Progress In Electromagnetics Research M, Vol. 75, 79–90, 2018

Hybrid Cross Approximation for the Electric Field Integral Equation

Priscillia Daquin, Ronan Perrussel, and Jean-René Poirier*

Abstract—The boundary element method is considered for solving scattering problems and is accelerated using the hierarchical matrix format. For this format, some matrix blocks, whose choice is based on geometrical criteria, are approximated by low-rank matrices using a robust compression method. In this paper, we validate the use of the hybrid cross approximation which is quite new in this area, and we apply it to several examples such as the scattering by a conducting sphere, by a rough (Weierstrass) surface and by a plane.

1. INTRODUCTION

In electromagnetics, it is common to consider the boundary element method (BEM) to solve the Maxwell equations describing scattering problems by a conducting surface. This method leads to an *a priori* algorithmic complexity in $\mathcal{O}(N^2)$, with N being the number of degrees of freedom (dofs). This number is related to the discretization step of the mesh, and it is commonly admitted that the mesh of a given surface needs to have a discretization step smaller than $\lambda/10$, with λ being the wavelength of the incident electromagnetic wave. This discretization step can even be smaller when the studied surface is not smooth, as the case with rough surfaces such as the sea. As a result, such simulations can become crippling with the increasing number of dofs.

In this paper, we consider the hierarchical matrix ($\mathcal{H}$) format to lessen the complexity of the BEM for the electric field integral equation (EFIE). Some chosen blocks of the matrix are approximated by low-rank matrices (r_k-matrices) using a compression technique, and consequently the overall numerical complexity is expected to be in $\mathcal{O}(N\log(N))$ instead of $\mathcal{O}(N^2)$. If it is now customary to use the fast multipole method (FMM) [1] or the adaptive cross approximation (ACA) [2] as a compression method to compute those r_k-matrices, the hybrid cross approximation (HCA) is still rarely used even though it has advantages over both the FMM and ACA [3]. We use the combination of the $\mathcal{H}$ format and the HCA to solve the EFIE in 3D. We then show the enhancement of the numerical complexity brought by this method as well as its validity. We finally apply this formalism to treat more challenging examples.

2. ELECTRIC FIELD INTEGRAL EQUATION (EFIE)

In this work, we consider the scattering of an incident electromagnetic wave by a perfectly conducting surface Γ. Let us denote $\mathbf{E}^{\text{inc}}$ as the incident electric field of frequency $f = \omega/2\pi$ and wavelength $\lambda = c/f$, and c as the speed of light in vacuum. The incident field $\mathbf{E}^{\text{inc}}$ induces surface currents $\mathbf{J}$ on Γ. The EFIE is written as [1]

$$\left(\nabla_{\text{r}}\frac{1}{\text{i}\omega_0}\int_\Gamma G(\mathbf{x}_\Gamma,\mathbf{y}_\Gamma)\nabla_{\text{r}}\cdot\mathbf{J}(\mathbf{y}_\Gamma)\text{d}\mathbf{y}_\Gamma - \text{i}\omega_0\mu_0\int_\Gamma G(\mathbf{x}_\Gamma,\mathbf{y}_\Gamma)\mathbf{J}(\mathbf{y}_\Gamma)\text{d}\mathbf{y}_\Gamma\right)\wedge\mathbf{n} = -\mathbf{E}^{\text{inc}}(\mathbf{x}_\Gamma)\wedge\mathbf{n}. \quad (1)$$

Received 28 May 2018, Accepted 29 July 2018, Scheduled 31 October 2018

* Corresponding author: Jean-René Poirier (poirier@laplace.univ-tlse.fr).
The authors are with the LAPLACE, Université de Toulouse, CNRS, Toulouse, France.

The discretization of this formulation leads to solving the linear system

$$ZI = V, \tag{2}$$

with $Z \in \mathbb{C}^{N\times N}$ the impedance matrix, $I \in \mathbb{C}^{N\times P}$ the unknown induced currents and $V \in \mathbb{C}^{N\times P}$ the right-hand side representing the excitation vectors, N the number of dofs, and P the number of right-hand sides ($P \ll N$). Those right-hand sides differ from each other by the value of the polarization direction of incident electric field $\mathbf{E}^{\text{inc}}$.

In the case of the EFIE with a triangular mesh, we get for $1 \leq i, j \leq N$ and $1 \leq k \leq P$ [1, 4]:

$$\begin{cases} z_{ij} = \displaystyle\int_{\Gamma\times\Gamma} g(\mathbf{x}_\Gamma, \mathbf{y}_\Gamma) \left(\mathbf{f}_i(\mathbf{y}_\Gamma) \cdot \mathbf{f}_j(\mathbf{x}_\Gamma) - \frac{1}{k_0^2} \nabla_{\mathrm{r}} \cdot \mathbf{f}_i(\mathbf{y}_\Gamma) \nabla_{\mathrm{r}} \cdot \mathbf{f}_j(\mathbf{x}_\Gamma) \right) \mathrm{d}\mathbf{y}_\Gamma \mathrm{d}\mathbf{x}_\Gamma, & (3) \\ e_{jk} = \dfrac{\mathrm{i}}{\omega\mu_0} \displaystyle\int_\Gamma \mathbf{E}_k^{\text{inc}}(\mathbf{x}_\Gamma) \cdot \mathbf{f}_j(\mathbf{x}_\Gamma) \mathrm{d}\mathbf{x}_\Gamma. & (4) \end{cases}$$

where $\mathbf{f}_i$ is the basis function regarding the ith edge of the mesh as defined by Rao et al. [4], $g = G$ the kernel function, and $\mathbf{E}_k^{\text{inc}}$ the incident electric field polarized in the kth direction.

