Efficient Multikernel Hierarchical Compression for Boundary Element Matrices

Damiano Franzò, Simon B. Adrian, Adrien Merlini, and Francesco P. Andriulli

Abstract – We present a new scheme that allows for the compression of operators of the combined field integral equation in the low-frequency regime using a hierarchical decomposition that leverages a unified pseudoskeleton approximation of both the single- and the double-layer Green's function at the quadrature points. Compared with a standard adaptive cross approximation, the numerical results show a reduced memory consumption without sacrificing computational run time.

1. Introduction

The boundary element method (BEM) gives rise to dense matrices, which are typically hierarchically compressed by so-called fast methods, such as the fast multipole method [1] or the adaptive cross approximation (ACA) [2, 3] to yield at most $\mathscr{O}(N)$ and $\mathscr{O}(N \log N)$ global complexity, respectively, for electrically small problems, where N is the number of unknowns. What makes the ACA appealing is its purely algebraic nature, which allows for a straightforward combination with existing BEM codes. A less commonly used algorithm, which is also of purely algebraic nature, is the pseudoskeleton approximation, also known as CUR [4, 5].

Regardless of whether the ACA or the CUR is used, when the underlying integral equation formulation consists of several operators, one has to either store and compress a single-system matrix of the linear combination of the operators or to separately discretize, store, and compress the system matrix of each operator. Although the first approach is more memory efficient, it makes it difficult, if not impossible, to allow for a combination with advanced preconditioning strategies [6, 7]. The second approach, on the other hand, results in an increased memory consumption, a slower execution of matrix vector products, and an increased compression time. Moreover, if material parameters are changed, the entire matrix must be rebuilt.

In this article, we present a new scheme that is compatible with advanced preconditioning strategies, that is, where each operator has its own discretization, however, by using a CUR-based single hierarchical decomposition leveraging a pseudoskeleton approximation of the single- and double-layer Green's function kernel for electrically small problems at the quadrature points, the memory cost is significantly reduced compared with traditional approaches. We achieve this by compressing the Green's function sampled at the quadrature points instead of compressing the weakly tested operators (e.g., as in a standard ACA approach) together with using a CUR-based decomposition. The latter allows to efficiently compute rows and columns of the compressed submatrices on the fly, so only a small portion of the system matrices needs to be stored, thus reducing the memory consumption. The weak discretization of the operators are then reconstructed through the usage of sparse mappings, which contain the Gaussian quadrature weights and the evaluated basis functions. The hybrid cross approximation (HCA), introduced in [8], leverages the compression of an interpolated kernel-generating function to efficiently handle single- and double-layer matrices. Although the objectives of the method we propose and those of the HCA exhibit some similarities in philosophy, compressing the Green's function matrix at the quadrature points as we propose allows for straightforward integration into existing $\mathscr{H}$-matrix-enabled frameworks and remains a purely matrix-based technique. Although the method we present targets electrically small problems, extensions to handle electrically larger problems are foreseen, by properly leveraging advanced preconditioning and compression techniques, and will be the subject of future contributions.

2. Background and Notation

The Helmholtz equation

$$\nabla^2 \varphi(\mathbf{x}) + k^2 \varphi(\mathbf{x}) = 0 \quad \mathbf{x} \in \Omega^+ \subset \mathbb{R}^d \tag{1}$$

with $d \in \{2, 3\}$ is used to model two-dimensional (2D) and three-dimensional acoustic and 2D electromagnetic scattering and radiation problems, where k is the wavenumber, Ω denotes the scatterer, $\Omega^+ = \mathbb{R}^3 \setminus \bar{\Omega}$, and in the context of acoustics, φ denotes the total pressure field. For the impenetrable scatterer, we have the sound soft boundary condition $\varphi = \varphi^{\text{sca}} + \varphi^{\text{inc}} = 0$ on the boundary $\Gamma = \partial\Omega$, where φ^{inc} is the incident and φ^{sca} is the scattered pressure field. To solve the sound soft scattering

Manuscript received 11 July 2023. This work was supported, in part, by the European Research Council through the European Union's Horizon 2020 research and innovation program (grant 724846, project 321), in part, by the European Innovation Council through the European Union's Horizon Europe research program (grant 101046748, project CEREBRO).

Damiano Franzò and Francesco Andriulli are with Dipartimento di Elettronica e Telecomunicazioni (DET), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy; e-mail: damiano.franzo@polito.it, francesco.andriulli@polito.it.

Simon Adrian is with Fakultät für Informatik und Elektrotechnik, Universität Rostock, 18059 Albert-Einstein-Str. 2, Rostock, Germany; e-mail: simon.adrian@uni-rostock.de.

