, since these two levels process small CBFM blocks.

TABLE 3.6: Details of 3-level CBFM-E applied to a forest simulation scene of $Nb_c = 162000$ cells. We display, in red, the CPU time needed to generate the CBFs and to construct Z^c for each level.

	size of $Z_{level}^{initial}$	N_{Blocks}	$Nbc_{Block,max}$	Size of Z_{level}^c	CBFs (sec)	Z_{level}^c (sec)
$1^{st}level$	486000	484	1863	17152	945	4174
$2^{nd}level$	17152	113	246	6134	5	3
$3^{rd}level$	6134	32	456	3879	5	2

At this stage, we note that the multilevel scheme helps to reduce significantly the CPU time but it does not enable us to run numerically larger simulation scenes, since we will always need to store the first level reduced matrix (Z_1^c) in our shared-memory workstation of 42 GB of RAM. We also see that the gain in CPU time achieved by the multilevel CBFM-E while resolving the final reduced matrix was partly mitigated by the great CPU time spent to generate the CBFs and the reduced matrix for the first level. Hence, a further improvement of the performances of the MLCBFM-E will pass by the enhancement of the different steps of the mono-level CBFM-E procedure. This will be done in the next chapter.

To complete the discussion of the multilevel CBFM-E and to confirm the observations made in the first example, we carry out a new simulation for a larger forest scene composed of 144 modeling 16 trees (see Figure 3.12). As for the previous example, we show in Figure 3.13 a comparison between the backscattered fields (polarization HH) obtained with the MLCBFM-E and those derived by the mono-level CBFM-E, and we summarize in Table 3.7 the performances of the two methods in terms of computing time and compression rate.

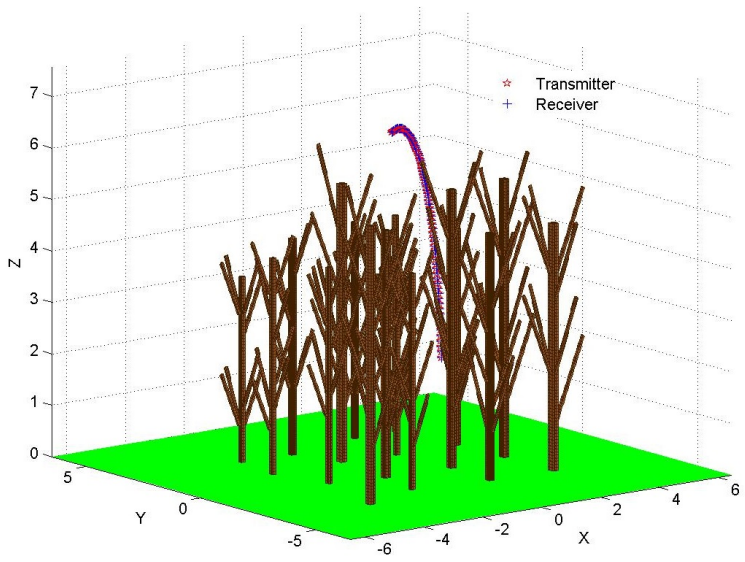
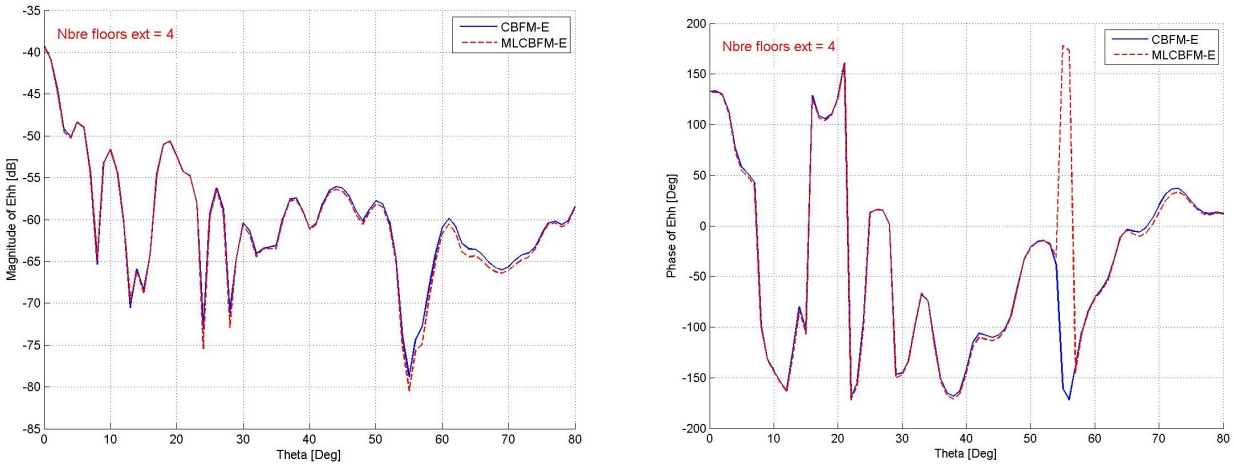


FIGURE 3.12: *A numerically large simulation scene comprised of* 16 *trees with* 8 *branches each and discretized into* $Nb_c = 314500$. *and divided initially to* 683 *blocks of height* $h_B = 1.3\lambda$.



(A) E_{HH} (B) ϕ_{HH}

FIGURE 3.13: Variations of the magnitude and phase of the scattered field (polarization HH) computed for $\theta_{i,s} = [0°; 1°; 80°]$ and $\varphi_{i,s} = 40°$ by the mono-level CBFM-E and a 2-level CBFM-E.

TABLE 3.7: Performances of the mono-level and the 2-level CBFM-E in terms of CPU time and compression rate. $(Z^c)^{-1}$ (sec) refers to the CPU time spent to solve the final system of linear equations associated to Z^c.

	Size of final Z^c	CR	ICR (%)	$(Z^c)^{-1}$ (sec)	Total CPU (sec)
CBFM-E	25744	36.65	2.72	10337	28402
MLCBFM-E	7596	124.2	0.8	626	20603

The comparison between the performances of the CBFM-E and MLCBFM-E, summarized in Table 3.7, confirms once again the relevancy of the implementation of a multilevel scheme of the CBFM-E,

since it enables us to achieve a compression rate which is almost 3 times higher than that obtained with the mono-level scheme. Thanks to the generalized multilevel extension, we also maintain the same level of accuracy going from the mono-level scheme to the multi-level one.

3.4 Conclusion

A generally extended multilevel scheme of the CBFM (ML-CBFM-E) has been developed in order to improve the performances of the conventional version especially in terms of memory resources. The concept of the ML-CBFM is based on an iterative application of the conventional CBFM process which results on a higher numerical compression of the electromagnetic problem compared to the one obtained with a mono-level CBFM-E.

As for the mono-level CBFM-E, ones should set correctly the number of incident plane waves N_{IPWs} depending on the height of the blocks composing the simulation scene at each level. Otherwise, the MLCBFM-E yields inaccurate results.

To ensure the accuracy of the MLCBFM-E, whatever the simulation scene is, the extension of the CBFM blocks is generalized and applied for the higher level since the blocks belong to the same scatterer. The ML-CBFM results are shown to be in good agreement with those obtained via the CBFM-E and the conventional MoM, which confirms the fact that the proposed method is not only computationally efficient, but is accurate as well.

Chapter 4

Efficiency enhancement techniques for the CBFM with Forest Scattering Modeling

Compared to a conventional Method of Moments, the Characteristic Basis Function Method (CBFM) has shown excellent performances both in terms of CPU time and required memory storage, while maintaining a satisfactory level of accuracy. We can now skip the conventional MoM as reference, and focus on the enhancement of the application of the CBFM in order to cope with the challenging computational burden posed by the computation of larger forest simulation scenes for higher frequencies.

Hence, in this chapter, we detail the various approaches adopted in order to reduce as much as possible the CPU time and the required storage, such as the application of the CBFM to a non-uniform mesh adapted to the dielectric properties of the forest simulation scene, and the hybridization of the CBFM with an other efficient iterative method namely the ACA (Adaptive Cross Approximation Algorithm). The main goal of these enhancement techniques is to produce an efficient CBFM code, adapted to the problem of forest scattering modeling and comparable in terms of computational performances to the iterative methods (ACA and MLFMM) while maintaining, as a direct method, the advantage of numerical stability.

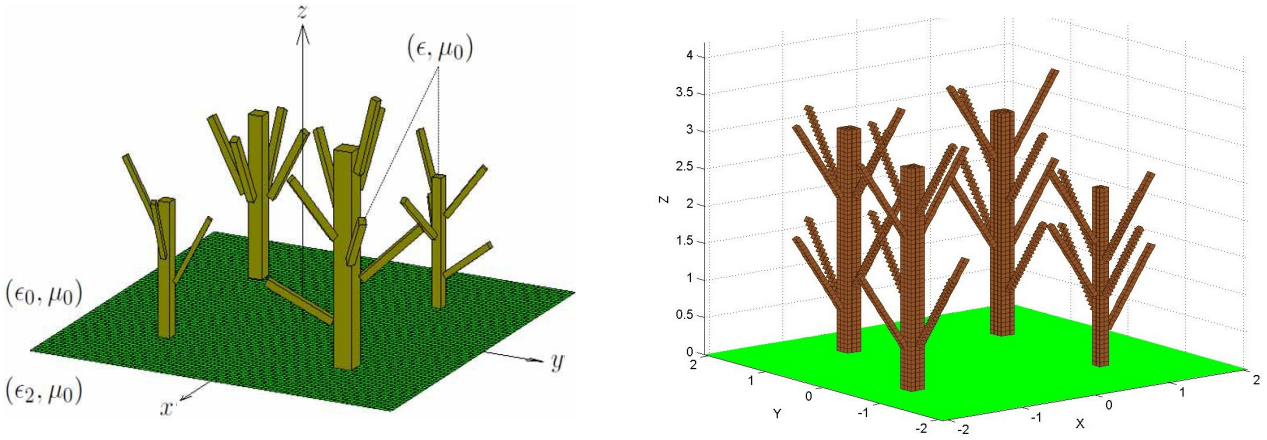
4.1 Application of the CBFM to an Adaptive Meshing

In this paragraph, we investigate the suitability and applicability of the CBFM process to a non-uniform mesh adapted to our 3D full-wave forest scattering model. In fact, this non-uniform mesh enables us to take advantage of the heterogeneous properties of a natural forest medium, in order to reduce the corresponding computational burden and, hence to reduce significantly the computation time and the memory resources needed to calculate the scattered fields.

Let us, first of all, explain and justify the need to implement an adaptive mesh for the 3D forest simulation scene and show the legitimacy of this approach and, then demonstrate its relevancy to the increase of the efficiency of the CBFM procedure in terms of CPU time and required storage.

4.1.1 Using Adaptive Meshing to describe a forest simulation scene

As detailed previously, in common with our previous works [24, 46], the tree trunks and primary branches, making up the under consideration forest simulation scene, are modeled as dielectric cylinders of square cross-section as shown in Figure 4.1a. To solve the volume integral equation and compute the electric field inside the scatterers, these tree-trunks and branches are divided into elementary cubic cells, of cell size T_c small enough so that the internal field is nearly uniform in each cell (see Figure 4.1b).



(A) Geometry of a forest

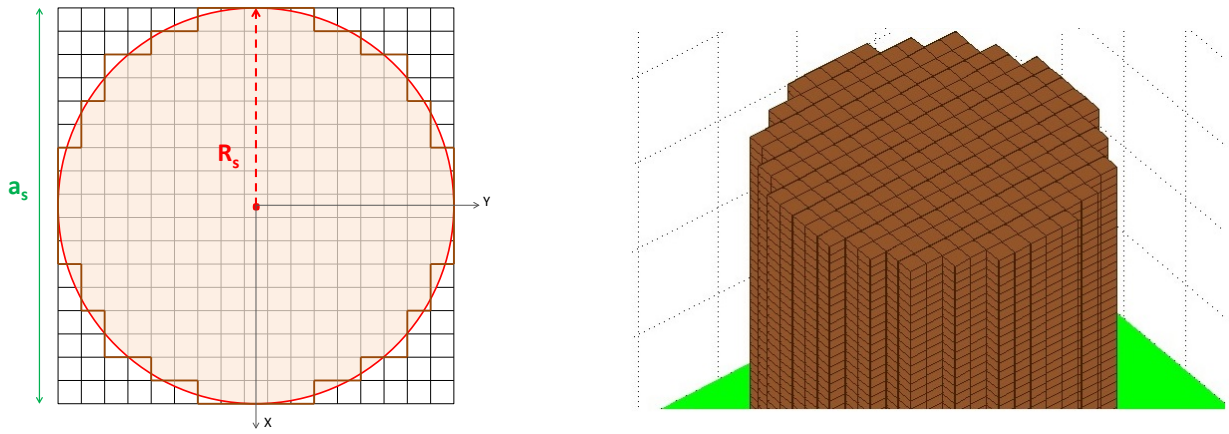
(B) Discretization into elementary cubic cells

FIGURE 4.1: Example of forest geometry and of discretization into elementary cubic cells. $\epsilon_1 = \epsilon_0$, ϵ_2 and ϵ are respectively the free space, ground and wood permittivities, and μ_0 is the free space permeability

The previous work reported in [24] demonstrates, by referring to [47], that the scattered fields from square and circular cross sections, having the same cross sectional areas (noted as c_s), are appropriately equivalent, provided that the condition $c_s < (\frac{\lambda_s}{5})^2$ is satisfied. Thus, this approximation loses its effectiveness for higher frequencies and larger tree trunks.

Therefore, we need to overcome this limitation in terms of frequency and tree trunk size, before addressing the issue of non-uniform mesh. For this purpose, we have modified the initial discretization code, in order to model the tree trunks and branches by using cylinders of circular cross section, while maintaining the simple cubic shape of the previously used elementary cells. In fact, given the radius R_s of the circular cross section scatterer, modeling the tree trunk, we compute the dimensions of the equivalent square cross section cylinder such as $\pi \times {R_s}^2 = {a_s}^2$ where a_s is the side of the square cross section. Next, we discretize conventionally the equivalent cylinder (of square cross section) into

elementary cubic cells of size T_c. Finally, we remove the cells located outside the contour defined by the radius R_s. Hence, the cell i is removed if the distance $d_i = \sqrt{{x_i}^2 + {y_i}^2} > R_s$. This simple, though effective approach is shown in Figure 4.2.



(A) Discretization approach

(B) Discretization of cylinder of circular cross section

FIGURE 4.2: Example of discretization of a cylinder of circular cross section modeling a tree trunk

Now, in order to confirm the accuracy of the observation made in [24] and [47], on the square and circular cross sections, and hence to prove the relevancy of the approach illustrated in Figure 4.2, we carry out two simulations with two different values of cross sectional areas c_s. Then, we compare the results obtained with a tree trunk of a square cross section of area c_s to those obtained with a tree trunk of a circular cross section having the same area c_s. The first simulation is performed with a tree trunk of cross-section area $c_{s,1} = 0.0036\ m^2$ and dielectric permittivity $\epsilon_s = 6.2+$ j 0.2 for a frequency $f = 300$ MHz. Therefore, the condition on c_s, cited above, is satisfied for this first simulation. For the second simulation, we increase c_s ($c_{s,2} = 0.0144\ m^2$) such as this condition is no longer complied with. The simulations conditions are listed in Table 4.1.

TABLE 4.1: Conditions of the application of the CBFM-E to 2 cylinders of square and circular cross section for two different values of cross section area c_s.

	f (MHz)	ϵ_s	$(\frac{\lambda_s}{5})^2$	c_s (m^2)	a_s (m)	R_s (m)
Simulation 1	300	6.2+ j 0.2	**0.00643**	0.0036	0.06	0.035
Simulation 2				0.0576	0.24	0.27

Figures 4.4 and 4.5 illustrates the magnitude of the backscattered field, computed for the square and circular cross section cylinder with the two different values of c_s. The backscattered field is presented, as shown in Figure 4.3 for $\theta_{i,s} = [10^\circ; 2^\circ; 80^\circ]$ and $\varphi_{i,s} = [0^\circ; 2^\circ; 360^\circ]$ for the two polarizations VV and HH.

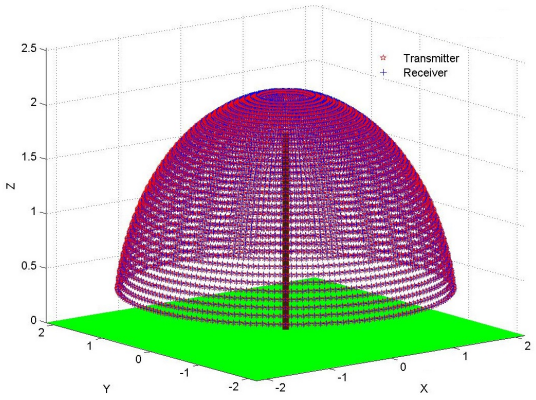
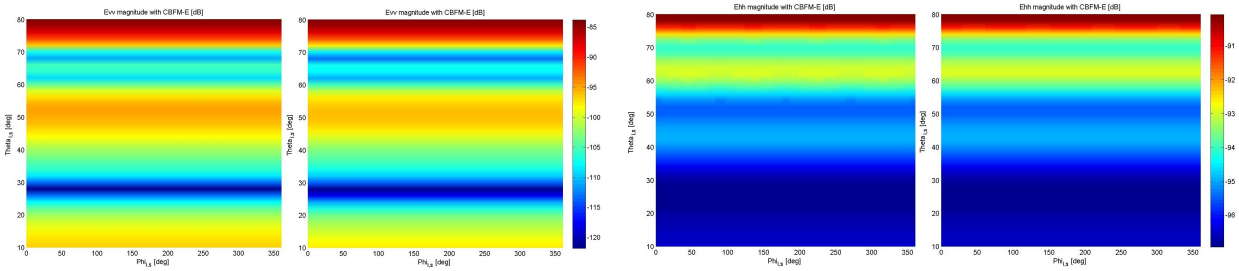
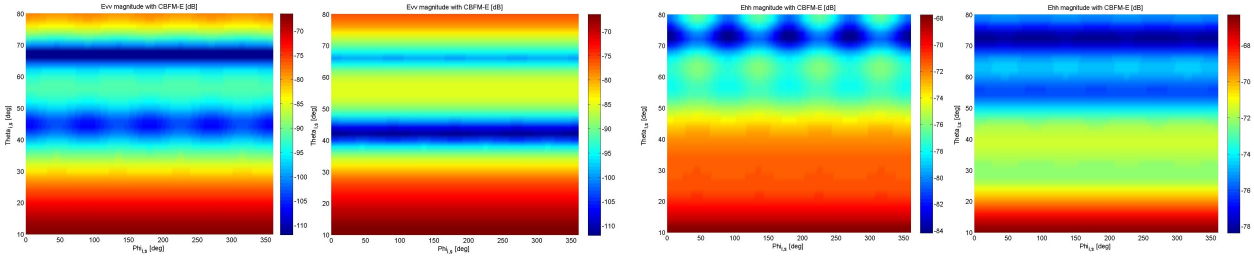


FIGURE 4.3: *The mono-static configuration used to compute the total backscattered electric field :* $\theta_{i,s} = [10°; 2°; 80°]$ *and* $\varphi_{i,s} = [0°; 2°; 360°]$



(A) E_{VV} (B) E_{HH}

FIGURE 4.4: Simulation 1 ($c_s = 0.0036\ m^2$) : Variations of the magnitude of the backscattered field computed with the CBFM-E for $\theta_{i,s} = [10°; 2°; 80°]$ and $\varphi_{i,s} = [0°; 2°; 360°]$. For each polarization (VV or HH), the figure on the left corresponds to the square cross section cylinder, and the figure on the right shows the results obtained with the circular cross section cylinder.



(A) E_{VV} (B) E_{HH}

FIGURE 4.5: Simulation 2 ($c_s = 0.0576\ m^2$) : Variations of the magnitude of the backscattered field computed with the CBFM-E for $\theta_{i,s} = [10°; 2°; 80°]$ and $\varphi_{i,s} = [0°; 2°; 360°]$. For each polarization (VV or HH), the figure on the left corresponds to the square cross section cylinder, and the figure on the right shows the results obtained with the circular cross section cylinder.

The results shown in Figure 4.4 and 4.5, attest to the fact that the approximation of a tree trunk by a cylinder of cubic cross section is only valid under a certain value of cross section area c_s and a certain frequency. Therefore, the adaptation of the mesh to the circular shape of a natural tree trunk,

enables us to keep a satisfactory accuracy while applying the CBFM to a more faithful representation of the forest.

Next, we focus on adapting the mesh to the dielectric properties of the cylinders modeling the tree trunks and branches. In fact, the implementation of the CBFM on a non-uniform mesh, depending on the dielectric permittivities of the trees, enables us to exploit the heterogeneity of the natural forest medium, in order to reduce the numerical size of the simulation scene. Therefore, we can handle larger forest patches for higher frequencies, while maintaining a satisfactory level of accuracy since we remain faithful to the dielectric reality of the forest.

In this paragraph, we explain our motivations to implement the CBFM on a non-uniform mesh in regards to the heterogeneity of a natural forest medium in terms of dielectric constant :

$$\epsilon_{r,s} = \epsilon'_{r,s} + \ j\ \epsilon''_{r,s} = \frac{\epsilon'_s}{\epsilon_0} + \ j\ \frac{\epsilon''_s}{\epsilon_0} \tag{4.1}$$

where $\epsilon'_{r,s} = Re(\epsilon_{r,s})$ is the real component, and $\epsilon''_{r,s} = Im(\epsilon_{r,s})$ is the imaginary component. The dissipation factor, or loss tangent, is defined as the ratio

$$\tan\Delta = \frac{\epsilon''_{r,s}}{\epsilon'_{r,s}} \tag{4.2}$$

Hence, we examine the dielectric properties of tree trunks from two perspectives. First, we look at the variations of the dielectric constant amongst the different trees making up the forest. And second, we study the behaviour of the electromagnetic wave inside the same tree, depending on the imaginary part of its dielectric constant, in order to demonstrate the relevancy of the implementation of a non-uniform mesh inside a single tree.

4.1.1.1 Non-uniform mesh among different trees

The electrical properties of wood have been the subject of numerous previous research works [48–51]. These studies affirm that the trees dielectric properties are strongly correlated with wood core moisture status. It was, also, proven that it exits a within-tree variability of ϵ_s in relation with wood chemistry and wood flux density depending on the species of the tree [48, 52, 53].

For instance, Figure 4.6 illustrates the variations of the complex relative dielectric constant as a function of depth into the stem for two types of trees (A and B). According to [48], Tree A, the taller dominant tree, was 19.7 m tall with a diameter at breast height (DBH) of 24.5 cm and tree B was a typical sub-dominant tree, 17.8 m tall with DBH of 23.2 cm. Furthermore, Figure 4.7 [53] and Table 4.2 [52] enable us to note a significant variability of the dielectric constant, particularly $\epsilon'_{r,s}$, depending on the species of the tree.

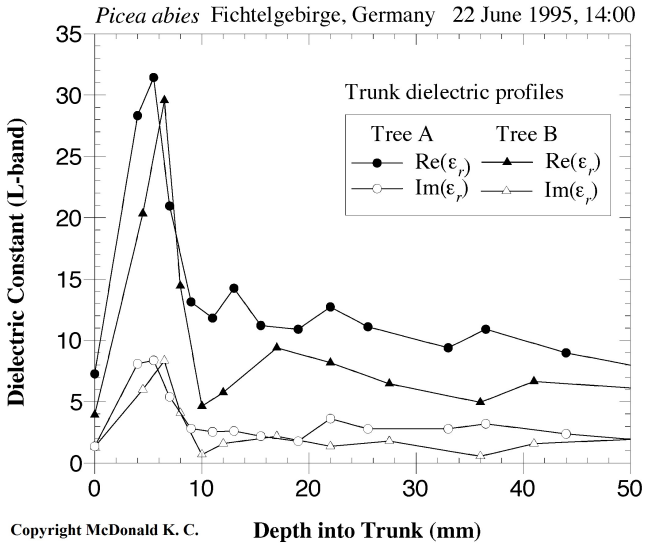
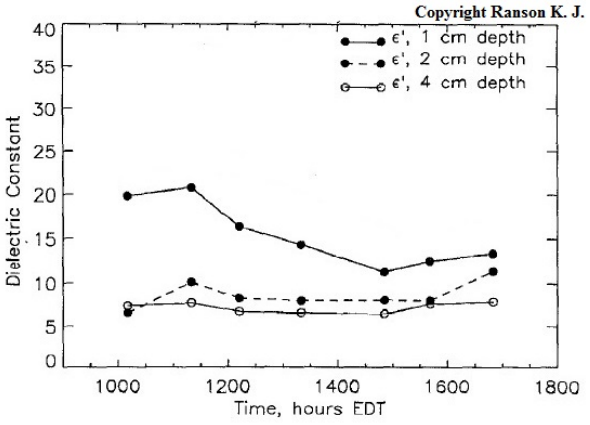
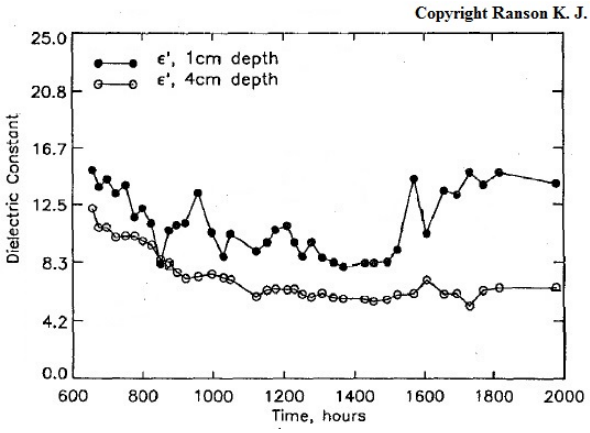


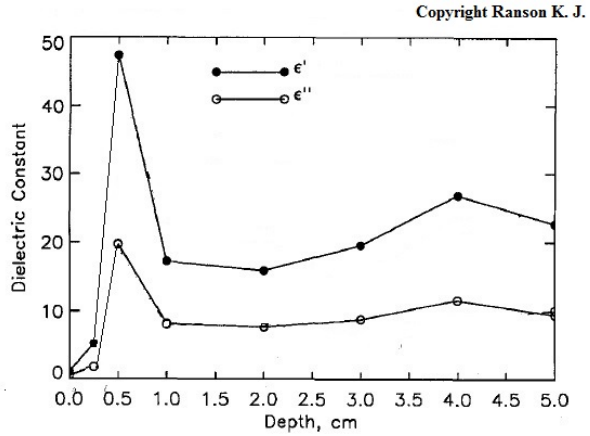
FIGURE 4.6: Complex relative dielectric constant as a function of depth into the stem for Tree A and Tree B. Measurements were recorded with a L band (1.25 GHz). Depth is measured from the outer side of the bark (0 mm) and progresses toward the stem center.



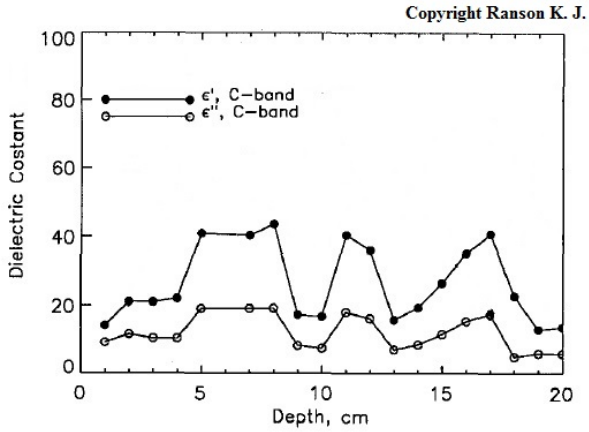
(A) Spruce tree



(B) Siberian Pine



(C) Siberian Fir



(D) White Pine

FIGURE 4.7: C-band dielectric measurements for different species of tree : Spruce, Siberian Pine, Siberian fir and White Pine

TABLE 4.2: Measured and calculated data on permittivity of trees at P and VHF bands [52, 54].

Autumn	Humidity, $m_g\%$	ϵ'_r calculated	measured ϵ'_r
Pine	67.80	15.5	23.58
Spruce	111.11	30.95	23.19
Larch	71.57	16.61	18.3
Winter			
Pine	33.50	6.8	16.45
Spruce	84.10	20.6	20.12
Larch	23.5	4.56	25.61

On the other hand, we know that the real part of the dielectric permittivity $\epsilon'_{r,s}$ is directly related to the cell size used to discretize the corresponding scatterer. In fact, in order to ensure the accuracy of the MoM solution, T_c must satisfy the following condition in relation with :

$$\textbf{Condition 1 :}\ T_c \leqslant \frac{\lambda_s}{D_\lambda} \text{ where } \lambda_s = \frac{\lambda}{\sqrt{\epsilon'_{r,s}}} \tag{4.3}$$

and D_λ is a constant chosen almost between 10 and 20, in order to ensure the accuracy of the MoM solution. Therefore, it is interesting to discretize the 3D forest simulation scene, by using a non-uniform mesh tailored on the basis of dielectric properties of the scatterers. This approach enable us to avoid over-discretization of the electromagnetic problem and thereby to reduce the burden on the associated computing time and memory resources.

For a sake of clarity, let us consider the example of a simple forest simulation scene composed of four trees trunks with four different values of $\epsilon'_{r,s}$, comprised between 4.56 and 20.6 as shown in table 4.3. These trees are illuminated by an electromagnetic plane wave of frequency $f = 400$ MHz. The adaptive discretization , depending on the real part of the dielectric constant of each tree, enables us to reduce significantly the numerical size of this forest simulation scene.

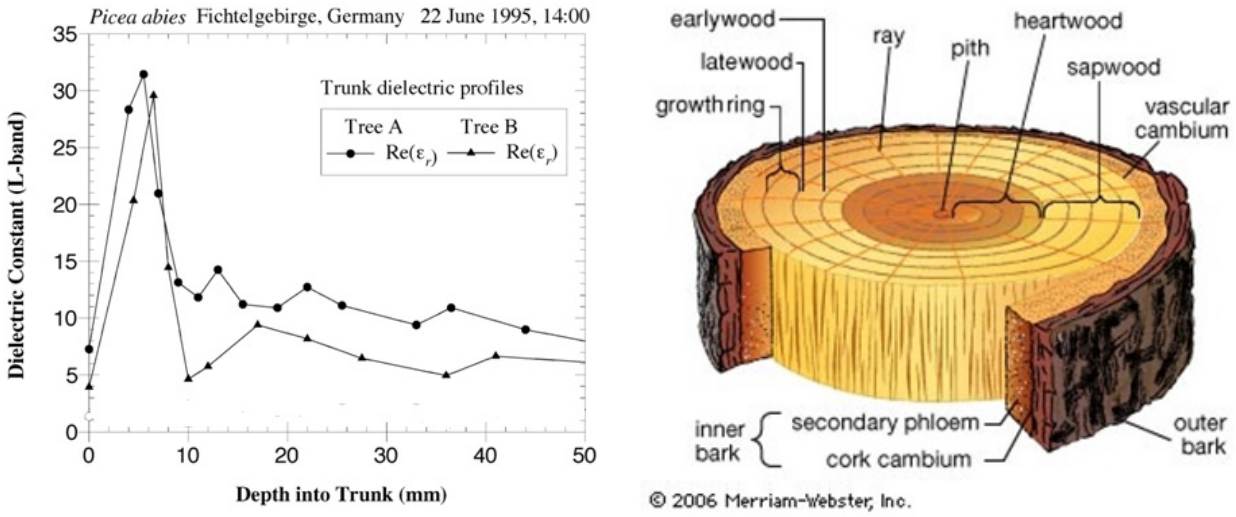
TABLE 4.3: The significant reduction of the electrical size of the forest simulation scene thanks to the adaptive meshing amongst the trees. D_λ is taken equal to 12 for this example.

	$\epsilon'_{r,s}$	$\frac{\lambda_s}{12}$	dimensions (m^3)	$T_{c,uniform}$ (m)	$T_{c,non-uniform}$ (m)
Tree 1	4.56	0.029	$0.120 \times 0.120 \times 2.400$	0.010	0.020
Tree 2	12.95	0.0173	$0.150 \times 0.150 \times 3.600$	0.010	0.015
Tree 3	17.80	0.014	$0.120 \times 0.120 \times 2.700$	0.010	0.010
Tree 4	20.60	0.01375	$0.180 \times 0.180 \times 4.500$	0.010	0.010
Total Number of cells Nb_c				300240	213000

4.1.1.2 Non-uniform mesh inside the same cylinder

Along the same lines, the implementation of a non-uniform mesh inside a single cylinder aims also to avoid an over-meshing of each tree and, hence, to reduce the initial electrical size of the forest simulation scene. This approach was motivated by two main reasons. The first is in relation with the composition of a natural tree trunk, which has a direct influence on $\epsilon'_{r,s}$ of the cylinder modeling this tree, and the second concerns the behaviour of an electromagnetic wave inside a material depending on the imaginary component of its dielectric constant ($\epsilon''_{r,s}$).

In fact, we observed the variations of the real part of the dielectric permittivity depending on the depth into the tree trunk (see Figure 4.8). Therefore, we noted the existence of a region of high peak dielectric constant value around 5 or 6 mm, which is explained by the fact that the inner bark is a living tissue. Hence, the peak observed corresponds to the well-hydrated living cells of the phloem and cambium tissues, the two main constituents of the inner bark [48].



(A) Variations of $\epsilon'_{r,s}$ inside a tree trunk

(B) A tree trunk cross section

FIGURE 4.8: Variations of the real component of the dielectric constant along with the depth into the tree trunk.

Consequently, it is interesting to take into consideration the variation of $\epsilon'_{r,s}$ with the depth inside the tree trunk while discretizing it into cells of size $T_c \leqslant \dfrac{\lambda_s}{D_\lambda}$ (see equation **??**). That would enable us to ensure the accuracy of the CBFM with a representation of the forest closer to the reality, without finely meshing the entire tree trunk. Figure 4.11 illustrates an example of adaptive mesh implemented for the tree B whose dielectric constant is presented in of Figure 4.8a.

