Centre de Mathématiques Appliqués
URM 7641 École Polytechnique CNRS
Route de Saclay, 91128 Cedex, France

Partial Differential Equations and Numerical Analysis
Centre de Mathématiques Appliquées - École Polytechnique

On The Hierarchical Matrices, Cross-Approximations, And Other Graph-Based Fast Methods For Linear Algorithmic and Memory Complexity For The Boundary Elements Method

Author: Pedro Ramaciotti [1]

October 2013

[1] ramaciotti@cmap.polytechnique.fr

Contents

1 Introduction

The goal of this work is to give a comprehensive, but also brief and straightforward introduction to the so-called fast methods for the resolution of integral equations arising in the solution partial differential equations from the perspective of the hierarchical matrix and the cross-approximation methods. It aims to propose a sequential understanding of the main ideas that have been developed in the field since the 1980s spread in several sub-branches of research.

The integral equation methods are a classical approach for the solution of partial differential equations related to wave propagation or potential phenomena in bounded and unbounded domains. They allow for the formulation of equivalent integral equations in the boundary of the domains instead of the more classical modeling through partial differential equations in the whole domains. A variational formulation and a Galerkin discretization based on a geometrical discretization of the boundary of the domains lead to the *Boundary Elements Method* (BEM). The BEM aims to solve the discretized variational formulation on the interphases of the domains rather than in their interior, as the *Finite Elements Method* (FEM) does. The main disadvantage is that, while the FEM leads to a sparse system matrix, the BEM leads to a full system matrix due to the non-local nature of the kernel of the integral equation arising from the integral representation. This constitutes a major difficulty in the implementation of the integral equation approach because the storage of the system matrix and the amount of operations required to solve the associated linear system (using iterative solvers with proper preconditioning) scale asymptotically as $\mathcal{O}(n^2)$ instead of $\mathcal{O}(n)$, as it is achieved with the FEM. This disadvantage leads rapidly to the impracticality of the integral equation approach as the size of the problem grows if no additional measures are taken. To take advantage of the benefits of the BEM over the FEM for certain applications (specially related to unbounded domains) the so-called fast methods must be implemented. These fast methods have been in constant development since their first introduction in the 1980s and provide means to deal with (store, operate, compute) system matrices with linear or linear-logarithmic complexity and algorithms to solve the related systems also with linear or linear-logarithmic arithmetic complexity. This work aims to introduce the development of the main ideas behind the fast methods for the BEM with a focus on the most general one, the hierarchical matrix method, articulating recent developments in many sub-fields in a single exposition. A special focus will also be kept in the *Cross-Approximation* (CA) methods and their kernel-independency over more classical kernel-dependent approximations.

This work is structured as follows:

- A contextual setting regarding the BEM is given showing the complexity problems inherent to the method and the need to study and develop fast methods.
- A development of the main ideas behind the fast methods, with a focus on hierarchical matrices and CA methods, are exposed; the main theorems and complexity estimates are provided.
- Some example computations are studied and exhibited, illustrating the concepts treated in this report.
- Finally, with the main ideas and applications in mind, a revision of the historical development of the methods is given.

2 Computational Complexity of the Boundary Element Method: An Example

2.1 Introduction

This section illustrates the main advantage of the BEM in the approximation of solutions to PDEs but also its main disadvantage as a motivation for the development and study of the so-called fast methods for the solution of the associated discrete *Boundary Integral Equations* (BIE). To this end a wave propagation problem in an unbounded domain is modeled exploiting the advantages of the integral equation approach. The BEM is used and the computational complexity is analyzed.

In the following, a sound propagation model is developed for an open space where a rigid obstacle has been placed.

The iso-entropic, time-harmonic variation of pressure p in a perfect fluid at rest is modeled by the Helmholtz equation:

$$-(\Delta + k^2)p = 0, \tag{1}$$

where $k = \omega/c$ is the wavenumber for a pulsation ω and the sound speed c.

Let us consider the situation where a rigid obstacle $\Omega_{int} \subset \mathbb{R}^3$ of boundary Γ is surrounded by an unbounded and perfect fluid at rest $\Omega_{ext} \subset \mathbb{R}^3$ such that $\overline{\Omega_{int}} \cap \overline{\Omega_{ext}} = \Gamma$ and $\overline{\Omega_{int} \cup \Omega_{ext}} = \mathbb{R}^3$.

Let as consider an incident pressure wave p_{inc} complying with the Helmholtz equation:

$$-(\Delta + k^2)p_{inc} = 0 \quad \text{in } \Omega_{ext}. \tag{2}$$

If we decompose the total pressure wave in an incident wave and a scattered wave $p = p_{inc} + p_{scat}$, and using (2), equation (1) imposes

$$-(\Delta + k^2)p_{scat} = 0 \text{ in } \Omega_{ext}.$$

Let $\hat{n}$ be the unit normal to Γ pointing towards Ω_{ext}, and let u be the displacement vector of the points of the obstacle immersed in the fluid of density ρ_{ext}. Then, on the boundary Γ, the following relation for the pressure p and the point displacement u holds [23]:

$$\rho_{ext}\,\omega^2 u \cdot \hat{n} = \frac{\partial p}{\partial \hat{n}} = \frac{\partial p_{inc}}{\partial \hat{n}} + \frac{\partial p_{scat}}{\partial \hat{n}}. \tag{3}$$

For a rigid obstacle the displacement of its point is null, i.e., $u = 0$, and equation (3) can be written as

$$\frac{\partial p_{scat}}{\partial \hat{n}} = -\frac{\partial p_{inc}}{\partial \hat{n}}. \tag{4}$$

Finally, the partial differential equation for the scattered part p_{scat} of the wave of an iso-entropic variation of pressure p produced by a wave p_{inc} incident on a rigid body of boundary Γ surrounded by a perfect fluid is

$$\begin{cases} -(\Delta + k^2)p_{scat} = 0 & \text{in } \Omega_{ext}, \\ \frac{\partial p_{scat}}{\partial \hat{n}} = -\frac{\partial p_{inc}}{\partial \hat{n}} & \text{on } \Gamma. \end{cases} \tag{5}$$

2.2 Integral Representation and Integral Equation

Definition 1 (Sommerfeld Radiation Condition) *A solution p to the Helmholtz equation is said to satisfy the Sommerfeld radiation condition (SRC) if it satisfies*

$$\lim_{\|x\|_2 \to \infty} \|x\|_2 \left(\frac{\partial}{\partial \|x\|_2} p(x) - ikp(x) \right) = 0$$

Lemma 1 (Green's Function for the Helmholtz Equation) *The Green's functions for the Helmholtz equation, i.e., the only function $G(x,y)$ that satisfies*

$$\begin{cases} -(\Delta_x + k^2)G(x,y) = \delta(x-y) \quad \mathcal{D}'(\mathbb{R}^3) \\ \\ \quad G(\cdot,y) \text{ satisfies the SRC} \end{cases}$$

$$\text{is } G(x,y) = \frac{e^{ik\|x-y\|_2}}{4\pi\|x-y\|_2}.$$

Demonstration Page 12, [26]. ■

Theorem 1 (Helmholtz Integral Representation Theorem) *Let q be a function such that*

$$\Delta q + k^2 q = 0 \quad \text{in } \Omega_{int},$$

$$\Delta q + k^2 q = 0 \quad \text{in } \Omega_{ext},$$

and such that it satisfies the Sommerfeld radiation condition. Let us define the jump functions over Γ as

$$[q] = q|_{int} - q|_{ext} \text{ and } \left[\frac{\partial q}{\partial \hat{n}}\right] = \left.\frac{\partial q}{\partial \hat{n}}\right|_{int} - \left.\frac{\partial q}{\partial \hat{n}}\right|_{ext}.$$

Then, $q|_{int}$, q_{ext}, $\frac{\partial q}{\partial \hat{n}}|_{int}$ and $\frac{\partial q}{\partial \hat{n}}|_{ext} \in C^0(\Gamma)$, and for $x \notin \Gamma$, q can be represented as

$$q(x) = \int\limits_\Gamma G(x,y) \left[\frac{\partial q}{\partial \hat{n}}(y)\right] d\Gamma(y) - \int\limits_\Gamma \frac{\partial}{\partial \hat{n}_y} \left(G(x,y)\right) [q(y)] \, d\Gamma(y).$$

Demonstration Theorem 3.1.1, page 110 [26]. ■

Theorem 2 (Traces' Theorem) *Let μ and λ be continuous over Γ and let q be*

$$q(x) = \int\limits_\Gamma G(x,y)\lambda(y) d\Gamma(y) - \int\limits_\Gamma \frac{\partial}{\partial \hat{n}_y} \left(G(x,y)\right) \mu(y) d\Gamma(y).$$

Then, the traces of q comply with

$$\begin{pmatrix} q|_{int} \\ \frac{\partial q}{\partial \hat{n}}|_{int} \end{pmatrix} = \begin{pmatrix} I/2 - D & S \\ -N & I/2 + D^* \end{pmatrix} \begin{pmatrix} \mu \\ \lambda \end{pmatrix} \text{ and } \begin{pmatrix} q|_{ext} \\ \frac{\partial q}{\partial \hat{n}}|_{ext} \end{pmatrix} = \begin{pmatrix} -I/2 - D & S \\ -N & -I/2 + D^* \end{pmatrix} \begin{pmatrix} \mu \\ \lambda \end{pmatrix},$$

where I is the identity operator and D, S, N and D^* are integral operators defined as follows:

$$D\mu(x) = \int\limits_\Gamma \frac{\partial}{\partial \hat{n}_y} G(x,y)\mu(y) d\Gamma(y),$$

$$S\lambda(x) = \int\limits_\Gamma G(x,y)\lambda(y) d\Gamma(y),$$

$$D^*\lambda(x) = \int\limits_{\Gamma} \frac{\partial}{\partial \hat{n}_x} G(x,y)\lambda(y) d\Gamma(y),$$

$$N\mu(x) = \int\limits_{\Gamma} \frac{\partial^2}{\partial \hat{n}_y \partial \hat{n}_x} G(x,y)\mu(y) d\Gamma(y).$$

Demonstration Theorem 3.1.2, page 113 [26]. ■

Using the two previous theorems the scattering problem (5) can be reformulated as follows. Let $\tilde{p}_{scat}$ be defined in $\Omega_{int} \cup \Omega_{ext}$ by:

$$\begin{cases} -(\Delta + k^2)\tilde{p}_{scat} = 0 & \text{in } \Omega_{ext} \\ \frac{\partial \tilde{p}_{scat}}{\partial \hat{n}}|_{ext} = -\frac{\partial p_{inc}}{\partial \hat{n}} & \text{on } \Gamma. \end{cases}$$

$$\begin{cases} -(\Delta + k^2)\tilde{p}_{scat} = 0 & \text{in } \Omega_{int} \\ \frac{\partial \tilde{p}_{scat}}{\partial \hat{n}}|_{int} = -\frac{\partial p_{inc}}{\partial \hat{n}} & \text{on } \Gamma. \end{cases}$$

Defined this way, it will result that $\tilde{p}_{scat} = p_{scat}$ in Ω_{ext}. Also, for $\tilde{p}_{scat}$, the jump relations are

$$\lambda(x) = \left[\frac{\partial \tilde{p}_{scat}}{\partial \hat{n}}\right](x) = 0 \text{ and } \mu(x) = [\tilde{p}_{scat}](x) \text{ for } x \in \Gamma.$$

If the function μ was known, then, by virtue of Theorem 1, the solution to the scattering problem could be computed as

$$p_{scat}(x) = \tilde{p}_{scat}(x) = -\oint\limits_{\Gamma} \frac{\partial G(x,y)}{\partial \hat{n}_y} \mu(y) d\Gamma(y) \text{ for } x \in \Omega_{ext}.$$

Function μ can in fact be calculated, since by virtue of Theorem 2 it satisfies the following integral equation:

$$\left.\frac{\partial \tilde{p}_{scat}}{\partial \hat{n}}(x)\right|_{ext} = -\frac{\partial p_{inc}}{\partial \hat{n}}(x) = -N\mu(x) = -\oint\limits_{\Gamma} \frac{\partial^2 G(x,y)}{\partial \hat{n}_x \partial \hat{n}_y} \mu(y) d\Gamma(y) \text{ for } x \in \Gamma. \quad (6)$$

2.3 Variational Formulation

A variational formulation for the integral equation (6) is provided to look for a solution. As a jump of the pressure (assumed to be for example in $H^1(\Omega_{ext} \cup \Omega_{int})$), μ can be proven to be in $H^{1/2}(\Gamma)$ and the traces of the derivatives of the pressure in $H^{-1/2}(\Gamma)$. The operator N can be proven to be continuous from $H^{1/2}(\Gamma)$ to $H^{-1/2}(\Gamma)$. The following theorem provides a variational formulation for the integral equation.

Theorem 3 (Variational Formulation for the N Integral Operator) *When $-k^2$ is not an eigenvalue of the associated interior Neumann problem for the Laplace equation, the integral operator N defined in Theorem 2 is continuous from $H^{1/2}(\Gamma)$ onto $H^{-1/2}(\Gamma)$ and the integral equation (6) admits the following variational formulation:*

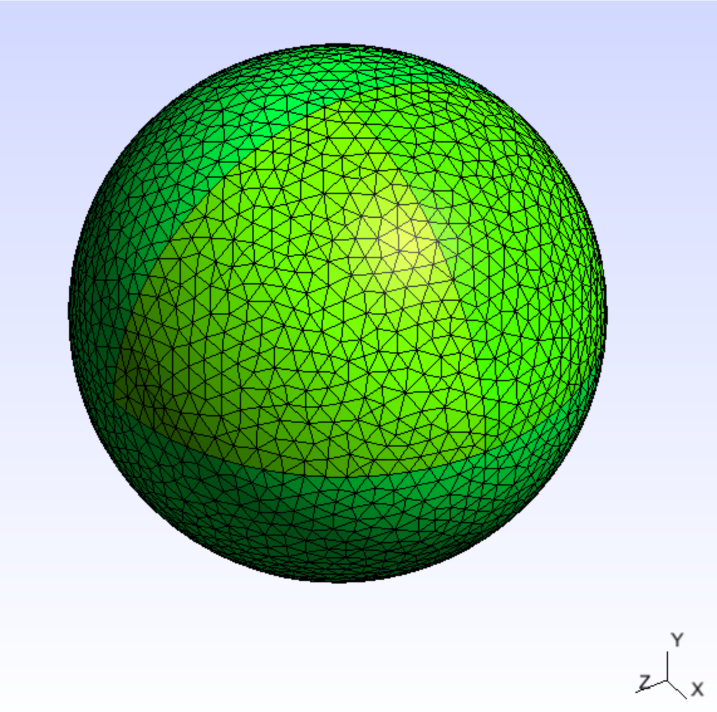


Figure 1: Triangular discretization approximation Γ_h the unit sphere.

$$\begin{cases} \textit{Find } \mu \in H^{1/2}(\Gamma) \textit{ such that } \forall \mu^t \in H^{1/2}(\Gamma): \\ \int\limits_{\Gamma\times\Gamma} G(x,y)\left(\overrightarrow{curl}_\Gamma \mu(y)\cdot \overrightarrow{curl}_\Gamma \mu^t(x)\right) d\Gamma(x)d\Gamma(y) \\ -k^2 \int\limits_{\Gamma\times\Gamma} G(x,y)\mu(y)\mu^t(x)\,\hat{n}_x\cdot\hat{n}_y\, d\Gamma(x)d\Gamma(y) \\ = -\int\limits_{\Gamma} \frac{\partial p_{inc}}{\partial \hat{n}_x}(x)\mu^t(x)d\Gamma(x). \end{cases} \tag{7}$$

Demonstration Theorem 3.4.2, page 143 [26]. ■

2.4 Galerkin Discretization and Computational Cost

If the surface Γ is assumed to be piecewise polygonal and it is approximated by a triangular discretization Γ_h where h indexes a discretization by the longest edge, the spaces $H^{1/2}(\Gamma)$ and $H^{-1/2}(\Gamma)$ can be approximated by finite-dimensional sub-spaces spanned by basis functions such as, e.g., Lagrange finite element functions P1, P2, etc. Non-constant finite elements (P1 or higher order) are needed so that their derivatives do not vanish in the variational formulation (7). Figure 1 exemplifies the approximation of the unit sphere by a triangular discretization Γ_h.

Using a finite-dimensional sub-space H_h of dimension N, spanned by a set of basis functions $\{\varphi_1, \varphi_2, ..., \varphi_N\}$, the variational formulation can be put in a linear system as

$$\mu(x) = \sum_{j=1}^{N} \mu_j \varphi_j(x) \text{ for } x \in \Gamma_h,$$

$$I = (\mu_1, \mu_2, ..., \mu_N)^T, \text{ and}$$

$$ZI = V,$$

where,

$$Z_{ij} = \int\limits_{\Gamma_h \times \Gamma_h} G(x,y) \left(\overrightarrow{curl}_{\Gamma_h} \varphi_j(y) \cdot \overrightarrow{curl}_{\Gamma_h} \varphi_i(x) \right) d\Gamma_h(x) d\Gamma_h(y)$$

$$-k^2 \int\limits_{\Gamma_h \times \Gamma_h} G(x,y) \varphi_j(y) \varphi_i(x) \, \hat{n}_x \cdot \hat{n}_y \, d\Gamma_h(x) d\Gamma_h(y),$$

and,

$$V_i = -\int\limits_{\Gamma_h} \frac{\partial p_{inc}}{\partial \hat{n}_x}(x) \varphi_i(x) d\Gamma_h(x).$$

The matrix Z is complex and symmetrical, and as such the storage required is of $N(N+1)/2$ complex numbers, thus, the storage complexity is of order $\mathcal{O}(N^2)$. If the system is to be solved by means of an iterative solver each iteration will require a matrix-vector multiplication, which will require N^2 sums and multiplications of complex floating point numbers, thus the computational complexity of the solver will be of oder $\mathcal{O}(N^2 N_{iterations})$.