Adrien Merlini is with Microwave department, IMT Atlantique, Technopôle Brest-Iroise, 29238 Brest cedex 03, France; e-mail: adrien.merlini@imt-atlantique.fr.

problem, we use the Brakhage–Werner equation combined field integral equation (CFIE) formulation [9, 10]

$$(\mathscr{I}/2 + \mathscr{D} - \mathrm{j}\eta\mathscr{S})\psi = -\varphi^{\mathrm{inc}} \quad \text{for } \mathbf{r} \in \Gamma \tag{2}$$

where ψ is a surface scalar field, j is the imaginary unity, $(\mathscr{I}f)(\mathbf{x}) = f(\mathbf{x})$ is the identity operator, and η is the coupling parameter of the single-layer $(\mathscr{S}f)(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{y})f(\mathbf{y})\mathrm{d}s(\mathbf{y})$ and the double-layer operator $(\mathscr{D}f)(\mathbf{x}) = \int_\Gamma \partial_{\mathbf{n}'}G(\mathbf{x},\mathbf{y})f(\mathbf{y})\mathrm{d}s(\mathbf{y})$, where $G(\mathbf{x},\mathbf{y}) = \mathrm{e}^{\mathrm{j}k\|\mathbf{x}-\mathbf{y}\|}/(4\pi\|\mathbf{x}-\mathbf{y}\|)$ is the free-space Green's function associated with (1). We will refer to $G(\mathbf{x},\mathbf{y})$ as kernel of $\mathscr{S}$ and to $\partial_{\mathbf{n}'}G(\mathbf{x},\mathbf{y}) = \nabla_{\mathbf{y}}G(\mathbf{x},\mathbf{y}) \cdot \mathbf{n}'$ as the kernel of $\mathscr{D}$. Once ψ is found, the scattered field φ^{sca} can be computed via $\varphi^{\mathrm{sca}} = (\mathscr{D} - \mathrm{j}\eta\mathscr{S})\psi$.

Using the BEM, we approximate $\psi \approx \sum_{n=1}^{N} v_n\lambda_n$ in terms of piecewise linear basis functions $\{\lambda_n\}_{n=1}^{N}$, where v_n is the unknown expansion coefficient and

$$\lambda_n(\mathbf{r}) = \begin{cases} 1 & \text{for } \mathbf{x} = v_n \\ 0 & \text{for } \mathbf{x} = v_{p\neq n} \\ \text{linear} & \text{otherwise} \end{cases} \tag{3}$$

where v_n is the nth vertex of the mesh. Leveraging a Galerkin approach, the discretized Brakhage–Werner equation becomes

$$(1/2\,[I] + [D] - \mathrm{j}\eta[S])[v] = -[\varphi^{\mathrm{inc}}] \tag{4}$$

where $[D]_{mn} = (\lambda_m, \mathscr{D}\lambda_n)_{L^2}$, $[S]_{mn} = (\lambda_m, \mathscr{S}\lambda_n)_{L^2}$, $[I]_{mn} = (\lambda_m, \lambda_n)_{L^2}$, $[\varphi^{\mathrm{inc}}]_m = (\lambda_m, \varphi^{\mathrm{inc}})_{L^2}$, and $[v]_n = v_n$ are the unknown expansion coefficients.

The matrices $[S]$ and $[D]$ are fully populated, resulting in $\mathscr{O}(N^2)$ complexity for memory storage and as complexity for the cost of a single matrix vector product (MVP). The goal of any fast method is, at most, to obtain a quasilinear complexity, that is, to reduce the storage and the MVP time complexity to $\mathscr{O}(N\log N)$.

To obtain such a complexity, $\mathscr{H}$ matrices can be used [11–14], where the fact is leveraged that the matrices $[S]$ and $[D]$ have submatrices $[B] \in \mathbb{C}^{t\times s}$ formed by rows and columns $\subset \{1, ..., N\}$, admitting a low-rank form (LRF), where the rank is $\mathscr{O}(1)$ in the low-frequency regime, and we used MATLAB notation to denote the submatrices. In this article, we leverage the pseudoskeleton form $[B](t,s) \approx [C][U][R]$, where $[C] \in \mathscr{C}^{n\times r}$, $[U] \in \mathscr{C}^{r\times r}$, $[R] \in \mathscr{C}^{r\times m}$ and r is the ε rank of the decomposition satisfying an error bound $\|[B](t,s) - [C][U][R]\|_2 \leq \varepsilon\|[B](t,s)\|_2$, thereby lowering the cost of the MVP and storage time complexity if $r \times (n+m) < n \times m$. This form arises typically from sampling rows $[R]$ and columns $[C]$ according to a certain pattern. The matrix $[U]$ is usually the inverse or pseudoinverse of the intersection between the sampled rows and columns. To identify suitable submatrices, a partitioning algorithm such as an octree must be used, which creates a hierarchical partition of row and column indices (for details, see [12]). In the following, we refer to a cluster as a set of either row or column indices; thus, a row and a column cluster together correspond to a specific submatrix.