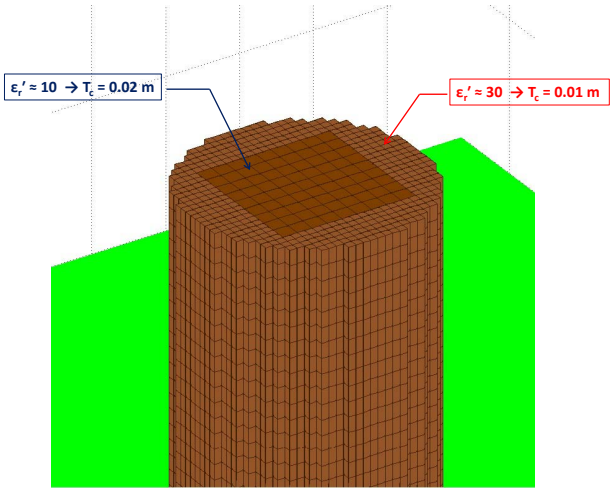


FIGURE 4.9: Example of non-uniform meshing depending on the value of $\epsilon_r^{'}$ inside the tree trunk. This example corresponds to a frequency $f = 300$ MHz and a constant $D_\lambda = 14$

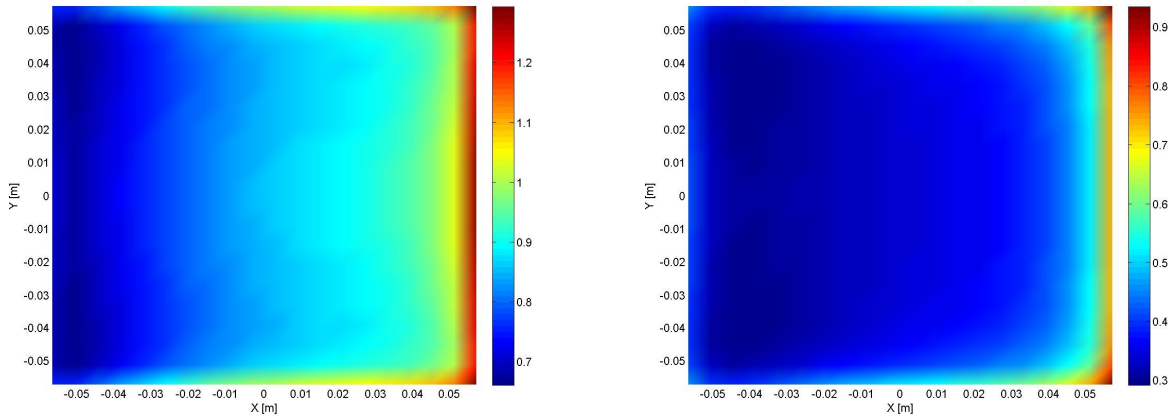
The other key motivation was in relation with the imaginary part of the dielectric constant $\epsilon_r^{''}$, which has also a determining influence on the cell size used to discretize the tree trunk. In fact, $\epsilon_r^{''}$ is closely linked to the skin effect that may be encountered on the outer part of the dielectric cylinder modeling a tree trunk. Let us recall that the skin depth δ is defined as the depth at which the electromagnetic field in a conducting material has decreased to $\frac{1}{e}$ of its initial value. Hence δ determines approximately the width of the area where the electric field is concentrated, when the depth is measured from the outer side of the scatterer. The skin depth δ is given by the expression below [55, 56] :

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}} = \frac{1}{\sqrt{\sigma\mu\pi f}} \;\Rightarrow\; \delta = \frac{67}{f(\text{ MHz})\sqrt{\epsilon_r^{''}}}\ (m) \tag{4.4}$$

where ω is the angular frequency, μ the permeability and σ the conductivity of the material, directly linked to the imaginary part $\epsilon_r^{''}$ of the dielectric constant. Hence, according to 4.4, the size of the skin depth region is inversely proportional to $\epsilon_r^{''}$. Figure 4.10 illustrates the variation of the electromagnetic field inside a dielectric cylinder of square-cross section modeling a tree trunk, computed with the CBFM-E for an incident plane wave with $\theta_i = 40^\circ$ and $\varphi_i = 0^\circ$. The results shown in Figure 4.10 confirm that the higher the value of $\epsilon_r^{''}$ is, the thinner the skin depth region is. On the other hand, it is known that the cell size, used to descritize the skin depth region, must be equal or less than $\frac{\delta}{5}$ if we aim to ensure the accuracy of the MoM solution :

$$\textbf{Condition 2 :}\ T_c \leqslant \frac{\delta}{D_\delta} \text{ where } \delta = \frac{67}{f(\text{ MHz})\sqrt{\epsilon_r^{''}}}\ (m) \text{ and } D_\delta \approx 5 \tag{4.5}$$

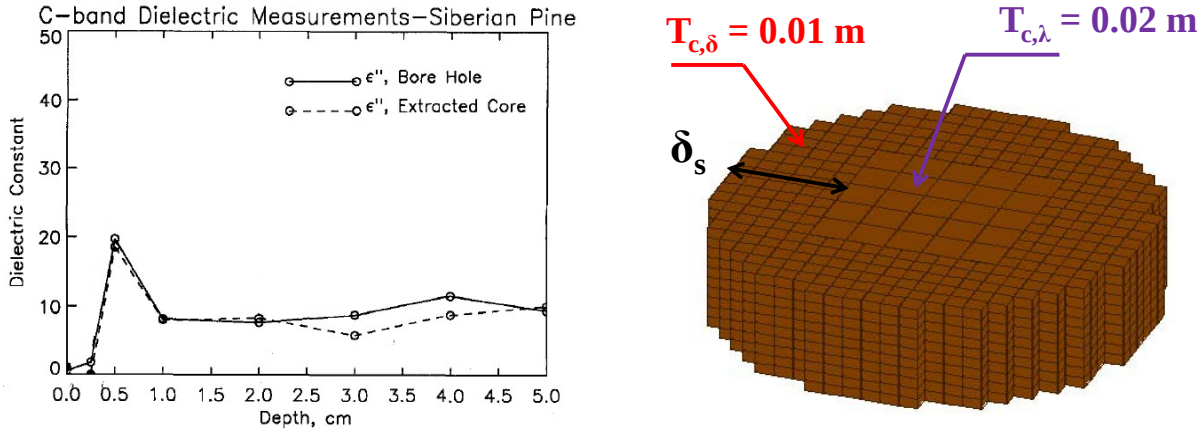
Hence, the idea was to take into account the rapid variations of the electric field inside the skin depth zone, while keeping the initial cell size T_c equal or less than $\frac{\lambda_s}{D_\lambda}$, with respect to condition 1 (see equation 4.3), for the rest of the scatterer.



(A) $\epsilon_r'' = 0.01$ (B) $\epsilon_r'' = 6$

FIGURE 4.10: Variation of the electric field inside a cylinder of square cross section computed by the CBFM-E, with 2 different values ϵ_r'', with a frequency $f = 300$ MHz, and for an incident plane wave defined by $\theta_i = 40°$ and $\varphi_i = 0°$.

This approach, considering non-uniform mesh inside the scatterer, as shown in Figure 4.11, is particularly interesting with high values of ϵ_r'' and with large tree trunks. It enables us a significant reduction of the needed memory resources even before starting the CBFM procedure. It remains to be seen whether the CBFM is suitable and insensitive to this non-uniform mesh, which will be discussed in the next paragraph.



(A) Example of variations of ϵ_r'' (B) Corresponding non-uniform mesh

FIGURE 4.11: Example of non-uniform mesh inside a tree trunk in order to take into account the variations of the electric field in the skin effect region. With $\epsilon_r'' = 20$, the skin depth for this scatterer, δ_s, is equal to 0.05 m. So the cell size in the δ_s zone, namely $T_{c,\delta}$, must be equal or less than 0.01 m.

At the end of this section dealing with the impact of the dielectric properties of the forest on the meshing approach, it is worthwhile mentioning, that we have not yet a clear "picture" on the dielectric constant in a forest medium at VHF and UHF frequency band [52]. In that sense, we have initiated

recently a cooperation with the LGEP laboratory (SUPELEC, Paris, France) in order to get, according to their research work, reliable measurements of the dielectric properties of the forest. In the meantime, we have adopted a wide range of values going up to $\epsilon_{r,s} = 30 + \mathrm{j}\ 20$. After all, the most important goal of this section is to ensure a sufficient meshing flexibility depending on the dielectric properties of a natural forest medium, in order to enable the users of the CBFM code to simulate larger forest simulation scenes with higher frequencies without incurring any extra costs both in terms of CPU time and memory.

4.1.2 Applicability of the CBFM to a non-uniform mesh

To carry out a comparison between the performances of the CBFM-E achieved by using a uniform and a non-uniform mesh, we begin this section by applying the CBFM-E to a simple example, namely scattering from two tree trunks modeled by two cylinders of circular cross section, whose dimensions are $[R_{s,1} = 0.24$ m ; $h_{s,1} = 3.3$ m] and [$R_{s,2} = 0.18$ m and $h_{s,2} = 2.1$ m], where R_s and h_s are, respectively, the radius and the height of each scatterer . The trees are set above ground, whose complex relative permittivity is $\epsilon_g = 5 + 3.6\mathrm{j}$. The cylinders have a complex relative permittivity $\epsilon_{s,1} = 7.2 + 0.04\mathrm{j}$ and $\epsilon_{s,2} = 24.6 + 0.06\mathrm{j}$. They are illuminated by an incident plane wave, at a frequency f=300 MHz. To compute the backscattered electric field using the CBFM-E, each cylinder is divided, along the vertical axis z, into blocks of height approximately equal to λ_s each (λ_s is the wavelength inside each scatterer). Since the cell size used to discretize the second cylinder (with the highest real permittivity ϵ'_s) must be less than $\frac{\lambda_{s,2}}{10} = 0.02$ m, we start by discretizing the entire simulation scene into $Nb_{c,1} = 61440$ cells of size $T_c = 0.015$ m. Therefore, in order to observe the behaviour of the CBFM-E that employs a non-uniform mesh based on the dielectric properties of the two cylinders, we discretize them following in two ways, namely using : (i) cell size $T_{c,1} = 0.03$ m for the first; and (ii) cell size $T_{c,2} = 0.015$ m for the second. This strategy results in a reduced number of cells, namely $Nb_{c,2} = 21400$. The non-uniform simulation model is shown in Fig. 4.12.

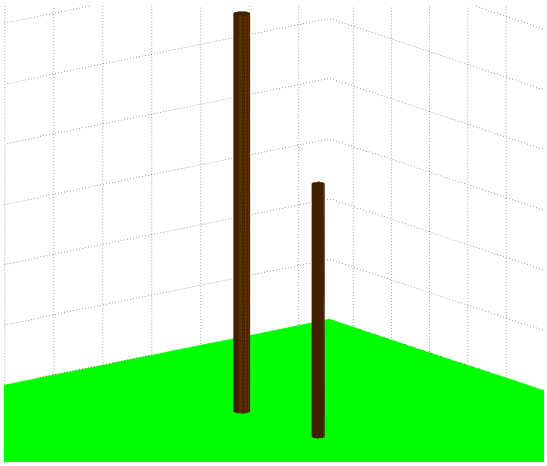
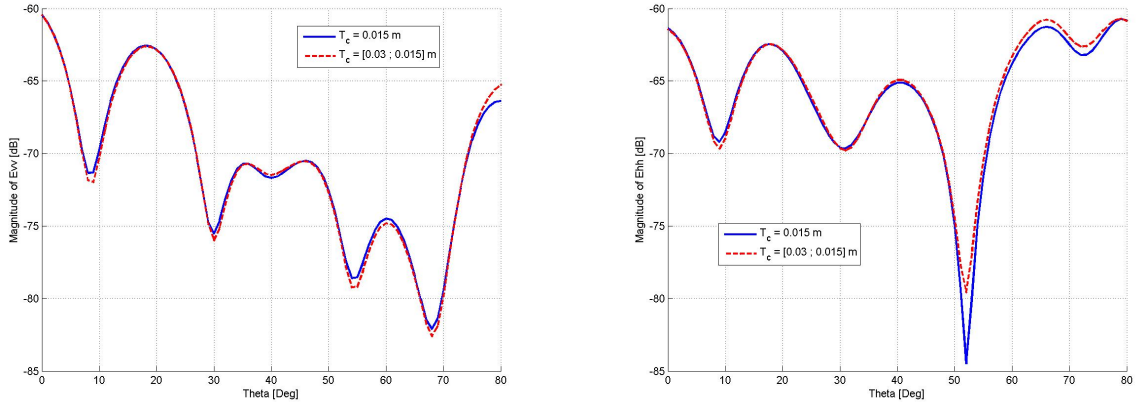


FIGURE 4.12: 2 vertical cylinders of relative permittivities $\epsilon_{s,1} = 7.2 + 0.04\mathrm{j}$ and $\epsilon_{s,2} = 24.6 + 0.06\mathrm{j}$ discretized respectively by using cell sizes $T_{c,1} = 0.03$ m and $T_{c,2} = 0.015$ m

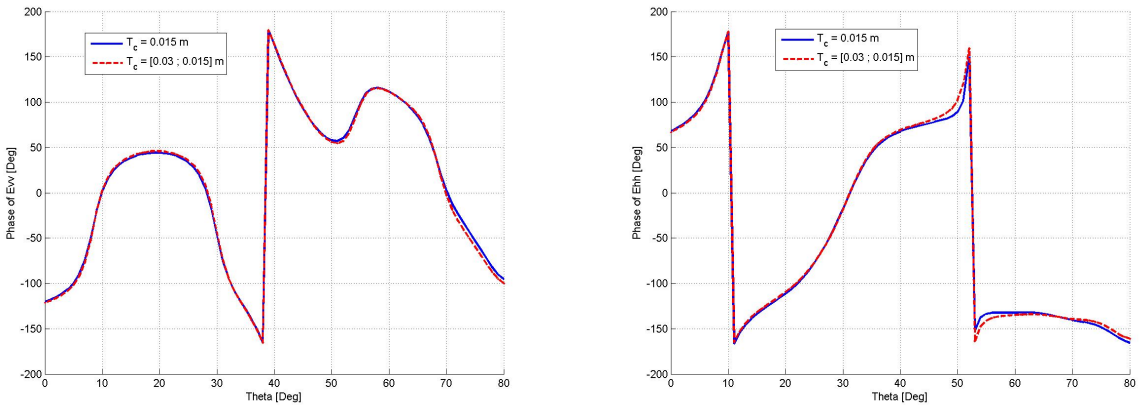
Figure 4.13 plots the magnitude of the backscattered electric fields obtained with the two simulations ($Nb_{c,1} = 61440$ and $Nb_{c,2} = 21400$). We see that the results obtained with the non-uniform mesh are identical to those derived by the CBFM-E using the uniform mesh. However, as expected, the computation time with $Nb_{c,1} = 61440$ is 6 times higher than that of the second simulation with $Nb_{c,2} = 21400$ (see Table 4.4).

TABLE 4.4: Difference between the uniform and non-uniform mesh in terms of CPU time and required memory storage.

T_c size (m)	Z size ($3 \times Nb_c$)	Z^c size	CPU time	CR
0.015	61440	1222	44 min 5 sec	150
[0.015 ; 0.03]	21400	806	7 min 48 sec	79



(A) E_{VV} (B) E_{HH}



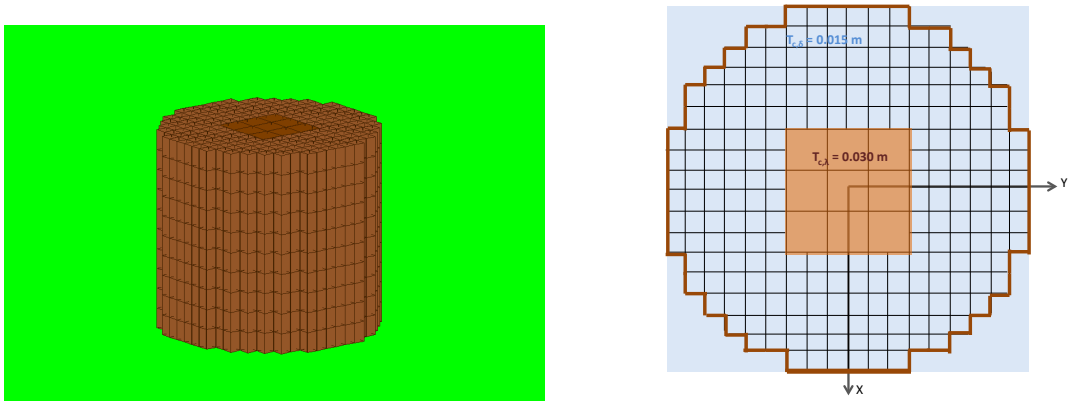
(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 4.13: Variations of the magnitude and phase of the backscattered fields, for both polarization VV and HH, computed with the CBFM-E with an uniform and a non-uniform meshes with $\theta_i = [0; 1; 80^\circ]$ and $\varphi_i = 40^\circ$

Additionally, the non-uniform mesh enables us to significantly reduce the memory consumption of the CBFM-E code since it reduces the numerical size of the electromagnetic problem. Nevertheless,

we notice that the first simulation achieves a better compression rate. Indeed CR_1 is equal to 150, while CR_2 is 79. This is due to the fact that we have used identical CBFM block heights for the two simulations. Indeed, as mentioned earlier, it has been demonstrated that the size of the CBFM blocks has a direct influence on the compression rate as well as on the computing time. Increasing the number of elements per block not only results in a significant increase of the compression rate but also, unfortunately, an increase of the CPU time [30]. To rectify this problem, and to enable the second simulation to achieve the same compression rate, we can simply divide the second cylinder into CBFM blocks of height $2\lambda_s$ rather than the present λ_s.

Next, we apply the CBFM-E to another simple example of a single tree trunk with a high value of tangent loss (and thus a high value of ϵ_r''), in order to check if the accuracy of the CBFM-E is maintained when it is applied on a non-uniform mesh inside the same scatterer. Hence, a single tree trunk is modeled by a cylinder of circular cross section whose dimensions are $R_s = 0.27$ m and $h_s = 4.5$ m. It has a complex relative permittivity $\epsilon_s = 7.6 + 8.9\text{j}$ ($\tan\Delta = 1.17$). As in the previous example, the cylinder is set above ground of complex relative permittivity $\epsilon_g = 5 + 3.6\text{j}$, and is illuminated by an incident plane wave, at f=300 MHz. Therefore, the selected value of ϵ_r'' corresponds to a skin depth inside this scatterer equal to $\delta_s = 0.075$ m. Then, we set up the constants D_λ and D_δ respectively to 10 and 5, which results in the implementation of a non-uniform mesh inside this single tree trunk. Thus, the skin depth zone is discretized into cells of size $T_{c,\delta} = \dfrac{\delta_s}{D_\delta} = 0.015$ m, and for the rest of the scatterer, we use a greater cell size, namely $T_{c,\lambda} = \dfrac{\lambda_s}{D_\lambda} = 0.03$ m. The first CBFM block of this single scatterer, discretized as such, is presented in Figure 4.14.



(A) Side view (B) Circular cross section

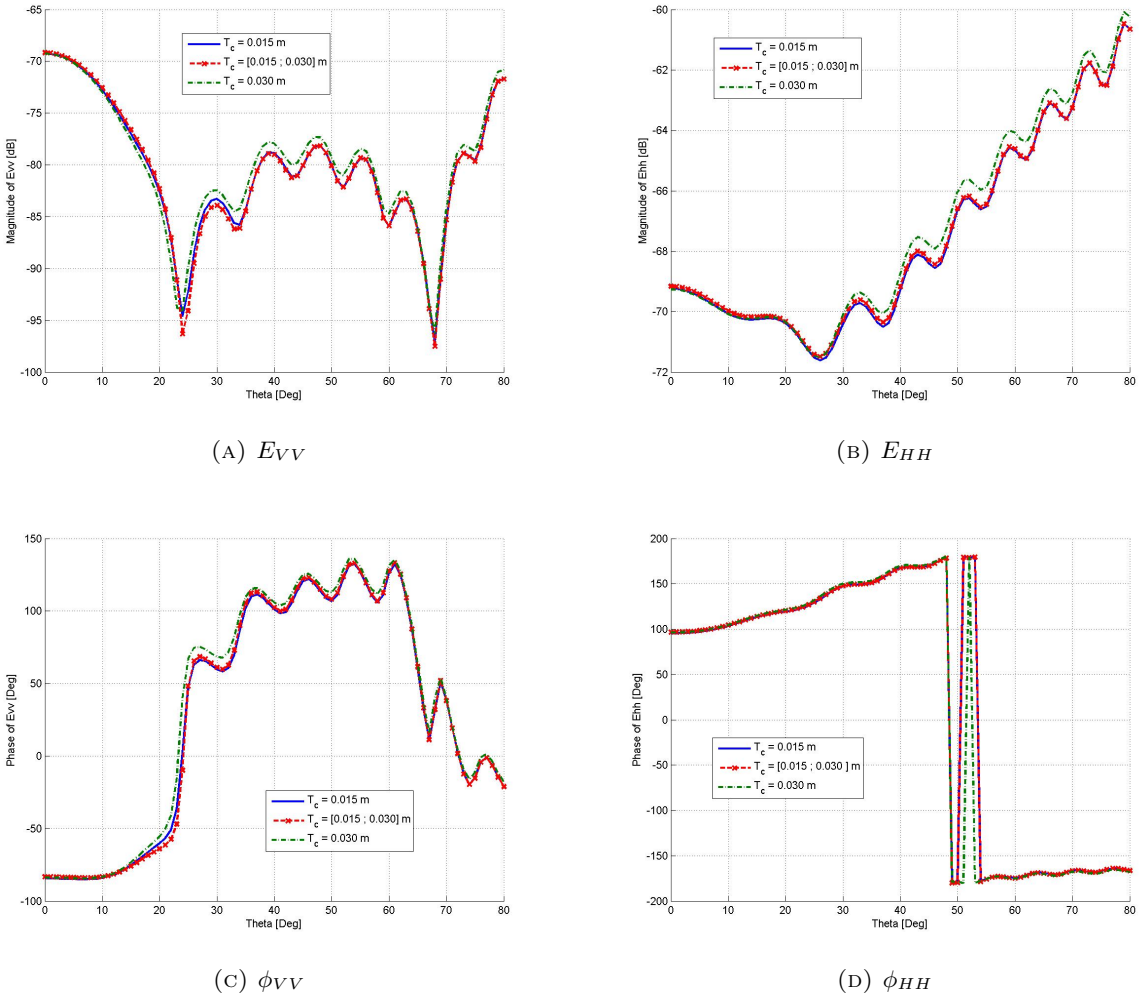
FIGURE 4.14: Non-uniform discretization of the first CBFM block of a tree trunk of a circular cross-section whose dimensions are $R_s = 0.27$ m and $h_s = 4.5$ m.

In order to demonstrate the relevancy of this approach, we compare, in Figure 4.15, the results obtained by applying the CBFM-E on the non-uniform mesh described above, to those obtained with two uniform cell sizes $T_c = 0.015$ m and $Tc = 0.030$ m. We present the performances of the CBFM-E in terms of CPU time and memory requirement, for these three simulations, in Table 4.5.

Note that, in order to quantify the contribution of the non-uniform mesh in the reduction of the electrical size of the simulation scene, we define G_{nu} as the ratio between the number of cells obtained with a uniform mesh which satisfies the conditions on the cell size (for example, $T_c \leqslant \frac{\lambda_s}{10}$ and $T_c \leqslant \frac{\delta_s}{5}$), and the number of cells arising from a non-uniform mesh, depending on the dielectric properties of the forest simulation scene.

$$G_{nu} = \frac{Nb_{c,uniform}}{Nb_{c,non-uniform}} = \frac{Nb_c(T_c = min_i(T^i_{c,\lambda}, T^i_{c,\delta}))}{Nb_c(T_c \in [T^i_{c,\lambda}; T^i_{c,\delta}])} \tag{4.6}$$

where $T^i_{c,\lambda}$ and $T^i_{c,\delta}$ are, respectively, the cell sizes which fulfil the conditions in relation with the wavelength inside the scatterer λ_s and the skin depth δ for each scatterer i.



(A) E_{VV} (B) E_{HH}

(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 4.15: Variations of the magnitude and phase of the backscattered fields, for both polarization VV and HH, computed with the CBFM-E with two uniform meshes ($T_c = 0.015$ m and $T_c = 0.030$ m) and a non-uniform mesh ($T_c = [0.015; 0.030]$ m) with $\theta_i = [0; 1; 80^\circ]$ and $\varphi_i = 40^\circ$

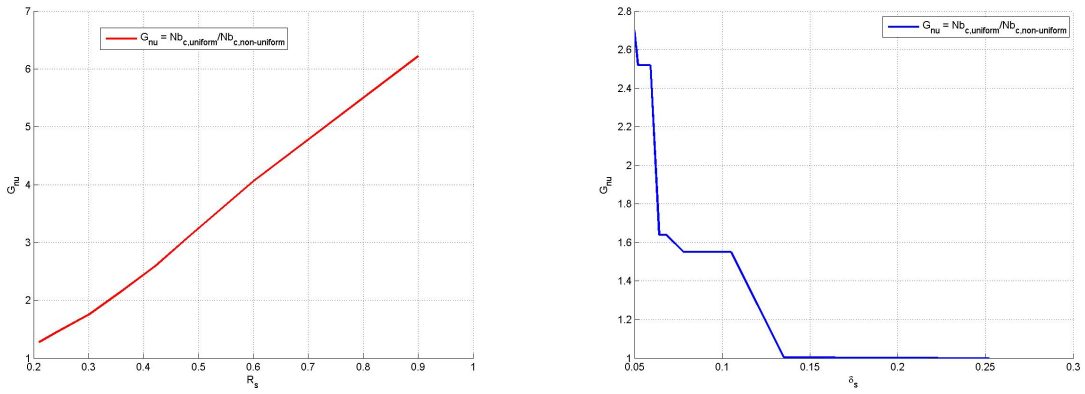
The first finding is that the non-uniform mesh, inside the scatterer, does not affected the accuracy of the CBFM-E solution, since we obtained results which match perfectly with those obtained for $T_c = 0.015$ m. Second, the results confirm that $T_c = 0.03$ m, which is in fact equal to $T_{c,\lambda}$ was not

sufficient to ensure the accuracy of the CBFM-E, since it does not enable it to take into account the rapid variations of the electric field inside the skin depth region. Hence the non-uniform mesh has enabled us to achieve a significant gain in terms of the electrical size of the simulation scene $G_{nu} = 2.87$, while maintaining the accuracy of the CBFM-E solution, thanks to its applicability to a non-uniform mesh.

TABLE 4.5: Performances of the CBFM-E in terms of CPU time and memory requirements when applied to the two uniform meshes defined by $T_c = 0.015$ m and $T_c = 0.030$ m and a non-uniform mesh with $T_c = [0.015; 0.030]$ m

T_c size (m)	Z size $(3 \times Nb_c)$	Z^c size	CPU time	CR
0.015	61440	1222	44 min 5 sec	150
[0.015 ; 0.03]	21400	806	7 min 48 sec	79
0.03	21400	806	7 min 48 sec	79

Furthermore, the non-uniform approach inside the same scatterer, is even more useful and relevant for large tree trunks with high values of ϵ_r''. In fact, as shown in Figure 4.16, the greater the ratio $\frac{R_s}{\delta_s}$, the greater G_{nu} is, and hence, the more gain in terms of memory resources and CPU time this approach brings.



(A) $G_{nu} = f(R_s)$ (B) $G_{nu} = f(\delta_s)$

FIGURE 4.16: The gain achieved by the non-uniform mesh increases with the tree trunk size (R_s) and decreases with the skin depth δ_s. Therefore, G_{nu} is directly proportional to the ratio $\frac{R_s}{\delta_s}$

We end this paragraph dealing with the applicability of the CBFM-E to a non-uniform mesh inside the same scatterer, by checking the value of D_δ required to ensure the accuracy of the CBFM-E solution for high values of ϵ_r''. For this purpose, we apply the CBFM-E to a single cylinder whose dielectric permittivity is equal to $\epsilon_s = 7.6 + 14.8$j, with different non-uniform meshes corresponding to different values of D_δ (see table 4.6). The initial $T_c = 0.03$ m complies with the condition on λ_s and is used in all cases to discretize the inner region which is not concerned by D_δ. Then, we

compare the associated results to those obtained by the application of the CBFM-E with a uniform cell size $T_c = min(T_{c,\lambda}, T_{c,\delta})$. Figure 4.17 shows a comparison of the backscattered fields obtained with the uniform $T_c = 0.01$ m with those derived with the different configurations of non-uniform mesh, summarized in table 4.6. In fact, a uniform mesh with $T_c = 0.01$ m is equivalent to the discretization of the entire scatterer, and not only the skin depth region, into cells of size $T_c \leqslant \frac{\delta}{5}$, resulting thus in a numerically large simulation scene of $Nb_c = 201600$ cells.

TABLE 4.6: Application of the CBFM-E to a cylinder of dimensions $R_s = 0.24$ m and $h_s = 4.5$ m, and relative dielectric permittivity $\epsilon_s = 7.6 + 14.8$ j for different values of D_δ

D_δ	T_c (m)	Nb_c	CPU time (sec)
1	0.030	7800	69
2	[0.030 ; 0.020]	19576	455
3	[0.030 ; 0.015]	45824	1410
5	[0.030 ; 0.010]	140032	13024

Figure 4.17 shows that a factor D_δ of 3 - 5 is sufficient to resolve the skin depth and, hence, to ensure the accuracy of the CBFM-E solution when applied to a dielectric 3D scatterer with a high value of ϵ_r''. Again, we can see that a uniform mesh with $T_c = T_{c,\lambda}$, although it is attractive in terms of computation burden, does not provide satisfactory results when compared to the uniform mesh corresponding to $T_c(D_\delta = 5)$.

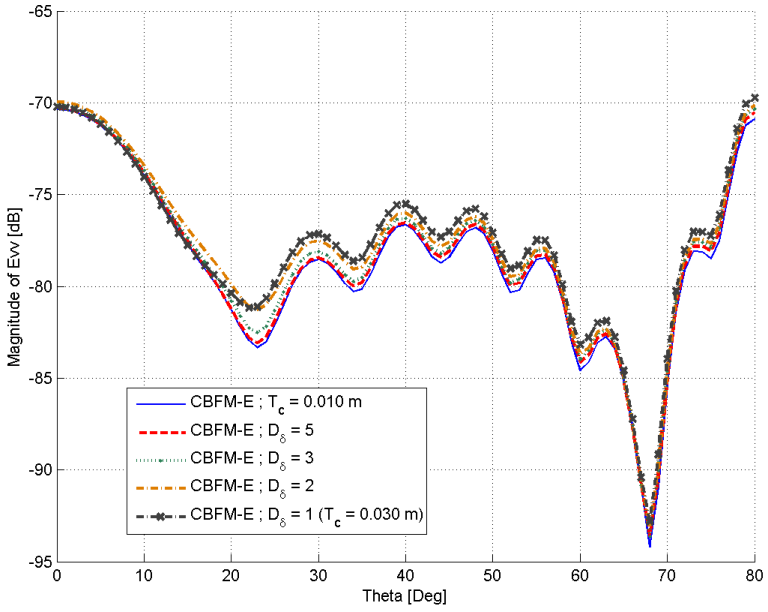


FIGURE 4.17: Comparison of the magnitude of the backscattered fields E_{VV}, obtained by applying the CBFM-E on a non-uniform mesh for different values of D_δ to that obtained with two uniform meshes $T_c = 0.030$ m (equivalent to $D_\delta = 1$) and $T_c = T_c(D_\delta = 5) = 0.010$. The results are plotted for $\theta_{i,s} = [0; 1; 80]^\circ$ and $\varphi_{i,s} = 0^\circ$.

Finally, in order to verify the accuracy of the CBFM solution while using a non-uniform mesh for a larger forest simulation scene, we apply the CBFM-E to a large electromagnetic problem simulating a forest patch composed of forty five cylinders modeling nine trees. The tree trunks go up to a height of 4.8 m and are approximately 2 m apart (see Fig. 4.18). The cylinders are placed over a dielectric ground with a complex relative permittivity of $\epsilon_g = 5 + 2.1\text{j}$, and the trees themselves have a complex relative permittivity $\epsilon_s = \epsilon'_s + \epsilon''_s$ j where $\epsilon''_s = 0.9$ and ϵ'_s vary from $\epsilon'_{s,min} = 2.9$ to $\epsilon'_{s,max} = 24.1$. As the previous example, the simulation scene is illuminated by an incident plane wave at a frequency f = 300 MHz.

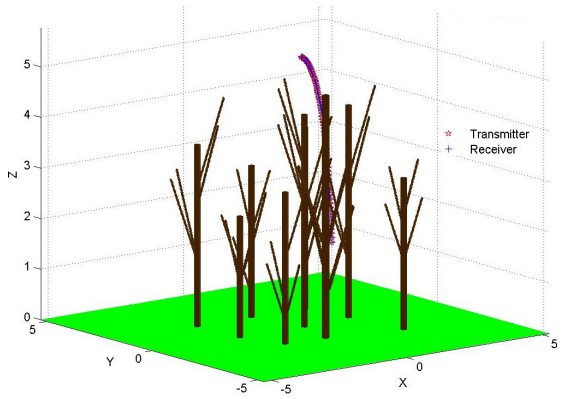
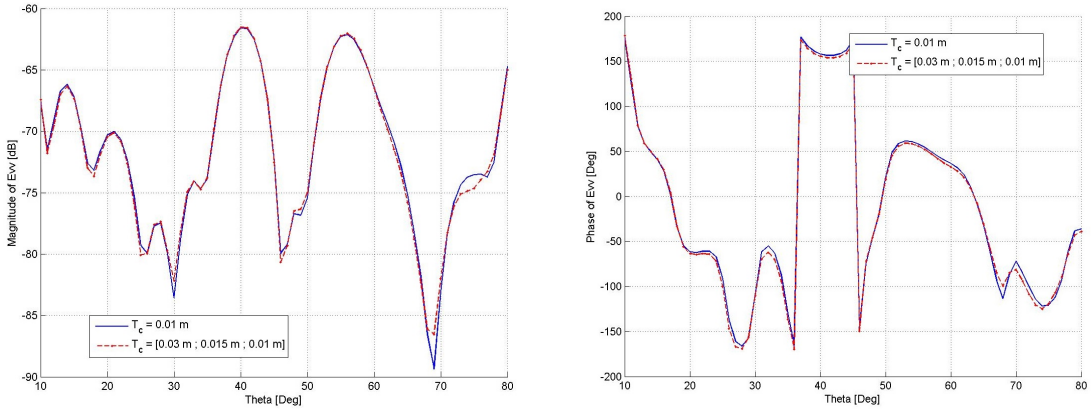


FIGURE 4.18: *A forest simulation scene composed of* 45 *cylinders modeling* 9 *trees with* 4 *branches each*



(A) E_{VV} (B) ϕ_{VV}

FIGURE 4.19: *Variations of the magnitude (a) and phase (b) of the backscattered fields* E_{VV}*, computed with the CBFM-E with an uniform and a non-uniform meshes with an azimuth angle* $\varphi_i = 40°$

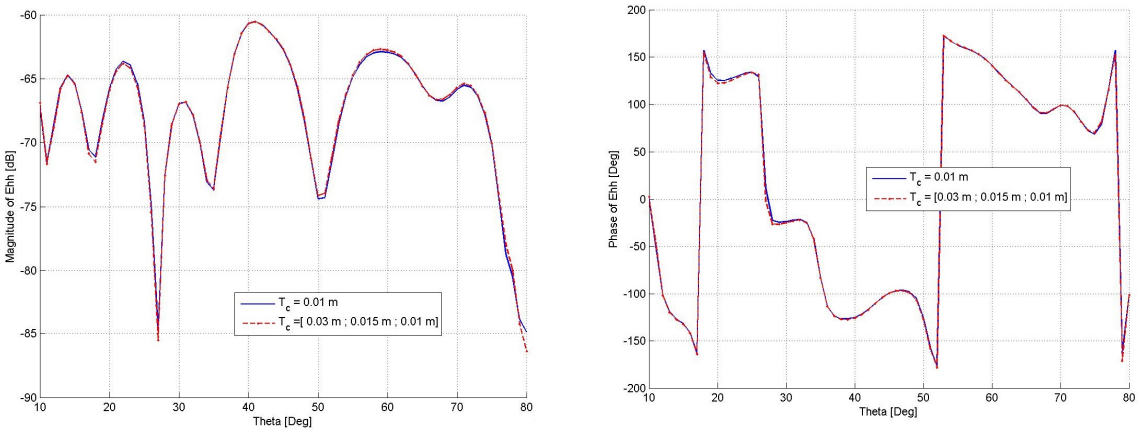
For this example, let us assume that the cell size used to discretize the scatterers must be less than $\frac{\lambda_s}{16}$. Thus, the highest real permittivity ϵ'_s, equal here to 24.1, corresponds to the lowest cell size $T_c = 0.01$ m. Therefore, we start by discretizing the entire simulation scene into $Nb_{c,1} = 565920$ cells of size $T_c = 0.01$ m. Then, once again, in order to observe the behaviour of the CBFM-E with a non-uniform mesh based on the dielectric properties of the 45 cylinders modeling the nine trees, we descritize them using three different cell sizes $T_{c,1} = 0.01$ m, $T_{c,2} = 0.015$ m and $T_{c,3} = 0.03$ m. Hence,

the simulation scene is now represented by an electromagnetic problem of $Nb_{c,2} = 171340$. The results obtained with the two simulations ($Nb_{c,1} = 565920$ and $Nb_{c,2} = 171340$) are presented in Table 4.7. Fig. 4.19 and Fig. 4.20 plot the magnitude of the backscattered electric fields obtained with the two simulations.