Matrix Z *a priori* requires a storage and a matrix-vector product complexity proportional to N^2. To lessen this complexity, the kernel g can be locally approximated by a degenerated kernel $\tilde{g}$, leading to the building of a data-sparse matrix stored in the $\mathcal{H}$ format.

3. HIERARCHICAL MATRICES ($\mathcal{H}$-MATRICES)

3.1. Clustering

The $\mathcal{H}$ representation [5] is a data-sparse matrix format that enables the memory storage and the complexity of the arithmetic operations, as the matrix-vector product, to scale as $N\log(N)$ instead of N^2. Consequently, it allows solving large-scale problems efficiently using the BEM.

The hierarchy is defined by a clustering technique of the dofs, which is performed recursively by successive bisections of the boxes bounding different dofs, until each bounding box contains a maximum of $n_{\max}$ dofs. This leads to a binary tree for the row and column indices of the matrix and a quadtree embodying the hierarchical matrix block structure.

The leaves are sorted as they do or do not respond to an admissibility condition over the distance between interacting clusters. A leaf is said admissible (resp. non admissible) if it represents "far" (resp. "near") interactions. The admissible leaves correspond to admissible matrix blocks that can be approached by r_k-matrices without a damaging loss of accuracy, while non-admissible blocks are not to be compressed.

3.2. Compression and Coarsening of a $\mathcal{H}$

Let us consider an admissible block $A|_{\hat{r}\times\hat{c}}$ of a $\mathcal{H}$ A, with $\hat{r}$ and $\hat{c}$ being the sets of dofs indices in two distant clusters. We can compress $A|_{\hat{r}\times\hat{c}}$ by replacing the kernel g with a degenerate kernel $\tilde{g}$ or by directly approximating $A|_{\hat{r}\times\hat{c}} \approx \tilde{A}|_{\hat{r}\times\hat{c}}$. Different approaches exist to achieve this aim, such as the FMM [1] or the ACA [6]. This last approach leads to a low-rank approximation of precision ε_{ACA} as

$$A|_{\hat{r}\times\hat{c}} \approx UV^H, \tag{5}$$

where $A|_{\hat{r}\times\hat{c}} \in \mathbb{C}^{m\times n}$, $U \in \mathbb{C}^{m\times k}$, $V \in \mathbb{C}^{n\times k}$, and k is the rank of the approximation.

It is then possible to lower the rank of the approximation UV^H by applying a truncated singular value decomposition which is adapted to format (5) (r_k-SVD), followed by a truncation at precision ε_{ACA}. This gives

$$A|_{\hat{r}\times\hat{c}} \approx \tilde{A}|_{\hat{r}\times\hat{c}} = U_{k'} X_{k'} V_{k'}^H, \tag{6}$$

with $U_{k'} \in \mathbb{C}^{m\times k'}$, $V_{k'} \in \mathbb{C}^{n\times k'}$, $X_{k'} \in \mathbb{R}^{k'\times k'}$ the diagonal matrix of the dominant singular values, and k' the new rank of the approximation $\tilde{A}|_{\hat{r}\times\hat{c}}$, $k' \leq k$.

Format (6) requires less memory than format (5) as soon as $k' < k$, and allows to obtain the Frobenius norm of $\tilde{A}|_{\hat{r}\times\hat{c}}$ quickly as it equals the Frobenius norm of the diagonal real matrix $X_{k'}$. Considering format (6) for the r_k-matrix is moreover useful for the coarsening explained below.

Once $\mathcal{H}$ is assembled, we can coarsen it by applying an algorithm consisting in the recursive agglomeration of two r_k-matrices into one r_k-matrix with precision $\varepsilon_{coars.}$ that controls the overall accuracy of the $\mathcal{H}$ [7]. It allows for simplifying the hierarchical structure of the $\mathcal{H}$ (see Figure 1) and lowering again its memory storage (see Figure 2).

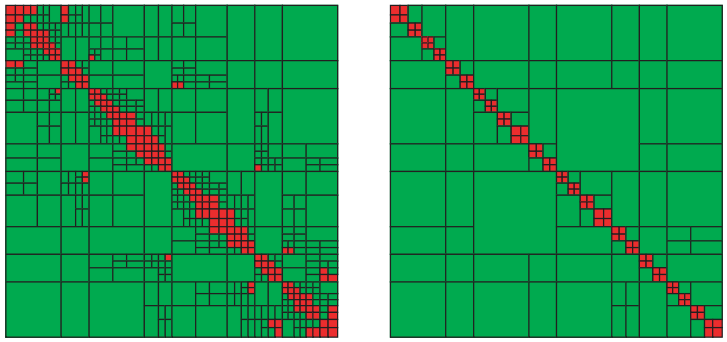

Figure 1. Hierarchical representation of the impedance matrix in the case of an incident wave ($f = 3\,\text{GHz}$) scattered by a metallic cylinder ($r = 0.1\,\text{m} = \lambda/100$), with 937 dofs, before and after coarsening ($\varepsilon_{coars.} = 10^{-14}$). (a) Before coarsening. (b) After coarsening.

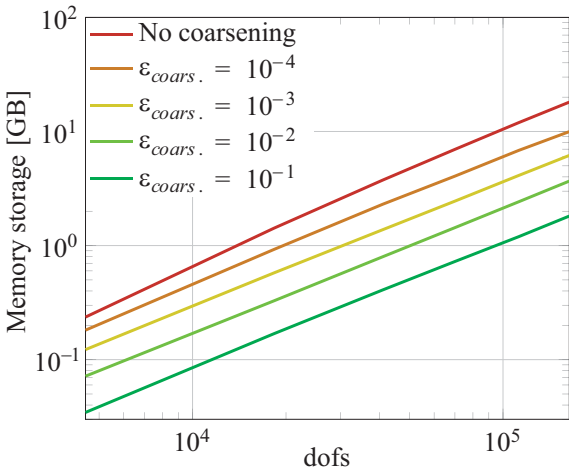

Figure 2. Memory storage of a $\mathcal{H}$ for different values of $\varepsilon_{coars.}$.