For compressible $[B](t,s)$, the associated integrands are nonsingular because the support of the expansion and of the testing functions are well separated. Because the integrand is nonsingular, a Gauss quadrature can be used for both the integrals over the domain of the testing and of the expansion function. Let $\mathscr{Q}$ be an integral operator with kernel $q(\mathbf{x},\mathbf{y})$. We indicate with $[Q_{\mathrm{ns}}]$ and $[Q_{\mathrm{s}}]$ the nonsingular and the singular interactions so that $[Q_{\mathrm{ns}}] + [Q_{\mathrm{s}}] = [Q]$. We note that

$$[Q_{\mathrm{ns}}]_{i,j} = \langle\lambda_i, \mathscr{Q}\lambda_j\rangle \approx \sum_{i'} w_{i'}\lambda(\mathbf{p}_{i'})\sum_{j'} w_{j'}\lambda(\mathbf{p}_{j'})q(\mathbf{p}_{i'} - \mathbf{p}_{j'}) \tag{5}$$

where $\mathbf{p}_{i'}$ and $\mathbf{p}_{j'}$ are the quadrature points associated with the integration over $\mathrm{supp}\,(\lambda_i)$ and $\mathrm{supp}\,(\lambda_j)$. By leveraging (5), we can rewrite $[Q_{\mathrm{ns}}]$ as a composition of matrix multiplications, leading to the matrix decomposition

$$[Q] = [W]^{\mathrm{T}}\,[\tilde{Q}]\,[W] + [Q_{\mathrm{s}}] + [Q_{\mathrm{c}}] \tag{6}$$

where $[Q_{\mathrm{s}}]$, $[W]$, and $[Q_{\mathrm{c}}]$ are sparse matrices whose values are zeros except $[W]_{ii'} = \lambda_i(\mathbf{p}_{i'})w_{i'}$, if $\mathbf{p}_{i'} \subset \mathrm{supp}(\lambda_i)$, $[Q_{\mathrm{c}}]_{ij} = -\left[[W]^{\mathrm{T}}\,[\tilde{Q}]\,[W]\right]_{ij}$, if $\langle\lambda_j, \mathscr{Q}\lambda_i\rangle$ is singular, $[Q_{\mathrm{s}}]_{ij} = \langle\lambda_j, \mathscr{Q}\lambda_i\rangle$, and if $\langle\lambda_j, \mathscr{Q}\lambda_i\rangle$ is singular and $[\tilde{Q}]_{i'j'} = (1 - \delta_{i',j'})\,q(\mathbf{p}_{i'} - \mathbf{p}_{j'})$, where $\delta_{i,j} = 1$, if $i = j$, and 0, otherwise, is a dense matrix containing the kernel evaluation on the quadrature points. This approach has been referred to as point based [15], and it is equivalent to replacing the traditional basis-to-basis interactions with point-to-point interactions. The correction matrix $[Q_{\mathrm{c}}]$ subtracts contributions that are considered twice in $[Q_{\mathrm{s}}]$ and $[\tilde{Q}]$. All the matrices above are sparse except $[\tilde{Q}]$, as it contains the kernel evaluations for quadrature points of nonsingular interactions.

3. A Multikernel Compression for the CFIE

We now consider $[Q] = [S]$ and $[Q] = [D]$, where we show that $[\tilde{D}]$ can be computed by reusing $[\tilde{S}]$, which we directly obtain, based on (5). Defining $\hat{G}(\mathbf{x},\mathbf{y}) = 4\pi G(\mathbf{x},\mathbf{y})$, we find that

$$\begin{aligned}\partial_{\mathbf{n}'}G(\mathbf{x},\mathbf{y}) = {} & G(\mathbf{x},\mathbf{y}) \\ & (\|\hat{G}(\mathbf{x},\mathbf{y})\|^2 - k\mathrm{j}\|\hat{G}(\mathbf{x},\mathbf{y})\|)(\mathbf{x}\cdot\mathbf{n}' - \mathbf{y}\cdot\mathbf{n}')\end{aligned} \tag{7}$$