We see, again, that the results obtained with the non-uniform mesh are identical to those derived by the CBFM-E using the uniform mesh. Furthermore, the non-uniform mesh allows us a significant reduction of the CPU time and the memory resources needed.

TABLE 4.7: Performances of the two simulations in terms of memory use and computing time

T_c size (m)	Z size ($3 \times Nb_c$)	CPU time
0.01	1697760	45 h 15 min
[0.01 ; 0.015 ; 0.03]	514020	4 h 12 min



(A) (B)

FIGURE 4.20: *Variations of the magnitude (a) and phase (b) of the backscattered fields E_{HH}, computed with the CBFM-E with an uniform and a non-uniform meshes with an azimuth angle $\varphi_i = 40°$*

We have demonstrated in this section that the accuracy of the results obtained by using the CBFM-E is not compromised when a non-uniform mesh is used. Consequently, we can apply the CBFM-E to larger forest areas while taking into account the dielectric heterogeneity of the trees as we decompose the computational domain by adapting the mesh to their dielectric permittivities. We observe that it is also interesting to investigate the behaviour of the CBFM-E when we use a non-uniform mesh inside each scatterer to capture the skin effect for high ϵ_s'' values (imaginary part of ϵ_s), without significantly increasing the numerical size of the electromagnetic problem.

Now that we have reduced the initial electrical size of the forest simulation scene, thanks to the robustness of the CBFM and its insensitivity to the non-uniform mesh, we focus on the enhancement of the different steps of the CBFM algorithm in order to reduce as much as possible the corresponding CPU time and required storage.

4.2 Sparse representation of the Macro-Basis Functions (MBFs)

As explained in chapter 2 dealing with the numerical procedure of the CBFM, the macro basis functions (MBFs) are determined for each block i by inverting its specific MoM matrix Z_{ii} of size $3N_i \times 3N_i$ where N_i is the number of cells in the block i (see equation 2.7). Simulations indicate that this step is a burdensome task particularly in terms of CPU time. Furthermore, the computing time corresponding to this part of the CBFM process, increases with the size and the number of the blocks decomposing the forest simulation scene.

Table 4.8 summarizes CPU time (in seconds) spent to generate the CBFs for different sizes of CBFM blocks. It includes the time to compute the elements of the matrix Z_{ii}, to compute the matrix $E_{ii}^{ref,IPWs}$ representing the plane waves excitations, to solve the corresponding matrix equation in order to generate the MBFs and finally the time allocated for the SVD and normalization applied to the MBFs in order to compute the CBFs. First, Table 4.8 enables us to define the most expensive steps, in terms of computing time, which are obviously the resolution of the matrix equation and then, from a certain value of N_i, the computation of the elements of Z_{ii} . Second, it clearly shows that increasing the size of the CBFM blocks leads to a dramatical increase in the CPU time associated to the generation of the MBFs of this block. However, we know that implementing the CBFM with small blocks results in a low compression rate, and thus also dramatically increases required storage.

TABLE 4.8: Total CPU time (sec) required to compute the Characteristic Basis Functions (CBFs) for the block i, namely $C^{(i)}$, depending on N_i which is the number of cells in this block (@ 300 MHz). It includes the time to compute Z_{ii} and $E_{ii}^{ref,IPWs}$, and to solve the associated matrix equation.

N_i	288	1200	2268	3072	4800
Z_{ii}	0	2	6	11	27
$E_{ii}^{ref,IPWs}$	0	0	0	1	1
Solving Z_{ii} equation	9	568	1056	2230	8407
SVD and normalization	1	13	32	88	304
Total CPU for $C^{(i)}$	10	583	1094	2330	8739

Hence, in order to reduce the burden on the CPU time, in the case of large CBFM blocks, previous studies [27, 43] have proposed to use the sparse representation of the MoM matrices to approximate the MBFs for each block instead of inverting 2.3. In fact, the sparse representation would reduce significantly the computation time as matrix inversion that costs $O((3N_i)^3)$ is replaced by $O(3N_i)$ vector-vector divisions (see equation 2.7), where N_i is the size of the CBFM block i. As mentioned in chapter 2, this approach can have a negative effect on the accuracy of the CBFM solution. The question, however, is whether or not this effect is so important that it affects the accuracy of the results obtained for the scattered field, since the ultimate goal of our 3D dielectric model is to compute the scattered fields by a forest medium.

In order to check the impact of the sparse representation on the accuracy of the CBFM solution, and to prove its relevancy to the reduction of the computation time, we apply the CBFM-E with and without the sparse representation of the MBFs to the large forest simulation scene described in Table 4.9 and illustrated in Figure 4.21.

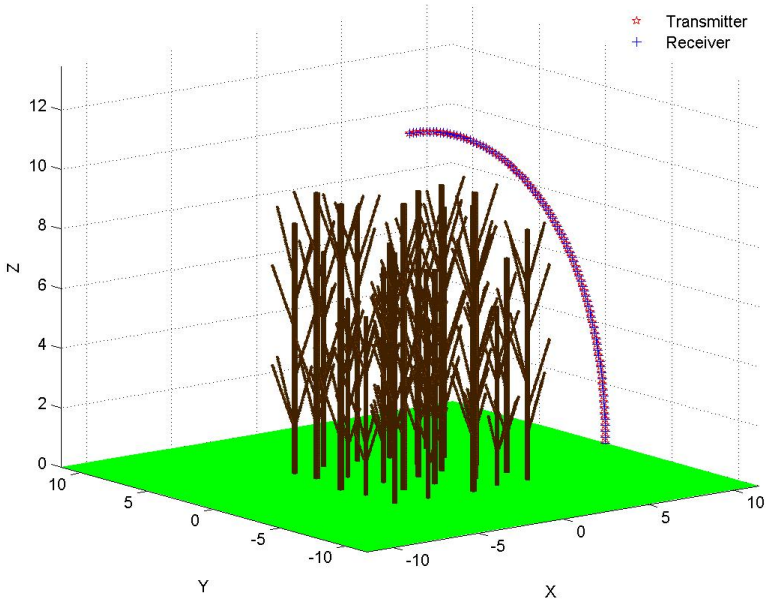


FIGURE 4.21: A large simulation scene composed of 225 cylinders modeling 25 trees with 8 branches each, placed over a real ground plane of relative permittivity $\epsilon_g = 5 + 2.1\text{j}$.

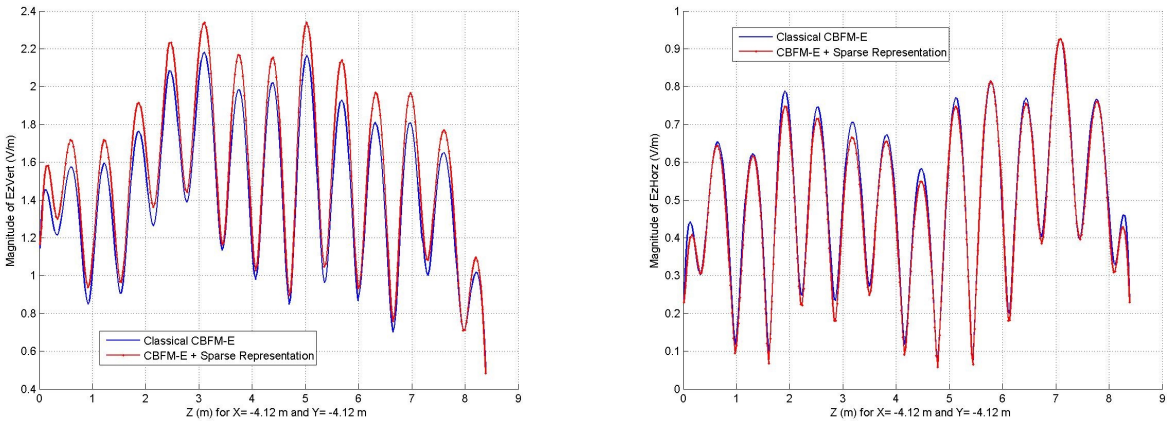
The variations of the magnitude of the components E_z^V and E_z^H of the electric field inside the first scatterer ($R_s = 0.150$ m and $h_s = 8.4$ m) are shown in Figure 4.22 for the incident plane wave defined by $\theta_i = 40^\circ$ and $\phi_i = 0^\circ$, and the backscatter results for both polarization VV and HH are presented in Figure 4.23 for $\theta_{i,s} = [0^\circ; 1^\circ; 80^\circ]$ and $\phi_i = 0^\circ$.

As can be seen, the sparse representation approach has a slight impact on the accuracy of the CBFM-E solution for the calculation of the electric field inside the tree trunks. However, given the targeted domain of application which is forest scattering modeling, we are more interested in the scattered fields, and Figure 4.23 shows that the results obtained with the modified CBFM-E for the backscattered fields match relatively well with those derived by the classical procedure of the CBFM-E.

At the same time, as shown in Table 4.10, the sparse representation of the MBFs enables us to significantly reduce the CPU time required to generate the CBFs. It is noteworthy that the modified CBFM-E achieves a slightly smaller compression rate compared to the classical CBFM-E, this does not, however, alter the fact that the sparse representation of the MBFs increases widely the efficiency of the CBFM-E without degrading its accuracy.

TABLE 4.9: The CBFM-E is applied with and without sparse representation to a large simulation scene of $Nb_c = 221400$ cells @ $f = 300$ MHz

Number of trees	25 with 8 branches each (225 cylinders)
Scatterers dimensions	$a_s \in [0.12, 0.15, 0.18]$ m; $h_s \in [6, 7.2, 8.4, 9.6]m$
Scatterers permittivities	$6.2 + 0.2\text{j} \leqslant \epsilon_s \leqslant 8.4 + 1\text{j}$
Ground permittivity	$\epsilon_g = 5 + 2.1\text{j}$
λ scatterer	$0.346 \text{ m} \leqslant \lambda_s \leqslant 0.401$ m
Cell size	$T_c = 0.03$ m
Number of Cells	$Nb_c = 221400$ cells
Number of CBFM blocks	$N_B = 458$
Number of extended floors	$Nber_{ext,floors} = 4$
Number of plane waves	$N_{IPWs} = 380$ ($\text{d}\theta = \text{d}\phi = 30^\circ$)

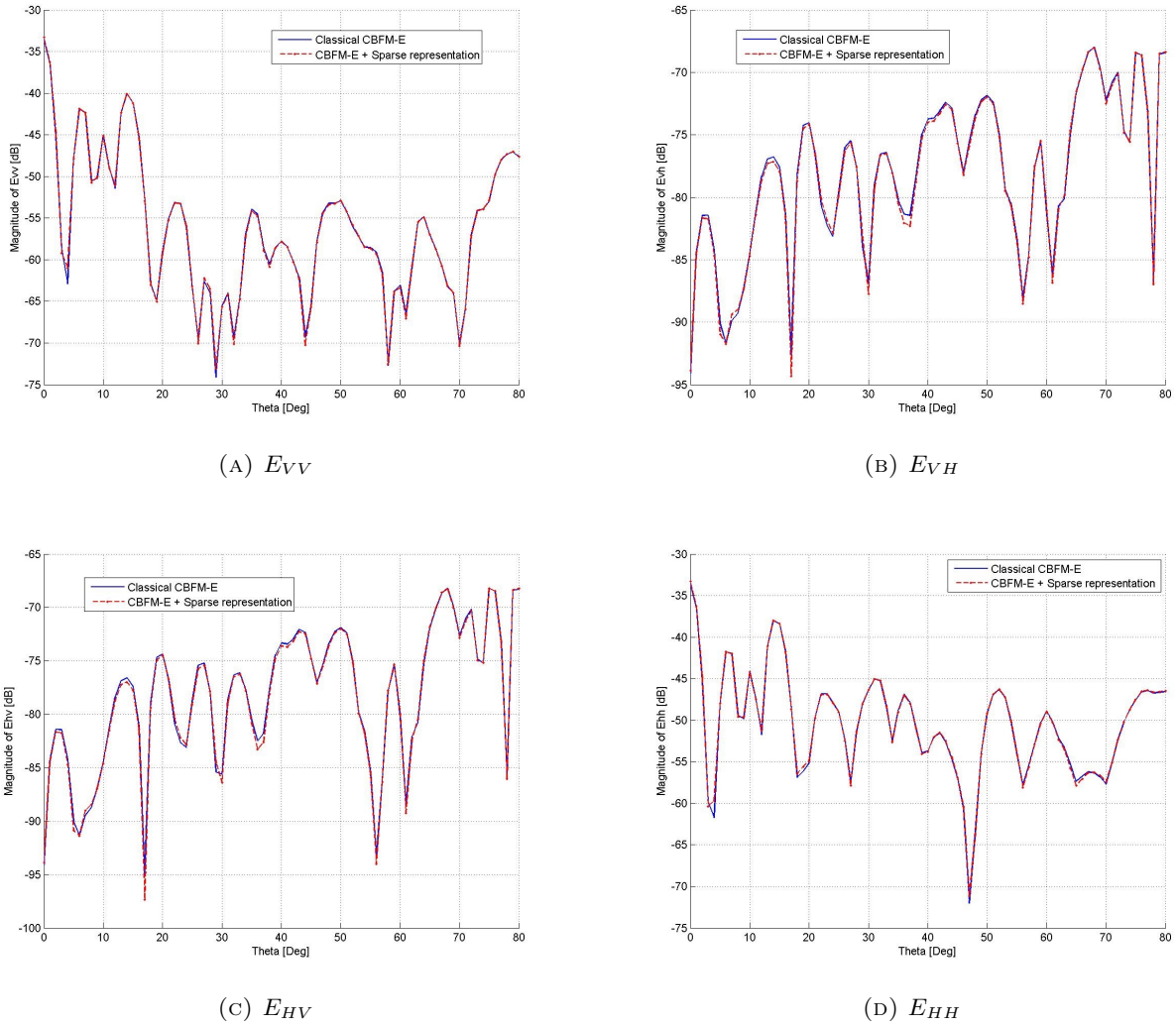


(A) E_z^V (B) E_z^H

FIGURE 4.22: Comparison of the electric field inside the scatterer (E_z^V and E_z^H) obtained by the CBFM-E with a sparse representation of the MBFs to that obtained with a classical CBFM-E.

TABLE 4.10: Performances of CBFM-E with sparse representation of the MBFs (modified CBFM-E) in terms of computation time and compression rate in comparison with the classical CBFM-E.

	Z_c size	CR	ICR (%)	CBFs	Z_c	Z_c^{-1}
Classical CBFM-E	24466	27.15	3.68	1 h 20 min	4 h 22 min	1 h 49 min
Modified CBFM-E	27122	24.5	4.08	2 min 48 sec	4h 25 min	2 h 18 min



(A) E_{VV} (B) E_{VH}

(C) E_{HV} (D) E_{HH}

FIGURE 4.23: Variations of the magnitude of the backscattered fields obtained by the CBFM-E with a sparse representation of the MBFs, and compared to that obtained with a classical CBFM-E.

In order to measure explicitly the saving of time due to the approximation done on the MBFs through the sparse representation approach, the gain G_{sr}^{i} has been defined as

$$G_{sr}^{i} = \frac{\text{Time to generate } E_i^{(\theta,\varphi)}}{\text{Time to generate } \widetilde{E}_i^{(\theta,\varphi)}} \tag{4.7}$$

As shown in Table 4.11, this gain increases with N_i the size of the CBFM blocks decomposing the simulation scene. Indeed, G_{sr}^{i} represents the ratio between $O((3N_i)^3)$ and $O(3N_i)$. This fact is a major advantage of the sparse representation approach, since we know that increasing the size of the CBFM blocks also significantly improves the compression rate achieved by the CBFM-E.

TABLE 4.11: Comparison between the classical and the modified CBFM-E in terms of CPU time and inverted compression rate ICR depending on N_i the size of the CBFM blocks. $\widetilde{ICR}$ (% is the ICR achived by the CBFM-E with the sparse representation of the MBFs).

N_i	690	1380	3450
$E_i^{(\theta,\varphi)}$ (min)	9	20	349
$\widetilde{E}_i^{(\theta,\varphi)}$ (min)	$\leqslant 1$	2	12
G_{sr}^i	≈ 9	10	29
ICR (%)	7.22	4.57	2.45
$\widetilde{ICR}$ (%)	8.77	5.44	2.72

However, despite the gain in terms of CPU time brought by this approximation, the time spent to generate the reduced matrix Z^c is still dominant in comparison to the time spent for the calculation of the CBFs (see Table 4.10). Hence, we focus in the next paragraph on reducing the CPU time required to compute Z^c by hybridizing the CBFM-E with the Adaptive Cross Approximation algorithm (ACA).

4.3 Hybridization of the CBFM with the ACA Algorithm

The Adaptive Cross Approximation (ACA) algorithm has shown good performance, in terms of compression, when dealing with coupling matrix blocks which represent well-separated MoM interactions. Consequently, in this subsection, we apply the ACA algorithm to the generation of the reduced matrix in the context of the CBFM to speed up the most expensive step in the CBFM in terms of the required CPU time

4.3.1 Outline of the Adaptive Cross Approximation algorithm (ACA)

The Adaptive Cross Approximation Algorithm is an adaptive and on-the fly rank-revealing block factorization of the rank-deficient sub-matrices [29]. It is based on the compression of the off-diagonal blocks of the MoM matrix representing well-separated interactions [57–59]. One of the advantages of the ACA algorithm is its purely algebraic nature, which makes it suitable for a great variety of application domain. Furthermore, the application of this algorithm does not require a complete knowledge of the initial electromagnetic problem, leading thus to a significant gain in term of memory cost and CPU time.

The ACA algorithm was introduced by Bebendorf, in [58], to solve static and low frequency problems, where the integral kernels are asymptotically smooth, with $O(N \log N)$ complexity. Later, the ACA was applied successfully to electromagnetic wave problems of moderate and large electrical size in [34, 37, 60, 61]. In this case, according to [34], the ACA algorithm produces very accurate results using only a small part of what are needed by conventional MoM both in terms of memory and total CPU.

Several recent studies [29, 39, 62] have also been interested in the integration of the ACA algorithm in the CBFM process in order to decrease the time spent while generating the final reduced matrix.

In fact, the initial MoM matrix, which represents remote interactions between the N cells composing the 3D forest scene, is a full dense matrix, as a result of the volumetric integral representation of the electric field inside the scatterers. However, the off-diagonal sub-blocks, which describe the interactions between well-separated original low-basis functions, are numerically rank-deficient and hence can be approximated by much lower-ranks matrices. This is where the ACA algorithm comes in, since it allows to approximate those rank-deficient MoM matrix blocks by a much-reduced set of column vectors [34, 58].

This algorithm works through an adaptive block factorization of the rank deficient sub-matrices. It enables to approximate the original MoM sub-matrix $Z^{m,n} \in \mathbb{C}^{m,n}$ by a much lower-rank matrix $\widetilde{Z}^{m,n}$ while respecting an accuracy level defined by the tolerance ϵ such as $||R^{m,n}|| = ||Z^{m,n} - \widetilde{Z}^{m,n}||_F \leqslant \epsilon||Z^{m,n}||_F$, where $R^{m,n}$ is termed as the error matrix and $||R^{m,n}||_F$ and $||Z^{m,n}||_F$ refer respectively to the Frobenius norm of the matrices $R^{m,n}$ and $Z^{m,n}$. The ACA algorithm approximates $Z^{m,n}$ through the following block factorization

$$\widetilde{Z}^{m,n} = U^{m\times r}V^{r\times n} = \sum_{i=1}^{r} U_i^{m\times 1}V_i^{1\times n} \tag{4.8}$$

where r indicates the effective rank of the matrix $Z^{m,n}$, $U^{m\times r}$ and $V^{r\times n}$ are two dense rectangular matrices respectively. The ACA works efficiently while it converges after r iterations with $r << min(m,n)$, then and only then it enables us to gain an appreciable speed advantage relative to a direct matrix fill method. Since the ACA algorithm does not require a priori full knowledge of the initial MoM matrix, instead of computing and storing $m \times n$ elements, the algorithm only requires to compute and store $(m+n) \times r$. This results in a significant reduction of the associated CPU time and memory storage required.

The ACA algorithm given in [34] is described below. The arrays $I = [I_1...I_r]$ and $J = [J_1...J_r]$ contain orderly selected row and column indexes of the matrix $Z^{m\times n}$. As shown in Figure 4.24, u_k is the k_{th} column of the matrix U and v_k is the k_{th} row of the matrix V.

We note that the algorithm proceeds by an iterative selection of the rows and columns that contribute the most to the information included in $Z^{m,n}$ so as to reduce in each iteration the approximation error. In fact, for instance, the row to be selected for the next iteration corresponds to the location of the maximum value of the last computed error column, and so forth. At the end of each iteration k, the convergence is checked through $||U_k||_F||V_k||_F \leqslant \epsilon||\tilde{Z}^{(k)}||_F$ where $||\tilde{Z}^{(k)}||_F^2 = ||\tilde{Z}^{(k-1)} + u_k v_k||_F^2$; u_k and v_k being respectively the k_{th} column of the matrix U and the k_{th} row of the matrix V. The ACA algorithm comes to end after r iterations when the tolerated approximation error ϵ is achieved (named also ϵ_{ACA}).

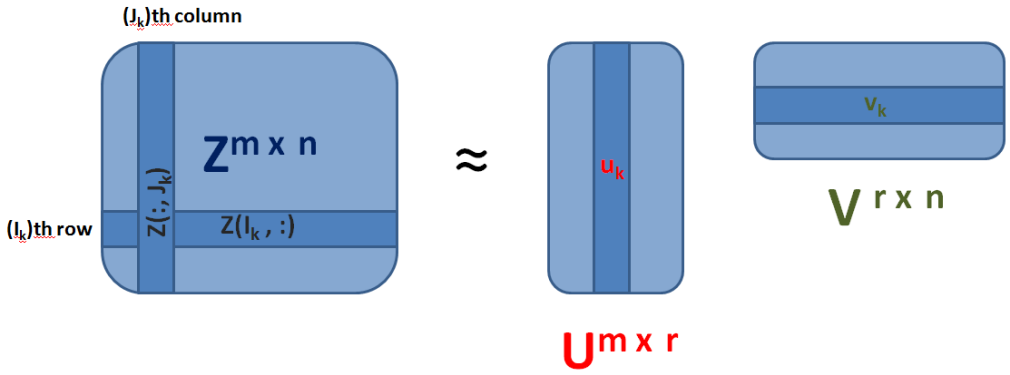


FIGURE 4.24: *The ACA algorithm approximates the matrix $Z^{m\times n}$ through a product form. The details of this algorithm are given below.*

Initialization (k=1):

1) Initialize the 1st row index $I_1 = 1$ and set $\widetilde{Z} = 0$.
2) Initialize the 1st row of the approximate error matrix : $\widetilde{R}(I_1, :) = Z(I_1, :)$.
3) Find the 1st column index J_1 : $|\widetilde{R}(I_1, J_1)| = max_j(|\widetilde{R}(I_1, j)|)$.
4) $v_1 = \widetilde{R}(I_1, :)/\widetilde{R}(I_1, J_1)$
5) Initialize the 1st column of the approximate error matrix : $\widetilde{R}(:, J_1) = Z(:, J_1)$.
6) $u_1 = \widetilde{R}(:, J_1)$.
7) $||\widetilde{Z}^{(1)}||^2 = ||\widetilde{Z}^{(0)}||^2 + ||u_1||^2||v_1||^2$.
8) Find 2nd row index I_2 : $|\widetilde{R}(I_2, J_1)| = max_i(|\widetilde{R}(i, J_1)|)$, $i \neq I_1$.

Iterations :

while $||u_k||||v_k|| \geqslant \epsilon||\widetilde{Z}^{(k)}||$ **do**

% Iteration k

1) Update (I_k)th row of the approximate error matrix : $\widetilde{R}(I_k, :) - \sum_{l=1}^{k-1}(u_l)_{I_k} v_l$.
2) Find kth column index J_k : $|\widetilde{R}(I_k, J_k)| = max_j(|\widetilde{R}(I_k, j)|)$, $j \neq J_1, ..., J_{k-1}$.
3) $v_k = \widetilde{R}(I_k, :)/\widetilde{R}(I_k, J_k)$.
4) Update (J_k)th column of the approximate error matrix : $\widetilde{R}(:, J_k) = Z(:, J_k) - \sum_{l=1}^{k-1}(v_l)_{J_k} u_l$.
5) $u_k = \widetilde{R}(:, J_k)$.
6) $||\widetilde{Z}^{(k)}||^2 = ||\widetilde{Z}^{(k-1)}||^2 + 2\sum_{j=1}^{k-1}|u_j^T u_k||v_k v_j^T| + ||u_k||^2||v_k||^2$.
7) Find next row index I_{k+1} : $|\widetilde{R}(I_{k+1}, J_k)| = max_i(|\widetilde{R}(i, J_k)|)$, $i \neq I_1, ..., I_k$.

end

Algorithm 1: The Adaptive Cross Approximation (ACA) algorithm

It is interesting to note that the ACA algorithm does not require a complete a priori knowledge of the original matrix. We need only to compute and store $r \times (m + n)$ terms of Z instead of computing the entire $m \times n$ terms. Hence, the ACA algorithm leads to a significant reduction of the memory storage in addition to that achieved for the CPU time. The latter scales as $O(r^2(m + n))$ compared to $O(m \times n)$ for the computation of the entire original matrix [34].

It was been confirmed in [34] that the ACA algorithm can be modular and very easily integrated into various MoM codes. Indeed, in [29], the ACA was used to speed up the generation of the reduced matrix Z^c in the context of the CBFM solution. In this section, we investigate the suitability of this approach to our 3D dielectric forest scattering model and particularly check the behaviour of the ACA algorithm toward the heterogeneity of a natural forest medium.

4.3.2 Fast reduced matrix generation using ACA

For a sake of clarity, let us assume that S_i CBFs are generated for each block i. Therefore, their use enables us to replace the initial system of linear equations of size $3N \times 3N$ by a compressed matrix Z^c of size $K \times K$ where N is the number of cells and K is the total number of the CBFs ($K = \sum_{i=1}^{M} S_i$ where M is the number of CBFM blocks). The matrix Z^c is derived by using the original MoM matrix and the total K CBFs, using the Galerkin method as depicted below

$$Z^c = \begin{pmatrix} Z^c_{11} & Z^c_{12} & Z^c_{13} & \cdots & Z^c_{1M} \\ Z^c_{21} & Z^c_{22} & Z^c_{23} & \cdots & Z^c_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z^c_{M1} & Z^c_{M2} & Z^c_{M3} & \cdots & Z^c_{MM} \end{pmatrix} \tag{4.9}$$

$$\text{where } Z^c_{i,j} = < C_i^T, Z^{MoM}_{i,j} C_j > \tag{4.10}$$

Here C_i^T is a $3N_i \times S_i$ matrix which contains the S_i CBFs corresponding to the N_i cells of block i. Z_{ij} $(i \neq j)$ is the coupling matrix linking block i to block j. The product $Z^c_{i,j}$ is then a $S_i \times S_j$ matrix. The reduced matrix generation is particularly expensive in terms of CPU time when the number and size of the CBFM blocks become large (see Table 4.12). On the other hand, it has been shown that the compression rate, CR, increases with the size of the CBFM blocks.

TABLE 4.12: CPU time needed to compute the terms of $Z^{MoM}_{i,j}$ and the product $< C_i^T, Z^{MoM}_{i,j} C_j >$ while generating the reduced matrix Z^c. Note that the total CPU time $CPU(Z^c)$ spent to compute Z^c is equal to $N_B^2 \times (CPU(Z^{MoM}_{i,j}) + CPU(Z^c_{i,j}))$ for an heterogeneous forest simulation scene, where N_B is the total number of CBFM blocks

$N_i = N_j$	897	1242	1932	2622	3312	4692	6072
$Z^{MoM}_{i,j}$ (sec)	$\leqslant 1$	1	3	6	14	27	38
$Z^c_{i,j}$ (sec)	$\leqslant 1$	$\leqslant 1$	1	3	6	14	26

To mitigate this problem, we suggest increasing the block size as much as possible to achieve a high compression rate, and subsequently apply the ACA algorithm to generate the reduced matrix, in order to significantly speed up this step. With the use of this new hybridized method, we benefit from the

high compression rate achieved by the CBFM without actually increasing the associated CPU time, thanks to the efficiency of the ACA algorithm.

4.3.2.1 Using ACA to approximate the submatrix $Z^c_{i,j}$

The ACA algorithm takes advantage of the rank-deficient nature of the coupling matrix blocks representing well-separated MoM interactions. In this work, we use it to approximate the submatrix $Z^c_{i,j}$, representing the interactions between the CBFM blocks i and j, before using it to compute the final reduced matrix. By using the ACA algorithm as detailed in [29], the submatrix $Z^c_{i,j}$, representing interactions between blocks i and j after compression, is approximated as

$$Z^c_{i,j} =< C_i^T, Z^{MoM}_{i,j} C_j > \approx < C_i^T, \tilde{Z}^{MoM}_{i,j} C_j > \tag{4.11}$$

$$\text{where } \tilde{Z}^{MoM}_{i,j} = U_i^{3N_i \times r} V_j^{r \times 3N_j} \tag{4.12}$$

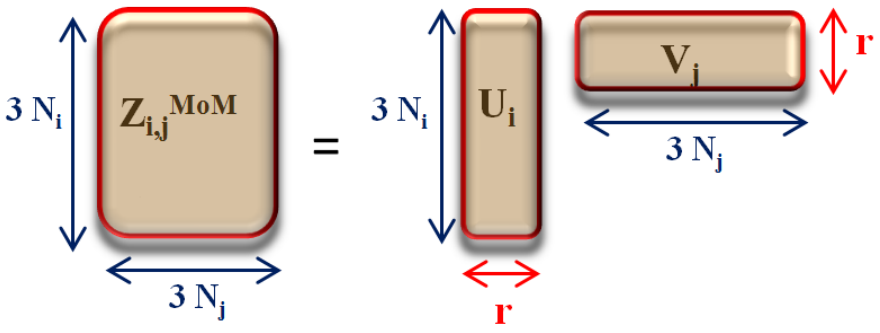


FIGURE 4.25: The ACA algorithm construct two dense rectangular matrices vectors

As explained earlier, $\tilde{Z}^{MoM}_{i,j}$ is a low-rank decomposition of $Z^{MoM}_{i,j}$. It is defined as a product of the two dense rectangular matrices $U_i^{3N_i \times r}$ and $V_j^{r \times 3N_j}$ where N_i and N_j are respectively the number of cells per block i and j and r is the effective rank of the matrix $Z^{MoM}_{i,j}$. This rank decreases when the distance separating the CBFM blocks i and j increases. The ACA algorithm works efficiently when blocks i and j are well separated and results in a rank r much lower than $min(3N_i, 3N_j)$. Hence the matrix $Z^c_{i,j}$ is efficiently computed as

$$Z^c_{i,j} =< C_i^T, U_i V_j C_j > \tag{4.13}$$

Hence, the hybridization of the CBFM-E with the ACA, enables us to reduce the CPU time spent to compute the matrix product $< C_i^T, Z^{MoM}_{i,j} C_j >$, but also help us to save time and memory space costs on the computation of $Z^{MoM}_{i,j}$. In fact, we only need to compute $(3N_i \times r) + (r \times 3N_j) = 3r \times (N_i + N_j)$ terms of $Z^{MoM}_{i,j}$ instead of computing the entire $3N_i \times 3N_j$. As example, let us consider two CBFM blocks i and j of size $N_i = 1356$ cells and $N_j = 2460$ cells. We assume here that these two blocks are

well separated such as the matrix $Z_{i,j}^{MoM}$ has an effective rank $r = 12 <<< min(3N_i, 3N_j)$. Thus, the integration of the ACA leads to a decrease the number of computed elements of $Z_{i,j}^{MoM}$ from 30021840 to 137376, which represents a significant gain both in terms of CPU time and memory space required to store these elements.

It remains to be seen whether the hybridization of the CBFM and the ACA yield a comparable level of accuracy to that obtained with the CBFM-E in comparison with the conventional MoM, when applied to our 3D dielectric forest simulation scene. The numerical accuracy and efficiency of the CBFM-E hybridized with the ACA is assessed in the next paragraph.

4.3.2.2 Primary results on combining CBFM-E and ACA

In order to check the accuracy of the hybridization of the CBFM-E with the ACA, we begin by applying the modified CBFM-E solution (CBFM-E + ACA) to a small simulation scene composed of 9 dielectric cylinders modeling a tree with 8 branches. The tree trunk is placed over a dielectric ground with a complex permittivity of $\epsilon_g = 5$+j2.1, and the tree itself has a complex relative permittivity of $\epsilon_r = 8.2$+j0.9. This homogeneous small simulation scene, presented in Figure 4.26, is illuminated by an incident plane wave, at a frequency $f = 300$ MHz.