In the case of wave propagation problems at least 10 triangles per wavelength are required to correctly sample the wave-like phenomena (5 triangles per wavelength is accepted in some far-fields applications). For a propagation domain Ω_{ext} with a given propagation speed c and a time-harmonic incident wave of frequency f the wavelength will be c/f and length of the edges in mesh will grow as $\mathcal{O}(1/f)$, the area of the triangles will grow as $\mathcal{O}(1/f^2)$ and the number of triangles as $\mathcal{O}(f^2)$. The number of degrees of freedom N can be number of triangle edges, triangle vertices, middle points, or other similar. The quantity of these geometrical elements grows with the same order than the number triangle faces as given by the Euler characteristic for the topology of the surface Γ, which implies that the storage complexity and the computational complexity has an asymptotic behavior of $\mathcal{O}(f^4)$. This renders the integral equation technique impractical for many applications if no additional measures are taken.

As an example, Figure 2 shows the required computer memory and the required number of complex floating point operations to perform a matrix-vector multiplication for the case of the computation of a sound wave of different frequencies scattered by a rigid unit sphere in open air, where the speed is around 343 m/s. The complex number representation is assumed to be *standard double* (IEEE-754), meaning that each one requires 16 bytes of storage. It can be seen in the figure that the slope of the complexities' curves is indeed 4 decades of complexity (both in storage and computation of a matrix-vector multiplication) per decade of frequency.

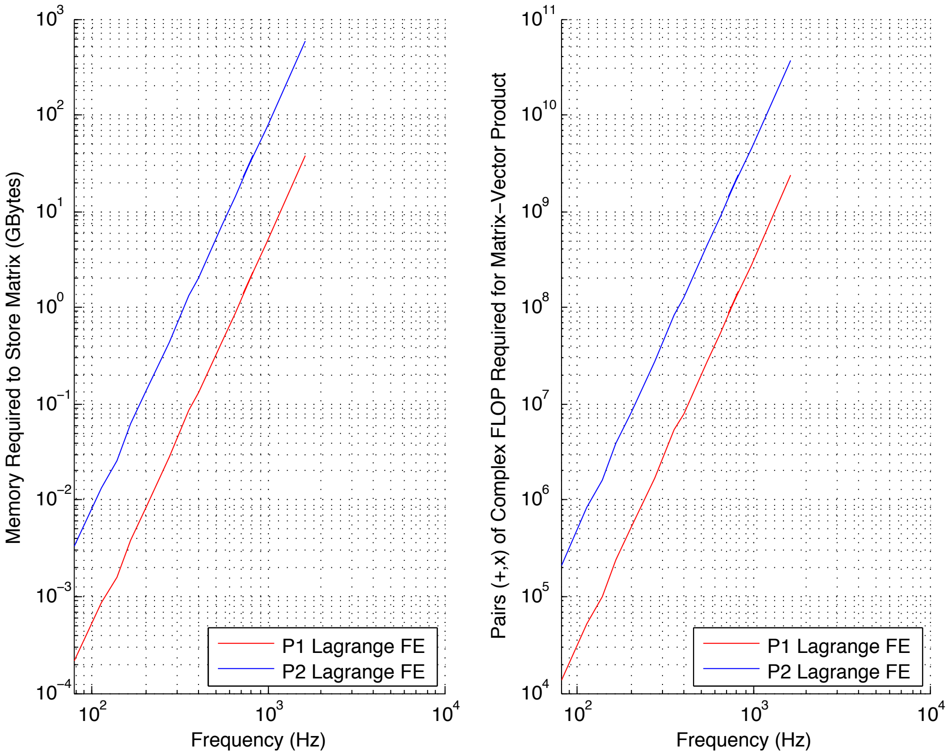


Figure 2: **Memory** required to store the interaction matrix for a rigid unit sphere in open air (left) and the **number of complex floating point operations** required to perform a matrix-vector multiplication (right), both respecting the criterion of 10 triangles per wavelength.

3 Hierarchical Matrices, Cross-Approximation And Other Fast Methods for the BIE

3.1 Introduction

In this chapter the main elements of the hierarchical method and the cross-approximation methods are sequentially constructed developing the complexity estimates required to store an approximation of the system matrix and to perform matrix-vector multiplications. A short link with other classical fast methods is discussed.

In a first part the core building elements of the fast methods, the degenerate approximation of an integral kernel, are shown and discussed. Their consequences are explored in terms of data-sparse representations and their advantages in storage and computation. In a second part the notational framework required to treat the structure of the system matrix in relation to the geometry of the problem is treated showing how it can be decomposed. Finally, in a third part of the chapter, the exploitation of the degenerate approximations of the kernels using the structure in the second part is treated together with error estimates and the algorithms to compute them.

3.2 Low-Rank Matrices and Their Relation with the Kernel of an Integral Operator

The main idea behind the fast methods for the BEM is the postulate that sub-matrices of the system matrix may be replaced by suitable approximations capable of reducing the storage complexity and the arithmetic complexity involved in relevant matrix operations. These approximations rely on the nature of the underlying problem, more precisely, on the associated kernels of the integral operators and their approximations. Their advantage arises from the consequences of the way the matrices they produce can be represented (and thus operated and stored, if needed). In this section, an introduction to the relation between low-rank representations and approximate integral operators is given, followed by a description and a discussion on the associated low-rank sub-matrices and the consequences in memory storage and computational complexity. A link will also be provided to the exploitation of these consequences through classical sub-techniques of these fast methods: the fast multipole method (FMM), the panel clustering method and the cross-approximation methods.

3.2.1 Matrices Arising from Degenerated Kernels

Using the integral equation approach a partial differential equation problem can be often formulated as an integral equation, e.g., for the single layer potential, in the form of

$$f(x) = \int_{\Gamma} G(x,y)u(y)d\Gamma(y),$$

which in turn can be formulated as a variational problem and then discretized via a Galerkin discretization, giving rise to equations of the type

$$\int_{\Gamma} f(x)\varphi_i(x)d\Gamma(x) = \int_{\Gamma\times\Gamma} G(x,y)u_h(y)\varphi_i(x)d\Gamma(y)d\Gamma(x),$$

where

$$u_h(x) = \sum_{j=1}^{N_h} \lambda_j\varphi_j(x),$$

and $\{\varphi_i\}_{i=1}^{N_h}$ is the set of basis functions that span de finite-dimensional sub-space of the space where the variational problem is given.

These equations can be put in a system of linear equations, i.e.,

$$\sum_{j=1}^{N_h} \lambda_j \left(\int\limits_{\Gamma\times\Gamma} G(x,y)\varphi_i(x)\varphi_j(y)d\Gamma(y)d\Gamma(x) \right) = \int\limits_{\Gamma} f(x)\varphi_i(x)d\Gamma(x),$$

which can be represented by a matrix equation for the system matrix Z:

$$ZI = V,$$

$$Z_{ij} = \int\limits_{\Gamma\times\Gamma} G(x,y)\varphi_i(x)\varphi_j(y)d\Gamma(y)d\Gamma(x),$$

$$V_i = \int\limits_{\Gamma} f(x)\varphi_i(x)d\Gamma(x).$$

For other integral operators the procedure is similar.

The main idea behind the fast methods for the BEM is that, at least for some couples of regions $\Gamma_x, \Gamma_y \subset \Gamma$ (chosen using a specific criterion that will be discussed), the kernel of the integral equation can be proven to be formulated as a series of terms formed by factors that account separately for the effect of the position x and y:

$$G(x,y) = \sum_{l=0}^{\infty} g_l(x)h_l(y).$$

If the series can be approximated by a suitable truncation up to the first k terms,

$$\tilde{G}(x,y) = \sum_{l=0}^{k-1} g_l(x)h_l(y),$$

then the sub-matrix $\tilde{Z} \in \mathbb{C}^{m\times n}$ of the system matrix corresponding to the integration over $\Gamma_x \times \Gamma_y$ could be approximated by a matrix whose elements are

$$\tilde{Z}_{ij} = \sum_{l=0}^{k-1} \left(\int\limits_{\Gamma_x} g_l(x)\varphi_i(x)d\Gamma(x) \right) \left(\int\limits_{\Gamma_y} h_l(y)\varphi_j(y)d\Gamma(y) \right).$$

This means that this portion of the system matrix could be written as the product of two matrices of lesser rank,

$$\tilde{Z} = A \cdot B^T,$$

the matrices $A \in \mathbb{C}^{m\times k}$ and $B \in \mathbb{C}^{n\times k}$ having elements

$$A_{il} = \int\limits_{\Gamma_x} g_l(x)\varphi_i(x)d\Gamma(x) \text{ and } B_{jl} = \int\limits_{\Gamma_y} h_l(y)\varphi_j(y)d\Gamma(y).$$

3.2.2 Low-Rank Matrices and Their Representation

The central element of the fast methods is the detection and approximation of the sub-matrices of the system matrix susceptible of being expressed in a more suitable way in the sense of storage and computational complexity. This chapter provides a formal framework to treat such matrices with the aim of exploiting these advantages.

Definition 2 (k-Rank Matrices) *We denote the set of $m \times n$ matrices having at most k linearly independent rows or columns by*

$$\mathbb{C}_k^{m\times n} = \{A \in \mathbb{C}^{m\times n} : rank(A) \leq k\}.$$

Theorem 4 (k-Rank Matrix Outer-Product Representation) *A matrix $A \in \mathbb{C}^{m\times n}$ belongs to $\mathbb{C}_k^{m\times n}$ if and only if there exist matrices $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$ such that*

$$A = UV^H,$$

where V^H denotes the conjugate transpose of V.

Demonstration Let us consider a matrix with rank less or equal than k, $A \in \mathbb{C}_k^{m\times n}$, i.e. the dimension of its column space is at most equal to k. Equivalently, the rows $a_1, a_2, ..., a_n$ of the matrix A belong to the column space spanned by a given basis $u_1, u_2, ..., u_k$:

$$a_j = \sum_{i=1}^{k} \tilde{v}_{kj} u_j.$$

This can be written as $A = UV^H$, where U is the matrix of columns $u_1, u_2, ..., u_k$ and V is the matrix of elements $v_{ji} = \tilde{v}_{ij}^*$ (where * denotes complex conjugation). Conversely, if $A \in \mathbb{C}^{m\times n}$ is a matrix that can be written as $A = UV^H$ with $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$ its columns belong to a space of dimension k or less, thus assuring $A \in \mathbb{C}_k^{m\times n}$. ■

The previous theorem tells us that k-rank matrices are a suitable representation, in their outer-product form, for matrices or sub-matrices arising from discrete integral operators where the kernel can be approximated in the way described in the previous section. The following definition distinguishes the cases where a k-rank matrix representation is advantageous in comparison to the classical full-matrix representation.

Definition 3 (Low-Rank Matrix) *A matrix $A \in \mathbb{C}^{m\times n}$ such that $A \in \mathbb{C}_k^{m\times n}$ for some k is called a low-rank matrix if the storage of its outer-product representation requires less elements than that of its full representation, i.e., a matrix for which k, m and n satisfy*

$$k(m+n) < mn.$$

If parts of the system matrix arising from a discrete integral operator can be approximated by low-rank matrices, the overall storage cost will be less than that of storing all its elements. In general, the size of the sub-matrices that cannot be approximated by low-rank matrices can be controlled, as it will be seen the following sections.

3.2.3 Basic Operations Involving Low-Rank Matrices

The main task to be carried after the application of a Galerkin discretization to the integral representation of partial differential equations is the solution of the linear equations associated to the system matrix. The main operation required to solve such linear systems is going to require matrix-vector multiplications in order to solve the linear system via iterative methods. Thus, a low-rank representation is desired to provide a matrix-vector multiplication using less arithmetic operations than a full matrix representation.

A matrix-vector multiplication of a matrix $A \in \mathbb{C}^{m\times n}$ by a vector $x \in \mathbb{C}^n$ requires $m\,(2n-1)$ arithmetic operations ($m\,n$ multiplications and $m\,(n-1)$ sums of floating complex numbers). If the matrix A is of rank k, and can thus be written as $A = UV^H$, then product Ax can be computed with $k\,(2n-1) + m\,(2k-1) = 2\,k\,(m+n) - m - k$ arithmetic operations $(k\,(m+n)$

multiplications and $k\,(m+n)-m-k$ sums of floating complex numbers). Assuming we have a suitable low-rank representation of the matrix A, then the product by a vector can be improved using this representation provided that

$$mn > k\,(m+n) - \frac{1}{2}k. \tag{8}$$

Other operation that is extensively used, especially in the determination of the quality of matrix approximations, is the computation of the Frobenius Norm. The Forbenius norm of a matrix A is computed as

$$\|A\|_F = \sqrt{trace(A^H A)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2},$$

and requires $2mn - m - n + 1$ operations (mn multiplications and $(m-1)(n-1)$ sums of floating complex numbers). If the matrix A has a k-rank approximation and can thus be written as $A = UV^H$ then the Frobenius norm can be computed as

$$\|A\|_F = \|UV^H\|_F = \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{k} \left(U_{1..m,i}\right)^H U_{1..m,j} \left(V_{1..n,i}\right)^H V_{1..n,j}},$$

where $U_{1..m,i}$ is the i-th column vector of matrix U. Such computation can be performed in $2k^2(m+n) - 2k + 1$ operations ($k^2(m+n+1)$ multiplications and $k^2(m+n-1) + 1 - 2k$ sums of floating complex numbers), thus the computation of the Frobenious norm of a k-rank matrix can be improved in speed provided that

$$mn > \left(k^2 + \frac{1}{2}\right)(m+n) - k \tag{9}$$

Condition (9) is stronger than condition (8). Improvement in the storage of sub-matrices of the system matrix and in the speed of computations provided the both conditions are met will prove to be critical tools in scaling a problem's size, and are the center of the fast methods for the BEM.

3.2.4 Connection with the Classical Fast Methods

The most prominent classical fast methods for the acceleration of the discretized boundary integral equation based on kernel decomposition (local existence of degenerate approximants to the kernel) are the Fast Multipole Method (FMM), the Panel Clustering Method and the Cross-Approximation techniques. This section relates the exposed ideas to the core mechanisms by which these different methods work. Other well known techniques based on different principles not discussed in this report are the Adaptive Integral Methods (AIM), the Pre-corrected Fast Fourier Transform (pFFT), and the wavelet compression techniques.

As stated above, the main idea behind the studied fast methods for the BEM is that, at least for some couples of regions $\Gamma_x, \Gamma_y \subset \Gamma$ the surface double-integral can be computed using a degenerate kernel approximant, thus allowing for the explotation of the advantages discussed. The selection of couples of subsets of Γ that allow for this approximation is related to the smoothness of the kernel $G(x,y)$, which in general can be assured for $x = y$, and it is dependent on the parameter $dist\{\Gamma_x, \Gamma_y\}$. In the following exposition of the classical fast methods using degenerate kernels it is assumed that the sub-matrices to be approximated by low-rank approximants has been performed using a criterion $dist\{\Gamma_x, \Gamma_y\} > \eta$ for a given parameter $\eta > 0$. Desirable properties of the approximations can also be derived from criteria considering the the sizes $diam(\Gamma_x)$ and $diam(\Gamma_y)$ of the sub-domains. A discussion on the subject is provided in the next section and an historical review of the development of the mentioned methods and techniques is given at the end of this report.

Fast Multipole Methods (FMM)

FMM formulate an approximation to the kernel such that matrix-vector multiplication operations involving approximated sub-matrices can take profit on its low-rank. The sub-matrix approximant is never store. In the following, a brief description of the method is exhibited. For further details the reader is referred to [11] for the application of the method to the case of the Helmholtz's equation.

Let m_x and m_y be the centroids of Γ_x and Γ_y respectively. The vector $x - y$ for $x \in \Gamma_x$ and $y \in \Gamma_y$ can be written as

$$x - y = (x - m_x) + (m_x - m_y) + (m_y - y).$$

Using Gegenbauer's Addition Theorem the Green's function for the Helmholtz equation can be written as

$$G(x,y) = \frac{e^{ik\|x-y\|_2}}{4\pi\|x-y\|_2} = \frac{ik}{16\pi^2} \lim_{L\to\infty} \int_{s\in\mathcal{S}} e^{iks\cdot(x-m_x)} T^L_{m_x-m_y}(s) e^{iks\cdot(m_y-y)} d\mathcal{S}(s),$$

where $\mathcal{S}$ is the unit sphere and the transfer function $T^L_{m_x-m_y}(s)$ is defined as

$$T^L_{m_x-m_y}(s) = \sum_{l=0}^{L} (2l+1) i^l h_l^{(1)}(k\|m_x - m_y\|) P_l\left(\cos(s,(m_x - m_y))\right).$$

In the previous definition $h_l^{(1)}$ is the spherical Hankel function of the first kind and order l, P_l is the Legendre polynomial of order l, and $\cos(s,(m_x - m_y))$ is the cosine of the angle between s and $m_x - m_y$.

When computing the interaction between two sub-domains, the transfer function can be truncated to a sufficiently large number L and the integration order can be changed to take advantage of the kernel decomposition:

$$\int_{\Gamma_x}\int_{\Gamma_y} G(x,y)\varphi_i(x)\varphi_j(y)d\Gamma_x(x)d\Gamma_y(y) =$$

$$\int_{\Gamma_y}\int_{s\in\mathcal{S}} \left[T^L_{m_x-m_y}(s)\left(\int_{\Gamma_x} e^{iks\cdot(x-m_x)}\varphi_i(x)d\Gamma_x(x)\right)\right] e^{iks\cdot(m_y-y)}\varphi_j(y)d\mathcal{S}(s)\Gamma_y(y).$$

The multipole decomposition then allows for matrix-vector multiplications without the need to compute the matrix and with less operations as follows. Every time the matrix is multiplied by a vector the operation is performed in sub-matrices corresponding to sub-domains of integrations. For the multiplications associated to the sub-matrix related to the interactions between $\Gamma_x \subset \Gamma$, first we compute

$$\mathcal{F}_{\Gamma_x}(s) = \int_{\Gamma_x} e^{iks\cdot(x-m_x)}\varphi_i(x)d\Gamma_x(x),$$

using a discretization of $\mathcal{S}$. Then for every other domain interacting with Γ_x we can compute

$$\int_{\Gamma_x}\int_{\Gamma_y} G(x,y)\varphi_i(x)\varphi_j(y)d\Gamma_x(x)d\Gamma_y(y) = \int_{\Gamma_y}\int_{s\in\mathcal{S}} \mathcal{F}_{\Gamma_x}(s)T^L_{m_x-m_y}(s)e^{iks\cdot(m_y-y)}\varphi_j(y)d\mathcal{S}(s)\Gamma_y(y),$$

using the precomputed values $\mathcal{F}_{\Gamma_x}(s)$ for the discretized unit sphere $\mathcal{S}$, if this other domain, depending on $dist(\Gamma_x, \Gamma_y)$, allows for a truncated multipole expansion.