By leveraging the decomposition (6), we obtain

$$\begin{aligned}[\tilde{D}] = {} & [H_{1x}][\tilde{G}_{\mathrm{d}}][H_{2x}] + [H_{1y}][\tilde{G}_{\mathrm{d}}][H_{2y}] + \\ & [H_{1z}][\tilde{G}_{\mathrm{d}}][H_{2z}] - [\tilde{G}_{\mathrm{d}}][M_x] - [\tilde{G}_{\mathrm{d}}][M_y] - [\tilde{G}_{\mathrm{d}}][M_z]\end{aligned} \tag{8}$$

where $[\tilde{G}_{\mathrm{d}}]_{i',j'} = [\tilde{S}]_{i',j'}(16\pi^2\,|\,[\tilde{S}]_{i',j'}\,|^2 - 4\pi k\mathrm{j}\,|\,[\tilde{S}]_{i',j'}\,|)$, $[M_p] = \mathrm{diag}(y_{i',p}n'_{i',p})$, $[H_{1p}] = \mathrm{diag}(n'_{i',p})$, and $[H_{2p}] = \mathrm{diag}(x_{i',p})$

with $p \in \{1, 2, 3\}$, $n_{i',p}$ are the pth components of the normal of the ith triangle, $x_{i',p}$ and $y_{i',p}$ are the pth components of the quadrature points $\mathbf{x}_{i'}$ and $\mathbf{y}_{i'}$ of the ith triangle, respectively.

Assume that $[B_\mathrm{s}]$ and $[B_{\tilde{G}_\mathrm{d}}]$ are rank-deficient submatrices of $[\tilde{S}]$ and $[\tilde{G}_\mathrm{d}]$ that are based on the same subset of rows and columns. Further assume the LRF of $[B_\mathrm{s}] \approx [C_\mathrm{s}][U_\mathrm{s}][R_\mathrm{s}]$ with $[C_\mathrm{s}] = [B_\mathrm{s}](:, \hat{J})$, $[R_\mathrm{s}] = [B_\mathrm{s}](\hat{I}, :)$, $[U_\mathrm{s}] = \left[[B_\mathrm{s}](\hat{I}, \hat{J})\right]^\dagger$, where $\hat{I}$ and $\hat{J}$ are index sets containing the row and column indices used in the factorization and the † denotes the Moore–Penrose pseudoinverse. With the same definitions, we denote the decomposition $[B_{\tilde{G}_\mathrm{d}}] \approx [C_{\tilde{G}_\mathrm{d}}][U_{\tilde{G}_\mathrm{d}}][R_{\tilde{G}_\mathrm{d}}]$, where we use the same index sets $\hat{I}$ and $\hat{J}$ for $[B_\mathrm{s}]$. The matrix decomposition of $[B_{\tilde{G}_\mathrm{d}}]$ can be related to $[B_\mathrm{s}]$ by

$$[C_{\tilde{G}_\mathrm{d}}][U_{\tilde{G}_\mathrm{d}}][R_{\tilde{G}_\mathrm{d}}] = [C_\mathrm{s}] \overline{\overline{*}} \, [U_{\tilde{G}_\mathrm{d}}] \overline{\overline{*}} \, [\mathrm{R}_s] \tag{9}$$

where, as in [16], $\overline{\overline{*}}$ is defined as

$$[A] \overline{\overline{*}} \, \mathbf{x} = 4\pi \sum_{j=1}^{N} (4\pi |[A]_{i,j}|^2 - kj|[A]_{i,j}|)[A]_{i,j} x_j \tag{10}$$

Also, (9) shows that once a decomposition of $[B_\mathrm{s}]$ has been obtained, the only additional information needed to approximate the submatrix $[B_{\tilde{G}_\mathrm{d}}]$ is the matrix $[U_{\tilde{G}_\mathrm{d}}]$, which can be computed in constant time. To obtain a memory-efficient compression, we note that the computation of $[C_\mathrm{s}]$ and $[R_\mathrm{s}]$ is extremely fast. Unlike in a traditional ACA scheme, where the actual system matrix is sampled, the direct evaluation of the Green's function at the quadrature points is much faster in terms of CPU time. For this reason, we will compute the matrix vector products with $[C_\mathrm{s}]$ and $[R_\mathrm{s}]$ on the fly. Due to (9), the same approach carries over to $[B_{\tilde{G}_\mathrm{d}}]$. Thus, in our approach, we store only the intersections $[B_\mathrm{s}](\hat{I}, \hat{J})$ and $[B_{\tilde{G}_\mathrm{d}}](\hat{I}, \hat{J})$ for each admissible cluster, which significantly reduces the stored memory. During each MVP $[C][U][R][x]$, rather than computing the pseudoinverse, which is subject to instabilities and depends on a truncation threshold, we use an lower–upper solver. Because the rank r is not known beforehand, we use the heuristic algorithm described in [5] to sample rows and columns: on each iteration, the indices I and J are chosen randomly, and the error is computed with respect to the projection into a random subspace of the previous iteration. If the error exceeds the demanded threshold, the sample size is doubled, and the algorithm continues with the next iteration. For each cluster, we have to compress the two submatrices $[B_\mathrm{s}]$ and $[B_{\tilde{G}_\mathrm{d}}]$. For stability reasons, we first start factorizing $[B_\mathrm{s}]$, and the obtained I_s and J_s are used as a starting point for factorizing $[B_{\tilde{G}_\mathrm{d}}]$.