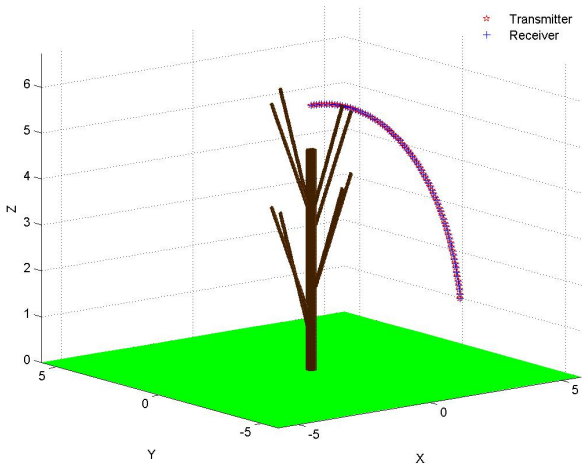
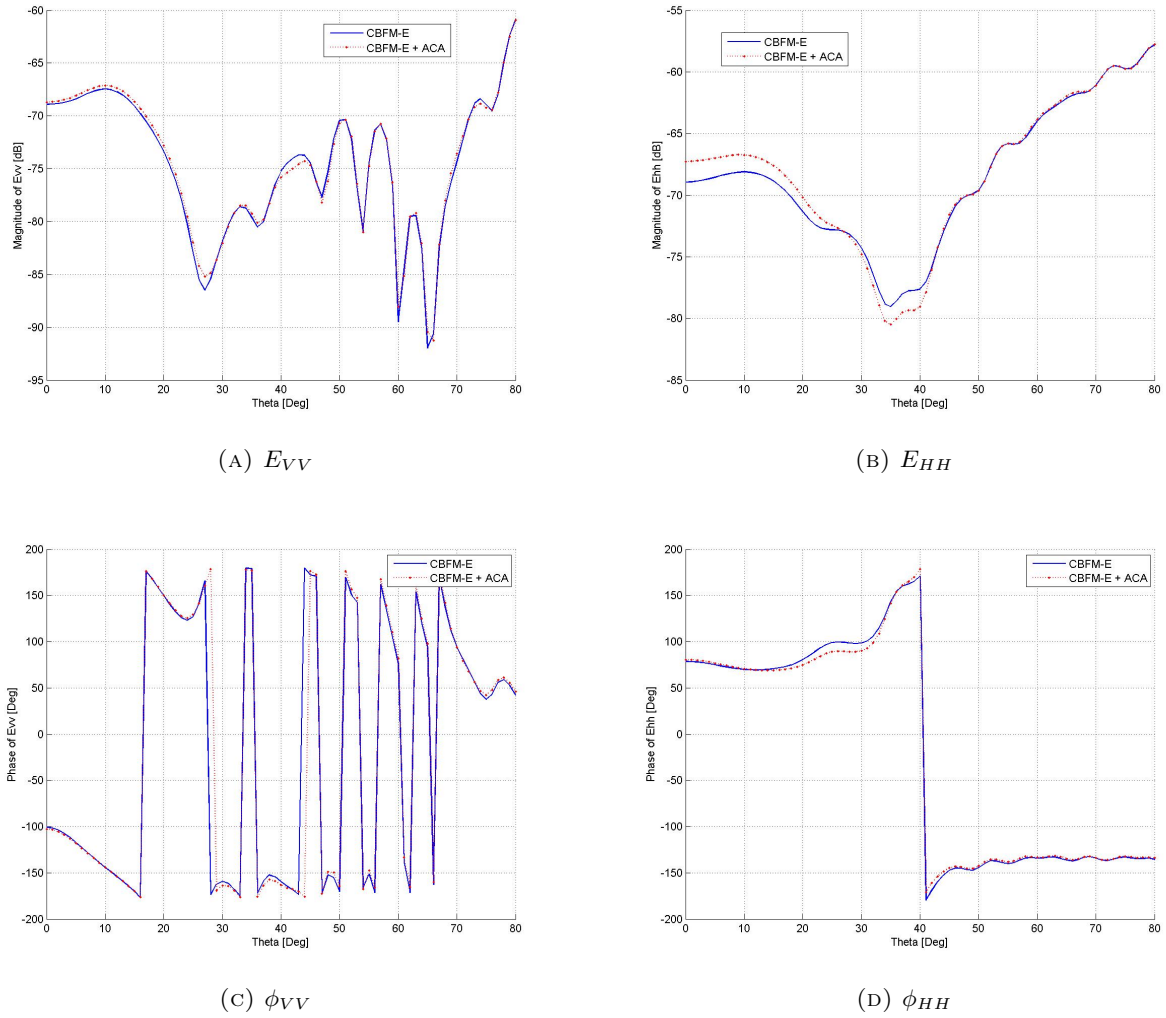


FIGURE 4.26: 9 dielectric cylinders modeling a tree with 8 branches, placed over a real ground plane of relative permittivity $\epsilon_g = 5 + 2.1$j, and discretized into $Nb_c = 13920$ cells.

Each cylinder is divided, along the vertical axis $\vec{z}$, into blocks of height equal almost to λ. Then, we apply the classical CBFM-E and the new hybridized version, with the ACA algorithm while generating the reduced matrix, and we compare the results to those derived by the conventional MoM. A threshold of 10^{-3} is used for the generation of the CBFs (SVD) in the CBFM-E, and the construction of U_i and V_j in the ACA. We also fixed the maximum number of iterations to $N_{iter,max} = 50$ for the ACA algorithm. Thus U_i and V_j, at the output of the ACA, are used to compute $Z_{i,j}^c$ if and only if the effective rank r of $Z_{i,j}^{MoM}$ is lower than $N_{it,max} = 50$.

In Figure 4.27, we can observe a satisfactory agreement between the CBFM-E combined with the ACA, the classical CBFM-E and the conventional MoM. Therefore, the accuracy of the CBFM solution is not compromised by the integration of the ACA algorithm to the generation of the reduced matrix Z^c, and this despite the fact that the ACA has well compressed all the off-diagonal sub-matrices of Z^c (see Figures 4.28 and 4.29).



(A) E_{VV} (B) E_{HH}

(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 4.27: Variations of the magnitude and phase of the backscattered fields, for both polarization VV and HH, computed with the classical CBFM-E, the CBFM-E hybridized with the ACA and the conventional MoM with $\theta_i = [0; 1; 80°]$ and $\varphi_i = 40°$. The ACA is applied with $\epsilon = 10^{-3}$ and $N_{iter,max} = 50$

Figure 4.28 illustrates the variations of $N_{iter,out}$ and the effective rank r depending on $d_{i,j}$ the distance between the CBFM blocks i and j. $N_{iter,out}$ is the number of iterations required by the ACA to converge, and r is the effective rank of $Z_{i,j}^{MoM}$ computed by means of a full SVD with a threshold of 10^{-3} on the normalized singular values. The size of the initial sub-matrix $Z_{i,j}^{MoM}$ is 2070 × 2070. As expected, we note that the effective rank of the coupling sub-matrix $Z_{i,j}^{MoM}$ decreases rapidly with the distance $d_{i,j}$ separating the CBFM blocks i and j. In fact, the ACA algorithm is computationally efficient starting from a separation distance of 2λ, thus it is useless, for instance, to apply the ACA

algorithm to the sub-matrices representing the interactions between two adjacent CBFM blocks. We can also see in Figure 4.28 that the number of iterations required so that the ACA algorithm converges, namely $N_{iter,out}$ is approximately equal to the effective rank of the sub-matrix under consideration. Indeed, the maximal difference between the two values $(r - N_{iter,out})$ is less than 0.4% of the initial size of the sub-matrix $Z_{i,j}^{MoM}$. Moreover, still on the basis of Figure 4.28, $N_{iter,out}$ is always higher than the effective rank r which ensures the accuracy of the CBFM-E solution later, since $\widetilde{Z}_{i,j}^{MoM}$ is still then faithful to the initial $Z_{i,j}^{MoM}$. It is therefore appropriate to keep only r, which will represents in the same time, for the rest of this work, the effective rank of $Z_{i,j}^{MoM}$ and the number of iterations needed by the ACA algorithm to approximate $Z_{i,j}^{MoM}$.

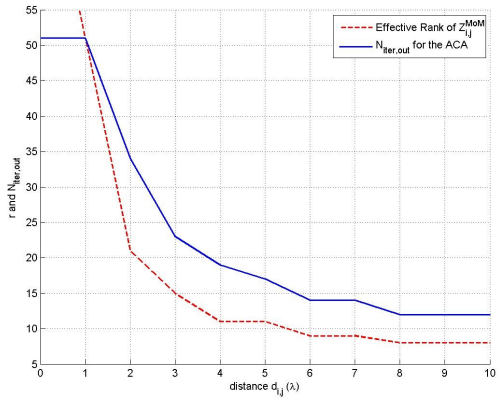
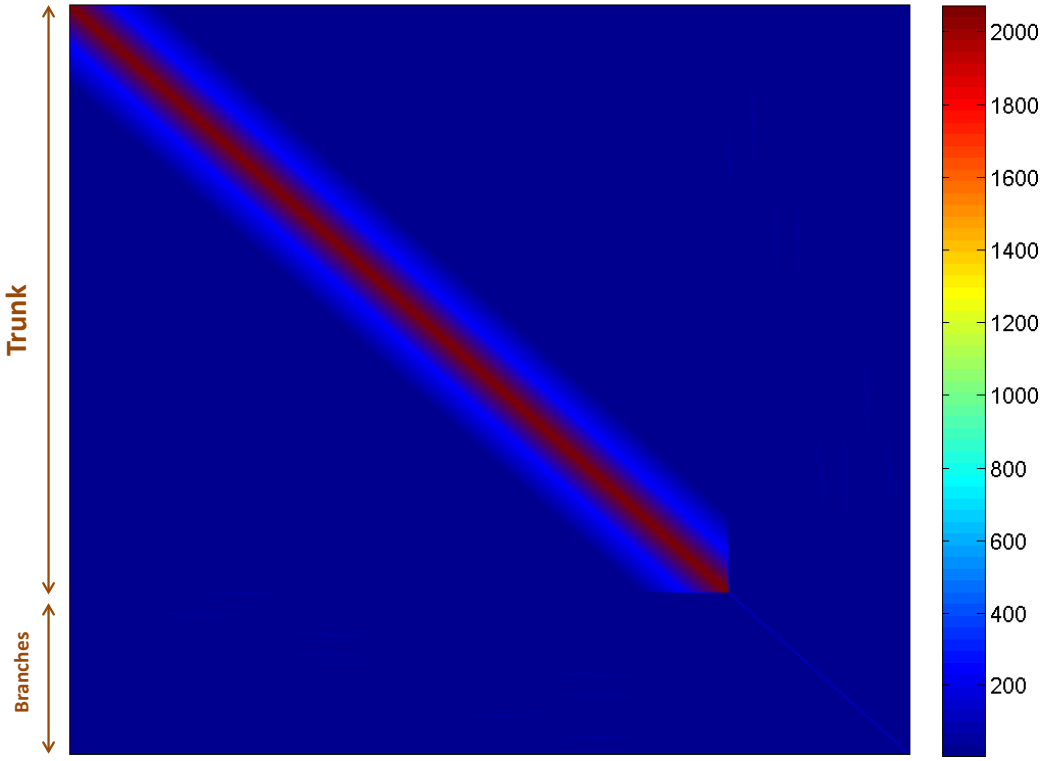


FIGURE 4.28: The distribution of the effective rank of $Z_{i,j}^{MoM}$ and the number of iterations $N_{iter,out}$ required by the ACA to construct $\widetilde{Z}_{i,j}^{MoM}$, depending on the distance between the blocks i and j.

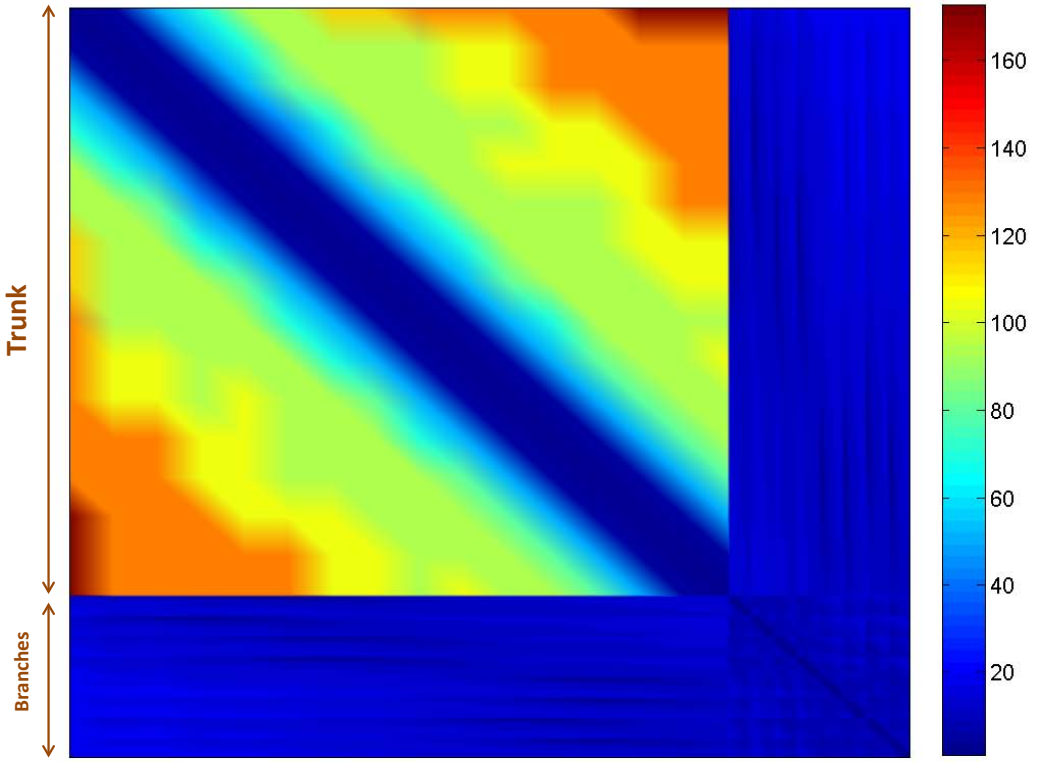
Figure 4.29a illustrates the distribution of r for $Z_{i,j}^{MoM}$ $(i \neq j$ and $i, j \leqslant 88)$, the off-diagonal sub-matrices of the initial $Z_{41760\times41760}^{MoM}$ representing the tree trunk and branches decomposed into 88 CBFM blocks. Figure 4.29b shows the sub-matrix compression rate $SCR_{ACA}^{i,j}$ achieved by the ACA for each coupling sub-matrix $Z_{i,j}^{MoM}$ $(i \neq j$ and $i, j \leqslant 88)$. $SCR_{ACA}^{i,j}$ is defined as the ratio between the size of $Z_{i,j}^{MoM}$ and the number of elements of $Z_{i,j}^{MoM}$ computed to construct U_i and Vj :

$$SCR_{ACA}^{i,j} = \frac{\text{Number of elements in } Z_{i,j}^{MoM}}{\text{Number of elements in } U_i \text{ and } V_j} = \frac{3N_i \times 3N_j}{(3N_i \times r) + (r \times 3N_j)} = \frac{3N_i \times N_j}{r \times (N_i + N_j)} \tag{4.14}$$

where N_i and N_j are the number of cells in the blocks i and j and r is the effective rank of $Z_{i,j}^{MoM}$. Obviously, given the radial size, namely R_s, of the tree trunk and the branches, the CBFM blocks composing the tree trunk are numerically much larger than those composing the branches ($Nbc_{i,trunk} = 690$ and $Nbc_{i,branch} = 40$). These two figures confirm that the sub-matrix compression rate $SCR_{ACA}^{i,j}$ is inversely related to the distance between the blocks i and j. Figure 4.29b particularly shows that $SCR_{ACA}^{i,j}$ depends also on the electrical size of the these two blocks. Indeed, the larger are the blocks i and j, the higher is the compression rate achieved by the ACA on $Z_{i,j}^{MoM}$.



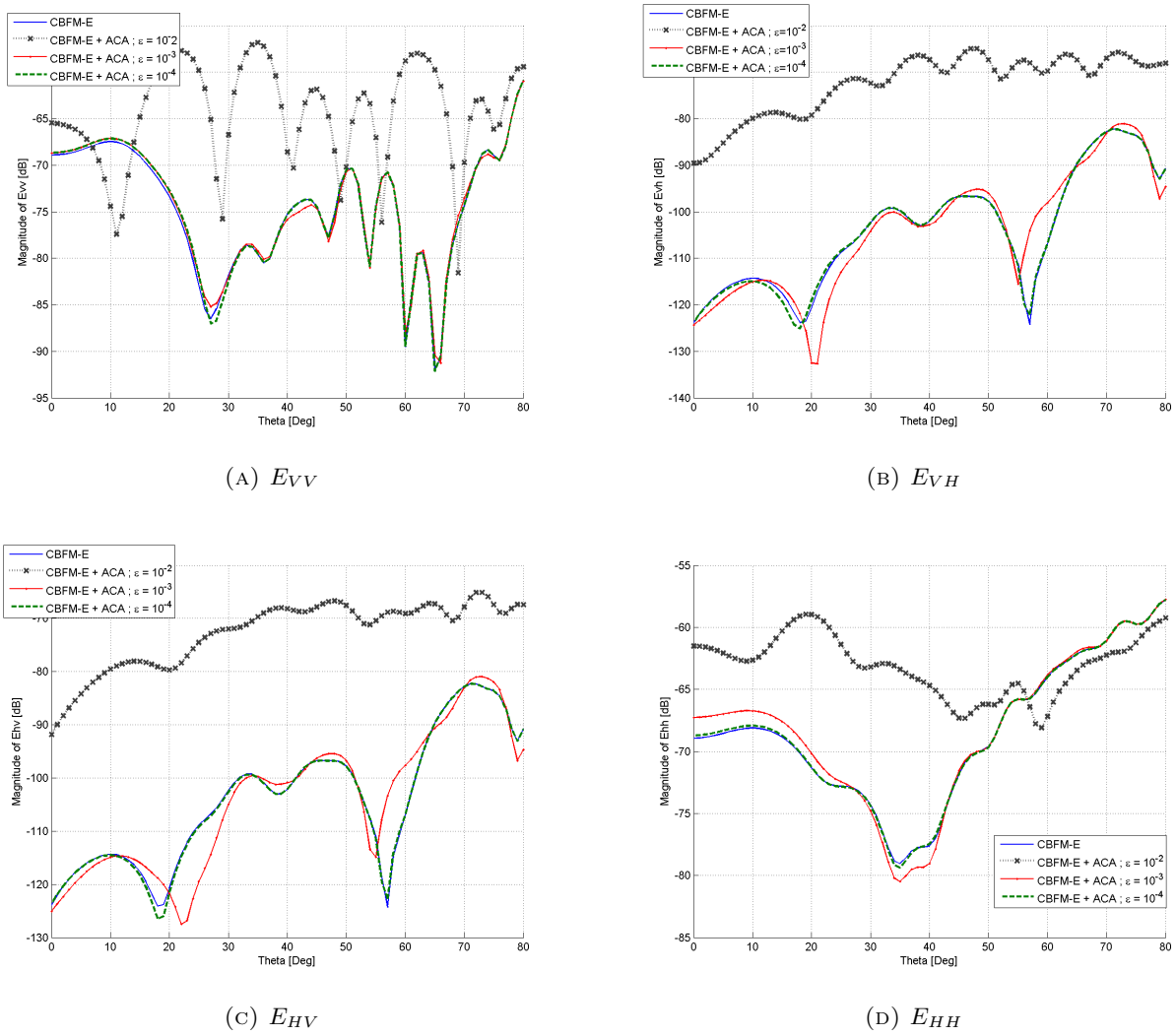
(A) Effective rank of $Z_{i,j}^{MoM}$



(B) Compression Rate on $Z_{i,j}^{MoM}$

FIGURE 4.29: Distribution of the effective rank r and the sub-matrix compression rate $SCR_{ACA}^{i,j}$ related to $Z_{i,j}^{MoM}$, depending not only on the distance separating the blocks i and j but also on their numerical sizes (N_i and N_j). The largest blocks (tree trunk) are of size $3N_i \times 3N_j = 2070 \times 2070$ and the smallest ones (branches) are of size 120×120. Note that we do not apply the ACA to the diagonal sub-matrices thus we put the compression rate at 1 for those blocks.

Despite the satisfactory level of accuracy, and the good compression rate brought by the hybridization of the ACA with the CBFM-E solution, Figure 4.27b shows a disturbing slight difference between the results obtained with the hybridized version and those obtained with a classical CBFM-E. In order to avoid any ambiguity, we carry out additional simulations on the same tree, with different values of ϵ_{ACA} and we observe the impact of this crucial parameter on the accuracy of the CBFM-E combined with the ACA. Figure 4.30 illustrates the variations of the backscattered fields, for the four polarizations (VV, VH, HV and HH) computed by the classical and the modified CBFM-E with $\epsilon_{ACA} = [10^{-2}, 10^{-3}, 10^{-4}]$. And Figure 4.31 shows the impact of this change in the value of ϵ_{ACA} on the sub-matrix compression rate $SCR^{1,j}_{ACA}$ ($j \leqslant 16$) achieved for sub-matrices the tree trunk.



(A) E_{VV} (B) E_{VH}

(C) E_{HV} (D) E_{HH}

FIGURE 4.30: Variations of the magnitude the backscattered fields computed with the classical CBFM-E and the CBFM-E hybridized with the ACA depending on the value of ϵ_{ACA}.

It appears that $\epsilon_{ACA} = 10^{-2}$ is not sufficient to ensure the accuracy of the ACA when applied to approximate $Z^{MoM}_{i,j}$. Yet, this value has been used in [29], for PEC objects in a context of analysis of large antenna systems, so that the relative error is expected to be in the order of few percent or less.

Hence, clearly this claim is no longer valid when the ACA is applied, as in our case, to 3D dielectric objects. Rather, it would be more safe to use a threshold equal to 10^{-3} or 10^{-4}.

On the other hand, we note in Figure 4.31 that decreasing the threshold ϵ_{ACA} has a modest impact on the compression rate. The difference between $N_{iter,out}^{(10^{-3})}$ and $N_{iter,out}^{(10^{-4})}$ does not exceed 0.53% of the initial size of $Z_{i,j}^{MoM}$ ($3N_i = 3N_j = 2070$). However, we may still prefer $\epsilon_{ACA} = 10^{-3}$ since it enables us to obtain a satisfactory level of accuracy with a minimum of computing tasks, and hence less memory space and CPU time. Finally, we can also observe the impact of the threshold value on the total CPU time in Table 4.13. It is clear that the ϵ_{ACA} has not any significant impact on the total CPU time, at least for this small example of $Nb_c = 13920$ cells. Note that both the classical and the modified CBFM-E enable us to reduce significantly the required CPU time comparing to the conventional MoM.

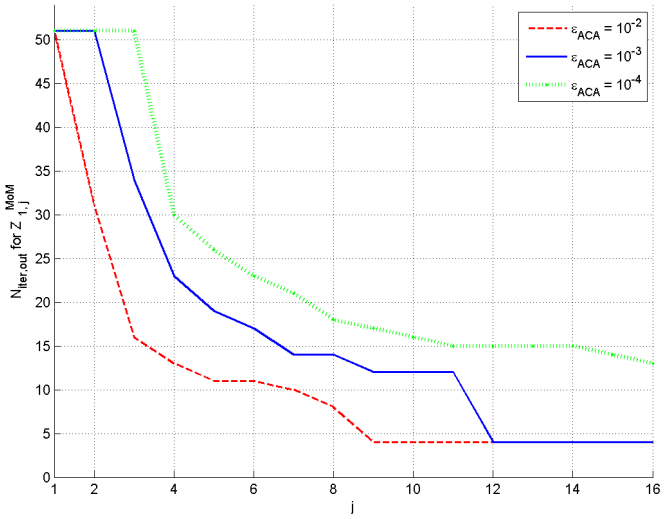


FIGURE 4.31: A large simulation scene composed of 225 cylinders modeling 25 trees with 8 branches each, placed over a real ground plane of relative permittivity $\epsilon_g = 5 + 2.1\text{j}$.

TABLE 4.13: Performances of the conventional MoM, the classical CBFM-E and the CBFM-E combined with the ACA, in terms of CPU time depending on the value of ϵ_{ACA}

	MoM	Classical CBFM-E	CBFM-E + ACA		
ϵ_{ACA}	–	–	10^{-2}	10^{-3}	10^{-4}
$Z_{i,j}^{MoM}$ (sec)	–	33	36	40	42
Total CPU time (min)	831	4.9	5.12	4.78	4.8

Next, in order to assess the performances of the CBFM-E when hybridized with the ACA both in terms of CPU time and memory storage, we move on to a larger forest simulation scene of size $Nb_c = 114770$ cells (see Figure 4.32) and we define the total gain in computing time G_{ACA}^{tot} as

$$G_{ACA}^{tot} = \frac{\text{Time to compute } Z^{MoM}}{\text{Time to compute } \widetilde{Z}^{MoM}} = \sum_{i=1}^{N_b}\sum_{j=1}^{N_b} G_{ACA}^{i,j} = \sum_{i=1}^{N_b}\sum_{j=1}^{N_b} \frac{\text{Time to compute } Z_{i,j}^{MoM}}{\text{Time to compute } \widetilde{Z}_{i,j}^{MoM}} \tag{4.15}$$

where N_b is the total number of CBFM blocks. To measure the efficiency of the hybridization of the CBFM-E with the ACA in terms of gain in required memory resources, we will use the sub-matrix compression rate $SCR_{ACA}^{i,j}$ defined earlier.

It is interesting to note, however, that the gain in required memory space achieved by the ACA while generating the reduced matrix, contrary to the total compression rate achieved by the CBFM-E, does not enable us to run larger simulation scenes with higher frequencies. In fact, this gain does not impact, in any way, the size of the final reduced matrix which corresponds to the required memory space. This size depends rather principally on the size of the CBFM blocks. However, the compression rate achieved by the ACA still important and useful since it decreases the number of elements to compute in $Z_{i,j}^{MoM}$. Hence, it is directly converted to a gain in CPU time.

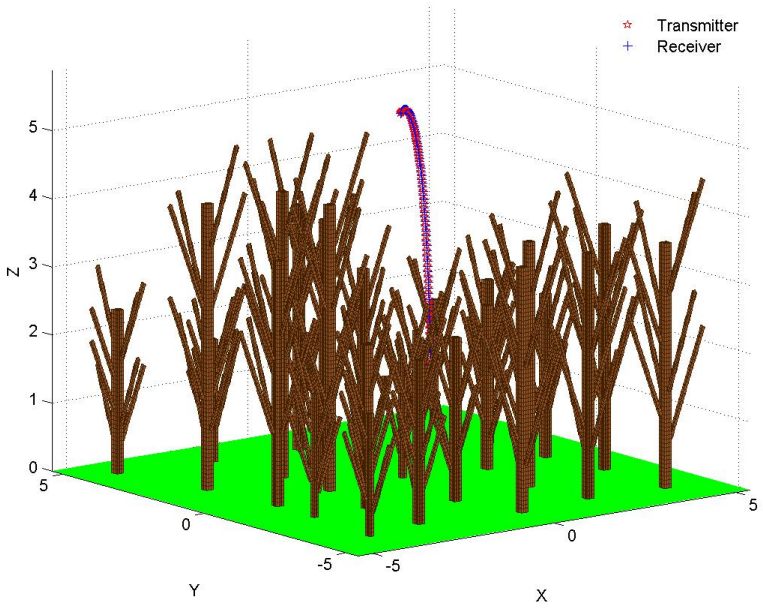


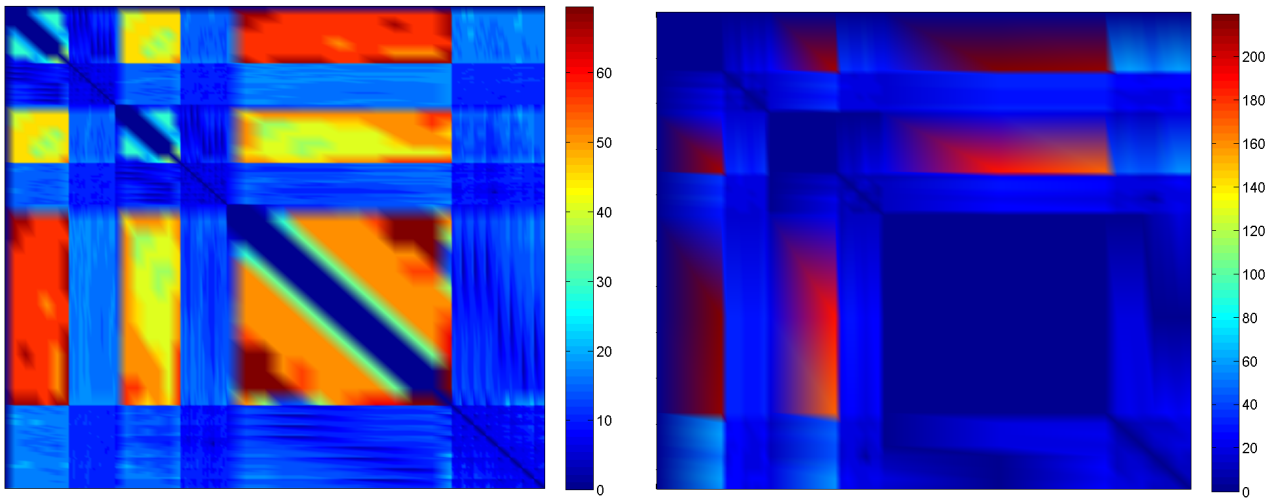
FIGURE 4.32: 225 dielectric cylinders modeling a forest simulation scene of 25 tree with 8 branches each, placed over a real ground plane of relative permittivity $\epsilon_g = 5 + 2.1\text{j}$, and discretized into $Nb_c = 114770$ cells.

In order to compare the performances of the classical CBFM-E with those achieved by the CBFM-E when combined to the ACA, we apply these two methods to the large simulation scene of Figure 4.32. Furthermore, we investigate the influence of the size of the CBFM blocks on the contribution brought by the ACA. The CPU time and memory storage requirements, for different values of h_B (height of the CBFM blocks) are summarized for the classical and modified CBFM-E in Tables 4.14 and 4.15.

TABLE 4.14: Performances of the classical CBFM-E and the CBFM-E combined with the ACA in terms of CPU time and compression rate (CR). The CPU time to compute the final reduced matrix Z^c and to solve the final linear system of equations are given.

h_B	Method	Size of Z^c	CR	CPU time for Z^c	CPU time for $(Z^c)^{-1}$
λ	Classical CBFM-E	33304	10.34	25 min 26 sec	3 h 56 min
	CBFM-E + ACA			19 min 56 sec	
$[2\lambda; 3\lambda]$	Classical CBFM-E	16978	20.23	34 min 27 sec	47 min
	CBFM-E + ACA			9 min 52 sec	
$[3\lambda; 5\lambda; 6\lambda]$	Classical CBFM-E	10944	31.46	44 min 45 sec	16 min
	CBFM-E + ACA			10 min 12 sec	

First, once again, we note the positive impact of the size of the CBFM blocks on the compression rate CR achieved by the CBFM-E solution. Furthermore, this impact results on a significant reduction of the total CPU time, since the final reduced matrix is increasingly smaller. This table shows also that increasing h_B enhances the proceeding of the ACA algorithm and substantially increases its contribution to the gain in CPU time. This is most likely due to the higher sub-matrix compression rate, $SCR^{i,j}_{ACA}$, achieved by the ACA for each sub-matrix $Z^{MoM}_{i,j}$ $(i \neq j)$. Indeed, Figure 4.33 shows the compression rate $SCR^{i,j}_{ACA}$ achieved for the three first trees, divided into 120 blocks of size $h_B = \lambda$ and then, divided into 28 blocks of size $h_B \in [3\lambda; 5\lambda; 6\lambda]$. We note that the maximum value of $SCR^{i,j}_{ACA}$ is much higher on the right (Figure 4.33b), which confirms that the larger the CBFM block, the higher the compression rate achieved for the corresponding MoM sub-matrix.

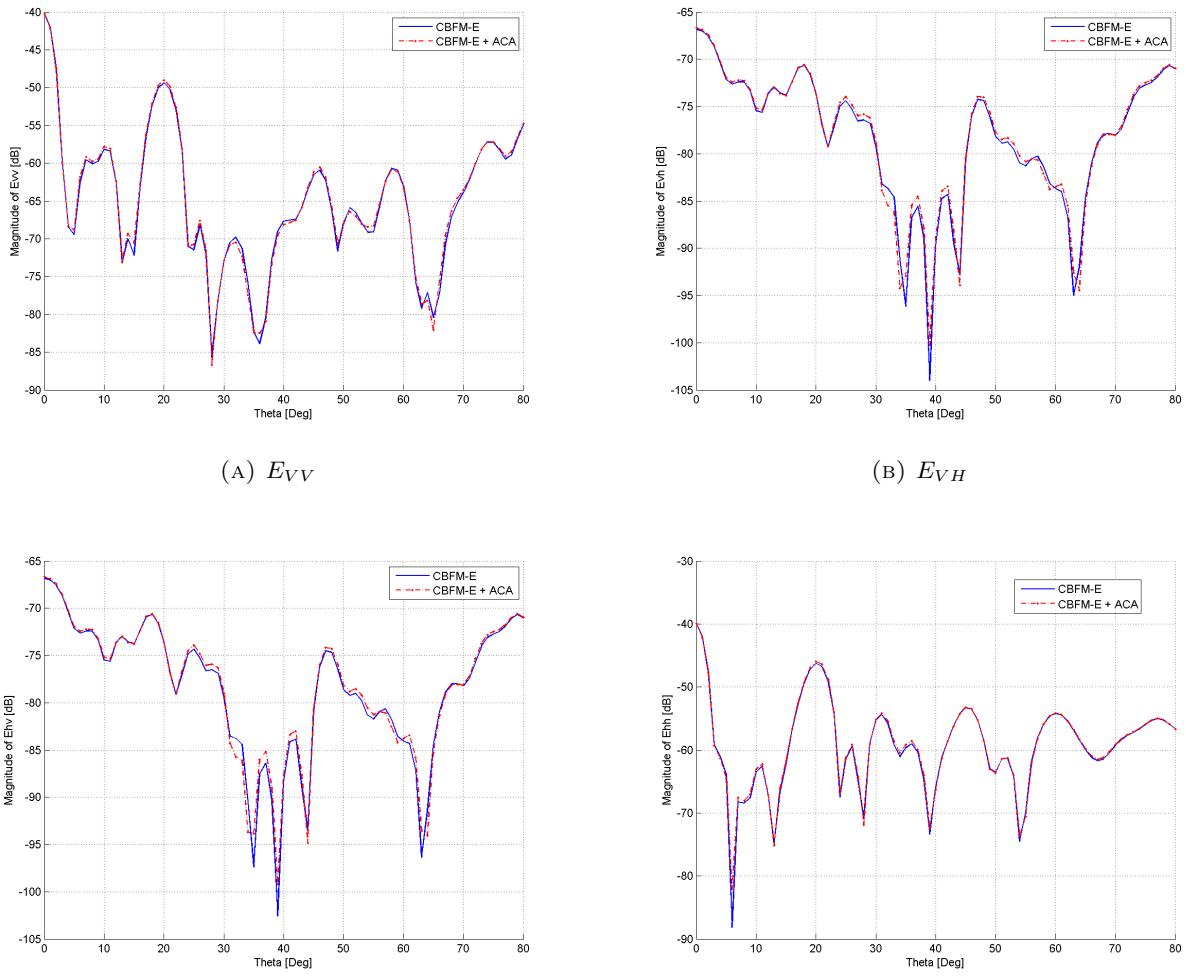


(A) $h_B = \lambda$ (B) $h_B = (3\lambda; 5\lambda; 6\lambda)$

FIGURE 4.33: Variation of the sub-matrix compression rate, $SCR^{i,j}_{ACA}$, achieved by the ACA for each sub-matrix $Z^{MoM}_{i,j}$ $(i \neq j)$. On the left, the three trees are divided into 120 blocks, and on the right, the same trees are now divided into only 28 larger blocks.

TABLE 4.15: The total gain in computing time G^{tot}_{ACA} achieved by the hybridization with the ACA, depending on the size of the CBFM blocks composing the same simulation scene.

h_B	λ	$[2\lambda; 3\lambda]$	$[3\lambda; 5\lambda; 6\lambda]$
G^{tot}_{ACA}	1.27	3.49	4.38



(A) E_{VV} (B) E_{VH}

(C) E_{HV} (D) E_{HH}

FIGURE 4.34: Variations of the magnitude the backscattered fields computed with the classical CBFM-E and the CBFM-E hybridized with the ACA for a mono-static configuration defined by $\theta_{i,s} = [0°; 1°; 80°]$ and $\phi_i = 45°$.

