Hierarchical matrix techniques for low- and high-frequency Helmholtz problems

Banjai, L ; Hackbusch, W

Abstract: In this paper, we discuss the application of hierarchical matrix techniques to the solution of Helmholtz problems with large wave number kappa in 2D. We consider the Brakhage–Werner integral formulation of the problem discretized by the Galerkin boundary-element method. The dense n x n Galerkin matrix arising from this approach is represented by a sum of an Formula -matrix and an Formula 2-matrix, two different hierarchical matrix formats. A well-known multipole expansion is used to construct the Formula 2-matrix. We present a new approach to dealing with the numerical instability problems of this expansion: the parts of the matrix that can cause problems are approximated in a stable way by an Formula -matrix. Algebraic recompression methods are used to reduce the storage and the complexity of arithmetical operations of the Formula -matrix. Further, an approximate LU decomposition of such a recompressed Formula -matrix is an effective preconditioner. We prove that the construction of the matrices as well as the matrix-vector product can be performed in almost linear time in the number of unknowns. Numerical experiments for scattering problems in 2D are presented, where the linear systems are solved by a preconditioned iterative method.

DOI: https://doi.org/10.1093/imanum/drm001

Hierarchical matrix techniques for low and high frequency Helmholtz problems

Lehel Banjai* Wolfgang Hackbusch†

12th December 2006

Abstract

In this paper, we discuss the application of hierarchical matrix techniques to the solution of Helmholtz problems with large wave number κ in two dimensions. We consider the Brakhage-Werner integral formulation of the problem, discretised by the Galerkin boundary element method. The dense $n \times n$ Galerkin matrix arising from this approach is represented by a sum of an $\mathcal{H}$-matrix and an $\mathcal{H}^2$-matrix, two different hierarchical matrix formats.

A well-known multipole expansion is used to construct the $\mathcal{H}^2$-matrix. We present a new approach to dealing with the numerical instability problems of this expansion: the parts of the matrix that can cause problems are approximated in a stable way by an $\mathcal{H}$-matrix. Algebraic recompression methods are used to reduce the storage and the complexity of arithmetical operations of the $\mathcal{H}$-matrix. Further, an approximate LU-decomposition of such a recompressed $\mathcal{H}$-matrix is an effective preconditioner. We prove that the construction of the matrices as well as the matrix-vector product can be performed in almost linear time in the number of unknowns. Numerical experiments for scattering problems in two dimension are presented, where the linear systems are solved by a preconditioned iterative method.

1 Introduction

Many physical problems (e.g. acoustics, electromagnetic scattering) require the solution of the Helmholtz equation (see [38]). We investigate the numerical solution of the Helmholtz equation by the boundary element method (BEM). In such methods the boundary is subdivided into n elements and the problem is reduced to the solution of an $n \times n$ linear system of equations. The corresponding matrix, B, is dense making direct methods for the solution of the system prohibitively expensive. To reduce the complexity from $\mathcal{O}(n^3)$ for the direct methods, or from $\mathcal{O}(n^2)$ for iterative methods, the so-called fast methods can be used (e.g. $\mathcal{H}$-matrices, panel clustering, FMM, wavelet methods [18, 28, 30, 33]). In these methods the matrix is represented by a data sparse format, reducing the cost of storage and matrix-vector multiplication to $\mathcal{O}(n \log^a n)$ for a small constant $a > 0$. The system is then solved using an iterative method. In this paper we describe how a combination of $\mathcal{H}$-matrix and $\mathcal{H}^2$-matrix techniques can be used to compress matrices arising from the discretisation of integral operators for the Helmholtz equation.

Two regimes of the Helmholtz problem are of interest: the high frequency and the low frequency regime. In the high frequency regime, the number of elements n is kept proportional to the wave number κ when working in two dimensions, and proportional to κ^2 when in three dimensions, i.e. $\kappa h = const$, where h is the mesh width. The condition $\kappa h = const$ insures that the accuracy of the approximation to the solution of the Helmholtz problem for different frequencies remains the same.