It is possible to optimize storage by only saving rows and columns, where the single layer is evaluated for I and J indices. One can use the subset of rows $I_s \subset I$ and columns $J_s \subset J$ to compute the single-layer MVP. Likewise, the double-layer MVP can be computed with the stored rows I and columns J by using (9) and (8). Our approach is based on two assumptions: 1) the ε rank of $[B_{\tilde{G}_\mathrm{d}}]$ is larger than that of $[B_\mathrm{s}]$; and 2) the same rows and columns can be used for the factorization. In the next paragraph, we will provide some considerations that support these assumptions.

Assume $g(\mathbf{x}_i, \mathbf{y}_i)$ to be the kernel function associated to the matrix $[B_g]$. We have

$$[B_g]_{i,j} = g(\mathbf{x}_i, \mathbf{y}_j) = g_p(\mathbf{x}_i, \mathbf{y}_j) + r^g_{\mathbf{y},p}(\mathbf{x}_i) \tag{11}$$

where g_p is the polynomial (Lagrange) interpolation of g of order p and $r^g_{\mathbf{y},p}$ is the associated remainder. Also, assume a pseudoskeleton approximation of rank $K_p = \sum_{l=0}^{p-1} \begin{pmatrix} l+2 \\ l \end{pmatrix}$

$$[B_g] = [C_g][U_g][R_g] + [E_g] \tag{12}$$

where $[E_g]_{i,j} = r^g_{K_p}(\mathbf{x}_i, \mathbf{y}_j)$ is the associated remainder. This definition can also be written as

$$[B_g]_{i,j} = [C_g](i, :)[U_g][R_g](:, j) + r^g_{K_p}(\mathbf{x}_i, \mathbf{y}_j) \tag{13}$$

It can be shown that $r^g_{K_p}$ is related to $r^g_{\mathbf{y},p}$ [17, eq. (6)]

$$|\, r^g_{K_p}(\mathbf{x}_i, \mathbf{y}_j) \,| \leq (1 + 2^{K_p}) \sup_{\mathbf{x}} |\, r^g_{\mathbf{y},p}(\mathbf{x}) \,| \tag{14}$$

that later will allow us to shift the results on the bound of the polynomial remainder to the one arising from the pseudoskeleton approximation. As a consequence, we exploit this connection to justify our method by showing that given a function $g_s(x, y) = \|\mathbf{x} - \mathbf{y}\|^{-s}$, for $s = 1$, the relative error of an associated polynomial interpolation will decrease faster to zero than for $s > 1$ and $p \to \infty$. To show this, we first recall that the kernels $g_s : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ are asymptotically smooth, so $C_{\mathrm{as1},g_s}, C_{\mathrm{as2},g_s} > 0$ and $s \in \mathbb{N}^*$ exist and we have [18]

$$|\partial_i^n g_s(x, y)| \leq C_{\mathrm{as1}g_s} \left(C_{\mathrm{as2},g_s} \|\mathbf{x} - \mathbf{y}\|\right)^{-n-s} n! \tag{15}$$

where $g_s(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^{-s}$ and ∂_i^n is the nth derivative with respect to x_i. The constants C_{as1} and C_{as2} are indicated with a subscript g_s to indicate the kernel dependency. For $s = 1$, we have $C_{\mathrm{as1},g_1} = C_{\mathrm{as2},g_1} = 1$ [8, section 2.1]. Furthermore, to show the faster convergence, we need $C_{\mathrm{as2},g_s} < 1$ for $s > 1$, which is guaranteed by the following lemma.

3.1 Lemma 1

Let $g_s(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^{-s}$ with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$, $\mathbf{x} \neq \mathbf{y}$ and $C_{\mathrm{as1},g_s} \in \mathbb{R}^+$. For $\mathbb{N} \ni s > 1$, we have $C_{\mathrm{as2},g_s} < 1$.