3.3. Arithmetic of $\mathcal{H}$-Matrices

As the $\mathcal{H}$ structure is decomposed into blocks of different sizes and formats, a specific arithmetic has been implemented to enable the solution of the linear system (2). It consists in classical arithmetic operations computed recursively within the levels of the hierarchy. The use of r_k-matrices for admissible leaves leads to approximations controlled by the same precision ε which was used for the compression.

The resulting arithmetic includes a $\mathcal{H}$-vector product, an approximate sum of two $\mathcal{H}-matrices$ and a formatted multiplication [6]. The algorithmic complexity of all these operations scales as $N\log(N)$ [8]. The computation time will be further enhanced on coarsened $\mathcal{H}-matrices$ than on $\mathcal{H}-matrices$ which are only locally compressed.

These operations allow for developing a $\mathcal{H}$-LU decomposition, which is an approximate recursive LU decomposition of a $\mathcal{H}$, computed with a prescribed precision ε_{LU} [8, 9].

4. HYBRID CROSS APPROXIMATION (HCA)

4.1. Principles

This method can be defined for a kernel function g such as

$$g(\mathbf{x}, \mathbf{y}) = D_{\mathbf{x}} D_{\mathbf{y}} \gamma(\mathbf{x}, \mathbf{y}), \tag{7}$$

where operators $D_{\mathbf{x}}$ and $D_{\mathbf{y}}$ are differential operators respectively according to $\mathbf{x}$ and $\mathbf{y}$, and γ is an asymptotically smooth function, especially being well suited for the ACA computation [6].

The HCA, first introduced in [3], is an algorithm based on the cross approximation at a functional level, i.e., directly on γ. It differs from the ACA, where the cross approximation is applied at the matrix level. The HCA is more dependent on the formulation than the ACA but does not require to know and use an explicit kernel expansion as for the classical FMM. Thus it can be applied to many kernels and discretization techniques with a slight adaptation.

The coefficients of matrix Z built with the Galerkin method are defined for $1 \leq i, j \leq N$ as

$$z_{ij} = \int_{\Gamma \times \Gamma} \varphi_i(\mathbf{x}) g(\mathbf{x}, \mathbf{y}) \psi_j(\mathbf{y}) \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y}, \tag{8}$$

g being the kernel function, ϕ_i the i^{th} basis function and ψ_j an appropriate test function.

For a pair of admissible subdomains $(\Omega_{\mathcal{I}}, \Omega_{\mathcal{J}})$, $\mathcal{I}$ (resp. $\mathcal{J}$) being the set of indexes of the dofs contained in $\Omega_{\mathcal{I}}$ (resp. $\Omega_{\mathcal{J}}$), with $n_{\mathcal{I}} = |\mathcal{I}|$ (resp. $n_{\mathcal{J}} = |\mathcal{J}|$) basis functions over $\Omega_{\mathcal{I}}$ (resp. $\Omega_{\mathcal{J}}$), we consider a degenerate kernel function $\tilde{g}$ to define an approximation $\tilde{Z}$ of Z as

$$\tilde{z}_{ij} = \int_{\Gamma \times \Gamma} \varphi_i(\mathbf{x}) \tilde{g}(\mathbf{x}, \mathbf{y}) \psi_j(\mathbf{y}) \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{y}. \tag{9}$$

4.1.1. Choice of the Pivots

The first step of the HCA consists in selecting pivot points belonging to $\mathcal{B}_{\mathcal{I}} \times \mathcal{B}_{\mathcal{J}}$, with $\mathcal{B}_{\mathcal{I}}$ (resp. $\mathcal{B}_{\mathcal{J}}$) being the bounding box of $\Omega_{\mathcal{I}}$ (resp. $\Omega_{\mathcal{J}}$). For this purpose it is possible to apply the ACA on a matrix whose entries are evaluations of γ on couples of points belonging to $\mathcal{B}_{\mathcal{I}} \times \mathcal{B}_{\mathcal{J}}$. These points can be randomly chosen points in the boxes $\mathcal{B}_{\mathcal{I}}$ and $\mathcal{B}_{\mathcal{J}}$, or coincide with interpolation points in the same boxes. Here we choose to consider the tensorized Chebyshev interpolation points of the bounding boxes $\mathcal{B}_{\mathcal{I}}$ and $\mathcal{B}_{\mathcal{J}}$ as in [3, 10].

To be more specific, let us consider $m_C \in \mathbb{N}^*$. The m_C^3 Chebyshev interpolation points of a bounding box $\mathcal{B}$ are defined as $\varphi_{ijk}^{(C)} = (p_i^{(C)}, p_j^{(C)}, p_k^{(C)})$ where $p_i^{(C)}$ is the i^{th} Chebyshev node of order m_C for $0 \leq i \leq m_C - 1$ (see Figure 3). Applying the ACA with a precision $\varepsilon_{\text{HCA}} = \varepsilon_{\text{ACA}}$ on those points in the bounding boxes $\mathcal{B}_{\mathcal{I}}$ and $\mathcal{B}_{\mathcal{J}}$ provides two lists of pivot points $(\mathbf{x}_i)_{i=1}^k$ in $\mathcal{B}_{\mathcal{I}}$ and $(\mathbf{y}_j)_{j=1}^k$ in $\mathcal{B}_{\mathcal{J}}$, with k the approximation rank provided by the ACA.