Hence increasing the size of the CBFM blocks engenders interestingly, in the same time, an increase in the compression rate CR achieved by the CBFM, and in $SCR^{i,j}_{ACA}$ achieved by the ACA. But we know, according to Chapter 3, that it also results in an additional cost in terms of CPU time, for the computing of Z^c (see Table 4.14). Hence, the hybridization of the CBFM with the ACA, combined to the use of large blocks, results in an enhancement of the performances of the CBFM in terms of memory storage without bearing the CPU related costs. Furthermore, the hybridization with the ACA does not affected the accuracy of the CBFM solution. Indeed, as can bee seen in Figure 4.34. Hence the

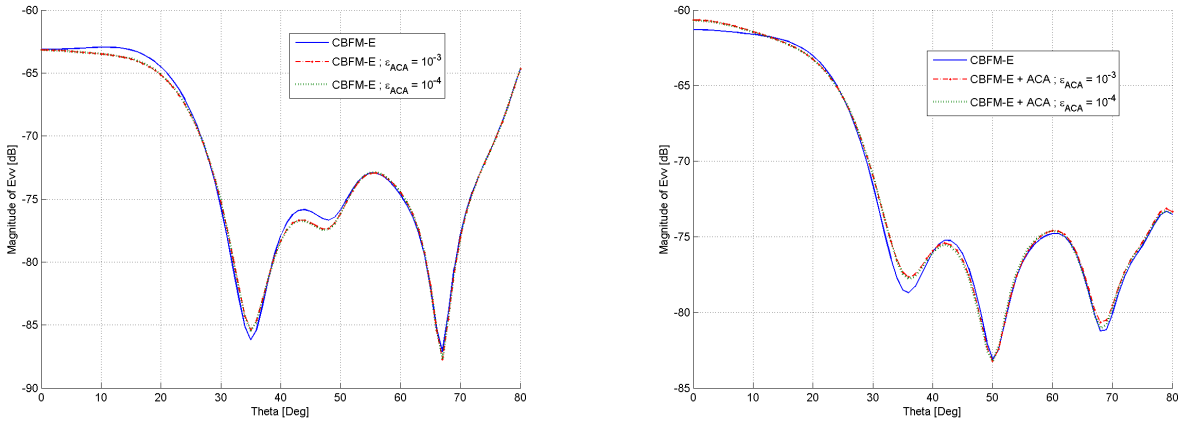
CBFM combined with the ACA yields accurate results with better performances than the conventional CBFM-E, both in terms of CPU time and memory use. Thus, the goal of the hybridization of the CBFM-E with the ACA has been achieved for this example.

Next, we investigate the behaviour of the ACA toward the heterogeneity of a natural forest simulation scene, when this iterative method is applied to approximate the sub-matrices $Z_{i,j}^{MoM}$ $(i \neq j)$. For this purpose, we apply the CBFM-E hybridized with the ACA to two cylinders of dimensions [$R_{s,1} = 0.24$ m; $h_{s,1} = 3.3$ m] and [$R_{s,2} = 0.18$ m; $h_{s,2} = 2.1$ m], while varying the real part of the dielectric constant of the second cylinder, such as we increase the difference in terms of ϵ' between these two cylinders. The simulations carried out are summarized in Table 4.16.

TABLE 4.16: The CBFM-E combined with the ACA is applied to two cylinders of dielectric constant $\epsilon_{s,1}$ and $\epsilon_{s,2}$. The two cylinders are separated by 2 m

Simulation	1	2	3	4	5
$\epsilon_{s,2}^{'}$	7.2	9.6	10.3	15	18.4
$\epsilon'_{s,2} - \epsilon'_{s,1}$	0	2.4	3.1	7.8	11.2

We display, in Figure 4.35, a comparison between the backscattered fields obtained by applying the conventional CBFM-E and those derived by the CBFM-E hybridized with the ACA for two values of ϵ_{ACA} (10^{-3} and 10^{-4}), for the simulations 1 and 2. Hence, we investigate the accuracy of the modified CBFM-E by comparing the results obtained with the maximum value of dielectric contrast ($\epsilon'_{s,2} - \epsilon'_{s,1} = 11.2$) to those obtained with a homogeneous simulation scene ($\epsilon'_{s,2} - \epsilon'_{s,1} = 0$).

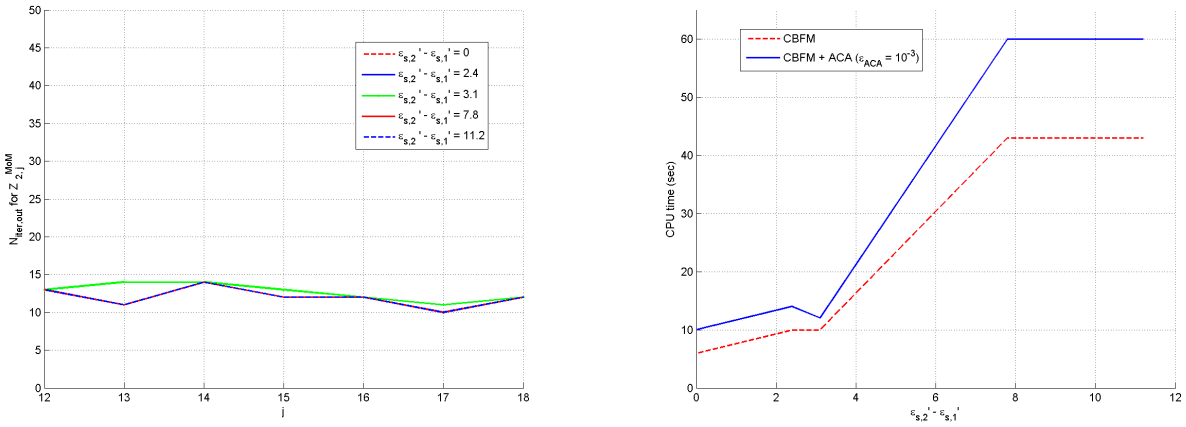


(A) $\epsilon'_{s,2} - \epsilon'_{s,1} = 0$ (B) $\epsilon'_{s,2} - \epsilon'_{s,1} = 11.2$

FIGURE 4.35: A comparison between the conventional CBFM-E and the CBFM-E hybridized with the ACA when applied to two homogeneous, then heterogeneous cylinders. The backscattered fields is presented for $\theta_{i,s} = [0°; 1°; 80°]$ and $\phi_i = 0°$.

The figure above shows that the heterogeneity of the simulation scene has not the slightest effect on the accuracy of the CBFM solution when combined with the ACA. The question, however, is whether

or not increasing $\epsilon'_{s,2} - \epsilon'_{s,1} = 0$ negatively affects the compression rate and the gain in CPU time achieved by the hybridization with the ACA. To answer this question, Figure 4.36a shows the sub-matrix compression rate, $SCR^{2,j}_{ACA}$, achieved by the ACA for each sub-matrix $Z^{MoM}_{2,j}$ $(12 \leqslant j \leqslant 18)$, depending on the dielectric constrast $\epsilon'_{s,2} - \epsilon'_{s,1}$. It shows is in a way the impact of the increase of the real part of the dielectric constant of the second tree, on the compression rate achieved by the ACA for the interactions of its blocks with the second block of the first tree, whose ϵ'_s is constant. On the other hand, Figure 4.38b illustrates the CPU time spent by the conventional and modified versions of the CBFM-E depending on the difference in terms of ϵ'_s between the two cylinders.

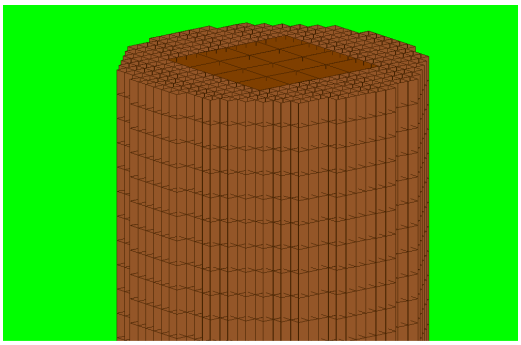


(A) $SCR^{2,j}_{ACA}$ (B) CPU time

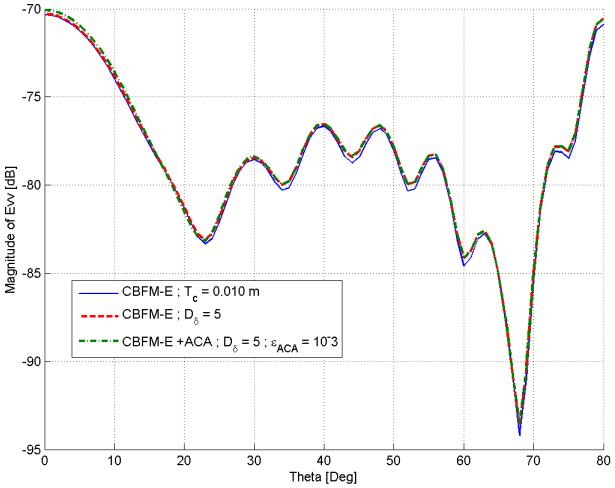
FIGURE 4.36: The performances of the CBFM-E combined with the ACA in terms of CPU time and sub-matrix compression rate depending on the value of $\epsilon'_{s,2} - \epsilon'_{s,1}$

First, we note that the heterogeneity of the simulation scene has not a significant impact on the sub-matrix compression rate. Then, even though it may appear that the increase in $\epsilon'_{s,2} - \epsilon'_{s,1}$ has a negative effect on the performances of the modified CBFM-E in terms of CPU time, it is not the case (see Figure 4.38b). Indeed, for this small example discretized into $Nb_c = 7960$ cells, and divided into blocks of a maximum of 520 cells, we can not expect a reduction of the CPU time by the ACA. We are interested, rather, in the variations of the ratio between the CPU time spent by the conventional CBFM and that of the modified version. This ratio still almost equal to the constant value of 1.4 despite the increase in $\epsilon'_{s,2} - \epsilon'_{s,1}$. Therefore, the hybridization with the ACA is apparently insensitive to the heterogeneity of a forest simulation scene both in terms of computing performances and accuracy.

Furthermore, it turns out that the accuracy of the CBFM-E combined with the ACA is not altered when the latter is applied on a non-uniform mesh inside the same scatterer. Indeed Figure 4.37 presents the backscattered fields by a single cylinder of dimensions $R_s = 0.24$ m; $h_s = 4.5$ m of dielectric permittivity $\epsilon_s = 7.6 + 14.8$ j. And, once again, the hybridization with the ACA enables us to reduce the CPU time spent while generating the final reduced matrix Z^c by a half without negatively impacting the accuracy of the CBFM-E solution. Indeed the conventional CBFM-E achieved the computation of Z^c in 3605 seconds while the modified version yield the same results in only 1722 sec.



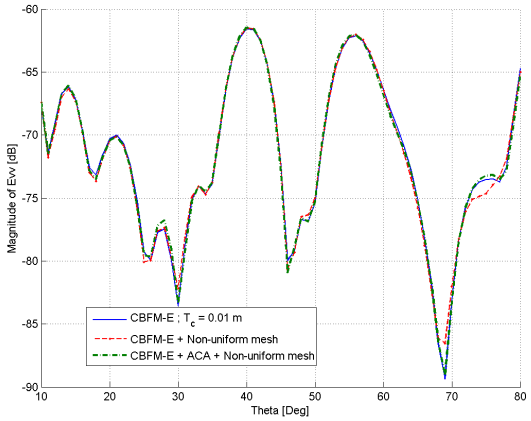
(A) Non-uniform mesh



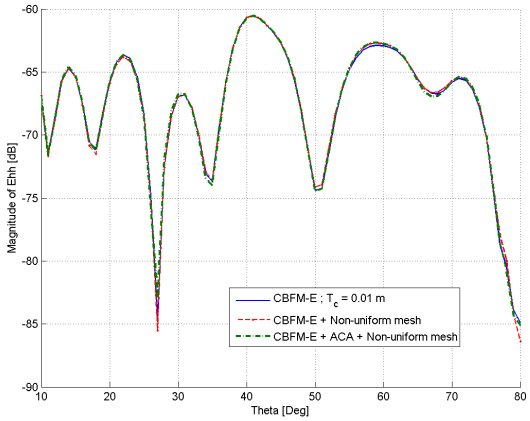
(B) E_{VV}

FIGURE 4.37: Application of the CBFM-E combined to the ACA on a non-uniform mesh inside the same scatterer. The magnitude of the backscattered field (polarization VV) corresponds to a mono-static configuration with $\theta_{i,s} = [0°; 1°; 80°]$ and $\phi_i = 0°$

Finally, Let us consider once again the example of heterogeneous forest simulation scene presented in Figure 4.18 and discretized using a non-uniform mesh into $Nb_c = 171340$ cells instead of $Nb_c = 565920$ cells in the case of a uniform mesh. We show, in Figure 4.38, the magnitude of the backscattered fields derived by the conventional CBFM-E and the CBFM-E combined with the ACA in comparison to the results obtained by the CBFM-E with the uniform mesh. The performances achieved by the both versions of the CBFM-E are illustrated in Table 4.17.



(A) E_{VV}



(B) E_{HH}

FIGURE 4.38: Variations of the magnitude of the backscattered fields (polarization VV and HH) computed by using the conventional and the modified versions of the CBFM-E on a uniform and a non-uniform mesh, for $\theta_{i,s} = [10°; 1°; 80°]$ and $\phi_i = 40°$.

TABLE 4.17: Performances of the three simulations described above in terms of memory storage required and computing time

	T_c size (m)	Z size ($3 \times Nb_c$)	CPU time
CBFM-E	0.01	1697760	45 h 15 min
CBFM-E	[0.01 ; 0.015 ; 0.03]	514020	4 h 12 min
CBFM-E + ACA		514020	1 h 26 min

The insensitivity of the ACA performances to the non-uniform mesh an to the heterogeneity of the forest scene allows us thus to benefit in the same time from the improvement in terms of CPU time and memory storage required brought by the non-uniform discretization of the initial simulation scene and from the gain in time achieved thanks to the hybridization of the CBFM-E with the ACA. And of the previous enhancement combine to optimize both the CPU time and the memory use performances of the conventional CBFM-E. In the next section, we focus on the enhancement of the CBFM-E code.

4.4 Acceleration approaches for the CBFM code proceeding:

In this subsection, we take an interest in the techniques used to accelerate the CBFM code without damaging the accuracy of the results. Indeed, once the CBFM-E algorithm was enhanced as much as possible using the approaches mentioned in the previous paragraphs (Sparse representation, non-uniform mesh, hybridization with the ACA), we focus on the improvement of the CBFM Fortran code itself. For this purpose, we first exploit the translation symmetry of the forest simulation scene by neglecting the effect of moving the CBFM blocks, along the axis $\vec{x}$, $\vec{y}$ and $\vec{z}$, on the characteristic basis functions generated at the beginning of the CBFM process. Next, we discuss the OpenMP and MPI parallel implementation of the CBFM-E code receptively on a shared and distributed memory workstations.

4.4.1 Neglecting the effect of the position of the CBFM block on its CBFs :

We noted, while starting the implementation of the CBFM-E, that, since two scatterers have the same dimensions and the same complex relative permittivity ϵ_s, even well-spaced, the application of the CBFM-E to them, results in the same number of CBFs for each two similar blocks ii and jj such as $Nb_{c,ii} = Nb_{c,jj}$. This observation is presented in Figure 4.39. It shows the normalized singular values obtained while applying the CBFM to the example described in table 4.18. As mentioned in this table, this simple example is composed of two cubic cylinders spaced by 2 meters along the axis $\vec{x}$. Each cylinder is divided, along the vertical axis $\vec{z}$, into 7 blocks.

This figure confirms that 2 identical CBFM blocks, in terms of size ($Nb_{c,i}$) and dielectric constant (ϵ_s) give both, after the singular value decomposition, the normalization and the application of the

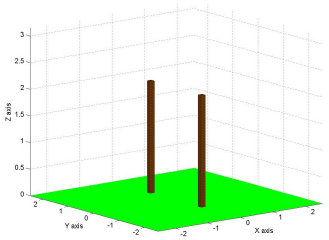
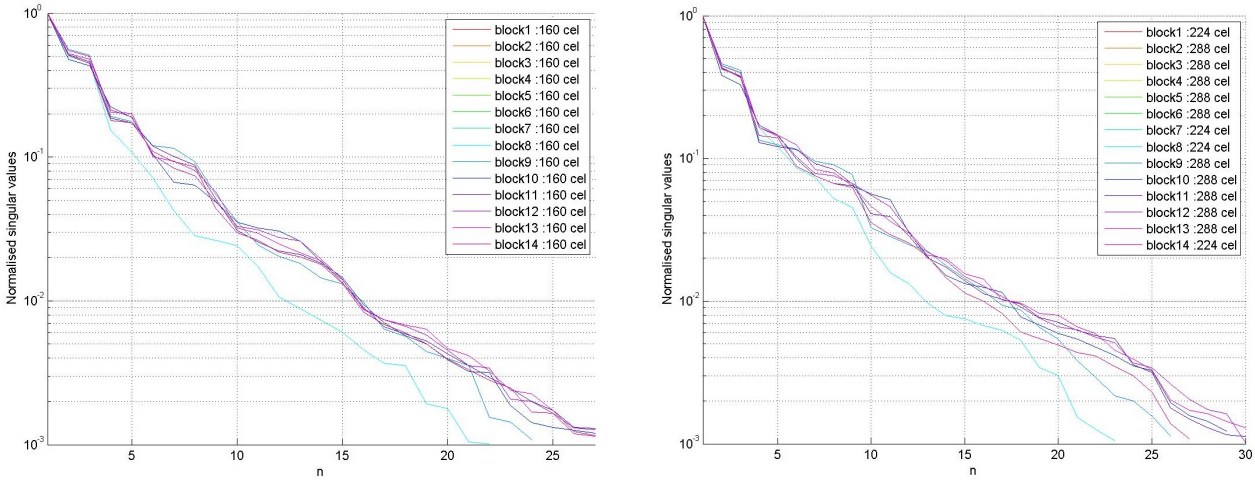
Simulation conditions		
	Frequency	$f = 2$ GHz
	Dimensions	0.12 m × 0.12 m × 2.1 m
	ϵ_s	9.6 + 0.01 j
	ϵ_g	5 + 2.1 j
	λ_s	0.323 m
	T_c	0.03 m
	Nb_c	2240
	N_B	14 (from bottom to top)
	N_{IPW}	182
	σ	10^-3

TABLE 4.18: The simulation conditions applied to 2 vertical cylinder placed over a real ground

threshold, the same number of CBFs upon condition that they have been extended by the same way while applying the CBFM-E. This observation stimulates an interest on the effect of the spatial shifting along $\vec{x}$, $\vec{y}$ and $\vec{z}$ on the fields inside the scatterers and consequently the scattered fields. We began then by studying the effect of a phase shifting by a constant angle Φ ($\exp^{-i\Phi}$) on the singular value decomposition (SVD) operation (see equation 2.8). We notice that the matrix S is always invariant and the vectors U and V are invariant or shifted in phase by the same angle Φ depending on the size of the matrix subjected to the SVD.



(A) CBFM-N (B) CBFM-E

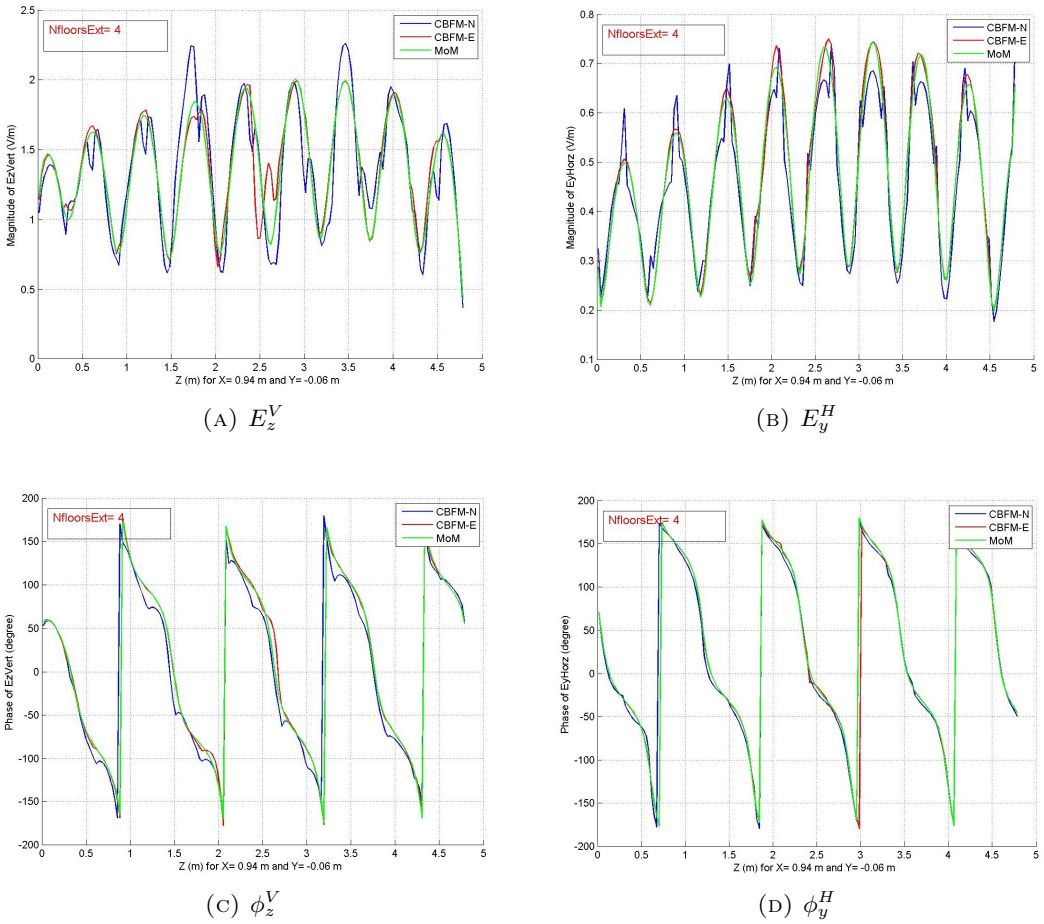
FIGURE 4.39: *The normalized singular values obtained with CBFM-N and CBFM-E using a threshold equal to* 10^{-3}

Consequently, we expect that, once the CBFs are computed for one single scatterer, they can be used for other identical scatterers, even if they are far away from the initial one (for which we have computed the CBFs). Copying the CBFs computed for previous similar blocks, instead of computing it again, will enable us to improve the performances of the CBFM in terms of computing time. On the

other hand, we hope that this technique will not impact negatively the accuracy of the results obtained with the CBFM solution.

Therefore, in order to check the validity of this approach, we apply the CBFM-E to two identical scatterers spaced by a distance Δ. Initially, Δ is equal to 2 m along the $\vec{x}$ axis. The magnitude and phase of the components E_z^V and E_y^H of the total electric field inside the first dielectric cylinder obtained for the incidence direction ($\theta_i = 30°$; $\varphi_i = 0°$) by the CBFM-N and the CBFM-E compared to the results given by the MoM are plotted in Figure 4.40. We display also the backscattered fields for a mono-static configuration defined by $\theta_i = \theta_s = [10°; 1°; 80°]$ and $\varphi_i = \varphi_s = 0°$ in Figure 4.41.

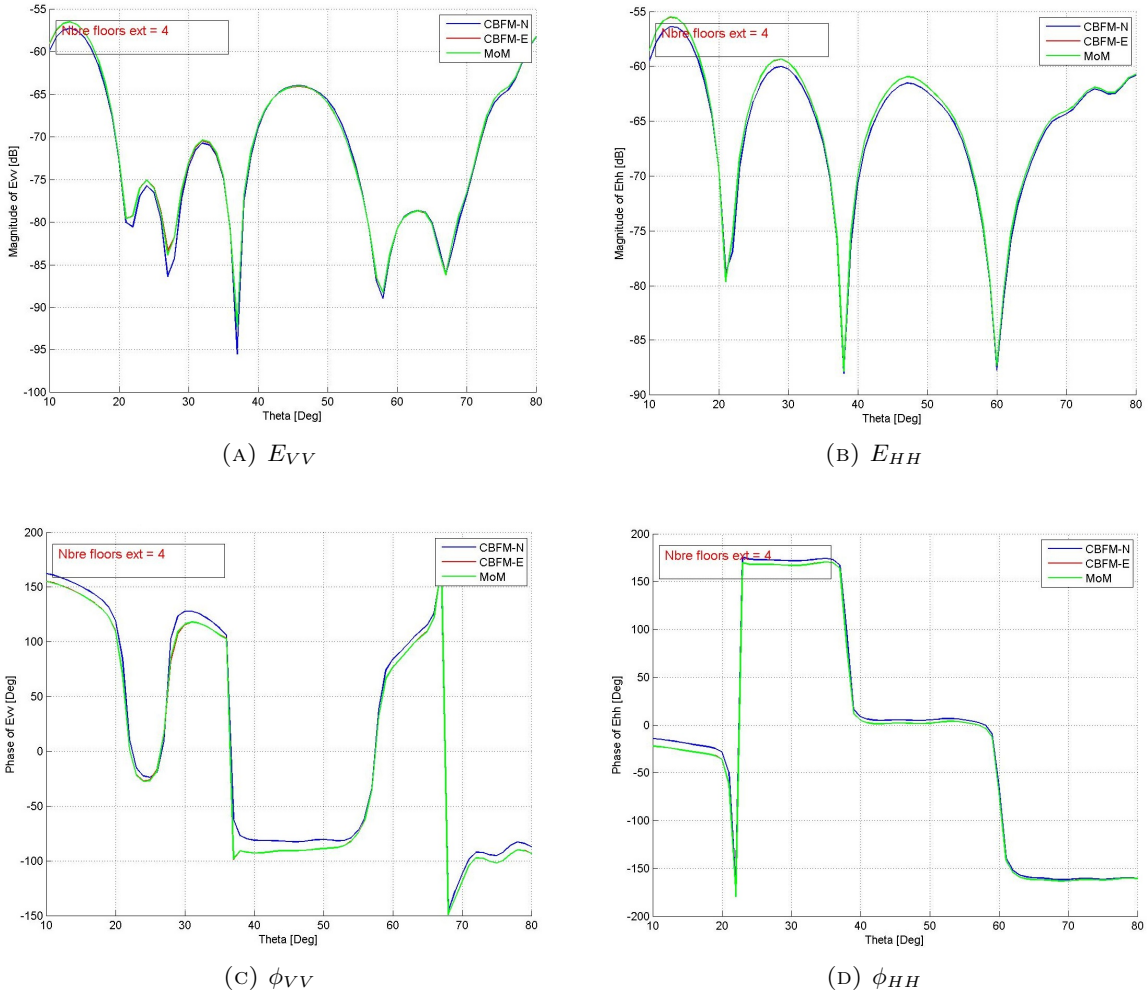
We notice that, with Δ equal to 2 m, using the CBFs generated for the first cylinder also for the second one, have not any influence on the accuracy of the results obtained with the CBFM-E.



(A) E_z^V (B) E_y^H

(C) ϕ_z^V (D) ϕ_y^H

FIGURE 4.40: *Variations of the magnitudes and phases E_z^V, E_y^H, ϕ_z^V and ϕ_y^H of the total electric field inside the first cylinder computed with the CBFM-N, the CBFM-E and the MoM for the incidence direction ($\theta_i = 30°$;$\varphi_i = 0°$).*



(A) E_{VV} (B) E_{HH}

(C) ϕ_{VV} (D) ϕ_{HH}

FIGURE 4.41: *Variations of the magnitude and phase of the backscattered fields, computed with the CBFM-N, the CBFM-E and the MoM with a mono-static configuration defined by* $\theta_i = \theta_s = [10°; 1°; 80°]$ *and* $\varphi_i = \varphi_s = 0°$

On the other side, as shows the table 4.19, this approach allows us to achieve, with this example, a slight computing time gain.

	CBFM-N	CBFM-E
Classical Version	2 *min* 11 *sec*	5 *min* 21 *sec*
New Version	1 *min* 29 *sec*	3 *min* 7 *sec*

TABLE 4.19: The computing time for the CBFM-N and the CBFM-E with and without neglecting the effect of the shifting along X, Y and Z on the CBFs (respectively the new version and the classical version)

Since the parallelism is implemented, for this version, only inside each cylinder, this gain depends on the degree of parallelism inside each scatterer and so, on the number of blocks by cylinder. The gain depends also on the size of the reduced matrix computed after CBFs generation. Indeed, increasing the number of blocks by cylinder increases consequently the degree of parallelism for each cylinder

which allows an optimal use of the 8 processors of my workstation but unfortunately it decreases in the same time the compression rate and so increases the size of the reduced matrix and consequently the computation time related to the resolution of the reduced equation system. In the next section, we detail the OpenMp and MPI parallelization of the CBFM code.

4.4.2 Parallelization of the CBFM code :

4.4.2.1 The Open MP parallelization of the CBFM

In order to accelerate the CBFM, we take advantage of the fact that the CBFs for each block are computed independently, as shown in Figure 4.42. Thus, the CBFM process is highly amenable to parallelization.

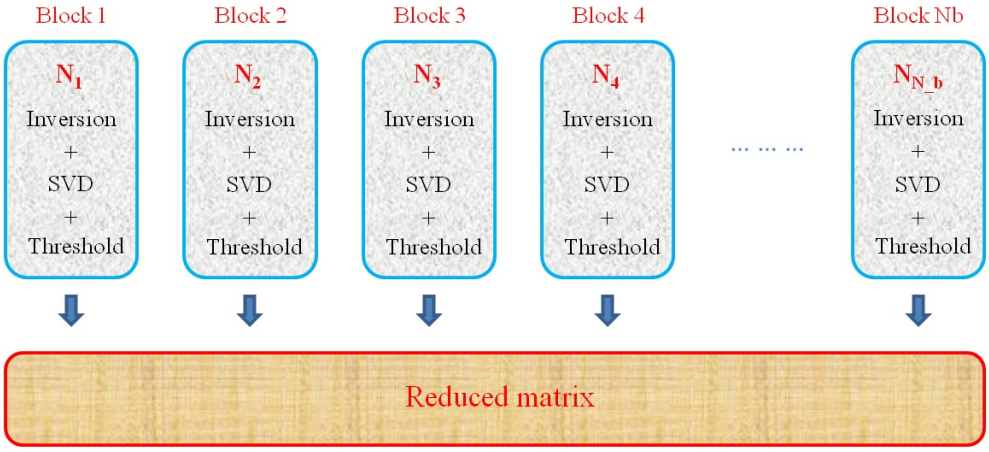
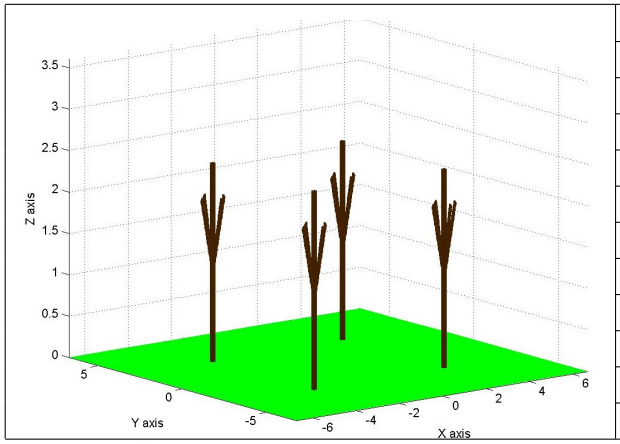


FIGURE 4.42: *The CBFs are computed separately for each block before generating the final reduced matrix.*

Toward this end, we implement, initially, openMP directives in a Fortran program running on a shared memory workstation equipped with an intel Xeon x5560 (8 tasks) at 2.8 GHz and 48 GB of RAM. The parallel implementation affects the creation of the CBFs for each block but also the generation of the reduced matrix and the reduced vector V_c representing the right hand-side term of the equation (2.11) and the computation of the final total field inside the scatterer for the different incident waves. In practical terms, the openMP directives are introduced since the code presents an iteration statement (a *for* loop) which is, fortunately, quite abundant in the CBFM code. To check the effect of this modification on the computing time, we apply the sequential and the parallel versions of the CBFM to the simulation example described by table 4.20.

Even with this quite small example ($Nb_c = 7040$ cells), we notice a significant difference in terms of computing time between the sequential and parallel versions. To confirm this observation, we compare the performances of the two versions while increasing the size of the scatterer under simulation. With a constant block height value (10 floors along the vertical axis $\vec{z}$), we increase consequently the total

number of blocks. The experiences are done with a single scatterer whose rectangular base dimensions are 0.18 m × 0.18 m with a complex relative permittivity of $\epsilon_s = 9.6 + 0.01\text{j}$ placed over a dielectric ground of complex relative permittivity $\epsilon_g = 5 + 2.1\text{j}$. The simulation conditions are listed in table 4.22. For each experience, corresponding to various heights h of the scatterer, we compute the ratio between the computing time with the sequential version and the one with the parallel version for the CBFM-N and the CBFM-E. The results are shown in Figure 4.43.



f	300 MHz
Dimensions	4 × [0.12 m × 0.12 m × 2.4 m]
ϵ_s	9.6 + 0.01 j
ϵ_g	5 + 2.1 j
λ_s	0.323 m
T_c	0.03 m
Nb_c	7040
N_B	80
N_{floors}	10
$Nber_{ext,floors}$	4
N_{IPW}	182
σ	10^{-3}

TABLE 4.20: The simulation conditions applied to 4 scatterers placed over a real ground

The results are provided in table 4.21.

	CBFM-N	CBFM-E	MoM
Sequential	3 *min* 29 *sec*	7 *min* 34 *sec*	46 *min* 47 *sec*
Parallel	1 *min* 33 *sec*	3 *min* 3 *sec*	

TABLE 4.21: The computing time with CBFM-N, CBFM-E and MoM with the sequential and parallel codes

Hence, the OpenMP parallelization of the CBFM-E code enables us to achieve a significant gain in computing time. Our CBFM code nevertheless remains limited in terms of numerical size of simulated forest simulation scene, essentially because of its implementation on a shared memory workstation. In the next paragraph, we move to a distributed memory configuration, and we discuss then the MPI parallelization of our CBFM code.

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7	Exp8	Exp9	Exp10	Exp11
h (m)	2.4	3.9	4.5	5.1	5.7	6.9	7.5	8.4	9.6	10.5	12
N_B	8	13	15	17	19	23	25	28	32	35	40
Nb_c	2880	4680	5400	6120	6840	8280	9000	10080	11520	12600	14400

TABLE 4.22: The simulation conditions while increasing the height h of the scatterer under simulation.

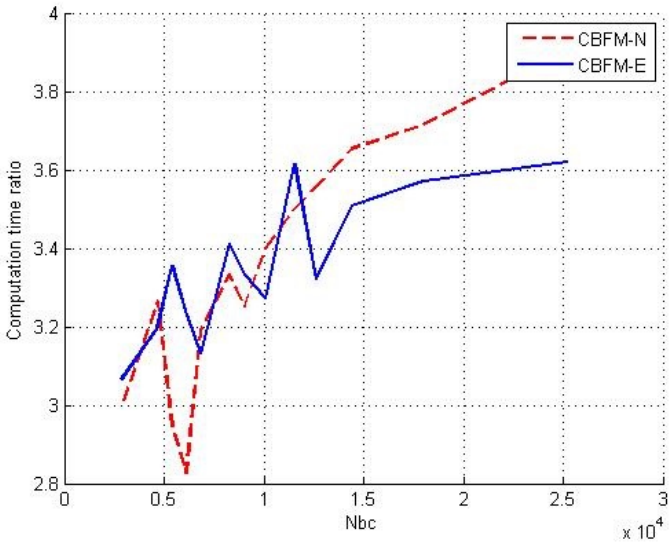


FIGURE 4.43: *The computation time ratio between the sequential and the parallel versions for the CBFM-N and the CBFM-E depending on the size of the scatterer.*

4.4.2.2 The MPI parallelization of the CBFM when applied to the 3D forest scattering modeling

Since the MPI parallelization is intended to have a significant effect on the performances of our CBFM code both in terms of CPU time and numerical size of problems under consideration, we present here our motivations for this technique, and then we detail the MPI implementation for each step of the CBFM code. Finally, we will end this subsection with some primary numerical results.