3.1.1 Proof: Let $i \in 1, 2, 3$. For proving the lemma, it is sufficient to consider the special case $x_j = y_j$ for $j \neq i$. Given the kernel $g_s(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^{-s}$, with $s \geq 1$, we have

$$|\partial_i^n g_s(\mathbf{x}, \mathbf{y})| = \|\mathbf{x} - \mathbf{y}\|^{-n-s}(n + s - 1)! \tag{16}$$

that can be substituted in the asymptotic condition (15), leading to

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\|^{-n-s}(n + s - 1)! \leq C_{\text{as}1,g_s} \\ \left(C_{\text{as}2,g_s}\|\mathbf{x} - \mathbf{y}\|\right)^{-n-s} n! \end{aligned} \tag{17}$$

that simplifies to

$$\prod_{l=n+1}^{n+s-1} l \leq C_{\text{as}1,g_s} C_{\text{as}2,g_s}^{-n-s} \tag{18}$$

that leads to a contradiction if $C_{\text{as}2,g_s} \geq 1$. □

Let $\mathbf{x} \in B_\tau \subset \mathbb{R}^3$ and $\mathbf{y} \in B_\sigma \subset \mathbb{R}^3$, where B_τ and B_σ are boxes as an octree would generate them. We introduce the auxiliary constants defined in [18] $\gamma_{g_s} := 1/(C_{\text{as}2,g_s}\text{dist}(B_\tau, B_\sigma))$, $C_{g_s} := C_{\text{as}1,g_s}/\left(C_{\text{as}2,g_s}\text{dist}(B_\tau, B_\sigma)\right)^s$, and $C_{\text{diam}} := \min\{\text{diam}(B_\tau), \text{diam}(B_\sigma)\}$. Let $\tilde{g_s}$ be an approximation of g_s obtained via an interpolation scheme. Assuming that $\text{dist}(B_\tau, B_\sigma) > 0$, the interpolation (absolute) error is bounded by [18, eq. (12)]

$$\begin{aligned} &\|g_s - \tilde{g_s}\|_{\infty, B_\tau \times B_\sigma} \leq 24\text{e}\Lambda_p C_{g_s} \\ &(1 + \gamma_{g_s} C_{\text{diam}})(p + 1)\left(1 + 2/(\gamma_{g_s} C_{\text{diam}})\right)^{-(p+1)} \end{aligned} \tag{19}$$

where Λ_p is the Lebesgue constant, which depends on the specific interpolation scheme used.

To simplify the notation, we introduce the constant $\beta = C_{\text{diam}}/(2\text{dist}(B_\tau, B_\sigma))$. We note that $(1 + \gamma_{g_s} C_{\text{diam}}) = (C_{\text{as}2,g_s} + 2\beta)/C_{\text{as}2,g_s}$ and $\left(1 + 2/(\gamma_{g_s} C_{\text{diam}})\right)^{-(p+1)} = \left(\beta/(\beta + C_{\text{as}2,g_s})\right)^{(p+1)}$ so that (19) becomes

$$\begin{aligned} &\|g_s - \tilde{g_s}\|_{\infty, B_\tau \times B_\sigma} \leq 24\text{e}\Lambda_p C_{g_s} \\ &\left(\frac{C_{\text{as}2,g_s} + 2\beta}{C_{\text{as}2,g_s}}\right)(p + 1)\left(\frac{\beta}{\beta + C_{\text{as}2,g_s}}\right)^{(p+1)} \end{aligned} \tag{20}$$

We divide (20) by $\|g_s\|_{\infty, B_\tau \times B_\sigma} = \text{dist}(B_\tau, B_\sigma)^{-s}$, yielding

$$\begin{aligned} &\tilde{\varepsilon}_p^{g_s} := \frac{\|g_s - \tilde{g_s}\|_{\infty, B_\tau \times B_\sigma}}{\|g_s\|_{\infty, B_\tau \times B_\sigma}} \leq 24\text{e}\Lambda_k(C_{\text{as}1,g_s}/C_{\text{as}2,g_s}^s) \\ &\left(\frac{C_{\text{as}2,g_s} + 2\beta}{C_{\text{as}2,g_s}}\right)(p + 1)\left(\frac{\beta}{\beta + C_{\text{as}2,g_s}}\right)^{(p+1)} \end{aligned} \tag{21}$$