Panel Clustering Methods

The so-called Panel Clustering Methods are similar to the FMM in that they rely in the identification of pairs of sub-domains $\Gamma_x, \Gamma_y \subset \Gamma$ separated enough so to assure smoothness of the kernel function, and then they seek to approximate it by a degenerate approximant thus allowing for the advantages of its low-rank features when performing matrix-vector multiplication operations. Also as with FMM, the approximant sub-matrices are never stored. The main difference between Panel Clustering Methods and FMM is the way in which they approximate the kernel; in the FMM it was approximated by a multipolar expansion based on the Gegenbauer's Addition Theorem, while in the Panel Clustering Methods the kernel is approximated for couples of sub-domains contained in axiparallel boxes (panels) where a Lagrange polynomial interpolation is performed for a grid of Chebyshev points to minimize the Runge's phenomenon present in interpolation. In the following, the approximation of the kernel is illustrated keeping a simple formalization, for further details the reader is referred to [31], Chapter 7, where a complete description of the method is given.

Let $B(\Gamma_x)$ and $B(\Gamma_y)$ the smallests axiparallel boxes containing the subsets $\Gamma_x, \Gamma_y \subset \Gamma$ respectively. Let $\Theta^m_{B(\Gamma_x)}$ be the set of m^3 Chebyshev points, a 3-dimensional tensorization of the set of m points, where each component of a point $\xi \in \Theta^m_{B(\Gamma_x)}$ is given by a Chebyshev point defined along the interval of one side of the axiparallel box $B(\Gamma_x)$. Let $L^m_\xi(x)$ be the tenderized Lagrange polynomial for that point $\xi \in \Theta^m_{B(\Gamma_x)}$, meaning that

$$L^m_\xi(x) = \begin{cases} 1 & \text{if } x = \xi, \\ 0 & \text{if } x = \tilde{\xi} \in \Theta^m_{B(\Gamma_x)} \text{ and } \tilde{\xi} \neq \xi. \end{cases}$$

Using the above-sketched polynomials a given function $f : B(\Gamma_x) \to \mathbb{C}$, sufficiently smooth, can be approximated as

$$f(x) \approx \left(\Pi^m_{B(\Gamma_x)} f\right)(x) = \sum_{\xi \in \Theta^m_{B(\Gamma_x)}} f(\xi) L^m_\xi(x).$$

A degenerate approximation $\tilde{G}$ of a kernel G can be computed as

$$\tilde{G}(x,y) = \left(\Pi^m_{B(\Gamma_y)} \Pi^m_{B(\Gamma_x)} G\right)(x,y) = \sum_{\xi_y \in \Theta^m_{B(\Gamma_y)}} \sum_{\xi_x \in \Theta^m_{B(\Gamma_x)}} G(\xi_x, \xi_y) L^m_{\xi_x}(x) L^m_{\xi_y}(y).$$

Cross-Approximations Methods

Cross-approximation methods also rely on the existence of a degenerate kernel but it is different from the previously exhibited methods in that they do not use knowledge of kernel other than the fact that a degenerate approximant exists. These methods provide low-rank approximations for sub-matrices using only a few entries of the original matrix relying on the knowledge that the kernel has a degenerate approximant rather than finding that explicit approximant. This characteristic allows for the use of previously written and tested BEM code, improving its storage and computational complexity. Another key feature of this methods in contrast with the previous ones is that they provide an explicit approximant to sub-matrices of the system matrix and not only an approximative low-rank method to perform matrix-vector multiplications. Together with a suitable structure, such as that provided by the hierarchical matrix method, these approximations can be used in matrix summation, multiplication and, using these operations, in matrix factorization and inversion. These possibilities can be exploited in the construction of pre-conditioners and in more complex integral equation (e.g., single-layer and double-layer, combined field integral equation).

The idea behind cross-approximation methods is to perform a rank-revealing decomposition of a matrix Z in order to construct consecutive approximants S_k growing in rank and diminishing the quantity $\|Z - S_k\|_F$. If the matrix Z is known to have a suitable low-rank approximation the number of computed consecutive approximants k required to achieve a good approximation could be low enough to take advantage in terms of storage and computational complexity as detailed in this section.

Let us consider a sub-matrix Z and let us notate $Z_{i\,1..n}$ the i-th row and $Z_{1..n\,j}$ the j-th column. Starting from a residue matrix $R_0 = Z$ a cross-approximation algorithm diminish by one the rank of R_0 to obtain a second residue matrix R_1 and so forth obtaining new residue matrices such that $rank(R_{k+1}) \leq rank(R_k)$. To diminish the rank, a column and a row are chosen for elimination in each step:

$$R_{k+1} = R_k - \left((R_k)_{i_k j_k}\right)^{-1} (R_k)_{1..m,\,j_k} (R_k)_{i_k,\,1..n}$$

The matrices that are subtracted to R_k in each step account for the difference $Z - R_k$ and are gathered in the approximant S_k: $Z = S_k + R_k$. The algorithm can be stopped either when a maximum rank k_{max} has been attained or when the approximant S_k is close enough to the matrix Z, i.e., $\|Z - S_k\|_F = \|R_k\|_F \leq \varepsilon$ for a specified tolerance ε resulting in $k_{max}(\varepsilon)$ steps. The approximant matrix $S_{k_{max}}$ is then available as:

$$S_{k_{max}} = \sum_{k=0}^{k_{max}-1} \left((R_k)_{i_k j_k}\right)^{-1} (R_k)_{1..m,\,j_k} (R_k)_{i_k,\,1..n}$$

A key observation is that the construction of S_k can be performed without using all the elements of the matrices R_k for previous steps. It is possible to store, for the matrices R_k, only the rows and columns that undergo change during the algorithm preserving a complexity that behaves asymptotically also as $\mathcal{O}(k(m+n))$ if $Z \in \mathbb{C}^{m \times n}$.

The choice of the pivots i_k and j_k in each step is a non-trivial task and it is related to the geometry of the problem. Also, if the approximation is to be determined for a fixed tolerance ε, the computation of the norm $\|Z - S_k\|_F$ in each step must be available with a complexity consistent with the other operations involved in the algorithm (i.e., $\mathcal{O}(k(m+n))$ if $Z \in \mathbb{C}^{m \times n}$). Finally, the number of steps required to achieved an approximant even under optimal pivot choices must be estimated a priori in order to assure that the cross-approximation approach is advantageous. These issues are currently the subject of active research and will be discussed briefly later on in this chapter. Further reading on the subject may be found in Bebendorf's book [6], chapter 3, and in Hackbusch's lecture notes [10], chapter 4. There are several methods based on cross-approximation of matrices, being the most prominent the *Adaptive Cross-Approximation* (ACA) methods, which set a given tolerance ε for the approximation an compute the approximant adaptively increasing its rank.

3.3 The Hierarchical Matrices Methods

The hierarchical matrices method allows for the exploitation of the advantages of low-rank matrices in storage and computational complexity. A system matrix associated to a discrete integral operator may not have a suitable low-rank representation, but many of its sub-matrices can be proven to have one. A hierarchical matrix is an abstract structure that allows for the division of the system matrix in sub-matrices, permitting the exploitation of low-rank approximations whenever possible, providing and algebraic structure in order to perform the matrix operations required to solve the system; most notably, matrix-vector multiplications used in iterative solvers. Hierarchical matrices are historically associated to the exploitation of low-rank approximations using the Panel Clustering, the Taylor expansion or the adaptive cross-approximation method. The Fast Multipole Method has relied in a similar but different technique also aiming to divide the system matrix; however, the procedure can also be described in term of hierarchical matrices.

3.3.1 Index Sets, Clusters and Cluster Trees

In this section a notational framework is to be developed. This notational framework will provide means to refer to the sub-matrices of the system matrix and to identify the matrix elements with the basis functions spanning the finite subspace of the function space where the variational version of the integral equation is defined.

Definition 4 (Index Set) *An index set $\mathcal{I} = \{1, 2..., n\} \subset \mathbb{N}$ is a set containing the indexes of the basis functions used in a Galerkin discretization.*

Definition 5 (Cluster) *A cluster is a set of indexes $t \subseteq \mathcal{I}$, where $\mathcal{I}$ is the index set related to a Galerkin discretization.*

Definition 6 (Sub-Vector Associated to a Cluster) *Let $x \in \mathbb{C}^m$ be a given vector and let $\mathcal{I}$ be the index set $\mathcal{I} = \{1, 2...m\}$. Let $t \subseteq \mathcal{I}$ be a cluster for the index sets $\mathcal{I}$. The sub-vector $x_t \in \mathbb{C}^{|t|}$ associated to the cluster t is the restriction of x to the indexes belonging to t.*

Given a discretization Γ_h of a surface Γ over which a Galerkin discretization base has been considered with indexes given by the index set $\mathcal{I}$, a cluster $t \subseteq \mathcal{I}$ can be associated with a spatial domain.

Definition 7 (Cluster Domain) *Let $\Omega_i \in \mathbb{R}^d$ be the support of the basis function φ_i. The support of the cluster t is then*

$$\Omega_t = \bigcup_{i \in t} \Omega_i.$$

The main idea behind the hierarchical matrix or other similar structures used in the fast methods relies on the concept of divide-and-conquer: to divide the system matrix up to the point where low-rank matrices can be used to provide an overall advantage. The key division is given at the level of the basis functions indexed by the index set $\mathcal{I}$, which is contained in the abstract structure of the cluster tree.

Definition 8 (Cluster Tree) *Let us define a tree $T_{\mathcal{I}} := (N, E)$ of clusters with a set of nodes N and a set of edges E. Let $S(t)$ be the set of successors of the node $t \in N$ and $\mathcal{L}(T_{\mathcal{I}}) \subset N$ the set of nodes that have no successor nodes (the leaves of the tree). The tree $T_{\mathcal{I}} := (N, E)$ is a cluster tree of an index set $\mathcal{I}$ if*

1. *$\mathcal{I} \in N$ is the root of the tree;*
2. *$\forall t \in N \backslash \mathcal{L}(T_{\mathcal{I}}) \left(t = \dot{\bigcup}_{t' \in S(t)} t' \neq \varnothing \right)$;*
3. *the degree of a node $t \in N$ is defined as the number of successors $deg(t) = |S(t)|$ for each node in $N \backslash \mathcal{L}(T_{\mathcal{I}})$ and it is bounded form bellow: $deg(t) \geq 2$.*

Remark 1 (Minimum Size n_{min} of a Leaf Cluster) *In practice it is useful to work with clusters having a minimal size $n_{max} > 1$ rather than dividing the index set up to singleton leaves. This number n_{min} is used in the construction of the cluster tree assuring that no division is to be carried out if it will produce clusters with less than n_{min} elements. This number controls the size of the sub-matrices that will not be approximated by low-rank matrices and that will thus have to be operated and stored as full matrices.*

The following set of definitions provides the required tools to characterize important properties of the cluster trees.

Definition 9 (Level of a Cluster in a Cluster Tree) *Let $T_{\mathcal{I}} \coloneqq (N, E)$ be a cluster tree for the index set $\mathcal{I}$.The level of a given cluster $t \in N$ in the cluster tree $T_{\mathcal{I}}$, notated as $level(t)$, is the distance to the root $\mathcal{I}$, i.e., the number of successions (application of S) from $\mathcal{I}$ required to reach the node t in the tree.*

Definition 10 (Level l of a Cluster Tree) *The level l of a cluster tree $T_{\mathcal{I}}$, notated as $T_{\mathcal{I}}^{(l)}$, is the set of the nodes $t \in N$ that have level equal to l:*

$$T_{\mathcal{I}}^{(l)} = \{t \in N; T_{\mathcal{I}} := (N, V), level(t) = l\}.$$

Remark 2 (Partitions of $\mathcal{I}$) *According to Definition 8, at every level l, $T_{\mathcal{I}}^{(l)}$ is a partition of $\mathcal{I}$ in the sense that*

$$\mathcal{I} = \dot{\bigcup_{t' \in T_{\mathcal{I}}^{(l)}}} t'.$$

.

Definition 11 (The Set of Levels That Have Leaves) *The set L of the levels of a tree $T_{\mathcal{I}}$ that have at least one leaf is defined as*

$$L = \left\{ l \in \mathbb{N}_0; \mathcal{L}(T_{\mathcal{I}}) \cap T_{\mathcal{I}}^{(l)} \neq \varnothing \right\}.$$

Definition 12 (Depth of a Cluster Tree) *The depth of a cluster tree $T_{\mathcal{I}}$, notated as $depth(T_{\mathcal{I}})$, is the number of different levels present in the tree, i.e.,*

$$depth(T_{\mathcal{I}}) = 1 + \max_{t \in N} \{level(t)\}.$$

Definition 13 (Balanced Cluster Tree) *A tree $T_{\mathcal{I}}$ is balanced if the quantity*

$$R := \min_{t \in N \setminus \mathcal{L}(T_{\mathcal{I}})} \{|t_1|/|t_2|; t_1, t_2 \in S(t)\}$$

is bounded from below independently of $|\mathcal{I}|$.

Lemma 2 (Number of Nodes $|N|$ in a Cluster Tree) *Let $T_{\mathcal{I}} \coloneqq (N, E)$ be a cluster tree for the index set $\mathcal{I}$ and let $q \coloneqq \min_{t \in N \setminus \mathcal{L}(T_{\mathcal{I}})} deg(t)$. Then, the number of nodes $|N|$ of the cluster tree $T_{\mathcal{I}}$ is bounded by the number of leaves as*

$$|N| \leq \frac{q|\mathcal{L}(T_{\mathcal{I}})| - 1}{q - 1} \leq 2|\mathcal{L}(T_{\mathcal{I}})| - 1.$$

Demonstration The number of nodes that have successor nodes is $|N| - |\mathcal{L}(T_{\mathcal{I}})|$. The number of nodes which are successors of some other node is at least $q(|N| - |\mathcal{L}(T_{\mathcal{I}})|)$. The total number of nodes is the numbers of nodes which are successor plus the root node, thus yielding

$$q(|N| - |\mathcal{L}(T_{\mathcal{I}})|) + 1 \leq |N|,$$

from which the desired bound for $|N|$ can be established: $|N| \leq (q|\mathcal{L}(T_{\mathcal{I}})| - 1)/(q - 1)$. Additionally, Definition 8 assures $q \geq 2$, which implies that $q/(q-1) \leq 2$ and $1/(q-1) \leq 1$, thus allowing for a simple bound for the number of nodes in the tree: $|N| \leq 2|\mathcal{L}(T_{\mathcal{I}})| - 1$. ■

Remark 3 (Storage Complexity of a Cluster Tree) *Since the number of leaves $|\mathcal{L}(T_{\mathcal{I}})|$ of a cluster tree $T_{\mathcal{I}}$ is bounded by $|\mathcal{I}|/n_{min}$, the previous lemma shows that the storage complexity of a cluster tree is linearly bounded by the cardinality $|\mathcal{I}|$ of $\mathcal{I}$: $|N| \leq 2|\mathcal{I}|/n_{min} - 1$.*

Lemma 3 (Maximum Depth of a Balanced Cluster Tree) *For a balanced cluster tree $T_{\mathcal{I}}$ with $q := \min_{t\in N\setminus\mathcal{L}(T_{\mathcal{I}})} deg(t)$ and $R := \min_{t\in N\setminus\mathcal{L}(T_{\mathcal{I}})}\{|t_1|/|t_2|; t_1, t_2 \in S(t)\}$ and for a node/cluster t of level l in the cluster tree it holds that $|t| \leq |\mathcal{I}|\xi^{-l}$, with $\xi = R(q-1)+1 \geq 1$, and that the depth of the tree is bounded by*

$$depth(T_{\mathcal{I}}) \leq 1 + \log_{\xi}(|\mathcal{I}|/n_{max}) \sim \log_{\xi}|\mathcal{I}|.$$

Demonstration (Taken from Bebendorf [6], page 31) Let us first find a lower bound for the ratio of the cardinalities of a cluster and one of its successor clusters in a cluster tree. Let us consider a non-leaf cluster $t \in N\setminus\mathcal{L}(T_{\mathcal{I}})$ and one of its successor cluster $t' \in S(t)$. Given $q := \min_{t\in N\setminus\mathcal{L}(T_{\mathcal{I}})} deg(t)$ and $R := \min_{t\in N\setminus\mathcal{L}(T_{\mathcal{I}})}\{|t_1|/|t_2|; t_1, t_2 \in S(t)\}$ a lower bound can be found for the ratio

$$\frac{|t|}{|t'|} = \frac{|t'| + \sum\limits_{s\in S(t), s\neq t'} |s|}{|t'|} = 1 + \sum_{s\in S(t), s\neq t'} \frac{|s|}{|t'|} \geq 1 + (|S(t)|-1)R \geq 1 + (q-1)R = \xi$$

Secondly, let us consider the path connecting the root and the highest level node in a tree of depth D, composed of edges $e_1, e_2, ...e_{D-1}$ and passing through the intermediate nodes $n_2, n_3, ...n_{D-1}$. From the previous result we know that

$$\xi|n_{l+1}| \leq |n_l|, \text{ for } l = 1, ..., D-1,$$

from which we obtain that

$$\xi^{D-1}|n_D| \leq |n_1| = |\mathcal{I}|,$$

which gives us the desired bound for the depth:

$$(D-1)\log\xi \leq \log\frac{|\mathcal{I}|}{|n_L|} \leq \log\frac{|\mathcal{I}|}{n_{min}}$$

$$\Rightarrow L \leq 1 + \frac{\log|\mathcal{I}|/n_{min}}{\log\xi} = 1 + \log_{\xi}\frac{|\mathcal{I}|}{n_{min}} \sim \log_{\xi}|\mathcal{I}|$$

■

A geometrical approach to construct a cluster tree for a set index $\mathcal{I}$ representing the basis of a Galerkin discretization is to subdivide the spatial domain $\Omega_{\mathcal{I}}$ in consecutive sub-domains generating successors for each cluster based on the geometrical information available. Classical choices are to start from the root $\mathcal{I}$ (with associated domain $\Omega_{\mathcal{I}}$) and subdivide this domain recursively in 8, 4 or 2 regular subdivisions assuring that clusters (nodes of the cluster tree) have cardinality larger than a number n_{min} set for the tree. This number assures that all the leaves of the tree, $t \in \mathcal{L}(T_{\mathcal{I}})$, have at least n_{min} elements. These cluster tree structures are known as *Octree* when using 8 subdivisions, *Quadtree* when using 4 subdivisions and *Binary Tree* when using 2 subdivisions.