4.1.2. Separation of Variables

We choose a first pivot point $(\mathbf{x}_1, \mathbf{y}_1)$ such as $\gamma(\mathbf{x}_1, \mathbf{y}_1) \neq 0$, and we define function γ_1, a first approximation of γ, as

$$\gamma_1(\mathbf{x}, \mathbf{y}) = \frac{\gamma(\mathbf{x}, \mathbf{y}_1) \gamma(\mathbf{x}_1, \mathbf{y})}{\gamma(\mathbf{x}_1, \mathbf{y}_1)}. \tag{10}$$

We note $r_1 = \gamma - \gamma_1$ the residue of this approximation, and we define, choosing a second pivot point $(\mathbf{x}_2, \mathbf{y}_2)$, a new approximation γ_2 of γ as

$$\gamma_2(\mathbf{x}, \mathbf{y}) = \gamma_1(\mathbf{x}, \mathbf{y}) + \frac{r_1(\mathbf{x}, \mathbf{y}_2) r_1(\mathbf{x}_2, \mathbf{y})}{r_1(\mathbf{x}_2, \mathbf{y}_2)}, \tag{11}$$

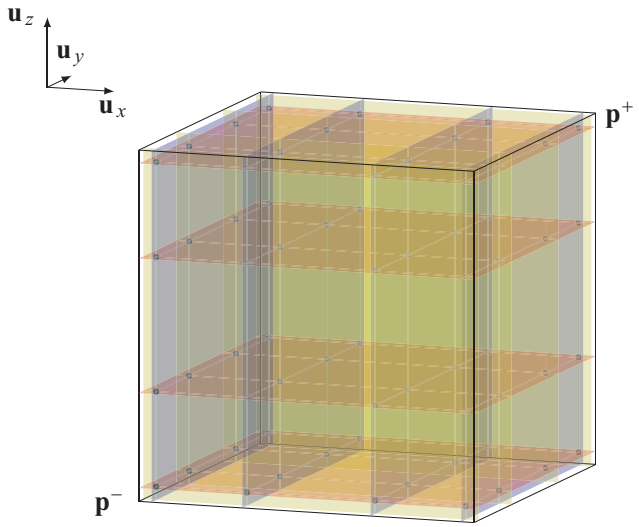

Figure 3. Grid of Chebyshev interpolation points on a cubic bounding box $\mathcal{B}$ ($m_C = 4$). Points $\mathbf{p}^+$ and $\mathbf{p}^-$ define the bounding box $\mathcal{B}$.

of residue $r_2 = \gamma - \gamma_2$. Repeating this process until the norm of the residue r_k is lower than ε_{HCA} leading to a degenerate kernel $\tilde{g}$ that can be written as $\tilde{g}(\mathbf{x},\mathbf{y}) = D_{\mathbf{x}} D_{\mathbf{y}} \tilde{\gamma}(\mathbf{x},\mathbf{y})$ with

$$\tilde{\gamma}(\mathbf{x},\mathbf{y}) = \begin{pmatrix} \gamma(\mathbf{x},\mathbf{y}_1) \\ \vdots \\ \gamma(\mathbf{x},\mathbf{y}_k) \end{pmatrix}^T M^{-1} \begin{pmatrix} \gamma(\mathbf{x}_1,\mathbf{y}) \\ \vdots \\ \gamma(\mathbf{x}_k,\mathbf{y}) \end{pmatrix} \tag{12}$$

where matrix $M = (m_{ij})_{1\leq i,j\leq k}$ is defined as $m_{ij} = \gamma(\mathbf{x}_i,\mathbf{y}_j)$, $(\mathbf{x}_i)_{i=1}^k$, and $(\mathbf{y}_j)_{j=1}^k$ is the pivot points resulting from the ACA algorithm proposed in Section 4.1.1.

The LU decomposition of M as $M = LU$ gives $M^{-1} = CD^T$ with $C = U^{-1} = (c_{ij})_{1\leq j\leq i\leq k}$ and $D = L^{-T} = (d_{ij})_{1\leq j\leq i\leq k}$. Thus we get

$$\tilde{\gamma}(\mathbf{x},\mathbf{y}) = \sum_{m=1}^{k}\sum_{n=1}^{k}\left(\sum_{l=1}^{k} c_{ml}d_{nl}\right)\gamma(\mathbf{x},\mathbf{y}_m)\gamma(\mathbf{x}_n,\mathbf{y}), \tag{13}$$

and then

$$\tilde{g}(\mathbf{x},\mathbf{y}) = \sum_{l=1}^{k}\left(\sum_{m=1}^{k} c_{ml} D_{\mathbf{x}}\gamma(\mathbf{x},\mathbf{y}_m)\right)\left(\sum_{n=1}^{k} d_{nl} D_{\mathbf{y}}\gamma(\mathbf{x}_n,\mathbf{y})\right). \tag{14}$$

Hence

$$\tilde{z}_{ij} = \sum_{l=1}^{k}\left(\sum_{m=1}^{k} c_{ml}\int_{\Gamma}\varphi_i(\mathbf{x}) D_{\mathbf{x}}\gamma(\mathbf{x},\mathbf{y}_m)\mathrm{d}\mathbf{x})\right)\left(\sum_{n=1}^{k} d_{nl}\int_{\Gamma}\psi_j(\mathbf{y}) D_{\mathbf{y}}\gamma(\mathbf{x}_n,\mathbf{y})\mathrm{d}\mathbf{y}\right). \tag{15}$$

Defining matrices $A = (a_{im})_{1\leq i\leq|\mathcal{I}|,1\leq m\leq k}$ and $B = (b_{jn})_{1\leq j\leq|\mathcal{J}|,1\leq n\leq k}$ such as

$$\begin{cases} a_{im} = \displaystyle\int_{\Gamma}\phi_i(\mathbf{x}) D_{\mathbf{x}}\gamma(\mathbf{x},\mathbf{y}_m)\mathrm{d}\mathbf{x}, & (16) \\ b_{jn} = \displaystyle\int_{\Gamma}\psi_j(\mathbf{y}) D_{\mathbf{y}}\gamma(\mathbf{x}_n,\mathbf{y})\mathrm{d}\mathbf{y}, & (17) \end{cases}$$

leads to

$$\tilde{z}_{ij} = \sum_{l=1}^{k}\left(\sum_{m=1}^{k} a_{im}c_{ml}\right)\left(\sum_{n=1}^{k} b_{jn}d_{nl}\right). \tag{18}$$

In other words, $\tilde{Z}$ is a r_k-matrix in format (5) approaching Z, as $\tilde{Z} = UV^H$ with $U = AC$ and $V = (BD)^H$. In order to obtain it in format (6) we only need to compute an r_k-SVD.