Clearly, we have an exponential convergence in K, where the speed of convergence is determined by the last factor. Because we have $C_{\text{as}2,g_1} = 1$ and due to Lemma 1, we have $C_{\text{as}2,g_s} < 1$ for $s > 1$, and we have

$$\left(\frac{\beta}{\beta + C_{\text{as}2,g_1}}\right)^{(p+1)} < \left(\frac{\beta}{\beta + C_{\text{as}2,g_s}}\right)^{(p+1)} \tag{22}$$

The result $\tilde{\varepsilon}_p^{g_s}$ can be interpreted as the supreme of the relative error of a polynomial interpolation of g_s of order p, which can be substituted in (14) on the right-hand side, leading to

$$\|[E_{g_s}] \oslash [B_{g_s}]\|_{\max} = \sup_{i,j} \frac{|r_{K_p}^g(\mathbf{x}_i, \mathbf{y}_j)|}{g_s(\mathbf{x}_i, \mathbf{y}_j)} \leq (1 + 2^{K_p})\,\tilde{\varepsilon}_p^{g_s} \tag{23}$$

where to simplify the notation, we set $[\tilde{E}_{g_s}] = [E_{g_s}] \oslash [B_{g_s}]$ to indicate the elementwise relative error matrix. By multiplying all terms of (23) by $\sqrt{mn}$, we obtain

$$\|[\tilde{E}_{g_s}]\|_{\text{F}} \leq \sqrt{mn}\|[\tilde{E}_{g_s}]\|_{\max} \leq \sqrt{mn}(1 + 2^{K_p})\,\tilde{\varepsilon}_p^{g_s} \tag{24}$$

Therefore, for each block, if a $\tilde{p}$ exists so that $\|[\tilde{E}_{g_s}]\|_{\text{F}} \leq \sqrt{mn}(1 + 2^{K_{\tilde{p}}})\,\tilde{\varepsilon}_{\tilde{p}}^{g_s} < \varepsilon$ for $s > 1$, then the exponential convergence properties of $\tilde{\varepsilon}_{\tilde{p}}^{g_s}$ will also guarantee that $\|[\tilde{E}_{g_1}]\|_{\text{F}} < \varepsilon$. Note, however, that the previously mentioned bound is often very pessimistic and the use of the previously mentioned criterion could lead to computational inefficiencies; thus, we opted for a heuristic approach for determining $K_{\tilde{p}}$, as explained at the beginning of this section. The validity of assumptions 1) and 2) for the use of heuristics is, as a consequence, a numerical conjecture.

4. Numerical Results

Consider as a canonical example the scattering from a sphere with radius 1 m, sound soft boundary condition, and a plane wave excitation with $\widehat{\mathbf{k}} = \widehat{\mathbf{z}}$, and frequency 20 Hz corresponding to $k \approx 0.36\,\text{m}^{-1}$. We study the asymptotic memory consumption and the time. As denoted in Section 2, we apply our algorithm to the CFIE (4). We compare the performance of our scheme with a standard ACA. Table 1 compares the timing between the two methods, while Figure 1 shows

Table 1. The timing performance of a standard ACA and our new formulation (multikernel)

N Time (min)		2562	10,242	40,962	163,842
ACA	Assembly	0.0675	0.3538	1.8326	9.1448
	Total	0.0690	0.3600	1.8605	9.2567
New	Assembly	0.0408	0.2171	1.0742	5.2420
	Total	0.1130	0.5204	2.4237	11.0048

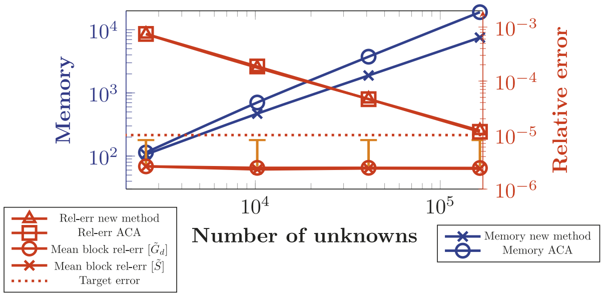

Figure 1. Memory usage and the relative error against an analytical solution of a standard ACA and our new method. The colored area indicates the standard deviation above the mean of the $[\tilde{G}_d]$ block relative error distribution.

memory consumption and the relative error. First, we note that the relative error of the solution displayed in the plot is with respect to the result provided by the analytical solution for the far field. Our formulation yields a similar accuracy as the standard ACA, while yielding memory savings. We note that for larger problems, the relative memory savings increase without jeopardizing the total run time. Our experiments have been performed on a cluster with a CPU Intel Xeon Gold 6238R and 1.5 TB of RAM. We used GMRES [19] as the numerical solver of the linear system, with a target accuracy of 10^{-5}. The target accuracy for the standard and new compression schemes is set to 10^{-5}. However, given the differences between compression algorithms, an identical target accuracy will lead to a lower level of compression for the reference ACA-based scheme. In practice, this means that a lower target accuracy can be set for the new scheme to obtain the same field accuracy, which leads to time and memory savings. The discussion on the choice of optimal target accuracy is omitted here because of space constraints.