Remark 4 (Balanced Trees and Principal Component Analysis (PCA)) *The mentioned construction of a cluster tree, called geometric bisection, may produce unbalanced trees. The subdivisions of domains is performed using Bounding Boxes, box-like domains of tensorized segments along the reference axes. Modifications to the geometric subdivision approach are used to avoid unbalanced trees if needed. For example, the Principal Component Analysis (PCA), in which the subdivision is performed also geometrically but according to new spatial axes oriented along the main directions of a set of points representative of the support of the basis functions.*

3.3.2 Block-Clusters

Since the elements in a system matrix arise from the bilinear operator in a variational formulation involving a discrete integral operator, they are related to the interaction of couples of basis functions through the kernel of the integral equation. In order to analyze blocks of sub-matrices of the system matrix it is useful to provide proper notations to link the concept of clusters and their interaction with basis functions belonging to other basis functions that may be they be contained in that cluster or in others.

Definition 14 (Block-Cluster) *Let us consider two (possibly the same) index sets $\mathcal{I}$ and $\mathcal{J}$ indexing basis functions spanning a finite-dimensional sub-space of a functional space where the variational formulation of the integral equation is to be solved. A block-cluster $t \times s$ is a set of pairs of indexes belonging to given index sets, $t \subseteq \mathcal{I}$ and $s \subseteq \mathcal{J}$.*

A sub-matrix of the system matrix represents the interaction of pairs of elements of the basis of the functional space of the variational formulation for the integral equation. Each element of a sub-matrix represents the interaction of a pair of functions. The block-clusters represent sets of interacting pairs of basis functions giving rise to a sub-matrix of the system matrix.

Definition 15 (Sub-Matrix Associated to a Block-Cluster) *Let $A \in \mathbb{C}^{m\times n}$ be a given matrix containing the pairwise interactions of the basis functions indexed by the index sets $\mathcal{I} = \{1,2...m\}$ and $\mathcal{J} = \{1,2...n\}$. Let $t \times s \subseteq \mathcal{I} \times \mathcal{J}$ be a block-cluster for the index sets $\mathcal{I}$ and $\mathcal{J}$. The sub-matrix $A_{t\times s} \in \mathbb{C}^{|t|\times|s|}$ associated to the block-cluster $t \times s$ is the restriction of A to the indexes belonging to $t \times s$.*

3.3.3 Geometrical Admissibility of Block-Clusters

The block-cluster provides a structure to identify sub-matrices of the system matrix with interactions of pairs of basis functions in the sense provided by the bilinear operator of the variational formulation of the integral equation. The core of the hierarchical matrix and the other fast methods for the solution of the boundary integral equations is to identify which sub-matrices (which block-clusters) can be represented in a more advantageous way. In the case of the hierarchical matrix method the key issue is the division of the system matrix in a hierarchy of sub-matrices identifying when they can be represented by a suitable low-rank approximation. This characteristic of a block-cluster is linked to the underlying integral equation, and reflects the fact that the Schwartz integral kernel $G(x,y)$ of the integral equation is singular only for $x = y$, situation that arises when $\Omega_t \cap \Omega_s \neq \emptyset$ for the block-cluster $t \times s$. This geometrical notion will be used to select block-clusters that have associated matrices with desirable properties in the sense that low-rank approximations can be fruitfully computed. The following criterion assures that the geometrical supports of two clusters constituting a block-cluster are disjoint. It will be shown later that this criterion also assures other features of the associated sub-matrix, which are desirable for computing low-rank approximations.

Definition 16 (Geometrical Admissibility of a Block-Cluster) *A block-cluster $t \times s \subseteq \mathcal{I} \times \mathcal{J}$ for the index sets $\mathcal{I} = \{1,2,...,m\}$ and $\mathcal{J} = \{1,2,...,n\}$ indexing the basis functions is said to be* **geometrically admissible** *if and only if*

$$\min\{diam(\Omega_t), diam(\Omega_s)\} \leq \eta\, dist(\Omega_t, \Omega_s), \tag{10}$$

for some $\eta > 0$.

The previous definition provides a concrete procedure with which to test wether a block-cluster is admissible or not. There still remains the problem of the complexity required to test the geometrical admissibility, especially in the computation of $dist(\Omega_t, \Omega_s)$, which requires $\mathcal{O}(|t| \cdot |s|)$ operations. The next definition proposes a new condition suitable for special cases of great interest that only requires $\mathcal{O}(|t| + |s|)$ operations.

Definition 17 (Geometrical Admissibility for Polygonal Supports) *Let Ω_t and Ω_s have piecewise polygonal supports (Ω_i is a polygon for all $i \in t \cup s$) and let m_t and m_s be the centroids of Ω_t and Ω_s respectively. Let us set*

$$\rho_t := sup\{\|x - m_t\|_2; x \in \Omega_t\} \text{ and } \rho_s := sup\{\|x - m_s\|_2; x \in \Omega_s\}.$$

A block-cluster $t \times s \subseteq \mathcal{I} \times \mathcal{J}$ for the index sets $\mathcal{I} = \{1, 2, ..., m\}$ and $\mathcal{J} = \{1, 2, ..., n\}$ indexing the basis functions is said to be **simply admissible** *if and only if*

$$2 \min\{\rho_t, \rho_s\} + \eta(\rho_t + \rho_s) \leq \eta \|m_t - m_s\|_2, \tag{11}$$

for some $\eta > 0$.

Lemma 4 () *Simply admissible block-clusters are geometrically admissible.*

Demonstration For two clusters $t \subset \mathcal{I}$ and $s \subset \mathcal{J}$ it is easy to check that

$$dist(\Omega_t, \Omega_s) \geq \|m_t - m_s\|_2 - \rho_t - \rho_s.$$

If the block-cluster $t \times s$ is simply admissible

$$\begin{aligned} 2 \min\{\rho_t, \rho_s\} + \eta(\rho_t + \rho_s) \leq \eta \|m_t - m_s\|_2 \quad &\Rightarrow 2 \min\{\rho_t, \rho_s\} \leq \eta \left(\|m_t - m_s\|_2 - \rho_t - \rho_s\right) \\ &\Rightarrow 2 \min\{\rho_t, \rho_s\} \leq \eta \cdot dist(\Omega_t, \Omega_s). \end{aligned}$$

Considering that $diam(\Omega_t) \leq 2\rho_t$ and $diam(\Omega_s) \leq 2\rho_s$ the desired bound is obtained. ■

It is clear that for a piecewise polygonal support condition (10) takes $\mathcal{O}(|t| \cdot |s|)$ operations, since the computation of the distance between Ω_t and Ω_s would require two nested loops computing the maximum distance between each pair of polygonal support pieces indexed by t and s. On the other hand it is cleat that condition (11) requires only $\mathcal{O}(|t| + |s|)$ operations, since the centroids can be computed with a loop over the polygons and the maximum radius form the centroids can be compute with a seconds loop over the polygons.

3.3.4 Block-Cluster Trees and Admissible Partitions of the System Matrix

The following definitions provides a formal structure to construct a subdivision of the system matrix in sub-matrices using the before-mentioned geometrical criterion and a cluster-tree in aims to identify which sub-matrices can be approximated by a low-rank approximation.

Definition 18 (Block-Cluster Tree) *Let $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ be cluster-trees for the index sets $\mathcal{I}$ and $\mathcal{J}$. Let $S_{\mathcal{I}}$ and $S_{\mathcal{J}}$ be the successor functions of each tree. The block-cluster tree $T_{\mathcal{I} \times \mathcal{J}} := (N, E)$ for the product $\mathcal{I} \times \mathcal{J}$ is constructed as follows:*

1. *The root of the tree is the set $\mathcal{I} \times \mathcal{J}$.*

2. *The vertices of the tree are block-clusters $t \times s \subseteq \mathcal{I} \times \mathcal{J}$.*

3. *The vertices of the tree are defined by the succession function $S_{\mathcal{I} \times \mathcal{J}}$:*

$$S_{\mathcal{I} \times \mathcal{J}}(t \times s) = \begin{cases} \varnothing, & \text{if } t \times s \text{ is geom. admissible or } S_{\mathcal{I}}(t) = \varnothing \text{ or } S_{\mathcal{J}}(s) = \varnothing, \\ S_{\mathcal{I}}(t) \times S_{\mathcal{J}}(s), & \text{else.} \end{cases}$$

Lemma 5 (Depth of a Block-Cluster Tree) *The depth $depth(T_{\mathcal{I}\times\mathcal{J}})$ of a block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ is bounded by the depths of the generating cluster trees:*

$$depth(T_{\mathcal{I}\times\mathcal{J}}) \leq \min\{depth(T_{\mathcal{I}}), depth(T_{\mathcal{J}})\}$$

Demonstration It is easy to see that condition 3 of the construction of a block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ specified in Definition 18 assures that $depth(T_{\mathcal{I}\times\mathcal{J}}) \leq \min\{depth(T_{\mathcal{I}}), depth(T_{\mathcal{J}})\}$. ■

The leaves $\mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})$ of the block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ form an **admissible partition** $P := \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})$ of the root block-cluster $\mathcal{I}\times\mathcal{J}$ in the sense given by the following definition.

Definition 19 (Admissible Partition) *An* **admissible partition** *P of $\mathcal{I}\times\mathcal{J}$ is a subset $P \subset \mathcal{P}(\mathcal{I}\times\mathcal{J})$ of the subsets of $\mathcal{I}\times\mathcal{J}$ that complies with the following statements:*

1. $\mathcal{I}\times\mathcal{J} = \bigcup\limits_{b\in P} b$.
2. $\forall b_1, b_2 \in P\,(b_1 \cap b_2 \neq \varnothing \Rightarrow b_1 = b_2)$.
3. $\forall b = t\times s \in P\,(b$ *is a geometrically admissible block-cluster or* $t \in \mathcal{L}(T_{\mathcal{I}})$ *or* $s \in \mathcal{L}(T_{\mathcal{J}}))$.

Definition 20 (Admissible and Non-admissible Leaves of a Block-Cluster Tree) *The set $\mathcal{L}^A(T_{\mathcal{I}\times\mathcal{J}})$ is the subset of leaves of $T_{\mathcal{I}\times\mathcal{J}}$ that are geometrically admissible:*

$$\mathcal{L}^A(T_{\mathcal{I}\times\mathcal{J}}) = \{t\times s \in \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}}); t\times s \text{ is geometrically admissible}\}.$$

The set $\mathcal{L}^{NA}(T_{\mathcal{I}\times\mathcal{J}}) = \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})\backslash\mathcal{L}^A(T_{\mathcal{I}\times\mathcal{J}})$ is the subset of leaves of $T_{\mathcal{I}\times\mathcal{J}}$ that are not admissible.

For an admissible partition P of $\mathcal{I}\times\mathcal{J}$ given by the block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ a common measure of complexity useful in the demonstration of several of the characteristics of hierarchical matrices is the sparsity constant.

Definition 21 (Sparsity Constant) *Let $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ be cluster trees for the index sets $\mathcal{I}$ and $\mathcal{J}$ and let $T_{\mathcal{I}\times\mathcal{J}}$ be the block-cluster tree for $\mathcal{I}\times\mathcal{J}$. Similarly to the number of elements in a given row of a matrix we denote the number of blocks $t\times s \in T_{\mathcal{I}\times\mathcal{J}}$ associated to a given cluster $t \in T_{\mathcal{I}}$ as*

$$c_{sc}^{row}(T_{\mathcal{I}\times\mathcal{J}}, t) := |\{s \subseteq \mathcal{J}; t\times s \in T_{\mathcal{I}\times\mathcal{J}}\}|.$$

Similarly to the number elements in a given column of a matrix we denote the number of blocks $t\times s \in T_{\mathcal{I}\times\mathcal{J}}$ associated to a given cluster $s \in T_{\mathcal{J}}$ as

$$c_{sc}^{col}(T_{\mathcal{I}\times\mathcal{J}}, s) := |\{t \subseteq \mathcal{I}; t\times s \in T_{\mathcal{I}\times\mathcal{J}}\}|.$$

The **sparsity constant** *of a block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ is then defined as*

$$c_{sc}(T_{\mathcal{I}\times\mathcal{J}}) := \max\left\{\max_{t\in T_{\mathcal{I}}} c_{sc}^{row}(T_{\mathcal{I}\times\mathcal{J}}, t), \max_{s\in T_{\mathcal{J}}} c_{sc}^{col}(T_{\mathcal{I}\times\mathcal{J}}, s)\right\}.$$

Remark 5 (Boundedness of the Sparsity Constant) *The sparsity constant $c_{sc}(T_{\mathcal{I}\times\mathcal{J}})$ can be kept bounded through a parametrized construction of the clusters trees $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ giving rise to the block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$. A method to do so is the Principal Component Analysis (PCA) seen in Remark 4.*

Lemma 6 (Storage Complexity of a Block-Cluster Tree) *Let $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ be cluster trees for the index sets $\mathcal{I}$ and $\mathcal{J}$ and let $T_{\mathcal{I}\times\mathcal{J}} := (N, E)$ be the block-cluster tree for $\mathcal{I}\times\mathcal{J}$ constructed with sparsity constant $c_{sc}(T_{\mathcal{I}\times\mathcal{J}})$. Then, the number of nodes $|N|$ of $T_{\mathcal{I}\times\mathcal{J}}$ is bounded as*

$$|N| \leq c_{sc}(T_{\mathcal{I}\times\mathcal{J}})\left(\frac{2}{n_{min}}\min\{|\mathcal{I}|, |\mathcal{J}|\} - 1\right).$$

Demonstration The number of nodes in a block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ can be represented as

$$|N| = \sum_{s\times t \in T_{\mathcal{I}\times\mathcal{J}}} 1.$$

If $depth(T_{\mathcal{I}}) \leq depth(T_{\mathcal{J}})$ then

$$|N| = \sum_{s\times t \in T_{\mathcal{I}\times\mathcal{J}}} 1 = \sum_{t\in T_{\mathcal{I}}} |\{s \in \mathcal{J}; t \times s \in T_{\mathcal{I}\times\mathcal{J}}\}| = \sum_{t\in T_{\mathcal{I}}} c_{sc}^{row}\left(T_{\mathcal{I}\times\mathcal{J}}, t\right).$$

Using Definition 21 and notating the number of nodes of $T_{\mathcal{I}}$ as $|T_{\mathcal{I}}|$ this means that

$$|N| \leq c_{sc}(T_{\mathcal{I}\times\mathcal{J}})|T_{\mathcal{I}}|.$$

Using now Lemma 2 and Remark 3 it is straightforward that

$$|N| \leq c_{sc}(T_{\mathcal{I}\times\mathcal{J}})\left(\frac{2}{n_{min}}|\mathcal{I}| - 1\right).$$

On the contrary, if $depth(T_{\mathcal{J}}) \leq depth(T_{\mathcal{I}})$ the bound would be

$$|N| \leq c_{sc}(T_{\mathcal{I}\times\mathcal{J}})\left(\frac{2}{n_{min}}|\mathcal{J}| - 1\right).$$

In any case, the bound of the lemma holds. ■

3.3.5 Hierarchical Matrices

In this section, the structure of the hierarchical matrix is presented for binary cluster trees $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ for the index sets $\mathcal{I}$ and $\mathcal{J}$ (a generalization to arbitrary cluster-trees is possible using nested dissection). The block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ is assumed to be generated using the above-mentioned geometrical admissibility condition.

Definition 22 (The Set of Hierarchical Matrices) *Let $T_{\mathcal{I}\times\mathcal{J}}$ be the block-cluster tree for the index sets $\mathcal{I}$ and $\mathcal{J}$ and let $P := \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})$ be an admissible partition of $\mathcal{I}\times\mathcal{J}$. The set of blockwise k-rank hierarchical matrices for the bock-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$ is the set*

$$\mathcal{H}(T_{\mathcal{I}\times\mathcal{J}}, k) = \left\{A \in \mathbb{C}^{\mathcal{I}\times\mathcal{J}}; \forall b \in P(b \textit{ is admissible } \Rightarrow rank(A_b) \leq k)\right\}.$$

In the following the elements of $\mathcal{H}(T_{\mathcal{I}\times\mathcal{J}}, k)$ will be called $\mathcal{H}$-matrices.

Theorem 5 (Storage Complexity of $\mathcal{H}$-matrices) *Let c_{sc} be the sparsity constant of the block-cluster tree $T_{\mathcal{I}\times\mathcal{J}}$. The number $N_{storage}(A)$ of elements to be stored for a $\mathcal{H}$-matrix $A \in \mathcal{H}(T_{\mathcal{I}\times\mathcal{J}}, k)$ is bounded by*

$$N_{storage}(A) \leq c_{sc}\max\{k, n_{min}\}\left(depth(T_{\mathcal{I}})|\mathcal{I}| + depth(T_{\mathcal{J}})|\mathcal{J}|\right).$$

If $T_{\mathcal{I}}$ and $T_{\mathcal{J}}$ are balanced cluster trees the previous bound can be further extended to a more easily computable number, by virtue of Lemma 3, as

$$N_{storage}(A) \leq c_{sc}\max\{k, n_{min}\}\left(|\mathcal{I}|\log_{\xi}|\mathcal{I}| + |\mathcal{J}|\log_{\xi}|\mathcal{J}|\right),$$

for a constant ξ depending on the construction parameters for the cluster trees, as specified in the lemma.