4.2. Application of the HCA Principles to the EFIE

The EFIE formulation is composed of two double integrals: the basis functions are vectorial in one of them while they are scalar in the other. Let us consider Eq. (3), denote $\gamma(\mathbf{x},\mathbf{y}) = g(\mathbf{x},\mathbf{y}) = G(\mathbf{x},\mathbf{y})$ and define matrices $\dot{A} \in \mathbb{C}^{|\mathcal{I}|\times k}$, $\dot{B} \in \mathbb{C}^{|\mathcal{J}|\times k}$, $A_p \in \mathbb{C}^{|\mathcal{I}|\times k}$, and $B_p \in \mathbb{C}^{|\mathcal{J}|\times k}$ with $p \in \{x,y,z\}$ whose coefficients are

$$\begin{cases} \dot{a}_{im} = \displaystyle\int_\Gamma \nabla_{\mathrm{r}} \mathbf{f}_i(\mathbf{x}) g(\mathbf{x},\mathbf{y}_m) \mathrm{d}\mathbf{x}, & (19) \\ \dot{b}_{jn} = \displaystyle\int_\Gamma \nabla_{\mathrm{r}} \mathbf{f}_j(\mathbf{y}) g(\mathbf{x}_n,\mathbf{y}) \mathrm{d}\mathbf{y}, & (20) \\ a_{im,p} = \left(\displaystyle\int_\Gamma \mathbf{f}_i(\mathbf{x}) g(\mathbf{x},\mathbf{y}_m) \mathrm{d}\mathbf{x}\right) \cdot \mathbf{u}_p, & (21) \\ b_{jn,p} = \left(\displaystyle\int_\Gamma \mathbf{f}_j(\mathbf{y}) g(\mathbf{x}_n,\mathbf{y}) \mathrm{d}\mathbf{y}\right) \cdot \mathbf{u}_p, & (22) \end{cases}$$

where $(\mathbf{u}_p)_{p\in\{x,y,z\}}$ is the canonical vector basis of $\mathbb{R}^3$. We have

$$\tilde{z}_{ij} = \sum_{l=1}^{k} \left(\sum_{p\in\{x,y,z\}} \left(\sum_{m=1}^{k} a_{im,p} c_{ml} \right) \left(\sum_{n=1}^{k} b_{jn,p} d_{nl} \right) - \frac{1}{k_0^2} \left(\sum_{m=1}^{k} \dot{a}_{im} c_{ml} \right) \left(\sum_{n=1}^{k} \dot{b}_{jn} d_{nl} \right) \right). \quad (23)$$

Hence

$$\tilde{Z} = \sum_{p\in\{x,y,z\}} U_p V_p^H + \dot{U}\dot{V}^H \quad (24)$$

with $U_p = A_p C \in (\mathbb{C})^{|\mathcal{I}|\times k}$, $V_p = (B_p D)^H \in (\mathbb{C})^{|\mathcal{J}|\times k}$, $\dot{U} = -1/k_0^2 \dot{A}C \in \mathbb{C}^{|\mathcal{I}|\times k}$ and $\dot{V} = (\dot{B}D)^H \in \mathbb{C}^{|\mathcal{J}|\times k}$. Building matrices $U \in \mathbb{C}^{|\mathcal{I}|\times 4k}$ and $V \in \mathbb{C}^{|\mathcal{J}|\times 4k}$ such as

$$\begin{cases} \hat{U} = \begin{bmatrix} U_x & U_y & U_z & \dot{U} \end{bmatrix}, & (25) \\ \hat{V} = \begin{bmatrix} V_x & V_y & V_z & \dot{V} \end{bmatrix}, & (26) \end{cases}$$

we finally get

$$\tilde{Z} = UV^H. \quad (27)$$

4.3. Setting of $\varepsilon_{\mathrm{HCA}}$ and m_C

We applied the HCA algorithm on the case of a conducting sphere of radius $r = 1\,\mathrm{m}$ at the frequency $f = 300\,\mathrm{MHz}$ with several numbers of dofs. On one hand, we set parameter $\varepsilon_{\mathrm{HCA}}$ to 10^{-5} — which is "very" accurate [3] — in order to evaluate the influence of the choice of the parameter m_C on the matrix precision. On the other hand, we set parameter m_C to 5 and vary the value of $\varepsilon_{\mathrm{HCA}}$. In both cases the matrix accuracy is evaluated performing a $\mathcal{H}$-vector product, which does not induce any other approximation than those from the $\mathcal{H}$ compression. More precisely, a random vector $\mathbf{x}$ is considered, and we perform the matrix-vector product $Z\mathbf{x} = \mathbf{b}$ and the $\mathcal{H}$-vector product $\tilde{Z}\mathbf{x} = \tilde{\mathbf{b}}$; the "accuracy" corresponds to $\|\mathbf{b} - \tilde{\mathbf{b}}\|_2$. The results of these simulations can be seen in Figures 4 and 5.