5. References

1. L. Greengard and V. Rokhlin, “A Fast Algorithm for Particle Simulations,” *Journal of Computational Physics*, **73**, 2, December 1987, pp. 325-348.
2. K. Zhao, M. N. Vouvakis, and J. F. Lee, “The Adaptive Cross Approximation Algorithm for Accelerated Method of Moments Computations of EMC Problems,” *IEEE Transactions on Electromagnetic Compatibility*, **47**, 4, November 2005, pp. 763-773.
3. M. Bebendorf and S. Rjasanow, “Adaptive Low-Rank Approximation of Collocation Matrices,” *Computing*, **70**, February 2003, pp. 1-24.
4. S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkil, “A Theory of Pseudoskeleton Approximations,” *Linear Algebra and Its Applications*, **261**, 1–3, August 1997, pp. 1-21.
5. H. Lopez-Menchon, A. Heldring, E. Ubeda, and J. M. Rius, “A GPU Parallel Randomized CUR Compression Method for the Method of Moments,” *Computer Physics Communications*, **287**, June 2023, p. 108696.
6. A. Merlini, Y. Beghein, K. Cools, E. Michielssen, and F. P. Andriulli, “Magnetic and Combined Field Integral Equations based on the Quasi-Helmholtz Projectors,” *IEEE Transactions on Antennas and Propagation*, **68**, 5, May 2020, pp. 3834-3846.
7. S. B. Adrian, A. Dely, D. Consoli, A. Merlini, and F. P. Andriulli, “Electromagnetic Integral Equations: Insights in Conditioning and Preconditioning,” *IEEE Open Journal of Antennas and Propagation*, **2**, October 2021, pp. 1143-1174.
8. S. Börm and L. Grasedyck, “Hybrid Cross Approximation of Integral Operators,” *Numerische Mathematik*, **101**, 2, June 2005, pp. 221-249.
9. H. Brakhage and P. Werner, “Über das Dirichletsche Außenraumproblem für die Helmholtzsche Schwingungsgleichung,” *Archiv der Mathematik*, **16**, December 1965, pp. 325-329.
10. A. Burton and G. Miller, “The Application of Integral Equation Methods to the Numerical Solution of Some Exterior Boundary-Value Problems,” *Proceedings of the Royal Society of London A: Mathematical and Physical Sciences*, **323**, 1553, June 1971, pp. 201-210.
11. W. Hackbusch, “A Sparse Matrix Arithmetic Based on *H*-Matrices. Part I: Introduction to *H*-Matrices,” *Computing*, **62**, 2, April 1999, pp. 89-108.
12. M. Bebendorf, *Hierarchical Matrices*, Springer, Berlin, 2008.
13. P. Daquin, R. Perrussel, and J. R. Poirier, “Hybrid Cross Approximation for the Electric Field Integral Equation,” *Progress In Electromagnetics Research M*, **75**, October 2018, pp. 79-90.
14. M. Bauer and M. Bebendorf, “Block-Adaptive Cross Approximation of Discrete Integral Operators,” *Computational Methods in Applied Mathematics*, **21**, 1, January 2021, pp. 13-29.
15. K. C. Donepudi, J. Song, J.-M. Jin, G. Kang, and W. C. Chew, “A Novel Implementation of Multilevel Fast Multipole Algorithm for Higher Order Galerkin’s Method,” *IEEE Transactions on Antennas and Propagation*, **48**, 8, August 2000, pp. 1192-1197.
16. F. P. Andriulli and G. Vecchi, “Helmholtz-Stable Fast Solution of the Combined Field Integral Equation,” Fourth European Conference on Antennas and Propagation, Barcelona, Spain, April 12–16, 2010, pp. 1–4.
17. M. Bebendorf, “Approximation of Boundary Element Matrices,” *Numerische Mathematik*, **86**, 4, October 2000, pp. 565-589.
18. S. Börm and L. Grasedyck, “Low-Rank Approximation of Integral Operators by Interpolation,” *Computing*, **72**, January 2004, 3–4, pp. 325-332.
19. Y. Saad and M. H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, **7**, 3, July 1986, pp. 856-869.