Demonstration(Taken from [14], Chapter 2) The total number of elements to be stored resides in the sub-matrices associated to the leaves of $T_{\mathcal{I}\times\mathcal{J}}$, which can be separated by their geometric admissibility. Let $N^A(t\times s)$ be the maximum number of elements required to store the sub-matrix associated to an admissible leaf $t\times s$ and let $N^{NA}(t\times s)$ be the maximum number of elements required to store a non-admissible leaf $t\times s$. Separating the elements associated to admissible and non-admissible leaves yields

$$N_{storage}(A) \le \sum_{t\times s\in\mathcal{L}^A(T_{\mathcal{I}\times\mathcal{J}})} N^A(t\times s) + \sum_{t\times s\in\mathcal{L}^{NA}(T_{\mathcal{I}\times\mathcal{J}})} N^{NA}(t\times s).$$

It is easy to see that $N^{NA}(t\times s)\le n_{min}\max\{|t|,|s|\}\le n_{min}(|t|+|s|)$. On the other hand Theorem 4 assures that $N^A(t\times s)\le k(|t|+|s|)$, which implies that

$$\begin{aligned} N_{storage}(A) &\le \sum_{t\times s\in\mathcal{L}^A(T_{\mathcal{I}\times\mathcal{J}})} k(|t|+|s|) + \sum_{t\times s\in\mathcal{L}^{NA}(T_{\mathcal{I}\times\mathcal{J}})} n_{min}(|t|+|s|) \\ &\le \sum_{t\times s\in\mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})} \max\{k,n_{min}\}|t| + \sum_{t\times s\in\mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})} \max\{k,n_{min}\}|s|. \end{aligned}$$

Furthermore, the sum over all the leaves of the block-cluster tree can be bounded by the sum over the leaves of one of the constituting cluster-trees using the sparsity constant, and this in turn can be bounded by the sum of all nodes in the levels containing leaves:

$$N_{storage}(A) \le \sum_{l\in L(T_{\mathcal{I}})}\sum_{t\in T_{\mathcal{I}}^{(l)}} c_{sc}\max\{n_{min},k\}|t| + \sum_{l\in L(T_{\mathcal{J}})}\sum_{s\in T_{\mathcal{J}}^{(l)}} c_{sc}\max\{n_{min},k\}|s|.$$

As noted in Remark 2, every level l of a cluster tree is a partition of the index set, and thus

$$N_{storage}(A) \le \sum_{l\in L(T_{\mathcal{I}})} c_{sc}\max\{n_{min},k\}|\mathcal{I}| + \sum_{l\in L(T_{\mathcal{J}})} c_{sc}\max\{n_{min},k\}|\mathcal{J}|.$$

Additionally, the number of levels containing leaves can be bounded by the depth of the tree giving the desired result. ■

In the rest of this section we define the most relevant operation involving the $\mathcal{H}$-matrices in the context of the resolution of integral equations, the matrix-vector product. We also present its arithmetic complexity.

Definition 23 (Vector Composition Using Clusters) *Let $T_{\mathcal{I}}$ be a cluster tree for the index set $\mathcal{I}$, and let $t\subset\mathcal{I}$ be one of the clusters of the tree. The composition $C_t^{\mathcal{I}}(x)$ of a vector $x\in\mathbb{C}^{|t|}$ is a vector in $\mathbb{C}^{|\mathcal{I}|}$ whose restriction $(C_t^{\mathcal{I}}(x))_t$ to the cluster t (in the sense of Definition 6) is equal to x and such that*

$$\forall i\in\mathcal{I}\left(i\notin t\Rightarrow (C_t^{\mathcal{I}}(x))_i = 0\right).$$

Definition 24 ($\mathcal{H}$-matrix Multiplication by a Vector) *Let $A\in\mathcal{H}(T_{\mathcal{I}\times\mathcal{J}},k)$. Let $P := \mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})$ be the admissible partition of the root block-cluster $\mathcal{I}\times\mathcal{J}$. The product Ax is computed as*

$$Ax = \sum_{t\times s\in P} C_t^{\mathcal{I}}(A_{t\times s}x_s).$$

Lemma 7 (Arithmetic Complexity of Matrix-Vector Product for $\mathcal{H}$-matrices) *The number of arithmetic operations $N_{mv}(A)$ required to multiply the $\mathcal{H}$-matrix $A\in\mathcal{H}(T_{\mathcal{I}\times\mathcal{J}},k)$ by $x\in\mathbb{C}^{|\mathcal{J}|}$ is bounded by*

$$N_{mv}(A)\le 2N_{storage}(A).$$

Demonstration(Taken from [14], Chapter 2) To perform the multiplication of an $\mathcal{H}$-matrix with a vector, $|\mathcal{L}(T_{\mathcal{I}\times\mathcal{J}})|$ matrix-vector multiplications must be made; some of them with full matrices and some of them low-rank matrices of rank k.

The cost $N_{storage}(A_{t\times s})$ of storing a full matrix $A_{t\times s}$ is of $|t|\cdot|s|$ elements. The cost $N_{mv}(A_{t\times s})$ of multiplying a full matrix with a vector is of $|t|\cdot|s| + |t|(|s|-1)$ operations. Thus, for a full matrix $A_{t\times s}$, $N_{mv}(A_{t\times s}) \leq 2N_{storage}(A_{t\times s})$.

The cost $N_{storage}(A_{t\times s})$ of storing a k-rank matrix $A_{t\times s}$ is of $k(|t|+|s|)$ elements. The cost $N_{mv}(A_{t\times s})$ of multiplying a k-rank matrix with a vector is of $2k(|t|+|s|) - |t| - k$ operations. Thus, for a k-rank matrix $A_{t\times s}$, we also have $N_{mv}(A_{t\times s}) \leq 2N_{storage}(A_{t\times s})$.

The cost of producing all the matrix-vector multiplications for the sub-matrices associated to the leaves is less than $2N_{storage}(A)$. ∎

3.4 Low-Rank Approximation of Matrices Arising in the Discretization of Integral Operators

In the first section of this chapter we analyzed what low-rank matrices, how they arise in the discretization of integral operators and how they reduce the computational complexity. In the second section we described a special type of structure, the $\mathcal{H}$-matrices, that can divide a matrix, based on the geometry of the integration domain and the support of the basis functions, into sub-matrices being either small enough or capable of being approximated by low-rank matrices. In the present and third section we will study when do low-rank approximations exist, how good are they, how to compute them and important features of these approximations related to the complexity of their construction and the relation between rank and precision.

3.4.1 The Existence of Low-Rank Approximations

Lesser rank approximants can be computed for every matrix . The quality of the approximation will be given by its rank, and wether if that approximant is a low-rank approximation will be dictated by compliance with Definition 3. The following theorem provides a method for constructing lesser rank approximations for any matrix giving a relation between the rank of the approximant and the precision of the approximation.

Theorem 6 (Best Approximation Via Singular Value Decomposition) *Let the matrix $A \in \mathbb{C}^{m\times n}$ have a Singular Value Decomposition noted as*

$$A = U\Sigma V^H$$

with unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ ($U^H U = I_{m\times m}$ and $V^H V = I_{n\times n}$) and diagonal matrix $\Sigma \in \mathbb{C}^{m\times n}$

$$\Sigma = diag(\{\sigma_1, .., \sigma_{min(m,n)}\}), \quad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{min(m,n)} \geq 0.$$

Let ε be a desired accuracy. If $\sigma_k > \varepsilon > \sigma_{k+1}$, then the matrix

$$R = \sum_{l=1}^{k} u_l \sigma_l v_l^T,$$

is the minimal rank approximation of A that fulfills $\|A - R\|_2 \leq \varepsilon$.

Demonstration The demonstration can be found in the book *Matrix Computations* [12], by Golub and Van Loan. ∎

Even for a matrix known to have a low-rank approximation a singular value decomposition is a costly operation. For a $m\times n$ matrix the computational complexity of performing a singular value decomposition is of $\mathcal{O}(mn^2)$ operations (assuming $m \geq n$), and its storage complexity is of $m^2 + n^2 + \min\{m, n\}$. The previous theorem is useful because it shows that lesser rank approximations exist but, even if low-rank approximants exist, the procedure it proposes is useless because its complexity overcomes the advantages of using low-rank approximations.

In general, approximating matrices of increasing rank and precision can be constructed using only selected elements of the original matrix. This allows for the computation of low-rank approximations with less operations and also avoids the need to compute all the original matrix entries. These techniques are known as skeleton-approximations or cross-approximation methods. The following definition provides the basic structure involved in cross-approximation methods.

Definition 25 (Cross or Skeleton-Approximation) *Let $A_{t\times s}$ be a matrix associated to the block-cluster $t \times s$. Let $\tilde{t} \subset t$ and $\tilde{s} \subset s$ be the sets of pivot rows and pivot columns of the matrix $A_{t\times s}$ such that there exists a matrix $S \in \mathbb{C}^{|\tilde{t}|\times|\tilde{s}|}$ that allows*

$$\|A_{t\times s} - A_{t\times\tilde{s}} S A_{\tilde{t}\times s}\|_2 \leq \varepsilon.$$

The matrix $R = A_{t\times\tilde{s}} S A_{\tilde{t}\times s}$ is a cross or skeleton approximation of precision ε and rank $\min\{|\tilde{t}|, |\tilde{s}|\}$.

The question remains wether if they exist, how to compute them, and if it can be done efficiently. The following theorem provides a positive answer to the first question.

Theorem 7 (Existence of Cross-Approximations for Matrices Admitting Low-Rank Approxima *Let $M \in \mathbb{C}^{m\times n}$ admit a k-rank approximation, i.e., for a given $\varepsilon > 0$ there exists $R \in \mathbb{C}_k^{m\times n}$ such that $\|M - R\|_2 \leq \varepsilon$. Then there exist k pivots such that a cross approximation $\tilde{M}$ can be constructed from M fulfilling*

$$\|M - \tilde{M}\|_2 \leq \varepsilon(1 + 4k)$$

Demonstration The demonstration can be found in the article *A theory of pseudo skeleton approximations* [13], by Goreinov, Tyrtyshnikov, Zamarashkin. ■

Given that cross-approximations exist for matrices having k-rank approximations the focus now turns to the existence of low-rank approximations. Once that matrices arising in BEM are proven to have low-rank approximations the issue of finding cross-approximations is addressed.

3.4.2 The Relation Between the Kernel, the Existence of Degenerate Approximants and Its Quality

In this section we explore the existence of degenerate kernel approximants for the BIE and the BEM, bounds for the quality of this approximation and the implications to the quality of the approximation made by the associated low-rank matrices. An important feature of the kernel used in the determination of its capacity to be degenerated is its **asymptotical smoothness**, which will be specified in the following definition. Let us consider for this the framework of a BIE, where an integral kernel G is integrated twice on a boundary $(d-1)$-dimensional surface $\Gamma \subset \mathbb{R}^d$.

Definition 26 (Asymptotically Smooth Kernel) *A function $G : \Gamma \times \mathbb{R}^d \to \mathbb{C}$ satisfying $G(x, \cdot) \in C^\infty(\mathbb{R}^d \backslash \{x\})$ for all $x \in \Gamma$ is called* **assymptotically smooth** *in Γ with respect to y if constants c and γ can be found such that for all $x \in \Gamma$ and all multi-indexes $\alpha \in \mathbb{N}_0^d$*

$$\left|\partial_y^\alpha G(x, y)\right| \leq c\, p!\, \gamma^p \frac{|G(x, y)|}{\|x - y\|_2^p} \text{ for all } y \in \mathbb{R}^d \backslash \{x\},$$

where $p = |\alpha|$.

In order to construct degenerate approximations of the kernel G let us consider a Taylor's expansion where possible. Being the only singularity of G at $x = y$, let us consider the Taylor's expansion of G in $\Gamma_x \times \Gamma_y$, with $\Gamma_x, \Gamma_y \subset \Gamma$ and $dist(\Gamma_x, \Gamma_y) > 0$.

Let ξ_x and ξ_y be the Chebyshev centers of Γ_x and Γ_y, i.e., the centers of the balls with minimum radii ρ_x and ρ_y such that they contain Γ_x and Γ_y:

$$\Gamma_x \subseteq B(\xi_x, \rho_x) \text{ and } \Gamma_y \subseteq B(\xi_y, \rho_y).$$

If G is asymptotically smooth in Γ_x with respect to y with constants γ and c, let us consider the following Taylor's series expansion of G for $(x, y) \in \Gamma_x \times \Gamma_y$:

$$G(x,y) = \sum_{|\alpha| \geq 0} \frac{1}{\alpha!} \left(\partial_y^\alpha G(x, \xi_y) \right) (y - \xi_y)^\alpha. \tag{12}$$

Equation (12) shows that the kernel G is degenerate for separate domains, i.e., $dist(\Gamma_x, \Gamma_y) > 0$. Truncating the Taylor's series a degenerate approximation can be obtained:

$$G_p(x,y) = \sum_{|\alpha| < p} \frac{1}{\alpha!} \left(\partial_y^\alpha G(x, \xi_y) \right) (y - \xi_y)^\alpha.$$

The degree k of this degenerate approximation of the kernel is at most the dimension of the polynomials in d variables with degree at most $p - 1$, i.e., $k \leq p^d$.

Imposing a slightly stronger condition on the separation between Γ_x and Γ_y, i.e., $\eta\, dist(\xi_y, \Gamma_x) \geq \rho_y$ for $\eta \in (0, 1)$ instead of simply $dist(\Gamma_x, \Gamma_y) > 0$, an interesting bound for the quality of the degenerate approximation can be found.

Lemma 8 (Precision of the Degenerate Kernel Approximation Using Taylor's Series) *Let G be an asymptotically smooth kernel on Γ_x with respect to y in the sense of Definition 26 with constants c and γ. Let $\eta \in (0, 1)$ be chosen so that $\gamma\sqrt{d}\eta < 1$. If be assume that $\eta\, dist(\xi_y, \Gamma_x) \geq \rho_y$ then accuracy of the approximation using a truncated Taylor's series is bounded as*

$$|G(x,y) - G_p(x,y)| \leq c \frac{(\gamma\sqrt{d}\eta)^p}{1 - \gamma\sqrt{d}\eta} \sup_{x \in \Gamma_x} \{|G(x, \xi_y)|\}, \text{ for all } (x, y) \in \Gamma_x \times \Gamma_y.$$

Demonstration(Taken from Bebendorf. Page 122, [6])

$$\begin{aligned}
|G(x,y)-G_p(x,y)| &= \left|\sum_{|\alpha|\geq p}\tfrac{1}{\alpha!}\left(\partial_y^\alpha G(x,\xi_y)\right)(y-\xi_y)^\alpha\right| \\
&\leq \sum_{|\alpha|\geq p}\tfrac{1}{\alpha!}\left|\partial_y^\alpha G(x,\xi_y)\right||(y-\xi_y)^\alpha| \\
&\leq c\,|G(x,\xi_y)|\sum_{|\alpha|\geq p}\tfrac{\gamma^{|\alpha|}|\alpha|!}{\alpha!\|x-\xi_y\|_2^{|\alpha|}}|(y-\xi_y)^\alpha| \\
&\leq c\,|G(x,\xi_y)|\sum_{l=p}^{\infty}\left(\tfrac{\gamma}{\|x-\xi_y\|_2}\right)^l\sum_{|\alpha|=l}\begin{pmatrix} l \\ \alpha \end{pmatrix}|(y-\xi_y)^\alpha| \\
&\leq c\,|G(x,\xi_y)|\sum_{l=p}^{\infty}\left(\gamma\sqrt{d}\tfrac{\|x-\xi_y\|_2}{\|x-\xi_y\|_2}\right)^l \\
&\leq c\,|G(x,\xi_y)|\sum_{l=p}^{\infty}\left(\gamma\sqrt{d}\eta\right)^l \\
&\leq c\,|G(x,\xi_y)|\,\tfrac{(\gamma\sqrt{d}\eta)^p}{1-\gamma\sqrt{d}\eta} \\
&\leq c\tfrac{(\gamma\sqrt{d}\eta)^p}{1-\gamma\sqrt{d}\eta}\sup_{x\in\Gamma_x}\{|G(x,\xi_y)|\}\,.
\end{aligned}$$

■

The previous theorem allows us to estimate a bound for the degree of degeneracy required to guarantee a given accuracy $|G(x,y)-G_p(x,y)|\leq\varepsilon$ using Taylor's expansions.

Let $s=\sup_{x\in\Gamma_x}\{|G(x,\xi_y)|\}$, then,

$$\varepsilon = c\,s\,\frac{(\gamma\sqrt{d}\eta)^p}{1-\gamma\sqrt{d}\eta}\Rightarrow \log\varepsilon = p\log(\gamma\sqrt{d}\eta)+\log\left(\frac{c\,s}{1-\gamma\sqrt{d}\eta}\right)$$

$$k\leq p^d\Rightarrow k\leq\left(\frac{\log\left(\frac{\varepsilon(1-\gamma\sqrt{d}\eta)}{c\,s}\right)}{\log(\gamma\sqrt{d}\eta)}\right)^d = \left(\log_{\gamma\sqrt{d}\eta}\left(\frac{\varepsilon(1-\gamma\sqrt{d}\eta)}{c\,s}\right)\right)^d \tag{13}$$

Remark 6 (Geometrical Admissibility Condition and Taylor's Expansions) *If G is asymptotically smooth with respect to both variables then the following condition suffices to assure the domain separation hypothesis of Lemma 8:*

$$\min\{\rho_x,\rho_y\}\leq\eta\,dist(\Gamma_x,\Gamma_y).$$

This condition is used in the construction of block-cluster trees as seen in the previous section. Once an integral kernel can be proven to be asymptotically smooth with constant γ, $\eta\in(0,1)$ must be chosen with the constraint $\eta\sqrt{d}\gamma<1$.

Remark 7 (On the Optimality of the Bound for the Degeneracy) *The bound (13) is not optimal in general but it is independent from the algebraic structure of the kernel function; it relies only on its asymptotical smoothness and on the integration regions through η. Depending on the particular case better bounds can be can be obtained, such as, e.g., when using the multipole expansion in the case of a Coulomb-type kernel $G(x,y)=\|x-y\|^{-1}$.*

Remark 8 (Degeneracy Degree / Rank of the Approximation for a Given Accuracy) *A key observation is that Lemma 8 assures, as shown by equation (13), that for a given approximation error ε there exists a degenerate kernel and a bounded degeneracy degree $k \lesssim |\log \varepsilon|^d$ that allows compliance with the specified error.*

Let us now explore how the accuracy of the kernel approximant affects the accuracy of the matrices that arise in the context of the mentioned framework of the Galerkin discretization described in the first chapter.