- Motivations It is first necessary to mention that the main reason we came up with this issue is the limitation of the multilevel CBFM-E when applied on a shared memory workstation. In order to understand this limitation, let us recall that the multilevel scheme of the CBFM-E consists on a recursive application of the monolevel CBFM in order to reduce as much as possible the size of the final matrix, before inverting it by using the conventional LU decomposition. The MLCBFM-E is thus assumed to be more efficient in terms of compression rate CR, and hence applicable to larger simulation scenes for higher frequencies. As explained in chapter 3, theoretically, this is exactly the main advantage of the MLCBFM-E compared to the mono-level CBFM-E.

Unfortunately, this theory clashes with the technical reality when the MLCBFM-E is implemented on a shared memory workstation. As shown in Figure 4.44, even if the MLCBFM-E yield a better compression rate after further iterations, the first level reduced matrix must be saved. In fact, this matrix will serves as initial 'MoM matrix' for the next application of the CBFM-E, and hence must be fully saved and available for the different CPUs involved in the computation of the CBFs for the next level.

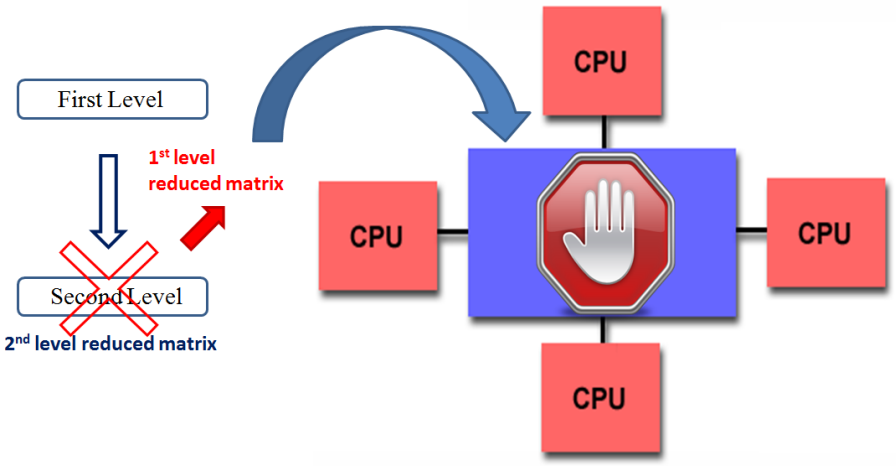
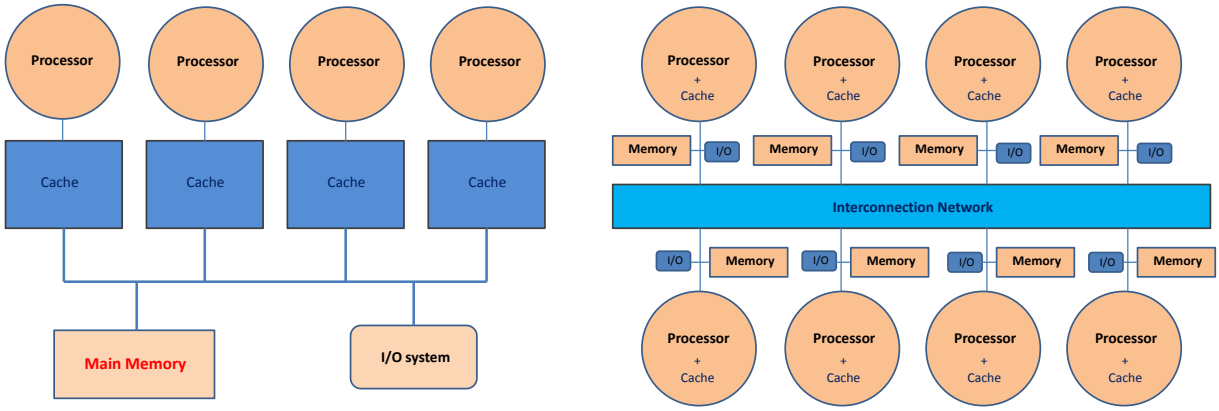


FIGURE 4.44: On a shared memory work station, even with the MLCBFM-E, the first level reduced matrix ($CR = CR_{mono-levelCBFM}$) must be saved. Thus, the MLCBFM-E is stopped as soon as this size exceeds the available memory resources.

Consequently, the MLCBFM-E, as the mono-level CBFM is stopped as soon as the size of the its first level reduced matrix exceeds the available memory resources. Therefore, the MLCBFM-E loses definitely its principal advantage over the classical CBFM-E, even if it maintains a superiority in terms of CPU time when both the CBFM-E and MLCBFM-E overcome the barrier of the memory space.

Obviously, this technical problem will be solved when the MLCBFM-E code is implemented on a distributed memory architecture, since none of the involved processors has to save the entire reduced matrix in its own private memory (see Figure 4.45).



(A) Shared memory (B) Distributed memory

FIGURE 4.45: The shared-memory and distributed-memory architecture.

Moreover, it is important to recall that the distributed-memory architecture itself presents important and relevant advantages from the point of view of memory resources and computing performances. Indeed, with a distributed-memory architecture, memory is directly proportional to the number of

processors. Hence, increasing the available memory resources requires simply an equivalent increase of the number of processors, and the finance cost, related to the memory, increases linearly with the needs, which is not the case with the shared-memory workstations. Furthermore, if the code under consideration is highly amenable to parallelization, its implementation on a distributed-memory architecture yields a significant reduction in the computing time, since each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain global cache coherency [63].

These facts, combined with the proven efficiency of the Message Passing Interface (MPI) [64] when used to parallelize the CBFM [40, 65], prompted us to MPI parallelize our CBFM-E code, once again, in order to improve its performances both in terms of CPU time and of management of the available memory resources [66, 67].

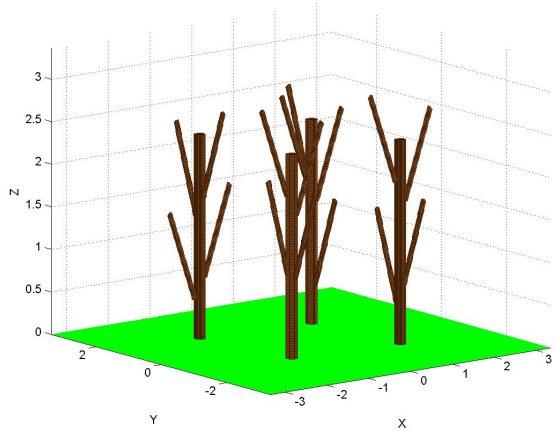


FIGURE 4.46: 20 dielectric cylinders modeling 4 trees with 4 branches each.

For a sake of clarity, let us first of all, consider the application of the CBFM-E to the simple example of an heterogeneous forest simulation scene, using 4 processors. As shown in Figure 4.46, the simulation scene is composed of 4 identical trees with 4 branches each, modeled by 20 dielectric cylinders. The dielectric constant of these trees ranges from $\epsilon_s = 4.2 + 0.1\text{j}$ and $\epsilon_s = 8.2 + 0.9\text{j}$. Each tree trunk is divided into 8 blocks of height $h_B = \lambda$, and each branch is divided into 2 blocks of height $h_B = 2\lambda$. Therefore, the entire simulation scene is divided into $N_b = 64$ CBFM blocks. The following provides a description of the MPI parallelization of each step of the mono-level CBFM procedure :

- Partitioning of the forest simulation scene : The parallelization of the CBFM code begins with an equitable distribution of the CBFM blocks such as each processor receive more or less the same amount of computing work. In fact, in order to ensure an optimal functioning of the parallel code, no processor shall have to wait for the others. The simple example of distribution of the 64 CBFM blocks between the 4 involved processors is shown in Figure 4.47. It should be recalled that, thanks to the translation symmetry mentioned at the beginning of the current subsection, the CBFs are only

computed for 3 blocks for each tree trunk. The computed blocks are, from the bottom to the top, the first, the second and the last of each tree trunk.

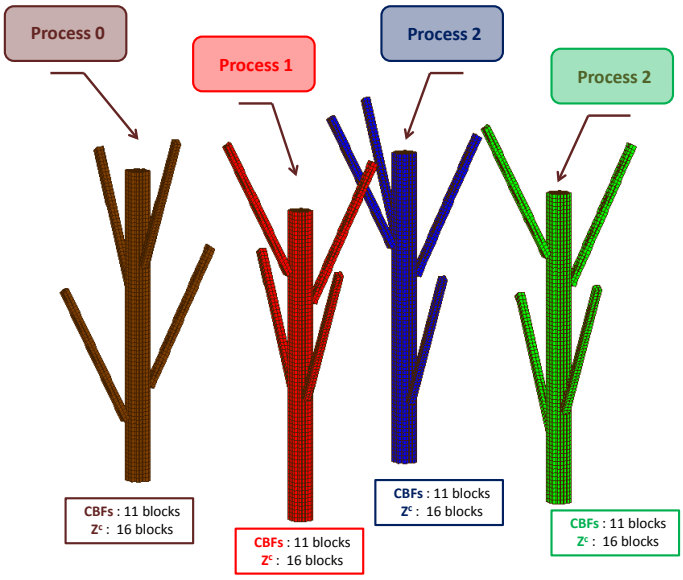


FIGURE 4.47: Example of a simple partitioning of the CBFM blocks between 4 Processes.

In a broader case, the challenge lies in how to allocate the entire N_b blocks to the processors, for a random and heterogeneous simulation scene, while taking into account the non-uniformity of the size of the CBFM blocks and bearing in mind the symmetries exploited to reduce the CPU time while generating the CBFs. We also need to think about the minimization of the time of communication between the processes while generating the reduced matrix Z^c. All these elements have been verified and taken into account while implementing the two partitioning algorithm 2 and 3 given below. The first one deals with the distribution of the total N_b blocks between the available processors and the second concerns only the allocation of the N_{nb} blocks, for which the CBFs are computed,where nb refers to new blocks. Once the simulation scene is distributed equally between the processes, we move to the next step which is the generation of the CBFs.

- Generation of the CBFs As explained above, since the CBFs for each block are computed independently, the CBFM algorithm is highly suitable to parallelization at this level. Hence, this step is carried out fully in parallel without any intercommunication between the N processors.

It is worth mentioning that the version of the CBFM-E, used in this work, and tailored to solve scattering problems, is more amenable to parallelization while generating the CBFs than that used with RFIC-Types [27]. Indeed, generating the "secondary" CBFs with the second version of the CBFM-E requires intercommunication between the processors [68].

- Construction of the reduced matrix Z^c For the second stage of the mono-level CBFM-E procedure, each process computes and stores only a segment of the final reduced matrix. To do it, it

Distribution of all the CBFM blocks between the N available processors:

1)Initialize the cursor K and the number of cells per process $Nb_{c,proc}$ to 0, and the number of the process n_{proc} to 1.
2)Initialize *permut* to 0
3)Compute the average number of cells per process as $Nb_{c,av} =$ ceiling(Nb_c/N)
while $K \leqslant N_b$ **do**
  1) Update $Nb_{c,proc}$ by adding the number of cells of the current block K :
  $Nb_{c,proc} = Nb_{c,proc} + Nb_{c,K}$
  if $Nb_{c,proc} \leqslant Nb_{c,av}$ **then**
    Allocate the block K to the current process n_{proc} and move to the next block $(K = K + 1)$.
  else
    Compute the excess as $Excess = Nb_{c,proc} - Nb_{c,av}$
    if $(Excess \leqslant 0.6 \times Nb_{c,K})$ *.and.* $(permut == 0))$ **then**
      1)Put *permut* to 1
      2)Allocate the block K to the current process n_{proc} and move to the next block $(K = K + 1)$ and to the next process $(n_{proc} = n_{proc} + 1)$
    else
      1)Move to the next process $(n_{proc} = n_{proc} + 1)$ and allocate the block K to this new process, then move to the next block $(K = K + 1)$.
      2)Do not forget to put *permut* back to 0.
    end
  end
end

Algorithm 2: Distribution of the N_b blocks between the N available processes.

Distribution of the CBFM blocks for which the CBFs are computed (The new blocks)

% Iterating on n_{proc}, the number of the process
while $n_{proc} \leqslant N$ **do**
  % Explore the $N_{b,proc}$ CBFM blocks allocated to process n_{proc} by the Algorithm 1.
  while $K \leqslant N$ **do**
    if K *is a new block* **then**
      1) Add the new block K to the new blocks which will be handled by the current process n_{proc} and move to the next block $(K = K + 1)$.
      2) Update the number of the latest block tested and added to n_{proc} a new block : $K_{last,n_{proc}} = K$.
    end
  end
  Check if the last new block $K_{last,n_{proc}}$ allocated to n_{proc} is identical to some of the following blocks allocated by the Algorithm 1 to the next processor $n_{proc} + 1$. If it is the case, allocate also $K_{last,n_{proc}}$ as a new block to the processor $n_{proc} + 1$ in order to avoid a supplementary communication burden.
end

Algorithm 3: Distribution of the N_{nb} new blocks between the N available processes

uses the CBFs computed by itself for its own CBFM blocks, and then communicates with the other processes in order to retrieve the CBFs of their CBFM blocks. The communications established between the processes in order to generate the reduced matrix are illustrated in Figure 4.48 for an example of 15 CBFM blocks distributed between 3 processors. This simple example corresponds to 2 different tree trunks divided into 8 and 7 blocks.

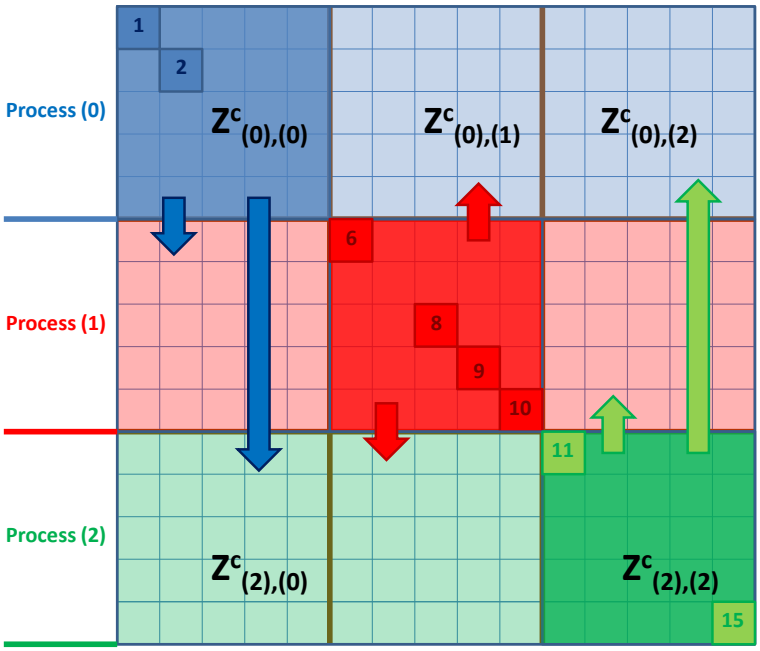


FIGURE 4.48: Generation of the reduced matrix Z^c whith a simulation scene divided into 15 CBFM blocks distributed between 3 processors. $Z^c_{(i),(j)}$ is the section of Z^c which is computed by using the CBFs computed by processors i and j.

Note that the highlighted blocks (1,2,6,8,9,10,11, and 15) are the new blocks, which means the blocks for which the CBFs are computed and stored by each of the 3 processors. For instance, process (0) has computed the CBFs of the blocks 1 and 2 (the two first blocks of the first tree trunk), hence it calculates independently the section of Z^c representing the interactions between the first 5 blocks. But, the processor (0) needs to communicate with process (1) and process (2) when it comes to computing the section of Z^c representing the interactions between its blocks $(1, 2, 3, 4, 5)$ and the rest of the simulation scene.

In order to reduce as far as possible the time spent on communications, the transfer of the CBFs between the processors is performed by moving them sequentially and cyclically from the processor (p) to the next processor $(p+1)$, as detailed in Algorithm 4. Thus, the matrix $CBF_k^{(p)}$ includes the CBFs available for the processor (p) in the iteration k. For example, with a total number of 5 processors, as shown in Figure 4.49, we have $CBF_1^{(0)} = CBF_3^{(2)}$ and $CBF_2^{(4)} = CBF_5^{(2)}$. Hence, as shown in Algorithm 4, for each iteration k, each processor uses the available matrix of CBFs to compute a section of Z^c, then sends it to the next processor and receives that of the previous one.

Initialization (k=0):
1) Each processor p uses its own matrix of CBFs to compute $Z^c_{(p),(p)}$. We recall that $Z^c_{(i),(j)}$ is the section of Z^c which is computed by using the CBFs computed by processors i and j.
2) Identify the previous processor $p_{prev} = (N + p - 1)\%N$.
3) Identify the following processor $p_{foll} = (p + 1)\%N$.

Iterations:
while $k \leqslant N$ **do**

% Iteration k

1) Each process send the matrix $CBF^{(p)}_{k-1}$ to p_{prev} and receive $CBF^{(p_{foll})}_{k-1}$, which is then equally $CBF^{(p)}_k$.

2) Process p uses $CBFs^{(p)}_k$ to compute the section $Z^c_{(p),(p_{org})}$ of Z^c, where $p_{org} = (p + k)\%N$ is the processor which computed originally the CBFs included in the current $CBF^{(p)}_k$.

end

Algorithm 4: Transfer of the CBFs between the N processors while computing the matrix Z^c. Note that % refers remainder of the division.

It is worth noting that this technique is commonly used to increase the efficiency of the MPI parallel implementation of matrix multiplication [69].

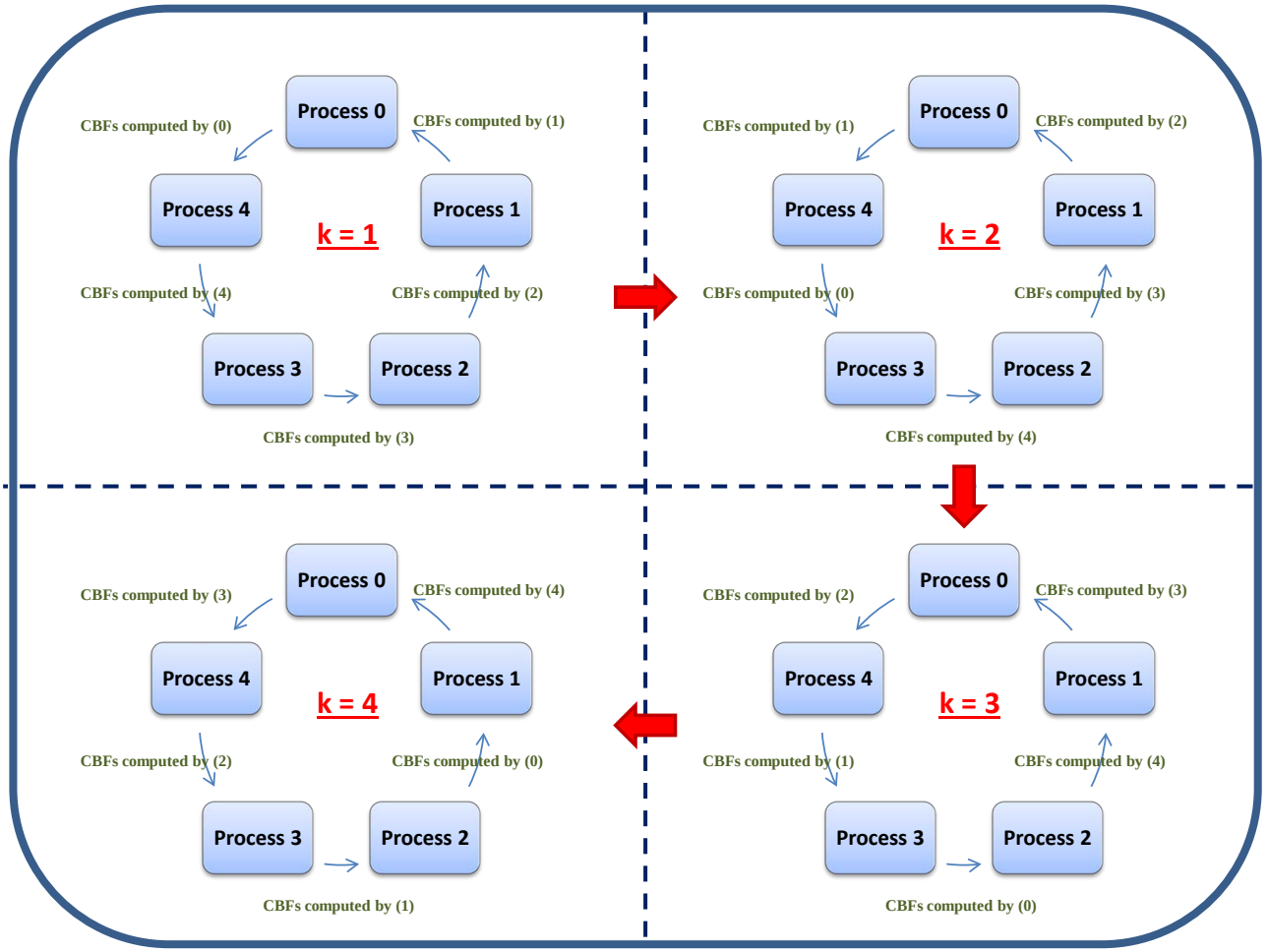


FIGURE 4.49: Example with 5 processors : The matrices of CBFs are cyclically moved from a processor to an another until finiching the construction the entire final reduced matrix Z^c

The communications between processors while constructing the reduced matrix Z^c are achieved through Point-to-Point Communication Routines such as 'MPI_Send' and 'MPI_Recv' and 'MPI_Sendrecv' and collective Communication routines such as 'MPI_Reduce'. At the end of this stage, each processor has a section of the final reduced matrix Z^c of size $3Nb_{c,proc} \times 3Nb_c$.

- Solving the final reduced matrix Z^c Finally, we solve the final system of linear equations corresponding to Z^c by using ScaLAPACK (Scalable Linear Algebra PACKage) subroutines [70]. For this purpose, the entire matrix Z^c is reordered in the cyclic format of ScaLAPACK as shown in Figure 4.50. This involves further communications between the processors in order to redistribute the reduced matrix Z^c.

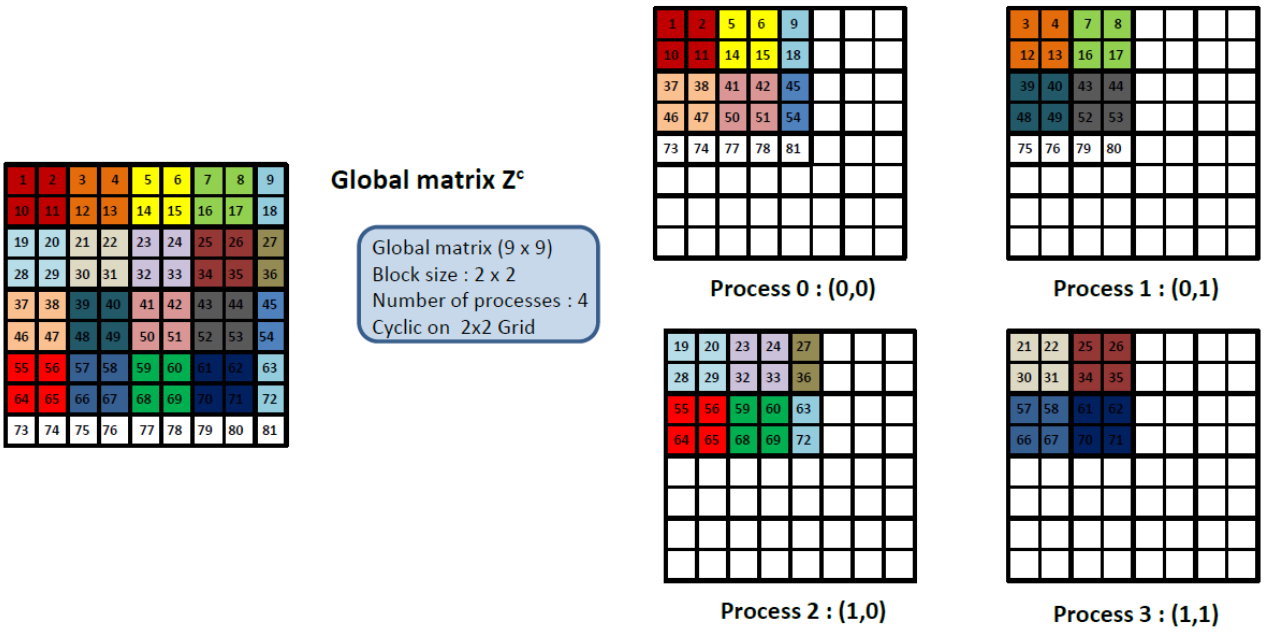


FIGURE 4.50: Example of a 9×9 matrix distributed between 4 processors in the cyclic format of ScaLAPACK.

4.4.2.3 Numerical results

It is interesting to recall that most of the simulations presented in this dissertation are carried out with an OpenMP parallelized FORTRAN code on a shared-memory workstation equipped with 8 processors and 42 GB of RAM. In order to demonstrate the important contribution of the OpenMP parallelization in the gain in CPU time achieved by the CBFM-E code, we go back once again to the forest simulation scene illustrated in Figure 4.18 and we conduct again the simulation described in Table 4.17 but without using the OpenMp subroutines. Hence, the new simulation is carried out on a single processor. The resulting CPU times corresponding to the generation of the CBFs, the computation of Z^c and the resolution of the associated system of linear equations are summarized in Table 4.23.

TABLE 4.23: Performances of the CBFM-E code in terms of CPU time with and without OpenMP parallelization

	Z size ($3 \times Nb_c$)	CBFs (min)	Z^c (min)	$(Z^c)^{-1}$ (min)
CBFM-E+ OpenMP	514020	44	23	9
CBFM-E		152	86	9

Hence the implementation of an OpenMP parallelized version of the CBFM-E enables us clearly to take advantage of the 8 available processors and to reduce thus significantly the CPU time spent by the CBFM-E to compute the backscattered fields.

Concerning the MPI parallelized code, we have not achieved yet the desired and expected performances because of some technical problems and of punctual unavailability of the distributed-memory workstation. It is worth to recall that the main objective of this parallelization was to increase the available memory storage of up to 80 nodes with 24 GB of RAM available on each node.

4.5 Conclusion

The CBFM-E solution showed in this Chapter a high adaptability to the different approaches introduced to enhance its performances both in terms of CPU time and memory storage required. This flexibility enables us to take advantage of the heterogeneity of a natural forest simulation scene in terms of dielectric properties through the implementation of a non-uniform mesh, and to benefit from the rank deficiency of the MoM sub-matrices that represent interactions between well-spaced blocks by hybridizing the CBFM-E with the ACA, thus adding the advantages of the CBFM-E to those of the powerful iterative method.

The nicest and concisely way to end this chapter consists on the following : Before Chapter 4, the computation of the backscattered fields by the large forest simulation scene in Figure 4.18 takes about 45 hours to achieve a compression rate CR of 163 (We keep 0.6 % of the initial matrix) and to yield accurate results comparing to the conventional MoM. After Chapter 4, the optimized CBFM-E achieves the same level of accuracy and a higher compression rate $CR \approx 195$ ($ICR = 0.5$ %) in only 46 minutes.

Chapter 5

Performance Comparison of the CBFM to Iterative Methods

We have demonstrated the absolute superiority of the CBFM-E over the conventional Method of Moments (MoM) both in terms of CPU time and memory storage required, while yielding the same level of accuracy. Indeed, the CBFM-E is considered as one of the most powerful available numerical methods to solve large electromagnetic simulation scenes. Being a direct method, its main competitors are iterative methods such as the Fast Multipole Method (FMM or FMA) and the Adaptive Cross Approximation Algorithm (ACA). As explained in the introduction of this report, a great deal has been written about the comparison between the direct and iterative methods in terms of computing time, memory requirement and accuracy. In this chapter, we investigate this issue in our case of application and we attempt to provide an element of an answer to the question : Direct or iterative ?, at least for our 3D dielectric forest scattering model. Hence, we compare the performances of the CBFM-E to those achieved by the ACA when applied to 3D large forest simulation scenes.

5.1 The Adaptive Cross Approximation Algorithm (ACA)

In this section, we apply the Adaptive Cross-Approximation (ACA) Algorithm, preceded by a hierarchical repartition of the elements of the initial full MoM matrix. The purpose is to provide a comparison between the performances of the CBFM-E and those achieved by the ACA in terms of accuracy, computing time and memory use, when applied in the context of our 3D dielectric forest scattering model. This comparison has been carried out through a cooperation with Dr. Xavier Juvigny from the ONERA (French Aerospace Lab). We thus could compare the two methods when applied to large dielectric forest simulation scenes.

5.1.1 H-Matrix representation of the MoM matrix and ACA

As mentioned earlier, The initial MoM matrix, representing remote interactions between the Nc_c cells composing the 3D forest scene, is a dense matrix, as a result of the volumetric integral representation of the electric field inside the scatterers. However, the off-diagonal sub-blocks, which describe the interactions between well-separated original low-basis functions, are numerically rank-deficient and hence can be approximated by much lower-ranks matrices. This is where the ACA algorithm comes in, since it allows to approximate those rank-deficient MoM matrix blocks by a much-reduced set of column vectors. The question now arises of what is then the difference between the application of this ACA in this chapter, and its use in the previous one. Indeed, the ACA was used previously in the context of the CBFM, only to speed up the computing of Z^c by approximating the sub-matrices $Z_{i,j}^{MoM}$ where i and j are two CBFM blocks. Here, though, the ACA is applied on the basis of hierarchical repartition of the initial MoM matrix in order to derive a final sparse compressed matrix. The latter will be finally solved using an iterative method such as the conjugate gradient method (CG) or the generalized minimal residual method (GMRES). Hence, the ACA here does not intervene only in the reduction of the CPU time but its major contribution is the compression rate achieved on the initial MoM matrix.

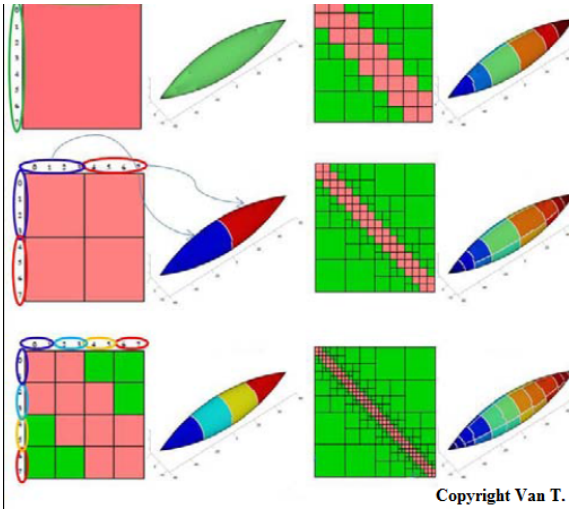


FIGURE 5.1: Hierarchical partitioning and the structure of the resulting H-matrix. Green sub-blocks are admissible and pink sub-blocks are inadmissible [37].

Therefore, a hierarchical partitioning is applied to the initial MoM matrix, in order to identify the matrix off-diagonal blocks most likely to undergo an efficient ACA compression. Sufficiently separated clusters determine sub-blocks that are numerically low rank, that is, suitable to an ACA application. These sub-blocks are called admissible [37]. On the other hand, sub-block corresponding to a pair of closely spaced clusters is called inadmissible and is not approximated. The threshold distance that sets out the admissible and inadmissible sub-blocks is a parameter, typically set to one wavelength of the frequency under consideration. Another parameter of the geometry decomposition process is the minimum block size. An inadmissible sub-block may be divided until it reaches that size. An example

of geometry decomposition, for the preparation of the application of the ACA, is illustrated in 5.1 [37]. More details are provided on the hierarchical matrices in [71].

Once the corresponding H-matrix is constructed, the ACA is applied as shown in Chapter 4, with the respect to an error tolerance ϵ. The final step consists on solving the final sparse system of linear equations by using an iterative solver such as the GMRES or the CG.

5.1.2 Comparison with the CBFM-E for 3D forest scattering modeling

In this paragraph, we carry out a simple comparison between the performances of the CBFM-E and those achieved by the ACA algorithm when applying the two methods to our 3D dielectric forest model described previously. The ACA code used was implemented in the language C++ by Dr. Juvigny (ONERA - High Performance Computing Department). A FORTRAN interface enables us to compare the CBFM-E and the ACA on the basis of the same meshing an the simulations were carried out on a shared-memory workstation with 8 processors and 42 GB of RAM.

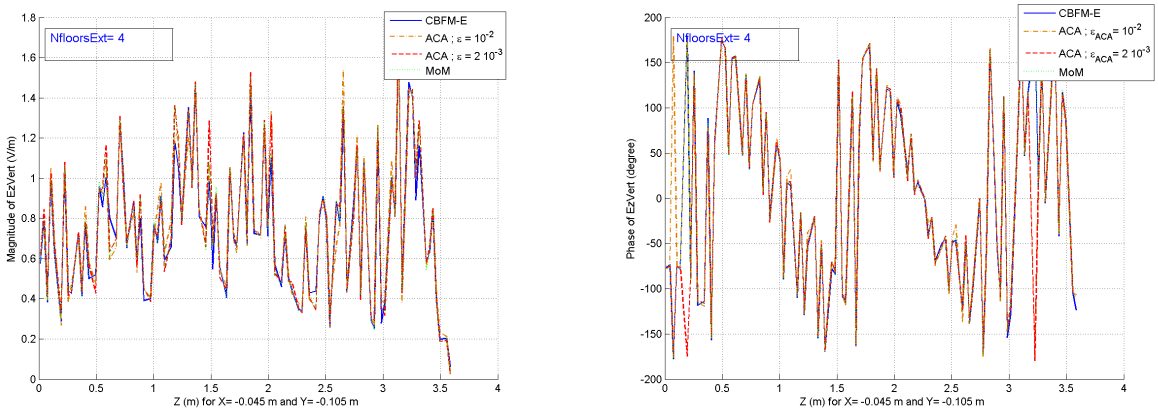
We start with the straightforward example of a single tree trunk modelled by a homogeneous dielectric cylinder placed over a real ground. The cylinder, of dielectric constant $\epsilon_s = 8.2 + 0.9$ j is illuminated by an incident plane wave of frequency $f = 300$ MHz, and therefore is discretized using a cell size $T_c = 0.03$ m into $Nb_c = 6240$ cells. The CBFM-E is applied after decomposition of the cylinder into 12 CBFM blocks of heigh $h_B = \lambda$.