These graphics validate the method accuracy. Figure 5 shows that for $\varepsilon_{\mathrm{HCA}} = 10^{-5}$, the relative error is really similar for $m_C = 4$ and $m_C = 5$, but the value $m_C = 3$ is already sufficient to obtain an error inferior to $\varepsilon_{\mathrm{HCA}}$. Thus for the particular value $\varepsilon_{\mathrm{HCA}} = 10^{-5}$, $m_C = 4$ seems to be a sufficient value.

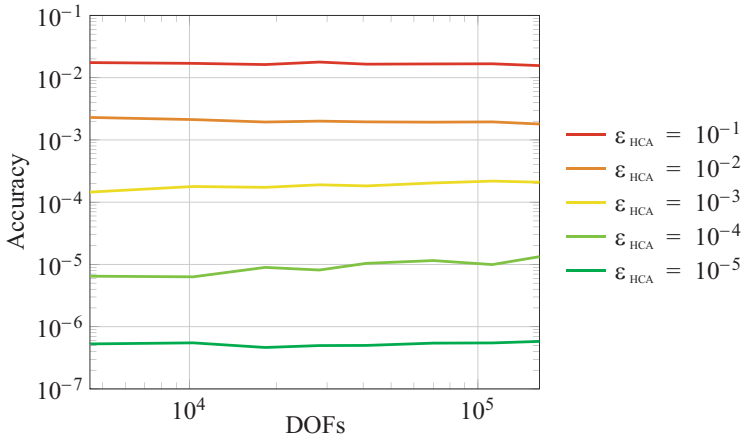

Figure 4. Accuracy of the HCA for different values of ε_{HCA} ($m_C = 5$).

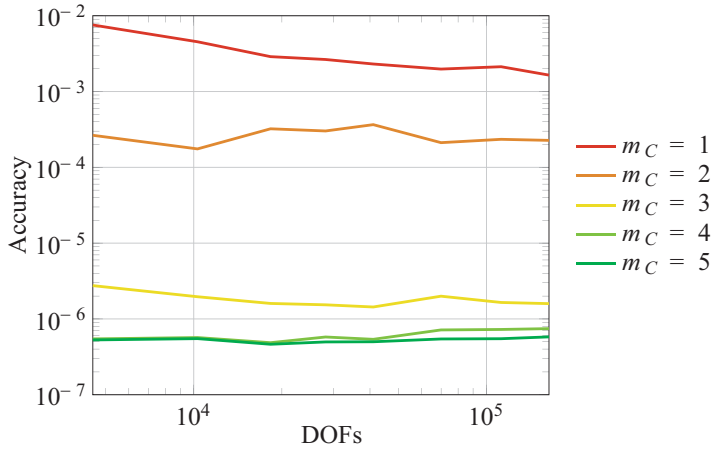

Figure 5. Accuracy of the HCA for different values of m_C ($\varepsilon_{\text{HCA}} = 10^{-5}$).

4.4. Time Repartition

The HCA algorithm can be divided into several steps: the building of $\tilde{\gamma}$ consisting in the ACA on the pivot points and a LU factorization (step 1), the building of matrices $\dot{A}$, $\dot{B}$, A_p, and B_p consisting in numerical integrations (step 2), the building of U and V^H consisting in matrix multiplications (step 3) and finally the r_k-SVD (step 4). Each of those steps depends differently on ε_{HCA}, m_C or both. We can evaluate their time repartition within the HCA algorithm for different values of N, as presented in Table 1 ($\varepsilon_{\text{HCA}} = 10^{-4}$, $m_C = 4$).

Table 1. Time ratio of each step within the HCA algorithm.

	$N = 4521$		$N = 18394$		$N = 69930$	
	t (s)	% ttot.	t (s)	% ttot.	t (s)	% ttot.
Building of $\tilde{\gamma}$ (step 1)	8.100	9.49	45.884	8.01	153.388	7.23
Integrations (step 2)	41.468	48.59	300.116	52.41	1223.888	57.67
Multiplications (step 3)	1.608	1.88	9.048	1.58	33.296	1.57
r_k-SVD (step 4)	34.176	40.04	217.540	37.99	711.484	33.53

This table allows us to notice that most of the HCA time is spent in the integration step. This finding is interesting, knowing that the HCA algorithm computes only simple integrals, versus double integrals for the ACA. We will be able to check this advantage over the ACA in the next section.

5. COMPUTATIONAL PERFORMANCES AND COMPARISON WITH ACA

We compare the performances of the HCA with the usual compression method ACA in terms of compression rate and computational time. The linear system is solved with the generalized minimal residual method (GMRES [11]) with the tolerance $\varepsilon_{\text{GMRES}}$. In this case the complexity of an iteration is $N\log(N)$ as for the $\mathcal{H}$-vector product. To improve the convergence, a right preconditioner is used and computed by the $\mathcal{H}$-LU decomposition with accuracy $\varepsilon_{\text{prec}}$. In order to build this preconditioner, we first coarsen the initial $\mathcal{H}$ with the precision $\varepsilon_{\text{prec}}$ and we apply afterwards a $\mathcal{H}$-LU decomposition on this simplified $\mathcal{H}$ with the same precision.

5.1. A Canonical Example: The Metallic Sphere

We first consider a metallic sphere of radius 1 m. We compute the relative error on the bistatic radar cross-section (RCS) between the numerical solution of the problem (2) using the ACA or the HCA algorithm and the analytic solution provided by Mie series. We set the value of ε_{HCA} according to m_C as $\varepsilon_{\text{HCA}} = 10^{-m_C}$, and $\varepsilon_{\text{GMRES}}$ to 10^{-12} in order to focus on the error due to the compression techniques.

It is shown on Figure 6 that the two methods have roughly the same behavior following the convergence rate of the BEM up to reach the desired accuracy given for the compression method (ε_{ACA} or ε_{HCA}).