Theorem 8 (Matrix Approximation Error Using Approximate Degenerate Kernels) *Let $A \in \mathbb{C}^{m\times n}$ be the matrix containing the pairwise interactions of the basis functions indexed by the index sets $\mathcal{I} = \{1, 2...m\}$ and $\mathcal{J} = \{1, 2...n\}$, as seen by the bilinear operator of the variational formulation used in a Galerkin discretization for a boundary integral equation, and let the kernel G of the integral equation be asymptotically smooth for both variables with constants c and γ. Let $t \times s \subseteq \mathcal{I} \times \mathcal{J}$ be a block-cluster for the index sets $\mathcal{I}$ and $\mathcal{J}$ such that $\min\{\rho_x, \rho_y\} \leq \eta\, dist(\Omega_t, \Omega_s)$ with a constant $\eta \in (0, 1)$ chosen so that $\gamma\sqrt{d}\eta < 1$ and let $\tilde{G}$ be a degenerate kernel for $(x, y) \in \Omega_t \times \Omega_s$. If it can be assured that $|G(x, y) - \tilde{G}(x, y)| \leq \varepsilon$, then the sub-matrix A_b associated to the block-cluster $b = t \times s$, and the matrix $\tilde{A}_b$ computed using $\tilde{G}$, comply with the following error estimates:*

$$|a_{b_{ij}} - \tilde{a}_{b_{ij}}| \leq \varepsilon \|\varphi_i\|_{L^1(\Gamma_h)} \|\varphi_j\|_{L^1(\Gamma_h)} \text{ for } i \in t \text{ and } j \in s,$$

$$\text{and } C > 0 \text{ such that } \|A_b - \tilde{A}_b\|_F \leq C\varepsilon,$$

$$\text{where } C = \sqrt{|t| \cdot |s|} \left(\max_{i\in\mathcal{I}\cup\mathcal{J}} \{\|\varphi_i\|_{L^1(\Gamma_h)}\} \right)^2 \text{ and } \varphi_i \text{ is basis function.}$$

DemonstrationTaken from Bebendorf, page 135 [6].

$$\begin{aligned}
|a_{b_{ij}} - \tilde{a}_{b_{ij}}| &= \left| \int\limits_{\Omega_t\times\Omega_s} \left(G(x, y) - \tilde{G}(x, y)\right) \varphi_i(x)\varphi_j(y) d\mu(x) d\mu(y) \right| \\
&\leq \int\limits_{\Omega_t\times\Omega_s} \left|G(x, y) - \tilde{G}(x, y)\right| |\varphi_i(x)| \, |\varphi_j(y)| d\mu(x) d\mu(y) \\
&\leq \varepsilon \|\varphi_i\|_{L^1(\Gamma_h)} \|\varphi_j\|_{L^1(\Gamma_h)}
\end{aligned}$$

The second estimates is evident from the definition of the Frobenius norm and from the first estimate. ■

3.4.3 Cross-Aproximation Methods

In the context of Galerkin discretizations of BIE, the bounds provided by the application of Lemma 8 and Theorem 8 assure us that given an accuracy ε there exists a degenerate kernel and a degree of degeneracy $k \lesssim |\log \varepsilon|^d$ such that kernel can be approximated with the specified accuracy. Given an admissible block-cluster b, as for an approximation $\tilde{Z}_b$ of Z_b such that $\|Z_b - \tilde{Z}_b\|_F \leq \varepsilon$, its existence can be assured if the integral kernel can be approximated with an accuracy of ε/C, where the constant C is the one from Theorem 8. If the block-cluster is admissible, and if the integral kernel is asymptotically smooth, then there exists a degenerate kernel approximation with degeneracy degree k with a bound, as shown in equation (13), of the type $k \lesssim |\log \varepsilon|^d$.

The idea behind cross-approximation methods is to perform a rank-revealing decomposition of the matrix Z_b, known to have a low-rank approximant, in order to construct consecutive approximations S_k growing in rank and diminishing the quantity $\|Z_b - S_k\|_F$ in each step. A cross-approximation method can be set to construct consecutive approximants up to a preset maximum rank k_{max} or until an accuracy ε is met, increasing the rank k adaptively up to a number $k_{max}(\varepsilon)$, case in which the cross-approximation method is called Adaptive Cross-Approximation (ACA) method.

Starting from a residue matrix $R_0 = Z_b$, the cross-approximation algorithms diminish the rank of R_0 to obtain a second residue matrix R_1 and so forth obtaining new residue matrices such that $rank(R_{k+1}) \leq rank(R_k)$. The difference between the original matrix and the residue is the approximant: $Z_b = S_k + R_k$. To diminish the rank in each step a column j_k and a row i_k are chosen for elimination:

$$R_{k+1} = R_k - \left((R_k)_{i_k,j_k}\right)^{-1} (R_k)_{1..m,\,j_k} (R_k)_{i_k,\,1..n}. \tag{14}$$

The matrices that are subtracted to R_k in all steps account for the difference $Z - R_k$ and are gathered in the approximant $S_k = Z - R_k$. The approximant matrix $S_{k_{max}}$ is then available as:

$$S_{k_{max}} = \sum_{k=0}^{k_{max}-1} \left((R_k)_{i_k,j_k}\right)^{-1} (R_k)_{1..m,\,j_k} (R_k)_{i_k,\,1..n}$$

As a sum of k_{max} unitary rank matrices, the matrix S_{kmax} is of rank at most k_{max}.

An important observation is that the construction of S_k can be performed without using all the elements of the matrices R_k of the previous steps. It is possible to store, for the matrices R_k, only the rows and columns that undergo change during the algorithm preserving a complexity that behaves asymptotically also as $\mathcal{O}(k^2(m+n))$ if $Z_b \in \mathbb{C}^{m \times n}$.

```
Let $k = 1$, $\Lambda = \emptyset$ and $\Lambda^c = \{1, 2, ..., m\}$;
while stop criterion not met do
    Choose a row $i_k \in \Lambda^c$;
    $\tilde{v}_k = (Z_b)_{i_k,1:n}$
    for $l = 1, ..., (k-1)$ do
        $\tilde{v}_k = \tilde{v}_k - (u_l)_{i_k} v_l$;
    end
    $\Lambda = \Lambda \cup \{i_k\}$ and $\Lambda^c = \Lambda^c \backslash \{i_k\}$;
    if $\tilde{v}_k$ does not vanish then
        Choose a column $j_k \in \{1, 2, ..., n\}$;
        $v_k = (\tilde{v}_k)_j^{-1} \tilde{v}_k$;
        $u_k = (Z_b)_{1:m,j_k}$;
        for $l = 0, ..., (k-1)$ do
            $u_k = u_k - (v_l)_{j_k} u_l$;
        end
        k=k+1;
    end
end
```

Algorithm 1: Structure of cross-approximation algorithms.

The Choice of the Pivots

It is easy to see that the procedure of cross-approximations algorithms produce residue matrices R_{k+1} eliminating selected rows and columns of R_k marked by the pivots. The choice of the the consecutive row pivots i_k and column pivots j_k is made so that the reduction in rank maximizes the reduction in the norm of the residue matrix $\|R_k\|_F = \|Z_b - S_k\|_F$. An ideal choice is the pivot pair (i_k, j_k) such that $R_{k_{i_k,j_k}}$ is the maximal entry in modulus. The inconvenient of this choice, known as full pivoting, is that it involves a $\mathcal{O}(mn)$ complexity. An heuristic solution is to choose a random fixed column j', choose in each step $i_k = \text{argmax}_i|R_{k_{i,j'}}|$ and then $j_k = \text{argmax}_j|R_{k_{i_k,j}}|$. This new method, known as partial pivoting, results in a $\mathcal{O}(m+n)$ complexity and requires that the a priori selected column is always updated.

Remark 9 (Partial Pivoting and Double Layer Potentials) *Certain boundary geometries can produce matrices that are not full when the kernel of a BIE is a double layer potential. In fact, in an cuboid box with adjacent faces formed by normal planes, a double layer potential can produce null blocks in the matrix associated to a block-cluster. If the column j' chosen a priori has null entries, the corresponding rows will not be chosen by the partial pivoting algorithm. To remedy this shortcoming of the partial pivoting algorithm it must be complemented with a second line of discernment for when no pivot rows i_k can be chosen selecting the maximum entry from the arbitrary row but the desired rank hasn't been met, in the case of CA algorithms, of the specified accuracy hasn't been achieved, in the case of ACA algorithms.*

The Stop Criterion

The algorithm can be stopped either when a maximum rank k_{max} has been attained or when the approximant S_k is close enough to the matrix Z: $\|Z_b - S_k\|_F / \|Z_b\|_F = \|R_k\|_F / \|Z_b\| \leq \varepsilon$ for a specified tolerance ε resulting in $k_{max}(\varepsilon)$ steps. While equation (13) shows the existence of a k-rank approximant of Z_b for a prescribed accuracy ε, this error could be possibly met with matrix whose rank was lower than k. In fact, it was mentioned that depending on the problem, bounds better than (13) could be obtained. It could then be desirable to test the relative error $\|R_k\|_F / \|Z_b\|_F$ as $rank(R_k)$ decreases, using an ACA method in hopes of reducing the number of required steps and thus the rank of the approximation.

Unfortunately, computing $\|R_k\|_F$ and $\|Z_b\|_F$ defeats the purpose of the cross-approximation methods in the context of a fast method for the boundary integral equation such as the hierarchical matrix method; on the one hand because it would require the knowledge of the matrix elements of Z_b and because it would imply a complexity of order $\mathcal{O}(mn)$. Instead of computing the matrix norm of Z_b and R_k an heuristically approach is commonly used founded in the following assumption: the cross-approximation algorithm monotonically diminishes the quantity $\|Z_b - S_k\|_F = \|R_k\|_F$. In fact it is easy to see that after $\min\{m, n\}$ steps the residue matrix will have zero norm. This assumption, although not true in general, is a common element in the current ACA methods, as the following lemmas show.

Lemma 9 (ACA Relative Error) *Let $Z_b \in \mathbb{C}^{m\times n}$ be a matrix to be decomposed as $Z_b = S_k + R_k$ using a cross-approximation algorithm of the kind of Algorithm 1. Let us suppose that there exists a number $\delta \in (0,1)$ such that for every k we have that $\|R_{k+1}\|_F \leq \delta\|R_{k+1}\|_F$. Then, the following condition is enough to assure that $\|Z_b - S_k\|_F / \|Z_b\|_F \leq \varepsilon$:*

$$\frac{1}{|(R_k)_{i_k,j_k}|}\|(R_k)_{1..m,j_k}\|_2 \, \|(R_k)_{i_k,1..n}\|_2 \leq (1-\delta)\frac{\varepsilon}{1+\varepsilon}\|S_k\|_F. \tag{15}$$

Demonstration In a cross-approximation algorithm, once we have chosen the pivots i_k and j_k for the step k, we know that the next residue matrix R_{k+1} can be computed as stated in equation (14), which implies that

$$\|R_k\|_F \leq \|R_{k+1}\|_F + \frac{1}{|(R_k)_{i_k,j_k}|}\|(R_k)_{1..m,j_k}(R_k)_{i_k,1..n}\|_F.$$

Assuming that there exists a number $\delta \in (0,1)$ such that for every k we have that $\|R_{k+1}\|_F \leq \delta\|R_{k+1}\|_F$, and considering that $\|(R_k)_{1..m,j_k}(R_k)_{i_k,1..n}\|_F \leq \|(R_k)_{1..m,j_k}\|_2 \, \|(R_k)_{i_k,1..n}\|_2$, the previous statement implies that

$$\|R_k\|_F \le \frac{1}{1-\delta}\|(R_k)_{1..m,\,j_k}\|_2\ \|(R_k)_{i_k,\,1..n}\|_2.$$

If equation (15) holds, then

$$|\ R_k\|_F \le \frac{\varepsilon}{1+\varepsilon}\|S_k\|_F.$$

Together with the fact that $\|S_k\|_F \le \|Z_b\|_F + \|R_k\|_F$ this yields that

$$\|R_k\|_F \le \varepsilon\|Z_b\|_F,$$

that is the desired results. ■

Remark 10 (Usefulness and Limitations of the Lemma 9) *Condition (15) can be computed with complexity in the order of $\mathcal{O}(k^2(m+n))$ (being S_k in $\mathbb{C}_k^{m\times n}$) and doesn't need previous knowledge of all the entries of neither R_k nor Z_b. However, the hypothesis of Lemma 9, i.e., that there exists a number $\delta \in (0,1)$ such that for every k we have that $\|R_{k+1}\|_F \le \delta\|R_{k+1}\|_F$, is a strong assumption and it cannot be proved in general. However, in practice, it is a sufficiently good condition, specially when applied with additional heuristic conditions to ignore the first realizations of the inequality. For a further discussion on the usefulness and the limitations of the application of the results of the lemma, the reader is referred to [25].*

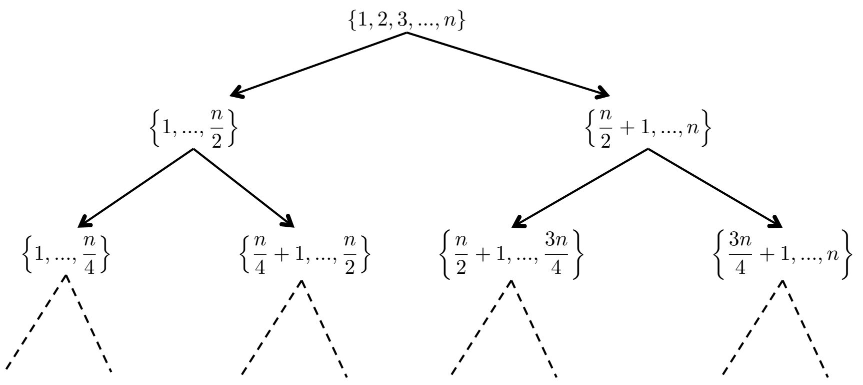
While the hypotheses of Lemma 9 can be proven for some case, it is a non-trivial task. Often, an heuristic alternative is chosen: to check to convergence of the increment of the sequence. Since we known that $\|R_k\|_F$ eventually vanishes to zero, an alternative stop criterion is to check the convergence of the increment of the sequence considering that the difference between R_{k+1} and R_k is a unitary rank matrix:

$$\|R_{k+1} - R_k\|_F = \|\left((R_k)_{i_k,j_k}\right)^{-1}(R_k)_{1..m,\,j_k}(R_k)_{i_k,\,1..n}\|_F \le \frac{1}{|(R_k)_{i_k,j_k}|}\|(R_k)_{1..m,\,j_k}\|_2\|(R_k)_{i_k,\,1..n}\|_2.$$

The alternative stop criterion for a specified relative tolerance ε would then be:

$$\text{Stop when } \frac{|(R_0)_{i_0,j_0}|}{|(R_k)_{i_k,j_k}|}\frac{\|(R_k)_{1..m,\,j_k}\|_2\|(R_k)_{i_k,\,1..n}\|_2}{\|(R_0)_{1..m,\,j_0}\|_2\|(R_0)_{i_0,\,1..n}\|_2} \le \varepsilon. \tag{16}$$

This heuristic criterion is widely used (prominent examples can be seen in [33], and a wide collection of numerical results in [10], Chapter 4) even though it can be proven to fail in assuring a maximum error of ε in the matrix approximant (an example of this can be seen in [25]). It is common to use this criterion with additional heuristic strategies.



$\{1,2,3,...,n\}$
$\left\{1,...,\frac{n}{2}\right\}$
$\left\{\frac{n}{2}+1,...,n\right\}$
$\left\{1,...,\frac{n}{4}\right\}$
$\left\{\frac{n}{4}+1,...,\frac{n}{2}\right\}$
$\left\{\frac{n}{2}+1,...,\frac{3n}{4}\right\}$
$\left\{\frac{3n}{4}+1,...,n\right\}$

Figure 3: First three levels of the cluster tree

4 Example Computations

4.1 An Elemental Case

This section exemplifies some of the concepts developed in the previous chapters through a simple example. Let us consider the following integral equation for a known function $f : [0,1] \to \mathbb{R}$:

$$\int_0^1 \log(|x-y|)\, u(y)dy = f(x)$$

A standard discretization scheme is Galerkin's method where we solve the integral equation projected onto the n-dimensional space $V_n = span\{\varphi_1, \varphi_2, ...\varphi_n\}$:

$$\sum_{j=1}^{n} \int_0^1 \int_0^1 \log(|x-y|)\, u_j \varphi_j(x)\varphi_i(y)dxdy = \int_0^1 f(x)\varphi_i(x)dx.$$

Let us split the $[0,1]$ interval in $n = 2^p$ intervals for p sufficiently large and let us considerate a P0 Lagrange basis to span the space V_n. The P0 basis function are

$$\varphi_i(x) = \begin{cases} 1 & \text{if } (i-1)/n < x < i/n \\ 0 & \text{otherwise} \end{cases}$$

Using these basis functions the elements of the system matrix can be computed as

$$A_{ij} = \int_{\frac{i-1}{n}}^{\frac{i}{n}} \int_{\frac{j-1}{n}}^{\frac{j}{n}} \log(|x-y|)\, dxdy.$$

The index set for the basis of V_n is $\mathcal{I} = \{1,2,...,n\}$. Selecting a minimal size n_{min} for a cluster we can construct a cluster-tree $T_{\mathcal{I}}$ dividing in two each cluster/node of the tree of root $\mathcal{I}$, provided that p is large enough. Figure 3 shows the first 3 levels (level 0, 1 and 2) of the cluster tree.

Following Definition 18 the block-cluster tree can be constructed as it is shown in Figure 4 for the first 3 levels (level 0, 1 and 2).