As shown in Figures 5.2 and 5.3, the results obtained are compared to those derived by the ACA with four different values of ϵ $(10^{-2}; 10^{-3}; 2\ 10^{-3}; 10^{-4})$. Figure 5.2 shows the variations of the component E_z^V of the electric field inside the cylinder under simulation, and Figure 5.3 presents the variations of the magnitude and phase of the backscattered fields (polarization VV) for $\theta_{i,s} = [1°; 1°; 80°]$ and $\varphi_{i,s} = 0°$. Finally, Table 5.1 summarizes the performances achieved by both CBFM-E and ACA in terms of computing time and compression rate.

	ICR (%)	CPU time (sec)
CBFM-E	2.61	141
ACA ($\epsilon = 10^{-1}$)	7.29	29 + 806
ACA ($\epsilon = 10^{-2}$)	11.39	30 + 940
ACA ($\epsilon = 2\ 10^{-3}$)	12	30 + 1036
ACA ($\epsilon = 10^{-4}$)	15.9	32 + 1312
Conventional MoM	–	5761

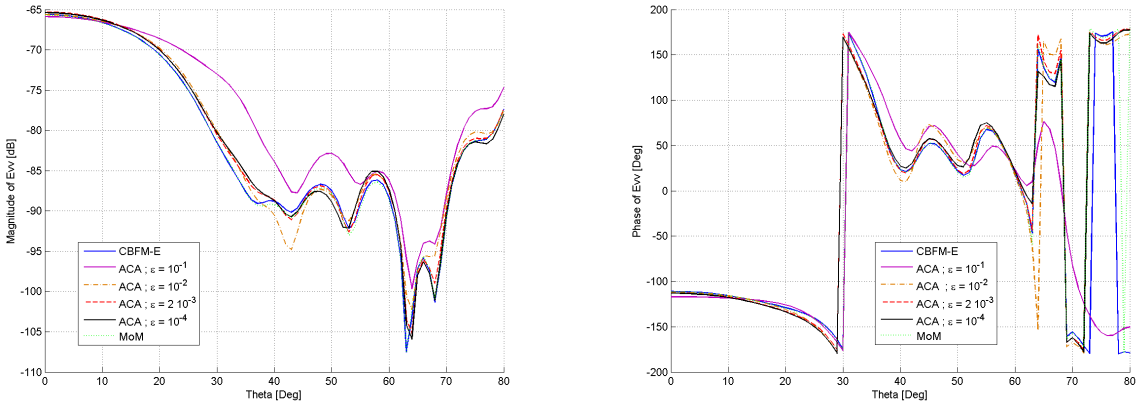
TABLE 5.1: The computing time and inverted compression rate (ICR) achieved by the CBFM-E and ACA (with four value of ϵ_{ACA}) compared to the conventional MoM. The CPU time spent by the ACA is presented in this form : CPU(H-Matrix+ACA) + CPU(GMRES)

First, concerning the accuracy of the solution, Figures 5.2 and 5.3 show that the results obtained with the ACA match relatively well with those derived by the CBFM-E and the conventional MoM starting from $\epsilon_{ACA} = 10^{-3}$ which confirms the observations made on ϵ_{ACA} in the previous chapter. However, as can bee seen in Table 5.1, the CBFM-E goes ahead of the ACA in terms of memory consumption and CPU time. Indeed, the high compression rate achieved by the CBFM-E is due to the use of large CBFM blocks ($Nb_{c,block} = 736$) and to the absence of branches. In fact, the branches impact negatively the total compression rate because of its small cross-section resulting in a small number of cells per block.



(A) E_Z^V (B) ϕ_Z^V

FIGURE 5.2: Variations of the magnitude an phase of the component E_Z^V of the electric field inside the cylinder under consideration, computed for $\theta_i = 40°$ and $\varphi_i = 0°$, with the conventional MoM, the CBFM-E and the ACA.



(A) E_{VV} (B) phi_{VV}

FIGURE 5.3: Variations of the magnitude the backscattered fields computed with CBFM-E and the ACA depending on the value of ϵ_{ACA}.

On the other hand, it is worthwhile mentioning that we have distinguished the CPU time for the construction of the H-Matrix and the application of the ACA, and that spent to solve the final sparse

matrix by using a GMRES, because the latter was not optimized yet. Indeed, the simple implementation of a preconditioner could improve significantly the performances of the GMRES in terms of CPU time. Thus, we can see that the ACA algorithm could be much faster, and comparable to the CBFM-E in terms of CPU time if the GMRES is optimized.

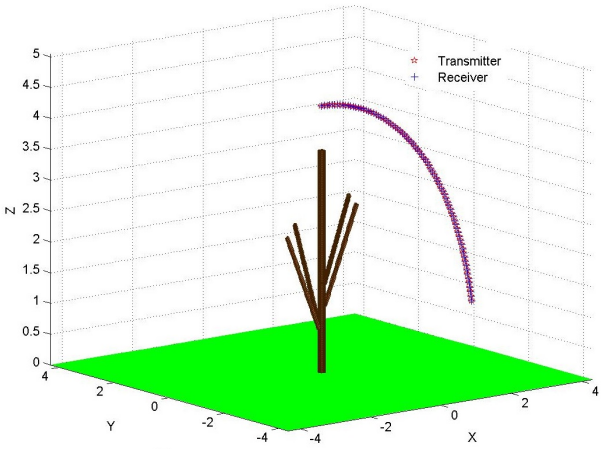
Now, we move to the example of a tree trunk with 4 branches in order to investigate the impact of the structure and meshing approach of our 3D model on the performances of the hierarchical repartition and the ACA. On that point, let us recall that the the ACA algorithm requires that the integral kernels be asymptotically smooth, to be successfully applied [34]. The example under consideration is composed of 5 dielectric cylinders modeling a homogeneous tree of dielectric permittivity $\epsilon_s = 8.2+0.9\text{j}$ and illuminated by an incident plane wave of frequency $f = 300$ MHz. The simulations are carried out by using the conventional MoM, the CBFM-E and the ACA with a tolerance $\epsilon = 10^{-3}$, for the three configurations shown in Figure 5.4, depending on β_{br}, the angle between the tree trunk and each branch. The corresponding results derived for $\theta_{i,s} = [1^\circ; 1^\circ; 80^\circ]$ and $\varphi_{i,s} = 0^\circ$, are also displayed in the same figure, and the performances of the ACA compared to those achieved by the CBFM-E and the MoM are summarized in Table 5.2.

		ICR (%)	CPU time (sec)
ACA	Config. 1 ($\beta_{br} = 30^\circ$)	17.6	9 + 94
	Config. 2 ($\beta_{br} = 0^\circ$)	15.9	6 + 57
	Config. 3 ($\beta_{br} = 90^\circ$)	78	1187 + 245
CBFM-E	–	12.85	27
MoM	–	–	553

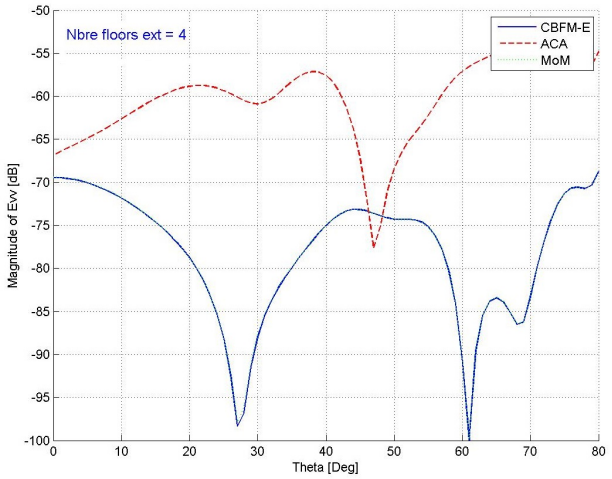
TABLE 5.2: The computing time and inverted compression rate (ICR) achieved by the ACA for the three configurations under consideration in comparison to the performances of the CBFM-E and the conventional MoM. The CPU time is still presented in the form : CPU(H-Matrix+ACA) + CPU(GMRES).

Therefore, curiously, we note that both performances and accuracy of the ACA solution critically depend upon the inclination angle of the branch to the vertical. This can be partly explained by the asymptotic discontinuity of the used mesh between the tree trunk and the branch (see Figure 5.5), which introduces an irregularity into the smoothness of the integral kernels, the latter being particularly important to ensure the accuracy of the ACA solution. Indeed, it was stated in [37], while evoking the limitations of the ACA, that the iterations may appear to converge while the true error remains inaccurately large. Furthermore, it confirms that the EFIE (Electric Fields Integral Equation) does not satisfy the conditions stated in [59] and [72], for the integral equations with the kernels which are asymptotically smooth.

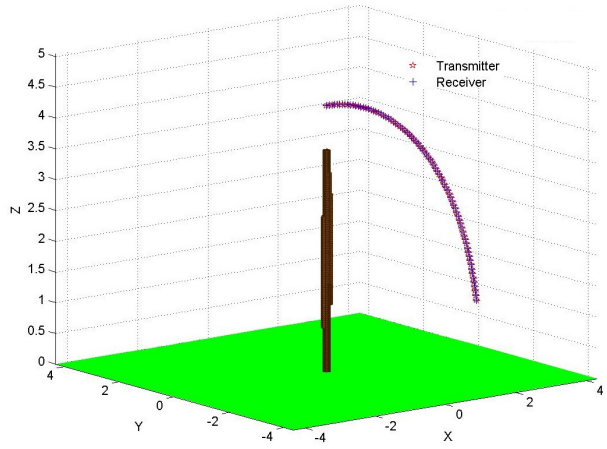
Nonetheless, we have attempt to adapt the meshing approach to the requirements of the ACA in terms of asymptotic smoothness, as shown in Figure 5.5. Then, we compare again the backscattered fields derived by the ACA algorithm to those obtained with the MoM and the CBFM-E.



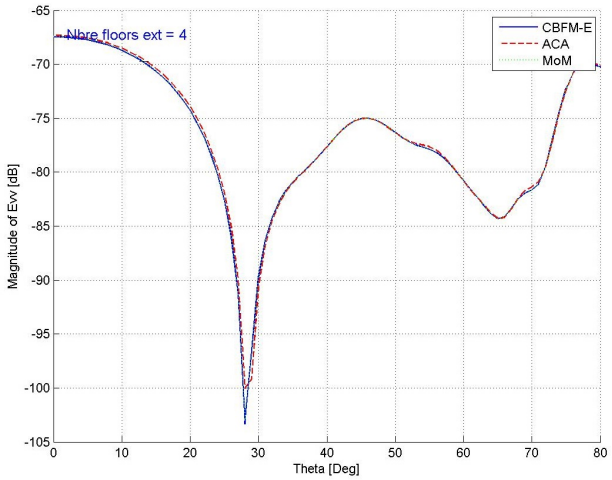
(A) Config. 1 : $\beta_{br} = 30^{\circ}$



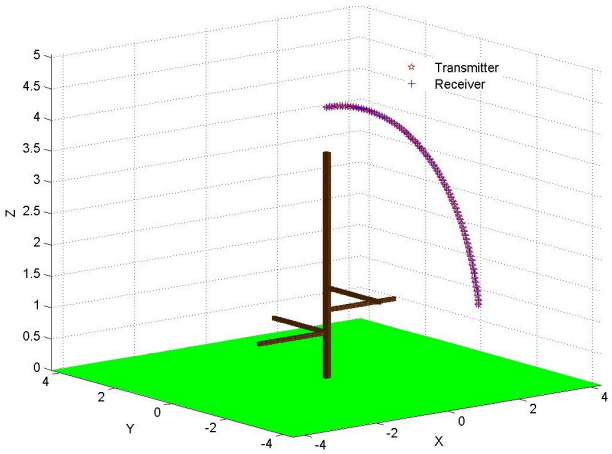
(B) Resulting E_{VV}



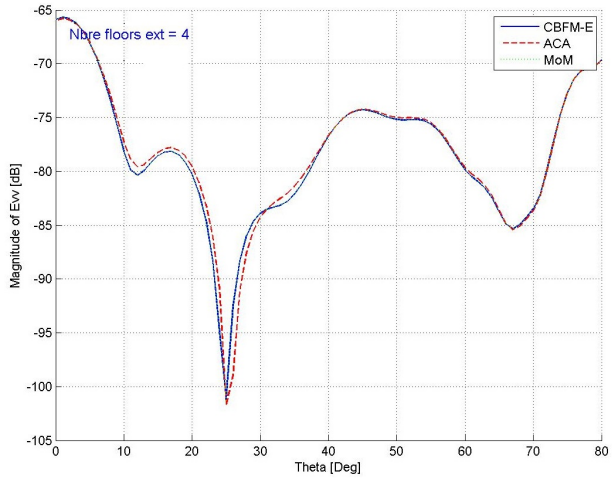
(C) Config. 2 : $\beta_{br} = 0^{\circ}$



(D) Resulting E_{VV}

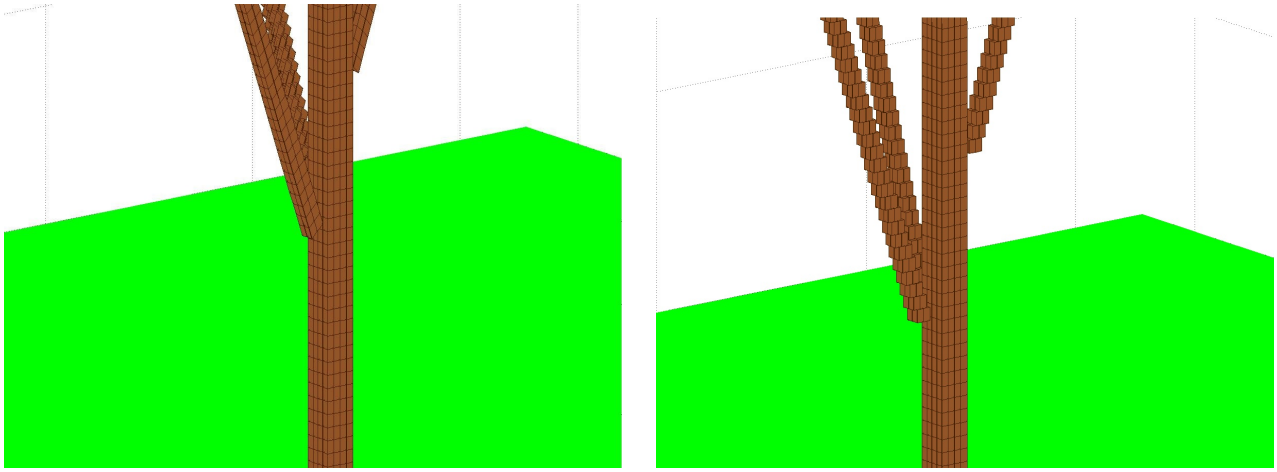


(E) Config. 3 : $\beta_{br} = 90^{\circ}$



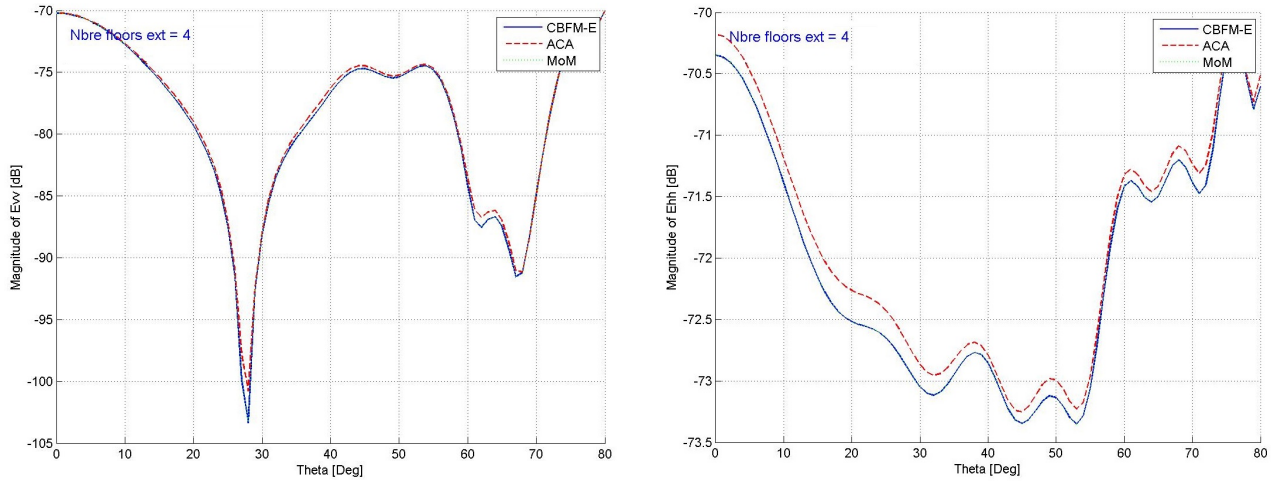
(F) Resulting E_{VV}

FIGURE 5.4: The variations of the magnitude of the backscattered field (polarization VV) computed by using the MoM, the CBFM-E and the ACA, for 3 different values of β_{br}, which is the angle between the tree trunk and the branch.



(A) Old Mesh (B) New Mesh

FIGURE 5.5: We attempt to adapt the meshing in order to guarantee the asymptotic smoothness of the integral kernels and hence to ensure the accuracy of the ACA solution.



(A) E_{VV} (B) E_{HH}

FIGURE 5.6: The variations of the backscattered fields (polarization VV and HH) computed by applying the ACA on the new mesh (see Figure 5.5), for $\theta_{i,s} = [1^\circ; 1^\circ; 80^\circ]$ and $\varphi_{i,s} = 0^\circ$.

	ICR (%)	CPU time (sec)
ACA	21.73	20 + 64
CBFM-E	12.85	26
MoM	–	616

TABLE 5.3: The performances of the ACA in comparison to those achieved by the CBFM-E and the conventional MoM when applied to the new mesh. The CPU time is still presented in the form : CPU(H-Matrix+ACA) + CPU(GMRES).

Obviously, the new meshing approach enables us to significantly improve the performances of the ACA in terms of CPU time and, more importantly, to ensure its accuracy. However The CBFM-E still remains more efficient both in terms of computing time and memory storage for this case of application, especially since this direct method clearly does not depend on the meshing approach used to discretize the simulation scene and does not suffer from any convergence problem, unlike the iterative methods.

We also noted that increasing the numerical size of the forest simulation scene is absolutely not in favour of the ACA solution, as the gap in efficiency widens between the CBFM-E and the ACA algorithm. Indeed, Table 5.4 summarizes the CPU time and inverted compression rate achieved by the ACA and the CBFM-E when applied successively to two trees with 8 branches discretized into $Nb_c = 12120$ cells and $Nb_c = 23760$ cells.

Tree	$Nb_c = 12120$		$Nb_c = 23760$	
	ICR (%)	CPU time (sec)	ICR (%)	CPU time
ACA	19.4	3513 + 1469	59.18	11438 + 1731
CBFM-E	10.06	172	6.60	390

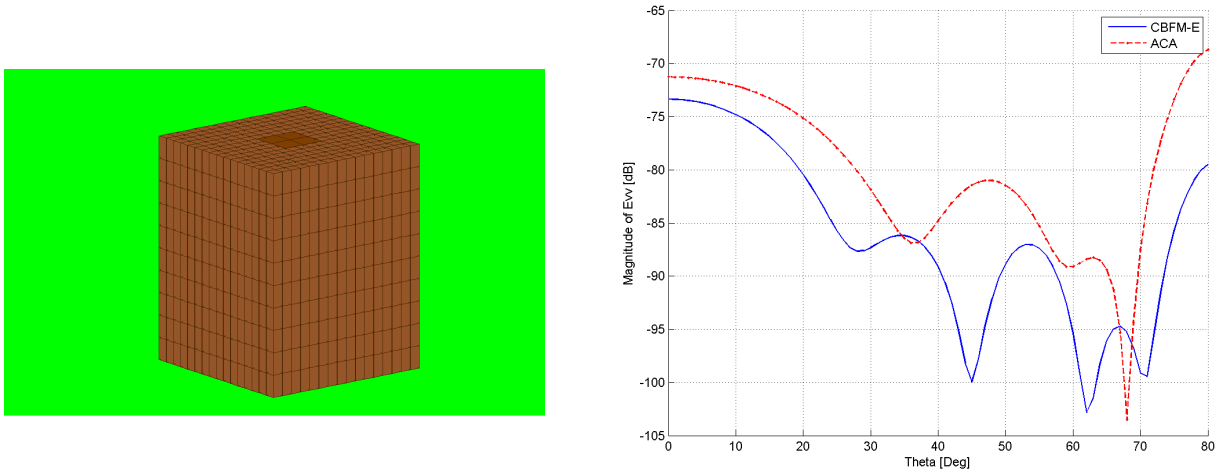
TABLE 5.4: The performances of the ACA in comparison to those achieved by the CBFM-E when applied to two trees discretized into $Nb_c = 12120$ and $Nb_c = 23760$ cells. The CPU time is still presented in the form : CPU(H-Matrix+ACA) + CPU(GMRES).

Moreover, the ACA algorithm maintains an ambiguous behaviour, in terms of compression rate and CPU time, towards the trees with branches even with the new mesh. Table 5.5 provides a comparison between the performances of the ACA with a quite large simulation scene composed of a single tree trunk discretized into $Nb_c = 35640$ cells (Scene 2), with those achieved with the previous example of $Nb_c = 23760$ cells (Scene 1).

	Scene 1 : $Nb_c = 23760$		Scene 2 : $Nb_c = 35640$	
	ICR (%)	CPU time (sec)	ICR (%)	CPU time
ACA	59.18	11438 + 1731	16.48	2479 + 3628
CBFM-E	6.60	390	1.68	777

TABLE 5.5: The performances of the ACA in comparison to those achieved by the CBFM-E when applied to a tree discretized into $Nb_c = 23760$ cells and a single tree trunk discretized into $Nb_c = 35640$. The CPU time is still presented in the form : CPU(H-Matrix+ACA) + CPU(GMRES).

Finally, we investigate the performances of the ACA algorithm when applied on a non-uniform mesh inside the same scatterer by applying it to the the single cylinder of dielectric constant $\epsilon_s = 6.75 + 6.7\text{j}$ (see Figure 5.7a. As shown in Figure 5.7b and in Table 5.6, fails the test of non-uniform mesh both in terms of accuracy and computing performance. This must be due, once again, to the sensitivity of the H-matrix partitioning and the ACA algorithm to the smoothness of the initial matrix kernel.



(A) 1^{st} block of the cylinder (B) E_{VV}

FIGURE 5.7: Application of the CBFM-E and the ACA on a non-uniform mesh inside the same scatterer. The magnitude of the backscattered field (polarization VV) corresponds to a mono-static configuration with $\theta_{i,s} = [0°; 1°; 80°]$ and $\phi_i = 0°$.

	ICR (%)	CPU time (sec)
ACA	21.79	1611 + 5518
CBFM-E	0.65	3138

TABLE 5.6: The performances of the ACA in comparison to those achieved by the CBFM-E when applied to a non-uniform mesh inside the same scatterer. The CPU time is still presented in the form : CPU(H-Matrix+ACA) + CPU(GMRES).

5.2 Conclusion

Hence, we demonstrated in this section that, despite of its proven performances both in terms of compression rate and reduction of CPU time [34], the ACA, preceded by the hierarchical partitioning of the MoM matrix, seems to be not adapted to our case of application because of the irregularities in the kernel of the initial MoM matrix. The limitations of the ACA in relation with its convergence were pointed out in [36, 37]. Since the CBFM-E, which based on a direct solver, is heir to the accuracy of the MoM, it does not suffer the convergence problems the ACA suffers from. Thus, the CBFM scored points against the ACA for this kind of application. One of our perspectives is to compare also the performances of CBFM-E to those achieved by the MLFMM [73], available under FEKO, when applied to our 3D dielectric forest model.

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

In this dissertation, a 3-D full-wave model, based on the integral representation of the electric field and dedicated to the analysis of bi-static scattering mechanisms by a forest in the VHF and UHF bands was efficiently enhanced. In order to overcome the limitation of the previous 3D model to small simulations scenes and low frequencies, we have developed, during this research work, a new model using basis functions adapted to the problem of interest, in the context of the Characteristic Basis Function Method (CBFM) and we investigated the suitability of this direct method for computing the electromagnetic fields inside and outside three-dimensional dielectric objects representing the tree trunks and branches.

The CBFM has shown great performances, when applied to the forest scattering modeling, both in terms of CPU time and memory resources needed. We have demonstrated that, to ensure a good level of accuracy by the CBFM-E (extended version of the CBFM), while computing the scattered field, without over-consuming memory resources and with a minimal CPU time, we must respect some ranges of value while setting CBFM parameters such as the height h of each block and the number of incident plane waves (N_{IPWs}) used to generate the macro-basis functions (MBFs). Once properly set, the CBFM-E is so efficient that it is able to treat in few minutes electromagnetic problems totally intractable with the classical MoM.

Furthermore, since the CBFM is a highly parallelizable decomposition domain method, we implemented and OpenMP version of the CBFM-E solution. This, in turn, makes it possible to deal with a substantial increase in the size of the scattering problems, up to several hundred of thousands of unknowns, and hence to solve problems involving much larger forest areas at higher frequencies.

We have also studied the suitability of the CBFM to the implementation of a non-uniform mesh which takes into consideration the heterogeneity of the forest medium in terms of dielectric properties. We found out that we can apply the CBFM-E to larger and more complex forest areas, while taking

into account the dielectric heterogeneity of the trees as we decompose the computational domain, by adapting the mesh to their dielectric properties.

Then, since we note that the Adaptive Cross Approximation (ACA) algorithm has shown good performance, in terms of compression, when dealing with coupling matrix blocks representing well-separated MoM interactions, we have applied the ACA algorithm to the generation of the reduced matrix in the context of the CBFM in order to speed up the most expensive step in the CBFM in terms of the required CPU time and we found out that the new hybridized algorithm is highly efficient, both in terms of computing time and memory use without compromising the accuracy of the CBFM solution. Furthermore, we compared the CBFM performances, when applied to forest scattering modeling, to other powerful iterative numerical methods such as the ACA algorithm (preceded here by the Hierarchical partitioning of the MoM matrix). Thus, the CBFM shows a wide superiority in terms of accuracy comparing to the ACA, at least when applied to our case of application.

Consequently, we have developed a powerful 3D forest electromagnetic scattering tool which allows us today to compute large forest electromagnetic problems in few minutes without worrying about the accuracy of the solution. On the other hand, we have demonstrated the efficiency and accuracy of the CBFM-E when applied to 3D dielectric objects in the context of the electric volumetric integral equation, and have consolidated thus its leading position in the computational electromagnetics, especially against the iterative solvers based numerical methods.

`We have thus achieved the two major challenges of this research work.`

6.2 Perspectives

Certainly, we have come a long way and a lot of enhancement has been done, but here is still a lot of work to do with the Characteristic Basis Function Method to improve further and further the performances of our 3D dielectric forest scattering model. The following provide some of the potential improvements, with a view to simulating numerically larger forest scenes :

- A 3D domain decomposition of the simulation scene, instead of the current decomposition along the vertical axis $\vec{z}$: This will enable us to overcome the limitation in terms of available memory per process, especially when simulating wide-section forest tree trunks.

- An Efficient MPI parallelization of the multilevel scheme of the CBFM : This entails optimized distribution of the CBFM blocks betwenn the available processors and reduced time of communication between them. For the multilevel scheme, in particular, we can draw on the previous research works dealing with the optimization of the parallelization of the multilevel Fast Multipole Method (MLFMM) [74–77].

- A Hybridization with the MLFMM in addition to that performed with the ACA in order compute the far-field interactions between distant CBFM blocks, as can bee seen in [42].

- Use of the sparsified Adaptive Cross Approximation algorithm instead of the conventional one to speed up the generation of the reduced matrix Z^c. Indeed, it has been confirmed in [78] that the sparsified algorithm achieves a considerable gain in efficiency over the conventional ACA.

- Application of the sparse representation of the macro-basis functions (MBFs) on a non-uniform mesh inside the same scatterer, in the case of high value of ϵ_s", in order to avoid the computational burden brought by the fact that the CBFM block initially scaled in $T_{c,\lambda}$ will be discretized into smaller cells of $T_{c,\delta}$.

Once the CBFM code optimized, it can be used to provide a data base for backscattered fields inversion for forest remote sensing, in order to retrieve the parameters of a forest medium [79].

Furthermore, the next step consists in the insertion of metallic targets in the 3D dielectric forest model. That will bring this simulation tool closer to the military application known under the name of FoPEN for Foliage Penetration. This new research work based on the volume-surface integral equation [80] will form the basis of a new PhD conducted in the L2E laboratory and funded by the French national research agency (ANR) and French Defence Agency (DGA).

Appendix A

Elements of the MoM matrix

The initial MoM matrix [19], which represents the interactions between the different cells composing the simulation scene, is of size $3N \times 3N$ where N is the total number of cells. It is written as follows

$$\underline{\mathbf{Z}} = \begin{bmatrix} [\mathrm{Z}_{xx}^{mn}] & [\mathrm{Z}_{xy}^{mn}] & [\mathrm{Z}_{xz}^{mn}] \\ [\mathrm{Z}_{yx}^{mn}] & [\mathrm{Z}_{yy}^{mn}] & [\mathrm{Z}_{yz}^{mn}] \\ [\mathrm{Z}_{zx}^{mn}] & [\mathrm{Z}_{zy}^{mn}] & [\mathrm{Z}_{zz}^{mn}] \end{bmatrix} \tag{A.1}$$

The coefficients Z_{pq}^{mn} of the MoM matrix are expressed as :

$$\mathrm{Z}_{pq}^{mn} = \delta_{mn}\delta_{pq} - \mathrm{Z}_{pq}^{\mathrm{s},mn} - \mathrm{Z}_{pq}^{\mathrm{r},mn} \tag{A.2}$$

The product $\delta_{mn}\delta_{pq}$ is equal to 1 only for the diagonal elements of the MoM matrix $\underline{\mathbf{Z}}$. δ_{mn} and δ_{pq} are the delta Kronecker functions defined as

$$\delta_{mn},\ \delta_{pq} = \begin{cases} 1 & \text{si} \quad m = n,\ p = q \\ 0 & \text{si} \quad m \neq n,\ p \neq q \end{cases} \tag{A.3}$$

$$\mathrm{Z}_{pq}^{\mathrm{s},mn} = k_0^2 \int_{V_{\mathrm{cell}}} \Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)\mathrm{G}_{pq}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n + \sum_{q=1}^{3} \frac{\partial^2}{\partial p_m q_m} \int_{V_{\mathrm{cell}}} \Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)\mathrm{G}_{pq}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n \tag{A.4}$$

$$\mathrm{Z}_{pq}^{\mathrm{r},mn} = k_0^2 \int_{V_{\mathrm{cell}}} \Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)\mathrm{G}_{pq}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n + \sum_{q=1}^{3} \frac{\partial^2}{\partial p_m q_m} \int_{V_{\mathrm{cell}}} \Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)\mathrm{G}_{pq}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n \tag{A.5}$$

$\overline{\mathbf{r}}_m$ is the coordinate vector of the observation point m , $\overline{\mathbf{r}}'_n$ is the coordinate vector of the center of the source cell n, $p, q = x, y, z$, k_0 is the number of wavelength in the air, G_{pq}^{s} is the component pq of the singular Dyadic Green's function, G_{pq}^{r} is the component pq of the regular Dyadic Green's function and $\Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}') = \dfrac{\epsilon(\overline{\mathbf{r}}') - \epsilon_0}{\epsilon_0}$ is the dielectric contrast.
We assume that :

$$\int_{V_{\mathrm{cell}}} \Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)\mathrm{G}_{pq}^{\psi}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n = \Delta\epsilon(\overrightarrow{r}')_n \int_{V_{\mathrm{cell}}} \mathrm{G}_{pq}^{\psi}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n)d\overline{\mathbf{r}}'_n \tag{A.6}$$

where $\psi = \mathrm{s}, \mathrm{r}$. If the domain Ω is homogeneous, the dielectric contrast $\Delta\epsilon(\overrightarrow{r}')(\overline{\mathbf{r}}'_n)$ does not depend on the position $\overline{\mathbf{r}}'_n$ and it is thus equal to $\Delta\epsilon(\overrightarrow{r}') = \epsilon_r - 1$.

A.1 Computing of the terms $\mathrm{Z}_{pq}^{\mathrm{s},mn}$

According to equation (A.4), the coefficients $\mathrm{Z}_{pq}^{\mathrm{s},mn}$ are obtained as follows

$$\mathrm{Z}_{xx}^{\mathrm{s},mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial x_m^2} \right) \int_{V_{\mathrm{cell}}} \mathrm{G}_{xx}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.7}$$

$$\mathrm{Z}_{yy}^{\mathrm{s},mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial y_m^2} \right) \int_{V_{\mathrm{cell}}} \mathrm{G}_{yy}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.8}$$

$$\mathrm{Z}_{zz}^{\mathrm{s},mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial z_m^2} \right) \int_{V_{\mathrm{cell}}} \mathrm{G}_{zz}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.9}$$

$$\mathrm{Z}_{xy}^{\mathrm{s},mn} = \left\{ \frac{\partial^2}{\partial x_m \partial y_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{yy}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.10}$$

$$\mathrm{Z}_{xz}^{\mathrm{s},mn} = \left\{ \frac{\partial^2}{\partial x_m \partial z_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{zz}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.11}$$

$$\mathrm{Z}_{yx}^{\mathrm{s},mn} = \left\{ \frac{\partial^2}{\partial y_m \partial x_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{xx}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.12}$$

$$\mathrm{Z}_{yz}^{\mathrm{s},mn} = \left\{ \frac{\partial^2}{\partial y_m \partial z_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{zz}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.13}$$

$$\mathrm{Z}_{zx}^{\mathrm{s},mn} = \left\{ \frac{\partial^2}{\partial z_m \partial x_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{xx}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.14}$$

$$\mathrm{Z}_{zy}^{\mathrm{s},mn} = \Delta\epsilon(\overrightarrow{r}')_n \left\{ \frac{\partial^2}{\partial z_m \partial y_m} \int_{V_{\mathrm{cell}}} \mathrm{G}_{yy}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.15}$$

Now, we have to compute the integrals on the cells of the components of the singular Dyadic Green's function. There are two possible cases : The first one is $m = n$ which means that the observation and source points are the same. Thus, the integration of the components of the Dyadic Green's function will present a singularity and the computing will be done as follows :

$$\mathrm{Z}_{pq}^{s,mn} = \begin{cases} \mathrm{PF} \left(\int_{V_{\mathrm{cell}}} \left(k_0^2 + \frac{\partial^2}{\partial p_m^2} \right) \mathrm{G}_{pq}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right) \Delta\epsilon(\overrightarrow{r}')_n & \text{si} \quad p = q, m = n \\ 0 \qquad \text{si} \quad p \neq q, m = n \end{cases} \tag{A.16}$$

If we approximate the integration on a cubic cell of size c_n by that on a spherical cell having the same center and of radius $a_n = c_n \sqrt[3]{\frac{3}{4\pi}}$, we can demonstrate that :

$$\mathrm{PF} \left(\int_{V_{\mathrm{cell}}} \left(k_0^2 + \frac{\partial^2}{\partial p_m^2} \right) \mathrm{G}_{pq}^{\mathrm{s}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right) = \frac{2}{3} \, e^{jk_0 a_n} (1 - jk_0 a_n) - 1 \tag{A.17}$$

Concerning the second case ($m \neq n$), we use also the same above approximation :

$$\int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{s}}_{pq}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n = \kappa^s_n \times \mathrm{G}^{\mathrm{s}}_{pq}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) \tag{A.18}$$

with :

$$\kappa^s_n = 4\pi \times \frac{\sin(k_0 a_n) - k_0 a_n \cos(k_0 a_n)}{k_0^3} \tag{A.19}$$

Thus we obtain :

$$\mathrm{Z}^{s,mn}_{xx} = \frac{e^{jk_0 r_{mn}}}{4\pi r^2_{mn}} \left(\tau_{mn} + k_0^2 \left(r_{mn} - \frac{(x_m - x'_n)^2}{r_{mn}} \right) - 3 \frac{(x_m - x'_n)^2}{r^2_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.20}$$

$$\mathrm{Z}^{s,mn}_{yy} = \frac{e^{jk_0 r_{mn}}}{4\pi r^2_{mn}} \left(\tau_{mn} + k_0^2 \left(r_{mn} - \frac{(y_m - y'_n)^2}{r_{mn}} \right) - 3 \frac{(y_m - y'_n)^2}{r^2_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.21}$$

$$\mathrm{Z}^{s,mn}_{zz} = \frac{e^{jk_0 r_{mn}}}{4\pi r^2_{mn}} \left(\tau_{mn} + k_0^2 \left(r_{mn} - \frac{(z_m - z'_n)^2}{r_{mn}} \right) - 3 \frac{(z_m - z'_n)^2}{r^2_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.22}$$

$$\mathrm{Z}^{s,mn}_{xy} = \mathrm{Z}^{s,mn}_{yx} = \frac{e^{jk_0 r_{mn}} \left(x_m - x'_n\right) \left(y_m - y'_n\right)}{4\pi r^3_{mn}} \left(-k_0^2 - \frac{3}{r_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.23}$$

$$\mathrm{Z}^{s,mn}_{xz} = \mathrm{Z}^{s,mn}_{zx} = \frac{e^{jk_0 r_{mn}} \left(x_m - x'_n\right) \left(z_m - z'_n\right)}{4\pi r^3_{mn}} \left(-k_0^2 - \frac{3}{r_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.24}$$

$$\mathrm{Z}^{s,mn}_{yz} = \mathrm{Z}^{s,mn}_{zy} = \frac{e^{jk_0 r_{mn}} \left(y_m - y'_n\right) \left(z_m - z'_n\right)}{4\pi r^3_{mn}} \left(-k_0^2 - \frac{3}{r_{mn}} \tau_{mn} \right) \kappa^s_n \, \Delta\epsilon(\overrightarrow{r}')_n \tag{A.25}$$

with : $r_{mn} = \sqrt{(x_m - x'_n)^2 + (y_m - y'_n)^2 + (z_m - z'_n)^2}$ is the distance between the center of the observation cell m and that of the source cell n and $\tau_{mn} = jk_0 - \dfrac{1}{r_{mn}}$.