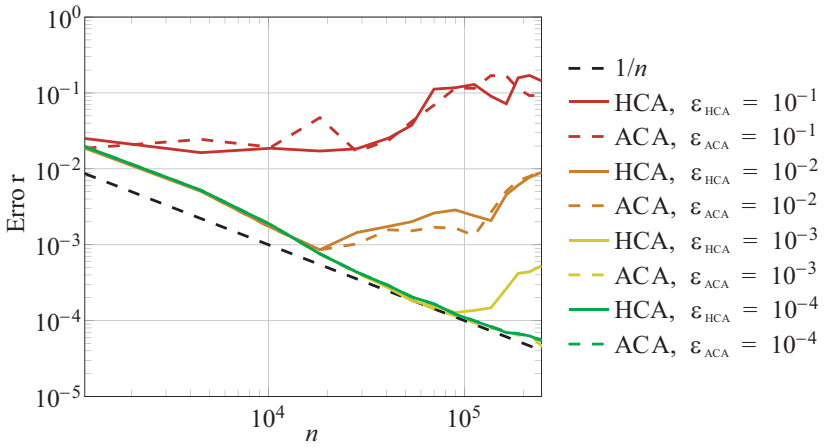

Figure 6. Relative error on bistatic RCS $|\sigma - \sigma_{\text{ref}}|/|\sigma_{\text{ref}}|$ ($f = 0.6\,\text{GHz}$).

We focus now on the compression rate, i.e., the ratio between the storages for the data-sparse format and for the full format, in Table 2. A mesh with 69930 unknowns is considered, and simulations are performed for frequencies 0.6 GHz and 1.2 GHz corresponding respectively to 20 and 10 points per wavelength. The involved parameters are $\varepsilon_{\text{ACA}} = \varepsilon_{\text{HCA}} = \varepsilon_{\text{GMRES}} = 10^{-4}$, $m_C = 4$, and $\varepsilon_{\text{prec}} = 10^{-2}$.

In Table 2 it is shown that

- the compression rates are roughly equivalent, and they are "very" close for the preconditioning matrix (which corresponds to a strong compression keeping only large singular values for the r_k-matrices);
- assembly rate seems to depend more on BEM properties (value of n_G) than on the compression method used;
- as expected the compression rate is better for low frequencies.

Table 2. Compression rates for the metallic sphere.

	$\lambda/10$ mesh				$\lambda/20$ mesh			
	ACA		HCA		ACA		HCA	
n_G	1	3	1	3	1	3	1	3
Assembly	11.02%	9.70%	11.02%	9.71%	8.90%	7.96%	8.90%	7.97%
Coarsening	7.19%	6.24%	7.23%	6.38%	5.18%	4.65%	5.55%	5.08%
Coarsening at $\varepsilon_{\text{prec}}$	3.94%	3.92%	3.94%	3.92%	2.18%	2.18%	2.18%	2.18

The corresponding computation times are provided in Table 3 for frequency 0.6 GHz.

Table 3. Computation times (s) for the metallic sphere.

	ACA		HCA	
n_G	1	3	1	3
Assembly	1275	4618	1415	2825
Coarsening	612	476	746	545
Coarsening at $\varepsilon_{\text{prec}}$	246	115	162	135
$\mathcal{H}$-LU decomposition	1397	870	998	919
GMRES	11.92	12.46	17.44	12.77

Following the observed compression rate and the time partition properties of both methods (see Table 1), we observe an advantage for the HCA in terms of assembly time when the number of Gauss-Legendre integration points increases which is the case when a good accuracy is desired. This result illustrates the advantage of the HCA in this case.

5.2. More Challenging Examples

As in the previous subsection, we set: $\varepsilon_{\text{ACA}} = \varepsilon_{\text{HCA}} = \varepsilon_{\text{GMRES}} = 10^{-4}$, $m_C = 4$ and $\varepsilon_{\text{prec}} = 10^{-2}$.

5.2.1. Weierstrass Rough Surfaces

Many electromagnetic problems and engineering applications are formulated in terms of scattering by a rough surface. A common example is the study of the remote sensing on the sea surface [12] or rough grounds.

We choose to model rough surfaces as 3D Weierstrass surfaces [13], which is a set of points $\varphi = (x, y, W(x, y)) \in \mathbb{R}^3$, where $W(x, y)$ is written as

$$W(x,y) = h \sum_{n=n_1}^{n_2} b^{(D_f-3)n} \sin(b^n)\big(x\cos(\psi_n) + y\sin(\psi_n)\big), \tag{28}$$

where h is the height of the surface, D_f the fractal dimension, n_1 (resp. n_2) the first (resp. last) scale, b characterizes the lacunarity of the fractal surface, $\psi_n = \frac{(2n-1)\pi}{n_2}$ and x (resp. y) running from 0 to the length l (resp. width w). These surfaces can be really irregular and thus need to be finely discretized in order to be accurately described. This can lead to big problems in terms of dofs, and thus the $\mathcal{H}$ format seems well suited for such a case. We study a case with the following parameters: $D_f = 2.5$, $n_1 = 0$, $n_2 = 5$ and $b = 2$. We use a mesh with 71689 unknowns which means 25 points per wavelength for the average step.

Compression rates and computation times are respectively reported in Tables 4 and 5. These results lead to conclusions similar to the case of the sphere. Other simulations confirm this conclusion.

Table 4. Compression rates for the rough surface (71689 unknowns).

	ACA		HCA	
n_G	1	3	1	3
Assembly	7.22%	6.18%	7.22%	6.17%
Coarsening	4.65%	3.99%	5.08%	4.45%
Coarsening at $\varepsilon_{\text{prec}}$	1.34%	1.33%	1.34%	1.33%

Table 5. Computation times (s) for the rough surface (71689 unknowns).