At level 2 of the block-cluster tree the first block-clusters susceptible of respecting a condition of geometrical admissibility appear. Let us focus in block-cluster $b = \{1,...,n/4\} \times$

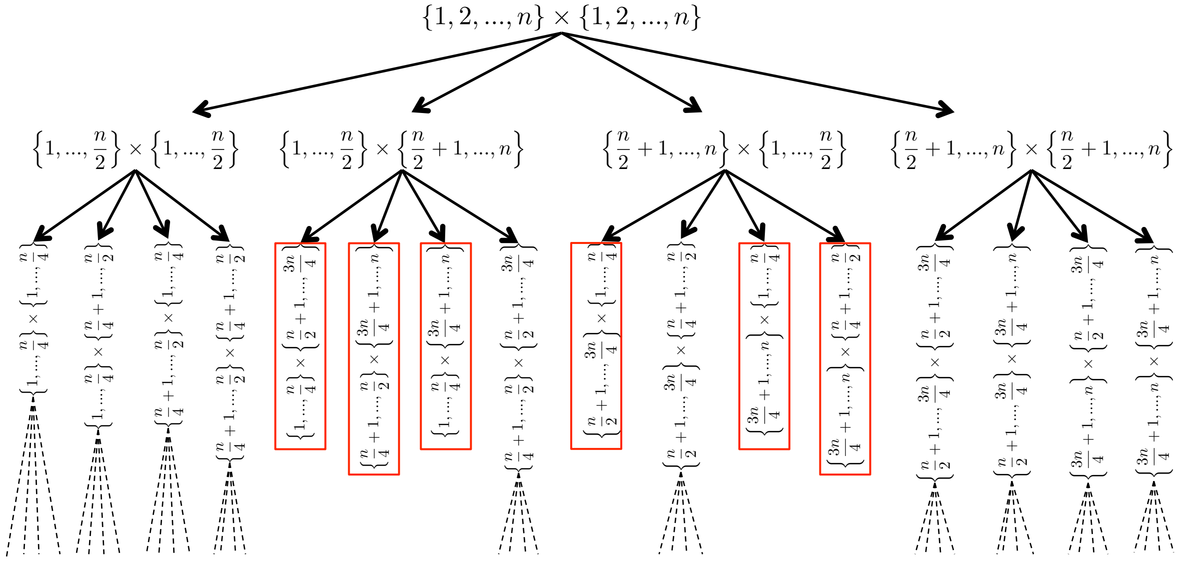

$\{1, 2, ..., n\} \times \{1, 2, ..., n\}$
$\left\{1, ..., \frac{n}{2}\right\} \times \left\{1, ..., \frac{n}{2}\right\}$
$\left\{1, ..., \frac{n}{2}\right\} \times \left\{\frac{n}{2}+1, ..., n\right\}$
$\left\{\frac{n}{2}+1, ..., n\right\} \times \left\{1, ..., \frac{n}{2}\right\}$
$\left\{\frac{n}{2}+1, ..., n\right\} \times \left\{\frac{n}{2}+1, ..., n\right\}$

Figure 4: First three levels of the block-cluster tree

$\{3n/4 + 1, ..., n\}$. This block-cluster can comply with geometrical admissibility conditions for $\eta \geq 1/2$, being among the first apparitions of sub-matrices selected to be approximated by a low-rank matrix.

The entries of the sub-matrix corresponding to the block-cluster $b = \{1, ..., n/4\} \times \{3n/4 + 1, ..., n\}$ can be computed as also its singular value structure. Let us consider the particular case of $p = 9$ and $n = 2^9 = 512$. Figure 6 shows the singular value structure of the sub-matrix $A_b \in \mathbb{C}^{128\times128}$ corresponding to the block-cluster $b = \{1, ..., 128\} \times \{385, ..., 512\}$.

As revealed by the singular value structure, the sub-matrix A_b is indeed suitable to be approximated by a low-rank matrix. Theorem 6 assures the existence of an approximation of consecutively growing rank that drastically lowers the difference between the matrix A_b and its low-rank approximate for the first 6 iterations. The procedure of Theorem 6 is, however, too expensive; it is of order $\mathcal{O}(n^3)$. One of the described cross-approximation methods must be used.

For any geometrically admissible block-cluster $b = t \times s$ the asymptotical smoothness can be checked since all geometrically admissible block-cluster comply with $dist(\Omega_t, \Omega_s) > 0$. It can be easily checked that the kernel complies with the definition of asymptotical smoothness:

$$\left|\frac{\partial^n}{\partial x^n} \log|x-y|\right| = \left|\frac{\partial^n}{\partial y^n} \log|x-y|\right| = \frac{(n-1)!}{|x-y|^n} \leq \frac{n!}{|x-y|^n} \leq c\gamma^n \frac{n!\,|\log|x-y||}{|x-y|^n},$$

with $\gamma = 1$ and $c = |\log(dist(\Omega_t, \Omega_s))|^{-1}$. This means that, for our chosen block-cluster $b = \{1, ..., 128\} \times \{385, ..., 512\}$, the choice of η is to be made such that $1/2 \leq \eta < 1$, for

$$\eta \in [1/2, 1) \Rightarrow b = \{1, ..., 128\} \times \{385, ..., 512\} \text{ is geometrically admissible.}$$

Being the the block-cluster b geometrically admissible, and being the kernel asymptotically smooth, it is assured that a lesser-rank approximation exists. In the following, a partial pivot CA and a full pivot cross-approximation are shown in Figure 7, illustrating the performance of the CA methods described in the previous chapter.

Finally, the results of a faster method based on a random choice of row pivots but an absolute-value-maximizing choice of column pivots, a semi-random pivoting CA, are shown in Figure 8.

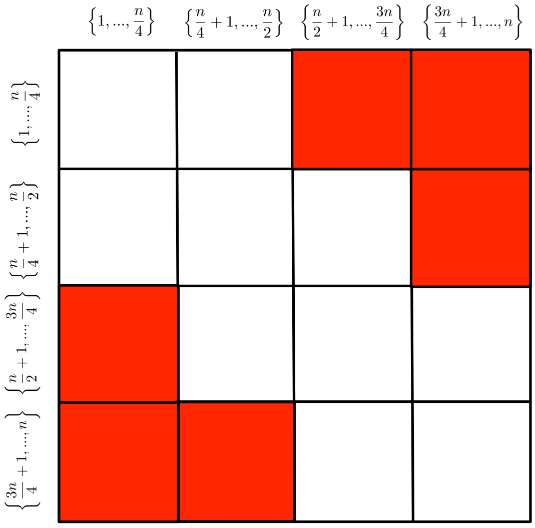

$\left\{1,...,\frac{n}{4}\right\}$
$\left\{\frac{n}{4}+1,...,\frac{n}{2}\right\}$
$\left\{\frac{n}{2}+1,...,\frac{3n}{4}\right\}$
$\left\{\frac{3n}{4}+1,...,n\right\}$
$\left\{1,...,\frac{n}{4}\right\}$
$\left\{\frac{n}{4}+1,...,\frac{n}{2}\right\}$
$\left\{\frac{n}{2}+1,...,\frac{3n}{4}\right\}$
$\left\{\frac{3n}{4}+1,...,n\right\}$

Figure 5: Sub-matrices associated to the partition made of the block-cluster in $T^{(2)}_{\mathcal{I}\times\mathcal{I}}$.

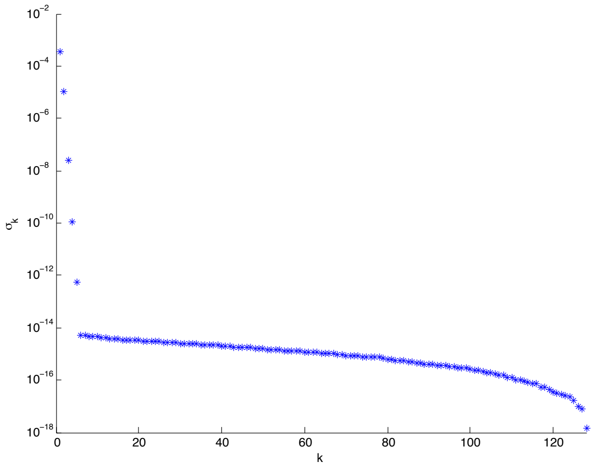


Figure 6: Singular value structure of the matrix.

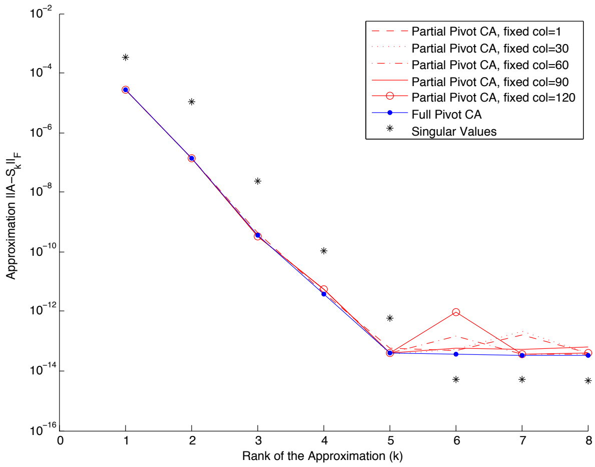


Figure 7: Comparison of the singular value structure with the full pivot CA and partial pivot CA using several fixed columns across the matrix.

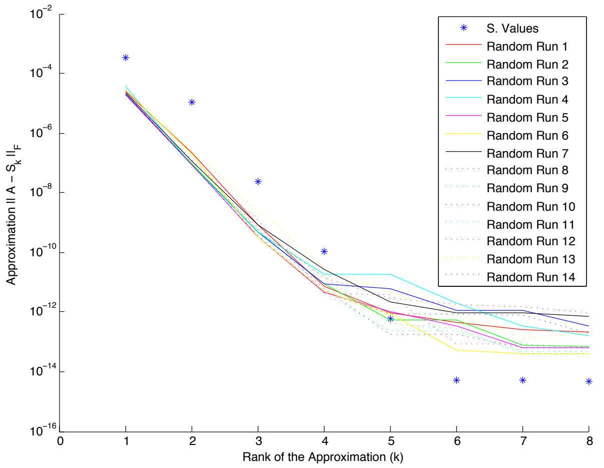


Figure 8: Comparison of the singular value structure with several realizations of semi-random pivoting CA.

4.2 Cross-Approximation for the BEM for the Electric Field Integral Equation

4.2.1 BEM for the Perfect Electric Conductor

In the following we consider the boundary integral equation for the scattering of an electromagnetic wave by a perfect electric conductor. Let Ω_{int} be a perfect electric conductor object (PEC) of boundary Γ immersed in open space Ω_{ext}. Let $\hat{n}$ be the unit vector normal to Γ pointing towards Ω_{ext}. In the absence of source electric or magnetic charges or currents, a time-harmonic electromagnetic field (E, H) in an isotropic conducting medium is governed by the Maxwell's equations

$$\begin{cases} i\omega\varepsilon E + curl H = 0, \\ \\ -i\omega\mu H + curl E = 0. \end{cases} \tag{17}$$

In equation (17) $\omega = 2\pi f$ is the pulsation at a given frequency f, ε and μ are the complex electrical permittivity and the magnetic permeability, and σ is the electric conductivity of the medium. The magnetic permeability $\mu = \mu_r\mu_0$ depends on the relative magnetic permeability $\mu_r \geq 1$ of the medium and that of the vacuum μ_0. The complex electrical permittivity $\varepsilon = \varepsilon_r\varepsilon_0 + i\sigma/\omega$ depends on the relative electrical permittivity $\varepsilon_r \geq 1$, on the electrical permittivity of the vacuum ε_0 and in the conductivity $\sigma \geq 0$ of the medium at the given pulsation ω (at time-harmonic regimes the conductivity accounts for the dielectric losses and for those associated to the induced electrical currents). In equation (17) it is implicit the time convention $E(x,t) = \Re\left(E(x)e^{-i\omega t}\right)$ and $H(x,t) = \Re\left(H(x)e^{-i\omega t}\right)$. Inside a perfect electric conductor (PEC) medium the field E and H are zero as the result of the limits process of taking the conductivity σ towards infinity.

The tangential components of the electric and magnetic fields remain continuous across a surface Γ of discontinuity that separates two regions where ε or μ are continuous. The boundary conditions for the electric and magnetic fields can be stated, for a boundary Γ that follows the discontinuity of ε or μ, as

$$[E \times \hat{n}]_\Gamma = 0 \text{ and } [H \times \hat{n}]_\Gamma = 0, \tag{18}$$

where $[\,]_\Gamma$ is the jump across Γ and $\hat{n}$ is the unitary normal to Γ.

Let us consider the problem of computing the electromagnetic field (E, H) scattered by a PEC object Ω_{int} immersed in an infinity vacuum Ω_{ext}, and illuminated by an incident electromagnetic plane wave (E_{inc}, H_{inc}).

Being a PEC ($E = 0$ inside Ω_{int}), and as a conclusion from equation (18), the trace of the total electric field $E+E_{inc}$ complies with $((E + E_{inc}) \times \hat{n})\,|_{ext} = 0$. If the incident electromagnetic wave (E_{inc}, H_{inc}) is a plane wave that satisfies the Maxwell's equations (17) then the satisfaction of those equations by the total field yields the partial differential equation problem for the scattered electromagnetic field:

$$\begin{cases} i\omega\varepsilon E + curl H = 0 & \text{in } \Omega_{ext}, \\ \\ -i\omega\mu H + curl E = 0 & \text{in } \Omega_{ext}, \\ \\ (E \times \hat{n})|_{ext} = -E_{inc} \times \hat{n} & \text{in } \Gamma. \end{cases} \tag{19}$$

Let us extend the values of the electromagnetic field (E, H) into Ω_{int} by those solution to an associated problem:

$$\begin{cases} i\omega\varepsilon E + curl H = 0 & \text{in } \Omega_{int}, \\ -i\omega\mu H + curl E = 0 & \text{in } \Omega_{int}, \\ (E \times \hat{n}|_{inc} = -E_{inc} \times \hat{n} & \text{in } \Gamma. \end{cases} \tag{20}$$

Let us also assume that the scattered electromagnetic field respects the Silver-Müller radiation condition, i.e., there exist values $C > 0$ and $R > 0$ such that

$$\left(\sqrt{\varepsilon}E - \sqrt{\mu}H \times \frac{x}{\|x\|_2}\right) \leq \frac{C}{\|x\|_2^2}, \ \forall \|x\|_2 > R.$$

If we denote the jump of the tangential fields in Γ as

$$j = (H \times \hat{n})|_{int} - (H \times \hat{n})|_{ext} \ \text{ and } \ m = (E \times \hat{n})|_{int} - (E \times \hat{n})|_{ext},$$

then it can be proved (Theorem 5.5.1, page 234 [26]), that the scattered electromagnetic field (E, H) can be written, for $x \notin \Gamma$ as

$$\begin{aligned} E(x) &= i\omega\mu \int_\Gamma G(x,y) j(y) d\Gamma(y) + \tfrac{i}{\omega\varepsilon}\nabla \int_\Gamma G(x,y) div_\Gamma j(y) d\Gamma(y) + curl \int_\Gamma G(x,y) m(y) d\Gamma(y), \\ H(x) &= -i\omega\varepsilon \int_\Gamma G(x,y) m(y) d\Gamma(y) - \tfrac{i}{\omega\mu}\nabla \int_\Gamma G(x,y) div_\Gamma m(y) d\Gamma(y) + curl \int_\Gamma G(x,y) j(y) d\Gamma(y), \end{aligned} \tag{21}$$

where $G(x,y)$ is the Green's function for the Helmholtz's equation. It can also be proved that, for $x \in \Gamma$,

$$\begin{aligned} (E \times \hat{n})|_{ext}(x) = \quad & -\tfrac{m(x)}{2} + \int_\Gamma \left(\tfrac{G}{\partial \hat{n}_x}(x,y) m(y) - \nabla_x G(x,y)(m(y) \cdot (\hat{n}_x - \hat{n}_y))\right) d\Gamma(y) \\ & + i\omega\mu \int_\Gamma G(x,y) j(y) \times \hat{n}_x d\Gamma(y) \\ & + \tfrac{i}{\omega\varepsilon} \int_\Gamma \left((\nabla_x G(x,y) \times (\hat{n}_x - \hat{n}_y))\, div_\Gamma j(y) + G(x,y) \overrightarrow{curl}_\Gamma div_\Gamma j(y)\right) d\Gamma(y). \end{aligned} \tag{22}$$

Considering that $m = 0$, the scattered field can be computed using equation (21) from j, which can be determined solving the integral equation arising from equation (22):

$$\begin{aligned} -(E_{inc} \times \hat{n})(x) = \quad & + i\omega\mu \int_\Gamma G(x,y) j(y) \times \hat{n}_x d\Gamma(y) \\ & + \tfrac{i}{\omega\varepsilon} \int_\Gamma \left((\nabla_x G(x,y) \times (\hat{n}_x - \hat{n}_y))\, div_\Gamma j(y) + G(x,y) \overrightarrow{curl}_\Gamma div_\Gamma j(y)\right) d\Gamma(y). \end{aligned} \tag{23}$$

It can be proven (Theorem 5.6.2, page 247 [26]), provided that $\mu\varepsilon\omega^2$ is not an eigenvalue of the associated problem, that the integral equation (23) admits the following variational formulation:

$$\begin{cases} \text{For } E_{inc} \times \hat{n} \in H^{-1/2}_{curl}(\Gamma), \text{ find } j \in H^{-1/2}_{div}(\Gamma) \text{ such that } \forall j^t \in H^{-1/2}_{div}(\Gamma), \\ -\int\limits_{\Gamma} (E_{inc}(x) \cdot j^t(x)) d\Gamma(x) = \quad -\frac{i}{\omega\varepsilon} \int\limits_{\Gamma\times\Gamma} G(x,y) div_{\Gamma} j(y) div_{\Gamma} j^t(x) d\Gamma(y) d\Gamma(x) \\ \qquad\qquad\qquad\qquad\qquad +i\omega\mu \int\limits_{\Gamma\times\Gamma} G(x,y)(j(y) \cdot j^t(x)) d\Gamma(y) d\Gamma(x), \end{cases}$$

where

$$H^{-1/2}_{curl}(\Gamma) = \left\{ E \in H^{-1/2}(\Gamma), rot_{\Gamma} E \in H^{-1/2}(\Gamma) \right\}$$

and

$$H^{-1/2}_{div}(\Gamma) = \left\{ E \in H^{-1/2}(\Gamma), div_{\Gamma} E \in H^{-1/2}(\Gamma) \right\}.$$

To discretize the problem we consider a piecewise triangular mesh Γ_h approximation of Γ, where h indexes the longest edge of the mesh. We take as basis functions the Rao-Wilton-Glisson (RWG) functions associated to each edge of the triangular mesh. For each edge we arbitrarily define a direction of flow, flowing from one side, triangle T^+, to the other side of the edge, triangle T^-. Let S^+ be the vertex of triangle T^+ that it's not over the common edge, and similarly let S^- be the vertex of triangle T^- that it's not over the common edge. The RWG basis function associated to an edge n, of length l_n, of the triangular mesh Γ_h is defined as

$$j_n(x) = \begin{cases} \frac{l_n}{2\,area(T^+_n)}(x - S^+_n) & \text{if } x \in T^+_n, \\ -\frac{l_n}{2\,area(T^-_n)}(x - S^-_n) & \text{if } x \in T^-_n, \\ 0 & \text{otherwise.} \end{cases}$$

The variational formulation can be then discretized expressing the solution as

$$j(x) = \sum_{n=1}^{N_h} \alpha_n j_n(x), \text{ for } x \in \Gamma_h,$$

where N_h is the number of edges of the triangular mesh Γ_h. The discretized formulation is put into a system of linear equations:

$$ZI = V,$$

where

$$I = (\alpha_1, \alpha_2, ..., \alpha_{N_h})^T,$$

$$V_i = -\int\limits_{\Gamma_h} (E_{inc}(x) \cdot j_i(x)) d\Gamma_h(x),$$

and

$$Z_{ij} = \quad -\frac{i}{\omega\varepsilon} \int\limits_{\Gamma_h\times\Gamma_h} G(x,y) div_{\Gamma} j_j(y) div_{\Gamma} j_i(x) d\Gamma_h(y) d\Gamma_h(x)$$

$$+i\omega\mu \int\limits_{\Gamma_h\times\Gamma_h} G(x,y)(j_j(y) \cdot j_i(x)) d\Gamma_h(y) d\Gamma_h(x).$$

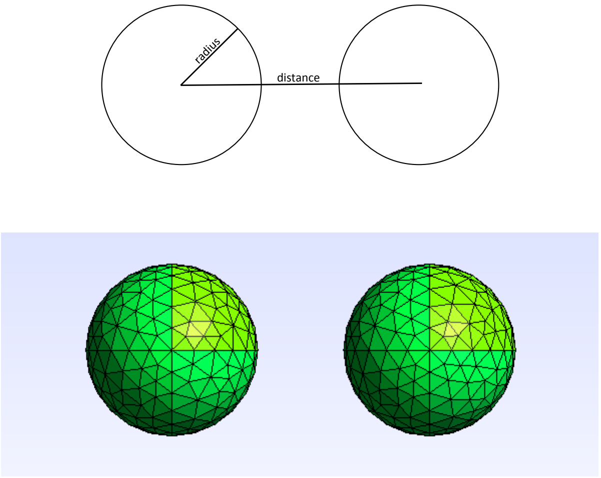


Figure 9: Diagram (top) and mesh (bottom) of the case study, consisting in two unit balls separated by a variable distance.