A.2 Computing the terms of regular interactions $\mathrm{Z}^{\mathrm{r},mn}_{pq}$

In this section, we detail the computing of the coefficients $\mathrm{Z}^{\mathrm{r},mn}_{pq}$. Based on (A.5), we obtain :

$$\mathrm{Z}^{r,mn}_{xx} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial x_m^2} \right) \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{xx}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial x_m z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zx}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.26}$$

$$\mathrm{Z}^{r,mn}_{xy} = \left\{ \frac{\partial^2}{\partial x_m \partial y_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{yy}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial x_m z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zy}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.27}$$

$$\mathrm{Z}^{r,mn}_{xz} = \left\{ \frac{\partial^2}{\partial x_m \partial z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zz}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.28}$$

$$\mathrm{Z}^{r,mn}_{yx} = \left\{ \frac{\partial^2}{\partial y_m \partial x_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{xx}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial y_m z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zx}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.29}$$

$$\mathrm{Z}^{r,mn}_{yy} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial y_m^2} \right) \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{yy}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial y_m z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zy}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.30}$$

$$\mathrm{Z}^{r,mn}_{yz} = \left\{ \frac{\partial^2}{\partial y_m \partial z_m} \int_{V_{\text{cell}}} \mathrm{G}^{\mathrm{r}}_{zz}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.31}$$

$$\mathrm{Z}_{zx}^{r,mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial z_m^2} \right) \int_{V_{\text{cell}}} \mathrm{G}_{zx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial z_m x_m} \int_{V_{\text{cell}}} \mathrm{G}_{xx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.32}$$

$$\mathrm{Z}_{zy}^{r,mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial z_m^2} \right) \int_{V_{\text{cell}}} \mathrm{G}_{zy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n + \frac{\partial^2}{\partial z_m y_m} \int_{V_{\text{cell}}} \mathrm{G}_{yy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.33}$$

$$\mathrm{Z}_{zz}^{r,mn} = \left\{ \left(k_0^2 + \frac{\partial^2}{\partial z_m^2} \right) \int_{V_{\text{cell}}} \mathrm{G}_{zz}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.34}$$

The coefficients $\mathrm{Z}_{pq}^{r,mn}$ include the exponentials $e^{j\nu(x_m - x'_n)}$, $e^{j\eta(y_m - y'_n)}$, et $e^{j\gamma_0(z_m + z'_n)}$. The derivations $\dfrac{\partial}{\partial x_m}$, $\dfrac{\partial}{\partial y_m}$, et $\dfrac{\partial}{\partial z_m}$ are equivalent to the multiplication of these exponentials respectively by $j2\pi\nu$, $j2\pi\eta$, et $j\gamma_0$:

$$\mathrm{Z}_{xx}^{r,mn} = \left\{ \left(k_0^2 - 4\pi^2\nu^2 \right) \int_{V_{\text{cell}}} \mathrm{G}_{xx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\nu\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.35}$$

$$\mathrm{Z}_{xy}^{r,mn} = \left\{ -4\pi^2\nu\eta \int_{V_{\text{cell}}} \mathrm{G}_{yy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\nu\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.36}$$

$$\mathrm{Z}_{xz}^{r,mn} = \left\{ -2\pi\nu\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zz}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.37}$$

$$\mathrm{Z}_{yx}^{r,mn} = \left\{ -4\pi^2\nu\eta \int_{V_{\text{cell}}} \mathrm{G}_{xx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\nu\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.38}$$

$$\mathrm{Z}_{yy}^{r,mn} = \left\{ \left(k_0^2 - 4\pi^2\eta^2 \right) \int_{V_{\text{cell}}} \mathrm{G}_{yy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\eta\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.39}$$

$$\mathrm{Z}_{yz}^{r,mn} = \left\{ -2\pi\eta\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{zz}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.40}$$

$$\mathrm{Z}_{zx}^{r,mn} = \left\{ \left(k_0^2 - \gamma_0^2 \right) \int_{V_{\text{cell}}} \mathrm{G}_{zx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\nu\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{xx}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.41}$$

$$\mathrm{Z}_{zy}^{r,mn} = \left\{ \left(k_0^2 - \gamma_0^2 \right) \int_{V_{\text{cell}}} \mathrm{G}_{zy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n - 2\pi\eta\gamma_0 \int_{V_{\text{cell}}} \mathrm{G}_{yy}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.42}$$

$$\mathrm{Z}_{zz}^{r,mn} = \left\{ \left(k_0^2 - \gamma_0^2 \right) \int_{V_{\text{cell}}} \mathrm{G}_{zz}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n \right\} \Delta\epsilon(\overrightarrow{r}')_n \tag{A.43}$$

The integration of the regular Dyadic Green's function terms on a cubic cell of size c_n is giben by :

$$\int_{V_{\text{cell}}} \mathrm{G}_{pq}^{\mathrm{r}}(\overline{\mathbf{r}}_m, \overline{\mathbf{r}}'_n) d\overline{\mathbf{r}}'_n = \int \int_{-\infty}^{+\infty} \kappa_n^r \left(\nu, \eta \right) g_{pq}^{\mathrm{r}}(\nu, \eta, z_m, z'_n) e^{j2\pi \left[\nu \left(x - x' \right) + \eta \left(y - y' \right) \right]} d\nu d\eta \tag{A.44}$$

with :

$$\kappa_n^r \left(\nu, \eta \right) = c_n^3 \mathrm{sinc} \left(\pi\nu c_n \right) \mathrm{sinc}(\pi\eta c_n) \, \mathrm{sinc} \left(\gamma_0 \tfrac{c_n}{2} \right) \tag{A.45}$$

Finally, this results in these expressions :

$$\begin{aligned} \mathrm{Z}_{xx}^{r,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int \int_{-\infty}^{+\infty} \kappa_n^r \left(\nu, \eta \right) \left\{ \left(k_0^2 - 4\pi^2\nu^2 \right) CR^{TE} - 2\pi\nu\gamma_0 C_\nu \right\} \\ &\times \; e^{j\gamma_0 \left(z_m + z'_n \right)} e^{j2\pi \left[\nu \left(x - x' \right) + \eta \left(y - y' \right) \right]} d\nu d\eta \end{aligned} \tag{A.46}$$

$$\begin{aligned} \mathrm{Z}_{xy}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{-4\pi^2\nu\eta CR^{^{TE}} - 2\pi\nu\gamma_0 C_\eta\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.47}$$

$$\begin{aligned} \mathrm{Z}_{xz}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{-2\pi\nu\gamma_0 CR^{^{TM}}\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.48}$$

$$\begin{aligned} \mathrm{Z}_{yx}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{-4\pi^2\nu\eta CR^{^{TE}} - 2\pi\eta\gamma_0 C_\nu\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.49}$$

$$\begin{aligned} \mathrm{Z}_{yy}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{\left(k_0^2-4\pi^2\eta^2\right) CR^{^{TE}} - 2\pi\eta\gamma_0 C_\eta\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.50}$$

$$\begin{aligned} \mathrm{Z}_{yz}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{-2\pi\eta\gamma_0 CR^{^{TM}}\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.51}$$

$$\begin{aligned} \mathrm{Z}_{zx}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{\left(k_0^2-\gamma_0^2\right) C_\nu - 2\pi\nu\gamma_0 CR^{^{TE}}\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.52}$$

$$\begin{aligned} \mathrm{Z}_{zy}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{\left(k_0^2-\gamma_0^2\right) C_\eta - 2\pi\eta\gamma_0 CR^{^{TE}}\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.53}$$

$$\begin{aligned} \mathrm{Z}_{zz}^{r,\,mn} &= \Delta\epsilon(\overrightarrow{r}')_n \int\int_{-\infty}^{+\infty} \kappa_n^r\left(\nu,\eta\right)\left\{\left(k_0^2-\gamma_0^2\right) CR^{^{TM}}\right\} \\ &\times\ e^{j\gamma_0\left(z_m+z_n'\right)} e^{j2\pi\left[\nu\left(x-x'\right)+\eta\left(y-y'\right)\right]} d\nu d\eta \end{aligned} \tag{A.54}$$

Bibliography

[1] Yousif Ali Hussin, Robin M Reich, and Roger M Hoffer. Estimating splash pine biomass using radar backscatter. *Geoscience and Remote Sensing, IEEE Transactions on*, 29(3):427–431, 1991.

[2] Jean-Michel Martinez and Thuy Le Toan. Mapping of flood dynamics and spatial distribution of vegetation in the amazon floodplain using multitemporal sar data. *Remote sensing of Environment*, 108(3):209–223, 2007.

[3] A Beaudoin, T. Le Toan, C.C. Hsu, H. C. Han, J.A Kong, and R.T. Shin. Simulation of forest backscatter as a function of forest and ground parameters. In *Geoscience and Remote Sensing Symposium, 1992. IGARSS '92. International*, volume 2, pages 1212–1214, May 1992. doi: 10.1109/IGARSS.1992.578394.

[4] L. Villard, P. Borderies, T. Le Toan, T. Koleck, and C. Albinet. Topography effects on forest radar scattering, consequences on biomass retrieval. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 60–63, July 2010. doi: 10.1109/IGARSS.2010.5650646.

[5] F. Garestier and T. Le Toan. Forest modeling for height inversion using single-baseline insar/pol-insar data. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(3):1528–1539, March 2010. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2032538.

[6] G. Smith and L. M H Ulander. A model relating vhf-band backscatter to stem volume of coniferous boreal forest. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(2):728–740, Mar 2000. ISSN 0196-2892. doi: 10.1109/36.842002.

[7] Fawwaz T Ulaby, Kamal Sarabandi, Kyle McDonald, Michael Whitt, and M Craig Dobson. Michigan microwave canopy scattering model. *International Journal of Remote Sensing*, 11(7):1223–1253, 1990.

[8] Tsenchieh Chiu and Kamal Sarabandi. Electromagnetic scattering from short branching vegetation. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(2):911–925, 2000.

[9] AT. Manninen and L. M H Ulander. Forestry parameter retrieval from texture in carabas vhf-band sar images. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(12):2622–2633, Dec 2001. ISSN 0196-2892. doi: 10.1109/36.974997.

[10] J. E S Fransson, F. Walter, and L. M H Ulander. Estimation of forest parameters using carabas-ii vhf sar data. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(2):720–727, Mar 2000. ISSN 0196-2892. doi: 10.1109/36.842001.

[11] H. Israelsson, L. M H Ulander, J.IH. Askne, J. E S Fransson, P.-O. Frolind, A Gustavsson, and H. Hellsten. Retrieval of forest stem volume using vhf sar. *Geoscience and Remote Sensing, IEEE Transactions on*, 35(1):36–40, Jan 1997. ISSN 0196-2892. doi: 10.1109/36.551932.

[12] M. Magnusson and J. E S Fransson. Combining carabas-ii vhf sar and landsat tm satellite data for estimation of forest stem volume. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International*, volume 4, pages 2327–2331 vol.4, Sept 2004. doi: 10.1109/IGARSS.2004.1369752.

[13] A. J. Gatesman, C. J. Beaudoin, R. H. Giles, J. Waldman, and W. E. Nixon. VHF/UHF imagery and RCS measurements of ground targets in forested terrain. In E. G. Zelnio, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4727 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 80–90, August 2002.

[14] Andrew J Gatesman, Thomas M Goyette, Jason C Dickinson, Jerry Waldman, Jim Neilson, and William E Nixon. Physical scale modeling the millimeter-wave backscattering behavior of ground clutter. Technical report, DTIC Document, 2001.

[15] T.K. Sjogren, V.T. Vu, M.I Pettersson, D. Murdin, A Gustavsson, L.M.H. Ulander, and Feng Wang. Forest clutter suppression for moving target detection in uhf dual channel sar. In *Synthetic Aperture Radar (APSAR), 2013 Asia-Pacific Conference on*, pages 581–584, Sept 2013.

[16] K. Sarabandi, M. Dehmollaian, and H. Mosallaei. Hybrid fdtd and single-scattering theory for simulation of scattering from hard targets camouflaged under forest canopy. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(8):2072–2082, Aug 2006. ISSN 0196-2892. doi: 10.1109/TGRS.2006.872091.

[17] Defense Update. Darpa's foliage-penetrating radar tested on the a160t hummingbird. URL http://defense-update.com/products/a/a160_forrester251009.html.

[18] Src Inc. Forester. URL http://www.srcinc.com/what-we-do/radar-and-sensors/forester-radar.html.

[19] Sami Bellez. *Contribution à l'analyse de la diffusion bistatique par un milieu forestier dans les bandes VHF et UHF à l'aide d'une représentation intégrale du champ électrique et de mesures en chambres anéchoïques*. PhD thesis, Paris 6, 2010.

[20] Thuy Le Toan, S Quegan, MWJ Davidson, Heiko Balzter, Philippe Paillou, K Papathanassiou, S Plummer, F Rocca, S Saatchi, H Shugart, et al. The biomass mission: Mapping global forest biomass to better understand the terrestrial carbon cycle. *Remote sensing of environment*, 115 (11):2850–2860, 2011.

[21] Huy Nguyen, Helene Roussel, and Walid Tabbara. A coherent model of forest scattering and sar imaging in the vhf and uhf-band. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(4): 838–848, 2006.

[22] Y. Ziade, H. Roussel, M. Lesturgie, and W. Tabbara. A coherent model of forest propagation - application to detection and localization of targets using the dort method. *Antennas*

and Propagation, IEEE Transactions on, 56(4):1048–1057, April 2008. ISSN 0018-926X. doi: 10.1109/TAP.2008.919150.

[23] Y. Ziade, H. Nguyen, H. Roussel, and W. Tabbara. Medeling of propagation of electromagnetic waves in forest media application to detection and localization of targets. In *Information and Communication Technologies, 2006. ICTTA '06. 2nd*, volume 2, pages 2140–2145, 2006. doi: 10.1109/ICTTA.2006.1684734.

[24] S. Bellez, C. Dahon, and H. Roussel. Analysis of the main scattering mechanisms in forested areas: An integral representation approach for monostatic radar configurations. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(12):4153–4166, 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2009.2023663.

[25] Tie-Jun Cui and Weng Cho Chew. Fast evaluation of sommerfeld integrals for em scattering and radiation by three-dimensional buried objects. *Geoscience and Remote Sensing, IEEE Transactions on*, 37(2):887–900, Mar 1999. ISSN 0196-2892. doi: 10.1109/36.752208.

[26] R. Mittra, V. V S Prakash, and J. Yeo. Some novel techniques for efficient analysis of large arrays and frequency selective radomes. In *Antennas and Propagation, 2003. (ICAP 2003). Twelfth International Conference on (Conf. Publ. No. 491)*, volume 2, pages 462–465 vol.2, 2003. doi: 10.1049/cp:20030112.

[27] Bijilash Babu. *Efficient Domain Decomposition Methods for the Electromagnetic Analysis of Dielectric Objects*. PhD thesis, Dublin City University, 2010.

[28] R. Mittra and K. Du. Characteristic basis function method for iteration-free solution of large method of moments problems. *Progress In Electromagnetics Research B*, 6:307–336, 2008.

[29] Rob Maaskant. *Analysis of Large Antenna Systems*. PhD thesis, Technische Universiteit Eindhoven, 2010.

[30] E. Lucente, A. Monorchio, and R. Mittra. An iteration-free mom approach based on excitation independent characteristic basis functions for solving large multiscale electromagnetic scattering problems. *Antennas and Propagation, IEEE Transactions on*, 56(4):999–1007, 2008. ISSN 0018-926X. doi: 10.1109/TAP.2008.919166.

[31] R. Coifman, V. Rokhlin, and S. Wandzura. The fast multipole method for the wave equation: a pedestrian prescription. *Antennas and Propagation Magazine, IEEE*, 35(3):7–12, 1993. ISSN 1045-9243. doi: 10.1109/74.250128.

[32] Jiming Song, Cai-Cheng Lu, and Weng Cho Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *Antennas and Propagation, IEEE Transactions on*, 45(10):1488–1493, 1997. ISSN 0018-926X. doi: 10.1109/8.633855.

[33] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. Aim: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems. *Radio Science*, 31(5):1225–1251, 1996. ISSN 1944-799X. doi: 10.1029/96RS02504. URL http://dx.doi.org/10.1029/96RS02504.

[34] Kezhong Zhao, Marinos N. Vouvakis, and J-F Lee. The adaptive cross approximation algorithm for accelerated method of moments computations of emc problems. *Electromagnetic Compatibility, IEEE Transactions on*, 47(4):763–773, Nov 2005. ISSN 0018-9375. doi: 10.1109/TEMC.2005.857898.

[35] P. Poschmann, L. Komzsik, and S. Mayer. Direct or iterative? a dilemma for the user. In *MSC Software, 1998 Americas Users' Conference*, pages 1–8, 1998. URL http://web.mscsoftware.com/support/library/conf/amuc98/index.cfm.

[36] Jaime Laviada, Raj Mittra, Marcos R Pino, and Fernando Las-Heras. On the convergence of the aca. *Microwave and Optical Technology Letters*, 51(10):2458–2460, 2009.

[37] T. Van, L.R.C. Suzuki, D. Latypov, J. Von Holle, T. Voss, T. Van, G. Wilson, G. Antilla, and M. Warren. Fast algebraic methods in computational electromagnetics. In *Aerospace and Electronics Conference (NAECON), Proceedings of the IEEE 2010 National*, pages 230–236, July 2010. doi: 10.1109/NAECON.2010.5712951.

[38] Noreen Jamil. A comparison of direct and indirect solvers for linear systems of equations. *International Journal of Emerging Sciences*, 2(2):310, 2012.

[39] R. Maaskant, R. Mittra, and A. Tijhuis. Fast analysis of large antenna arrays using the characteristic basis function method and the adaptive cross approximation algorithm. *Antennas and Propagation, IEEE Transactions on*, 56(11):3440–3451, Nov 2008. ISSN 0018-926X. doi: 10.1109/TAP.2008.2005471.

[40] Jaime Laviada, Marcos R. Pino, Raj Mittra, and Fernando Las-Heras. Parallelized multilevel characteristic basis function method for solving electromagnetic scattering problems. *Microwave and Optical Technology Letters*, 51(12):2963–2969, 2009. ISSN 1098-2760. doi: 10.1002/mop.24767.

[41] J. Laviada, F. Las-Heras, M.R. Pino, and R. Mittra. Solution of electrically large problems with multilevel characteristic basis functions. *Antennas and Propagation, IEEE Transactions on*, 57 (10):3189–3198, Oct 2009. ISSN 0018-926X. doi: 10.1109/TAP.2009.2028603.

[42] Eliseo Garcia, Carlos Delgado, Iván González Diego, and Manuel Felipe Catedra. An iterative solution for electrically large problems combining the characteristic basis function method and the multilevel fast multipole algorithm. *Antennas and Propagation, IEEE Transactions on*, 56(8): 2363–2371, 2008.

[43] B. Babu and R. Mittra. Electromagnetic analysis of dielectric objects using the characteristic basis function method. In *Applied Electromagnetics Conference (AEMC), 2011 IEEE*, pages 1–4, Dec 2011. doi: 10.1109/AEMC.2011.6256855.

[44] G. Tiberi, E Lucente, R. Mittra, and A. Monorchio. Characteristic basis function method for analyzing em scattering by large conducting structures with apertures. *Antennas and Propagation Society International Symposium, 2008. AP-S 2008; IEEE*, pages 1–4, July 2008. doi: 10.1109/APS.2008.4619430.

[45] A.D. Yaghjian. Sampling criteria for resonant antennas and scatterers. *Journal of Applied Physics*, 79(10):7474–7482, 1996. ISSN 0021-8979. doi: 10.1063/1.362683.

[46] Sami Bellez, Hélène Roussel, Cyril Dahon, and Jean-Michel Geffrin. A rigorous forest scattering model validation through comparison with indoor bistatic scattering measurements. *Progress In Electromagnetics Research B*, 33:1 – 19, September 2011. doi: 10.2528/PIERB11063009. URL http://hal.upmc.fr/hal-00628199.

[47] Jamal S. Izadian, L. Peters, and Jack H. Richmond. Computation of scattering from penetrable cylinders with improved numerical efficiency. *Geoscience and Remote Sensing, IEEE Transactions on*, GE-22(1):52–61, Jan 1984. ISSN 0196-2892. doi: 10.1109/TGRS.1984.350579.

[48] K.C. McDonald, R. Zimmermann, and J.S. Kimball. Diurnal and spatial variation of xylem dielectric constant in norway spruce (picea abies [l.] karst.) as related to microclimate, xylem sap flow, and xylem chemistry. *Geoscience and Remote Sensing, IEEE Transactions on*, 40(9): 2063–2082, Sep 2002. ISSN 0196-2892. doi: 10.1109/TGRS.2002.803737.

[49] L. Hess and David S. Simonett. Dielectric properties of two western coniferous tree species. In *Geoscience and Remote Sensing Symposium, 1990. IGARSS '90. 'Remote Sensing Science for the Nineties'., 10th Annual International*, pages 503–503, May 1990. doi: 10.1109/IGARSS.1990.688535.

[50] D.P. Buhl and R.L. Rogers. Experimental characterization of the forest as a medium for hf radio propagation. In *Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE*, pages 1–4, July 2008. doi: 10.1109/APS.2008.4619793.

[51] Yu Song Meng, Yee Hui Lee, and Boon Chong Ng. Further study of rainfall effect on vhf forested radio-wave propagation with four-layered model. *Progress In Electromagnetics Research*, 99:149–161, 2009.

[52] Anatoly Kalinkevich, Boris Kutuza, and Leo P. Ligthart. Some aspects on the interpretation of radar images of pine forest at p and vhf bands. In *Synthetic Aperture Radar, 2012. EUSAR. 9th European Conference on*, pages 603–606, April 2012.

[53] K.J. Ranson, B.N. Rock, W.A Salas, K. Smith, and D.L. Williams. Analysis of the dielectric properties of trunk wood in dominant conifer species from new england and siberia. In *Geoscience and Remote Sensing Symposium, 1992. IGARSS '92. International*, volume 2, pages 1283–1285, May 1992. doi: 10.1109/IGARSS.1992.578415.

[54] AW. Glisson and D. Wilton. Simple and efficient numerical methods for problems of electromagnetic radiation and scattering from surfaces. *Antennas and Propagation, IEEE Transactions on*, 28(5):593–603, Sep 1980. ISSN 0018-926X. doi: 10.1109/TAP.1980.1142390.

[55] H.A Wheeler. Formulas for the skin effect. *Proceedings of the IRE*, 30(9):412–424, Sept 1942. ISSN 0096-8390. doi: 10.1109/JRPROC.1942.232015.

[56] Gabriel Caloz, Monique Dauge, Erwan Faou, and Victor Péron. On the influence of the geometry on skin effect in electromagnetism. *Computer Methods in Applied Mechanics and Engineering*, 200(9):1053–1068, 2011.

[57] Sergej Rjasanow. Adaptive cross approximation of dense matrices. In *Int. Association Boundary Element Methods Conf., IABEM*, pages 28–30, 2002.

[58] Mario Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4): 565–589, 2000.

[59] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003. ISSN 0010-485X. doi: 10.1007/s00607-002-1469-6. URL http://dx.doi.org/10.1007/s00607-002-1469-6.

[60] Lei Chen, Yufa Sun, and Shuaishuai Yang. Fast calculation of bistatic rcs for conducting objects using the adaptive cross approximation algorithm. In *Antennas, Propagation EM Theory (ISAPE), 2012 10th International Symposium on*, pages 999–1002, Oct 2012. doi: 10.1109/ISAPE.2012.6408943.

[61] J. Shaeffer. Direct solve of electrically large integral equations for problem sizes to 1 m unknowns. *Antennas and Propagation, IEEE Transactions on*, 56(8):2306–2313, Aug 2008. ISSN 0018-926X. doi: 10.1109/TAP.2008.926739.

[62] Wenchun Lu, Jun Hu, and Zaiping Nie. Accelerating the cbfm solution of 3d electromagnetic scattering by using mlfma and aca. In *Microwave Technology Computational Electromagnetics (ICMTCE), 2011 IEEE International Conference on*, pages 453–456, May 2011. doi: 10.1109/ICMTCE.2011.5915557.

[63] Barney B. Lawrence Livermore National Laboratory. Introduction to parallel computing, Aug 2014. URL https://computing.llnl.gov/tutorials/parallel_comp/#DistributedMemory.

[64] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*, volume 1. MIT press, 1999.

[65] Jaime Laviada, Raj Mittra, Marcos R Pino, and Fernando Las-Heras. A parallel implementation of the multilevel characteristic basis function method. In *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, pages 50–54. IEEE, 2009.

[66] Fei Dai, Guodong Han, and Changqing Gu. Parallel adaptively modified characteristic basis function method based on static load balance. *Journal of Electronics (China)*, 26(4):532–536, 2009.

[67] Hany Abd-El-Raouf, Raj Mittra, and Ji-Fu Ma. Solving very large em problems (10 9 dofs or greater) using the mpi-cbfdtd method. In *Antennas and Propagation Society International Symposium, 2005 IEEE*, volume 2, pages 18–21. IEEE, 2005.

[68] DJ Ludick and DB Davidson. Investigating efficient parallelization techniques for the characteristic basis function method (cbfm). In *Electromagnetics in Advanced Applications, 2009. ICEAA'09. International Conference on*, pages 400–403. IEEE, 2009.

[69] Dimitri Lecas CNRS/IDRIS France. Produit de matrices. URL http://www.idris.fr/data/cours/parallel/mpi/mpi_tp_C/tp5/produit_matrices.c.

[70] Jaeyoung Choi, James Demmel, Inderjiit Dhillon, Jack Dongarra, Susan Ostrouchov, Antoine Petitet, Ken Stanley, David Walker, and R Clinton Whaley. Scalapack: A portable linear algebra library for distributed memory computers—design issues and performance. In *Applied Parallel*

Computing Computations in Physics, Chemistry and Engineering Science, pages 95–106. Springer, 1996.

[71] Mario Bebendorf. *Hierarchical matrices*. Springer, 2008.

[72] Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Hierarchical matrices. *Lecture notes*, 21:2003, 2003.

[73] Eric Darve. The fast multipole method: numerical implementation. *Journal of Computational Physics*, 160(1):195–240, 2000.

[74] Ozgur Ergul and Levent Gurel. A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm. *Antennas and Propagation, IEEE Transactions on*, 57 (6):1740–1750, 2009.

[75] Ö Ergül and L Gürel. Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics. *Electronics Letters*, 44(1):3–5, 2008.

[76] Ozgür Ergül and Levent Gürel. Hierarchical parallelization of the multilevel fast multipole algorithm for the efficient solution of large-scale scattering problems. In *Proc. IEEE Antennas Propag. Soc. Int. Symp*, 2008.

[77] Bart Michiels, Jan Fostier, Ignace Bogaert, and Daniël De Zutter. Weak scalability analysis of the distributed-memory parallel mlfma. 2013.

[78] A Heldring, J.M. Tamayo, C. Simon, E. Ubeda, and J.M. Rius. Sparsified adaptive cross approximation algorithm for accelerated method of moments computations. *Antennas and Propagation, IEEE Transactions on*, 61(1):240–246, Jan 2013. ISSN 0018-926X. doi: 10.1109/TAP.2012.2215292.

[79] M. Kanj, C. Dahon, H. Roussel, and B. Duchene. Forest remote sensing: Inversion using a coherent scattering model and a genetic algorithm. In *Radio Science Meeting (Joint with AP-S Symposium), 2013 USNC-URSI*, pages 193–193, July 2013. doi: 10.1109/USNC-URSI.2013.6715499.

[80] Xiao-Chun Nie, Ning Yuan, Le-Wei Li, Yeow-Beng Gan, and Tat Soon Yeo. A fast volume-surface integral equation solver for scattering from composite conducting-dielectric objects. *Antennas and Propagation, IEEE Transactions on*, 53(2):818–824, 2005.

Abstract : A 3-D full-wave model, based on the integral representation of the electric field and dedicated to the analysis of bi-static scattering mechanisms by a forest in the VHF and UHF bands was efficiently enhanced. In order to overcome the limitation of a previous 3D model to small simulations scenes and low frequencies, we have developed, during this research work, a new model using basis functions adapted to the problem of interest, in the context of the Characteristic Basis Function Method (CBFM) and we investigated the suitability of this direct method for computing the electromagnetic fields inside and outside three-dimensional dielectric objects representing the tree trunks and branches. The CBFM has shown great performances, when applied to the forest scattering modeling, both in terms of CPU time and memory resources needed. Once properly set, the CBFM-E is so efficient that it is able to treat in few minutes electromagnetic problems totally intractable with the classical MoM. Consequently, we have developed a powerful 3D forest electromagnetic scattering tool which allows us today to compute large forest electromagnetic problems in few minutes without worrying about the accuracy of the solution. On the other hand, we have demonstrated the efficiency and accuracy of the CBFM-E when applied to 3D dielectric objects in the context of the electric volumetric integral equation, and have consolidated thus its leading position in the computational electromagnetics, especially against the iterative solvers based numerical methods.

Keywords : Forest scattering modeling, VHF-UHF, Volumetric integral equation, Method of Moments, Characteristic Basis Function Method (CBFM), Adaptive Cross Approximation algorithm (ACA), High performance computing (OpenMP/MPI)

Résumé : Cette thèse porte sur la modélisation et l'analyse de la propagation électromagnétique dans un milieu forestier dans les bandes VHF et UHF. L'objectif principal est le développement d'un modèle numérique "full-wave" tridimensionnel de diffusion par la forêt permettant de caractériser l'interaction d'une onde électromagnétique avec un milieu forestier. Un tel modèle s'avère, actuellement, un outil indispensable à l'analyse des mesures radar pour l'étude des paramètres caractéristiques de la forêt tels que la biomasse forestière, la hauteur des arbres et leur densité. La complexité numérique de ce modèle a limité son domaine d'application à de petites parcelles de forêt et aux basses fréquences. Pour pouvoir traiter de larges zones forestières tout en montant en fréquence, et s'approcher ainsi des besoins et exigences des utilisateurs potentiels de notre modèle, nous avons intégré à ce modèle une méthode numérique efficace dédiée à l'analyse de larges problèmes électromagnétiques. La méthode en question, connue sous le nom de Characteristic Basis Function Method (CBFM) était récemment développée dans le laboratoire de Communication et Electromagnétisme de l'université PennState dirigé par le Professeur Mittra. Après une optimisation et une adaptation au problème d'intérêt, la CBFM réalise d'excellentes performances et nous permet une diminution considérable du temps de calcul et des besoins en espace mémoire sans pour autant dégrader la qualité des résultats obtenus ou altérer la fidélité du modèle à la réalité du problème électromagnétique traité.

Mots clés : Diffusion électromagnétique par la forêt; VHF–UHF; Equation intégrale en volume; Méthode des moments (MoM); CBFM; Algorithme ACA; Calcul haute performance (OpenMP/MPI)

List of publications

- Ines Fenni, Sami Bellez, Muriel Darces, Hélène Roussel, Raj Mittra, 'Efficient Domain Decomposition Methods for the analysis of 3D Large Dielectric Problems Based on CBFM' , *Numelec 2012, Marseille.*

- Ines Fenni, Hélène Roussel, Muriel Darces, Raj Mittra, 'Parallelization of a multilevel CBFM applied to a complex electromagnetic problem', *18th French National Microwave Days, JNM 2013, Paris.*

- Ines Fenni, Hélène Roussel, Muriel Darces, Raj Mittra, 'Parallelized Multilevel Characteristic Basis Function Method Applied to Scattering Model for Forest Remote Sensing', *APS/URSI 2013, Orlando, USA*

- Ines Fenni, Hélène Roussel, Muriel Darces, Raj Mittra, 'Application of the Characteristic Basis Function Method (CBFM) on a non-uniform mesh to the solution of large-size Dielectric Scattering Problems', Convened Session, *EuCAP 2014, The Hague, Netherlands.*

- Ines Fenni, Hélène Roussel, Muriel Darces, Raj Mittra, 'Efficient Parallelization of Extended Multi-Level CBFM for the Analysis of Large 3D Dielectric Scattering Problems', *CEFC 2014, Annecy, France.*

- Ines Fenni, Hélène Roussel, Muriel Darces, Raj Mittra, 'Fast Computing of Large 3D Dielectric Forest Scattering Problems Using the Characteristic Basis Function Method with the Adaptive Cross Approximation Algorithm', *APS/URSI 2014, Memphis, USA.*

- Fenni, I; Roussel, H.; Darces, M.; Mittra, R., 'Fast Analysis of Large 3-D Dielectric Scattering Problems Arising in Remote Sensing of Forest Areas Using the CBFM,' *Antennas and Propagation, IEEE Transactions on , vol.62, no.8, pp.4282,4291, Aug. 2014.*