	ACA		HCA	
n_G	1	3	1	3
Assembly	1255	3851	1203	2003
Coarsening	555	352	565	428.0
Coarsening at $\varepsilon_{\text{prec}}$	147	100	157	202
$\mathcal{H}$-LU decomposition	406	322	483	429
GMRES	13.62	11.19	16.76	14.83

5.2.2. Fokker Plane RCS

We end this study by the scattering of a plane wave on a plane. The simulations will be performed at frequencies 0.6 GHz (77190 unknowns mesh) and 1.2 GHz (308760 unknowns mesh) with high mesh size of 8 points per wavelength. Only the case $n_G = 3$ is considered for $f = 1.2$ GHz. We consider computation times in Tables 6 and 7, and the corresponding RCS for $f = 0.6$ GHz is given Figure 7.

Tables 6 and 7 confirm the advantages of HCA in terms of computation times. It is even more obvious in Table 7. Note that this mesh is a very irregular one. It may be an explanation to the

Table 6. Computation times (s) for the Fokker plane (77190 unknowns, $f = 0.6$ GHz).

	ACA		HCA	
n_G	1	3	1	3
Assembly	4146	8195	2234	2878
Coarsening	572	492	683	543
Coarsening at $\varepsilon_{\text{prec}}$	240	241	196	179
$\mathcal{H}$-LU decomposition	7952	7718	6619	5721
GMRES	515	578	553	585

Table 7. Computation times for the Fokker plane (308760 unknowns, $f = 1.2$ GHz).

	ACA	HCA
Assembly (s)	47012	12812
Coarsening (s)	3795	3422
Coarsening at $\varepsilon_{\text{prec}}$ (s)	1402	949
$\mathcal{H}$-LU decomposition (s)	59527	34520
GMRES (s)	2647	2488

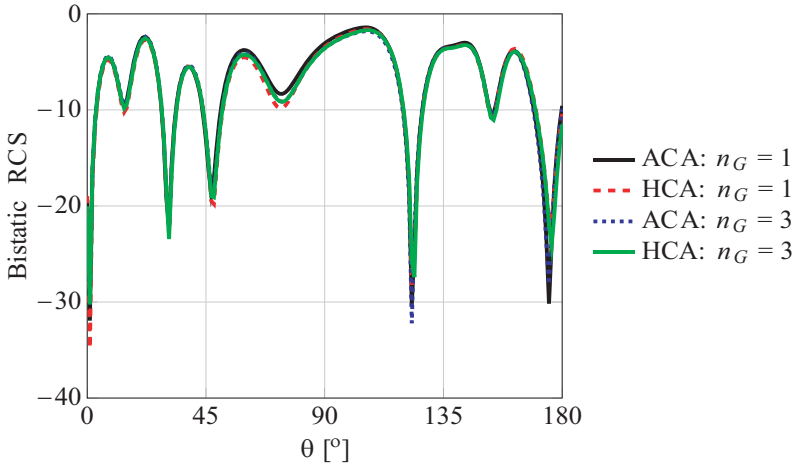

Figure 7. Bistatic RCS of the Fokker plane ($f = 0.6\,\text{GHz}$).

difficulties of ACA to get an optimal compression. Accuracy results in terms of RCS Figure 7 are "satisfying" as the bistatic RCS obtained with the HCA fits well with the one given by the ACA for both values of n_G.

6. CONCLUSION

In this paper we have presented the implementation of the hybrid cross approximation applied to scattering problems solved by the EFIE. This technique is an alternative to the adaptive cross approximation and has the advantage of being directly applied without any heuristics. Indeed, it does not depend on an evaluation of the error (which presents a risk of failure in the ACA algorithm). The obtained results show that the method could give equivalent results in terms of accuracy and improve the computation times.

REFERENCES

1. Darve, É., "The fast multipole method: Numerical implementation," *Journal of Computational Physics*, Vol. 160, 195–240, 2000.
2. Poirier, J.-R. and R. Perrussel, "Adaptive cross approximation for scattering by periodic surfaces," *Progress In Electromagnetics Research M*, Vol. 35, 97–103, 2014.
3. Börm, S. and Lars Grasedyck, "Hybrid cross approximation of integral operators," *Numerische Mathematik*, Vol. 101, 221–249, 2005.
4. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, Vol. 30, 409–417, 1982.
5. Bebendorf, M., "Hierarchical matrices — A means to efficiently solve elliptic boundary value problems," *Lecture Notes in Computational Science and Engineering*, Springer, 2008.
6. Bebendorf, M., "Approximation of boundary element matrices," *Numerische Mathematik*, Vol. 86, 565–589, 2000.
7. Grasedyck, L. and W. Hackbusch, "Construction and arithmetics of H-matrices," *Computing*, Vol. 70, 295–334, 2003.
8. Bebendorf, M., "A hierarchical LU decomposition-based preconditioners for BEM," *Computing*, Vol. 74, 225–247, 2005.
9. Grasedyck, L., "Adaptive recompression of H-matrices for BEM," *Computing*, Vol. 74, 205–223, 2005.

10. Siau, J., et al., "Hybrid cross approximation for a magnetostatic formulation," *IEEE Transactions on Magnetics*, Vol. 51, No. 3, 1–4, 2015.
11. Saad, Y. and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. and Stat. Comput.*, Vol. 7, No. 3, 856–869, 1985.
12. Berizzi, F. and E. Dalle-Mese, "Fractal analysis of the signal scattered from the sea surface," *IEEE Transactions on Antennas and Propagation*, Vol. 47, No. 2, 324–338, 1999.
13. Franceschetti, G., M. Migliaccio, and D. Riccio, "An electromagnetic fractal-based model for the study of fading," *Radio Science*, Vol. 31, 1749–1759, 1996.