4.2.2 Cross-Approximation for Separated Objects

Using the hierarchical matrix approach discussed in the previous chapter the integration domain could be divided into regions where cross-approximation techniques could be then applied. In this section, we will consider a slightly simpler case, that of separated scatterer objects. Let us assume that we have two separated scatterer objects and that their surfaces have been discretized in triangular meshes. Let us assume that the mesh of the first object has N_1 edges and that the mesh of the second object has N_2 edges. The system matrix of the associated BEM could be then represented as

$$Z = \begin{pmatrix} Z_{11} & Z_{12} \\ \\ Z_{21} & Z_{22} \end{pmatrix},$$

where $Z_{11} \in \mathbb{C}^{N_1 \times N_1}$, $Z_{21} \in \mathbb{C}^{N_1 \times N_2}$, $Z_{21} \in \mathbb{C}^{N_2 \times N_1}$ and $Z_2 \in \mathbb{C}^{N_2 \times N_2}$. Since the matrix Z is symmetric, we also know that $Z_{11} = Z_{11}^T$, $Z_{22} = Z_{22}^T$ and $Z_{12} = Z_{21}^T$. When introducing a second object in a scattering simulation, the added complexity is more than that associated to the matrix of the second object, i.e., Z_{22}, since it also incorporates the reactions between the basis functions on each object, i.e., matrices Z_{12} and Z_{21}.

Let us consider two PEC balls of unitary radius, centers separated by a variable distance and floating in the vacuum illuminated with a 500MHz incident plane wave. Let each ball have 710 edges ($N_1 = N_2 = 710$) of average length of $10cm$. Figure 9 shows the considered case.

The matrix Z_{12} (and thus Z_{21}) is a good candidate to be approximated by cross-approximation, allowing for a complexity close the sum of the complexities associated to each separated ball. Figure 10 shows the evolution of a low-rank approximation computed with a full pivoting cross-approximation algorithm for different separations between the metallic balls. It can be seen in the figure how larger distances quickly allow for improved approximations in the sense of achieving a smaller approximation error with lesser ranks.

Let suppose that for a given application, it is enough to approximate Z_{12}, in this particular case, with an absolute error of at most 10^{-2}. This means that depending on the distance (between the ones chosen for our example) the rank of the approximant could be between 2 and 25. This means that the number of elements to store could range from $710(710+1)+2(710+710) = 507650$ to $710(710+1)+25(710+710) = 540310$. Considering that no approximation at all yields a number of elements to store equal to $1420(1420+1)/2 = 1008910$, the cross-

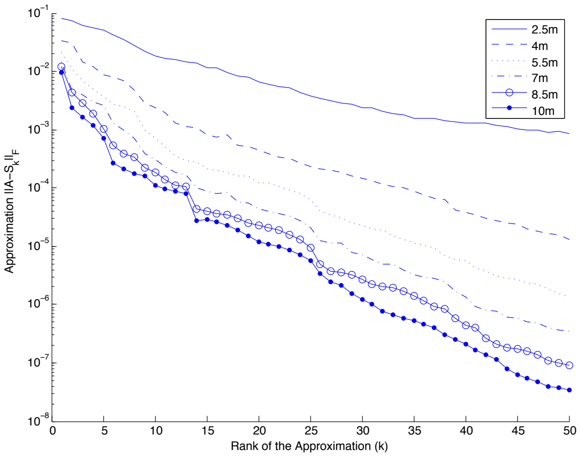


Figure 10: Error evolution of a cross-approximation of Z_{12} computed with a full pivoting cross-approximation algorithm for different separations between the unitary radii metallic balls.

approximation approach (even in this case where no hierarchical structures were used) means a cut in memory requirement between 46% and 49%. Even for the simple approach of using cross-approximation only for the sub-matrices of the interactions between the different objects in a scattering problem (without a hierarchical approach) implies a significant advantage. Applications to this simple case are also of interest, e.g., as in the case of a multi-static scattering by an object surrounded by multiple emitting/receiving antennas (air-tracking with radar networks, microwave medical imaging, security scanning with synthetic antennas, etc) or in the case of multiple inclusions (underground imaging of land mines in soils with other objects, photonic crystals, etc). To further improve the complexity below that of the sum of the complexity of the scattering problems associated to each separate object, the geometric division and hierarchical matrix approach (or other such as FMM, Panel Clustering, etc) can be used.

5 Historical Review

This last section offers a short description of the historical development of the methods and techniques described in this report, which are divided in four main parts: the apparition of the fast methods, the formalization of the hierarchical structures, the hierarchical matrices and the development of the cross-approximation techniques.

5.1 The Acceleration of Pairwise Interactions and the Fast Multipole Method

The ideas behind the acceleration of the resolution of the discretized integral equation are firstly related to the pairwise nature of the operation of a bilinear operator over a finite-dimensional basis spanning a discrete sub-space considered for a Galerkin discretization. In this sense the first advances in the acceleration and compression (in terms of memory requirements) can be traced back to the landmark Rokhlin's article of 1985: *Rapid solution of integral equations of classical potential theory* [29]. In this work a method to compute the mutual interactions of N elements with less than $\mathcal{O}(N^2)$ complexity is proposed. This methods introduces the concept of local interactions of a given element, from which other key concepts spring such as degenerate integral kernel approximations. Using local interactions and multipolar expansions based on the Gegenbauer's Addition Theorem the overall number of computations is effectively reduced. This idea proved to be fruitful for the computation for particle-like elements, first developed by Greengard and Rokhlin in 1987 [15], and it soon showed its advantages in the pairwise interactions used in collocation, Galerkin and Nystrom methods for the integral equations. The earliest developments in the field can be traced back to Greengard in 1988 [16] and many applications have been developed ever since for the boundary integral equation.

The application of this method based on a multipolar expansion of the kernel of a boundary integral operator is known as the Fast Multipole Method (FMM), and it achieves less-than-quadratic computational complexity provided that a geometrical partition of the integration domain has been done to exploit local interactions and provided that the kernel can be approximated by a multipolar expansion.

A comprehensive review on the history and application of the FMM can be consulted in [27].

5.2 The Panel Clustering Method and the Development of the Hierarchical Structures

A different type of method for the acceleration and compression of the boundary integral equations methods appeared shortly after the birth of the FMM. It was proposed in 1986 by Hackbusch and Nowak and it is known as the panel clustering method [20]. It also relies in the approximation of the kernel of the integral equations exploiting the computations for the local interactions by based on polynomial interpolation rather than in multipole expansion. The kernel of the integral operator is approximated by tensorized Lagrange polynomials over grids of Chebyshev points in axiparallel boxes dividing the integration domain. Efficient uses for the boundary integral equations were reported as early as 1989 also by Hackbusch and Nowak for the collocation method [21], and its use with the Galerkin method was developed in Sauter's doctoral thesis under Hackbusch supervision in 1992 [30] [22]. The use of axiparallel boxes developed into the formalization of a hierarchical geometrical subdivision of the integration domain and structures for its representation and use in acceleration and compression of the matrices arising in the discretization of integral equations. The goal for these structures and their formalizations was to provide efficient means to compute matrix-vector multiplications (used in iterative solvers) exploiting degenerate expansions of the kernel (in this case polynomial). Precedent of this direction of development towards hierarchical structures is found in 1986 given by Barnes and Hut in the computation of forces for a group of elements scattered in different positions [2]. A use directly related to integration was used in 1990 by Brandt and Lubrechet in the computation of integral transforms [8] much in the spirit of the panel clustering method but with emphasis on the structure of the matrix and the geometrical division. Some final developments before the introduction of what is now known as hierarchical matrices was done by Brandt and Venner in 1998 specifically for the boundary integral equations [9].

5.3 The Hierarchical Matrix

The hierarchical matrix is a formalization of the sub-matrix structure of a matrix and it is linked to the geometry of the problem from where it arises. The construction of the sub-matrix structure is determined by criteria related to the exploitable features arising from geometrical characteristics. In the case of the discretization of boundary integral equations these criteria relate to distance between integration sub-domains and their sizes, allowing for the approximation using degenerate kernels and thus resulting in low-rank matrices approximations for some of the sub-matrices of the hierarchical structure. The formalization of this structure allows for the development of an algebraic structure for the set of sub-matrices linked to a particular discretization problem, thus allowing matrix operations other than just the matrix-vector multiplication as it is the case for the FMM and Panel Clustering methods. A formalization of matrix summation, matrix-matrix multiplication and computation of matrix norms for hierarchical matrices provides means to carry out more complex matrix operations such as matrix factorizations and inversions. These approaches have been successfully used to compute matrix pre-conditioners with less than quadratic complexity. Another interesting but yet fairly unexplored field arises from the hierarchical matrix formalization, seeking to fit hierarchical rank structures in a matrix without a priori information such as the one provided by a related geometrical domain in the case of integral equations. A proper fit for a given matrix within a type of hierarchical matrix structure could then be used to perform operations in less than quadratic complexity. The first formalization of hierarchical matrices, denoted as $\mathcal{H}$-matrices, was given by Hackbusch in two articles in 1999 [17], setting the formalism for the structure and its algebraic framework, and in 2000 [18] providing applications to problems relating to multidimensional geometries and their uses in boundary integral methods. A wider generalization, called $\mathcal{H}^2$-matrices, was also proposed in 2000 by Hackbusch, Khoromskij and Sauter [19] capable of accounting for the exploitation made by FMM and Panel Clustering methods of the multi-level information (geometrical sub-matrix nesting) besides the geometrical features of the used subdomains. This feature, known as upward and downward pass in the FMM and Panel Clustering algorithms is associated to a recompression of the sub-matrices of a $\mathcal{H}^2$ using information on the nesting of the structure. The first application of this new variant to boundary integral equations was provided by Hackbusch and Börm in 2004 [7]. The first pre-conditioners computed from $\mathcal{H}$-matrices for the BEM where shown by Bebendorf in 2005 [5].

5.4 Cross-Approximation Methods

A novel method, called skeleton-approximation, related to the degenerate kernel approximation but different in nature, and contemporary to the development of hierarchical matrices, appeared in 1996 suggested by Tyrtyshnikov [32] and was formalized by Goreinov, Tyrtyshnikov and Zamarashkin in 1997 [13]. This method can be seen as a reduced model problem for the space of matrices spanned by a given number of unitary rank matrices; as such its origins can be traced to the apparition of the reduced model theory in the mechanical community in the late 1970s [1] [28].The method, later re-baptized as cross-approximation (CA), produced a low-rank approximation of a matrix using only the matrix entries without prior knowledge of the kernel function other than the fact that it accepted a degenerate approximant. An adaptive version, the adaptive cross-approximation (ACA) method, was then proposed by Bebendorf in 2000 [4] using geometrical information of the associated problem to derive properties of the rank-structure of the matrices to be approximated so to produce consecutive adaptive-rank matrix approximating a given matrix up to a given error. The first uses of the ACA method to the resolution of boundary integral equations are due to Bebendorf and Rjasanow in 2003 for the collocation method [3] and to Kurz, Rain and Rjasanow in 2002 for the 3D Galerkin BEM [24]. The development of ACA methods for different BEM is an active field since then. An early example of application to the BEM for the electromagnetic propagation was provided by Zhao, Vouvakis and Lee in 2005 [33].

The key feature of the CA methods in their application to the BEM for the boundary integral equations is that it can compute low-rank approximation of matrices up to a desired error without the need of developing kernel expansions. The approximations can be computed consulting only a few entries of the original matrix provided than it can be shown that degenerate approximants for the kernel exist (this condition can be assured checking certain smoothness conditions for the kernel). This feature allows for the re-utilization of existing BEM code (often extensively tested) in the construction of fast methods with the aid of hierarchical structures such as the $\mathcal{H}$-matrices.

References

[1] B. O. Almroth, P. Stern, and F. A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA Journal*, 16:525–528, May 1978.

[2] J. Barnes and P. Hut. A hierarchical $\mathcal{O}(n \log n)$ force calculation algorithm. *Nature*, (324):446–449, 1986.

[3] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.

[4] Mario Bebendorf. Approximation of boundary element matrices. *Numerical Mathematics*, 86(4):565–589, 2000.

[5] Mario Bebendorf. Hierarchical lu decomposition based on preconditioners for bem. *Computing*, 74:225–247, 2005.

[6] Mario Bebendorf. *Hierarchical Matrices.* Lecture Notes in Computational Science and Engineering. Springer, 2008.

[7] S. Börm and W. Hackbusch. *London Mathematical Society Lecture Notes*, chapter Approximation of boundary element operators by adaptive $\mathcal{H}$-matrices, pages 58–75. Cambridge University Press, 2004.

[8] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of the integral equations. *Journal of Computational Physics*, 90(2):348–370, 1990.

[9] A. Brandt and C. H. Venner. Multilevel evaluation of integral transforms with asymptotically smooth kernels. *SIAM Journal on Scientific Computing*, 19(2):468–492, 1998.

[10] Steffen Bröm, Lars Grasedyck, and Wolfgang Hackbusch. Lectures on hierarchical matrices. Technical report, Max Plank Institute for Mathematics, 2006.

[11] R. Coifman, V. Rokhlin, and S. Wandzura. The fast multipole method for the wave equations: a pedestrian prescription. *IEEE Antennas Propagation Magazine*, 35(3):7–12, 1993.

[12] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* John Hopkins Studies in the Mathematicas Sciences. John Hopkins University Press, 4th edition, 20013.

[13] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Applications*, 261:1–21, 1997.

[14] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. Technical report, Max Plank Institute for Mathematics, 2004.

[15] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.

[16] Leslie Greengard. The rapid evaluation of potential fields in particle systems. *Frontiers in Applied Mathematics*, 1988.

[17] W. Hackbusch. A sparse arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[18] W. Hackbusch. A sparse arithmetic based on $\mathcal{H}$-matrices. part ii: Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[19] W. Hackbusch, B. N. Khoromskij, and S. Sauter. *Lectures on Applied Mathematics*, chapter On $\mathcal{H}^2$-matrices, pages 9–29. Springer-Verlag, 2000.

[20] W. Hackbusch and Z. P. Nowak. On the complexity of the panel method. In *International Conference on Modern Problems in Numerical Analysis*, 1986.

[21] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerical Mathematics*, 54(4):463–491, 1989.

[22] W. Hackbusch and S. Sauter. On the efficient use of the galerkin method to solve fredholm integral equations. In *Proceeding of the 1992 International Symposium on Numerical Analysis, Part I*, volume 38, pages 301–322, 1993.

[23] Frank Ihlenburg. *Finite Element Analysis of Acoustic Scattering*. Springer, 1998.

[24] S. Kurz, O. Rain, and S. Rjasanow. The adaptive cross approximation technique for the 3d boundary element method. *IEEE Transaction on Magnetics*, 38(2):421–424, 2002.

[25] J. Laviada, R. Mittra, M. R. Pino, and F. Las-Heras. On the convergence of the aca. *Microwave and Optical Letters*, 51(10):2458–2459, 2009.

[26] Jean-Claude Nédélec. *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems*. Number 144 in Applied Mathematical Sciences. Springer, 2001.

[27] N. Nishimura. Fast multipole accelerated boundary integral equation methods. *Applied Mechanical Review*, 55(4):299–323, 2002.

[28] A. K. Noor and J. M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA Jounal*, 18(4):455–462, April 1980.

[29] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187–207, 1985.

[30] Stefan Sauter. *Über die Verwendung des Galerkinverfahrens zur Lösung Fredholmscher Integralgleichungen (On the use of Galerkin methods to solve Fredholm integral equations)*. PhD thesis, Chistian-Albrechts-Universität, 1992.

[31] Stefan Sauter and Christoph Schwab. *Boundary Element Methods*. Springer Series in Computational Mathematics. Springer, 2001.

[32] E. E. Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33(1-2):47–57, 1996.

[33] K. Zhao, M. N. Vouvakis, and J.-F. Lee. The adaptive cross approximation algorithm for accelerated method of moments computation of emc problems. *IEEE Transaction on Electromagnetic Compatibility*, 47:763–773, 2005.