*University of Zurich, Switzerland *lehelb@math.unizh.ch*

†Max-Planck Institute for Mathematics in the Sciences, Leipzig, Germany *wh@mis.mpg.de*

In the low frequency regime, however, κ is a small constant, and the number of elements n is varied depending on the accuracy that needs to be achieved. The latter problem has many similarities with the Laplace problem and can be solved in $\mathcal{O}(n \log^a n)$ operations by similar methods (see [11] and also the numerical results in this paper).

The high frequency problem presents a considerably more difficult challenge. The fast multipole method (FMM) has been used to accelerate the solution of the high frequency Helmholtz problem by a number of authors. Initially one or two level versions were recommended which gave $\mathcal{O}(n^{3/2})$ or $\mathcal{O}(n^{4/3})$ algorithms [40, 41], but recently multilevel implementations were reported on, both in 2D and 3D, with complexity $\mathcal{O}(n \log^a n)$ for some small constant a (see [4, 19, 37]). In this paper we will draw on the contribution due to the multipole community. In particular we use a well-known multipole expansion to construct the $\mathcal{H}^2$-matrix, the details we adopt being closest to the paper of Amini and Profit [4]. In an $\mathcal{H}^2$-matrix, a sub-block R of the dense Galerkin matrix B is replaced by an approximation of a special form:

$$R \approx USV^\top, \quad \text{where } (R)_{kj} = \begin{cases} (B)_{kj}, & \text{if } n_1 \le k \le n_2,\ m_1 \le j \le m_2, \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

For the high-frequency case it is essential that the matrix S is of special structure, e.g. diagonal or Toeplitz. This can be achieved by the use of multipole expansions. Unfortunately, for some sub-blocks, $\|S\|_\infty$ can become very large and numerical instability problems render the approximation (1.1) unusable. Numerical instability problems of the multipole expansion for the Helmholtz problem have been well documented (see [39]). Using the findings of [39] we detect the blocks for which the approximation (1.1) is unstable, and approximate these blocks by an $\mathcal{H}$-matrix which can be computed in a stable manner without the use of the multipole expansion. It is possible to do this efficiently, since these blocks stem from the discretization of parts of the boundary that are small compared to the wavelength. Therefore we approximate the Galerkin matrix B by a sum of an $\mathcal{H}^2$- and an $\mathcal{H}$-matrix:

$$B \approx \hat{B} = \hat{B}_{\mathcal{H}^2} + \hat{B}_{\mathcal{H}}.$$

This splitting has further positive implications. It allows for considerable savings in storage and the cost of the solution of the linear problem, as we explain next. Algebraic recompression techniques described in [25] can be used to significantly reduce the storage and the complexity of arithmetical operations of the $\mathcal{H}$-matrix, $\hat{B}_{\mathcal{H}}$. The LU-decomposition of such a recompressed $\mathcal{H}$-matrix can be computed efficiently using $\mathcal{H}$-matrix techniques as described in [8, 25]. Once the LU-decomposition is available, the $\mathcal{H}$-matrix can also be used as an effective preconditioner, reducing the number of iterations needed by the iterative solver significantly. A further new aspect of our proposed method is that we allow for interaction between clusters of different sizes, which is not usually the case in the fast multipole methods for the Helmholtz equation.

In this paper, we consider only the Helmholtz Dirichlet problem and use the classical Brakhage-Werner integral formulation [16]. We discretize the integral equation by the boundary element method (BEM) with piecewise constant basis functions and prove that for a given accuracy $\epsilon > 0$, the proposed algorithm has complexity $\mathcal{O}(\kappa \log \kappa \log n + n \log \kappa \log \frac{1}{\epsilon})$ for the construction of the $\mathcal{H}^2$-matrix and for the matrix-vector product. The $\mathcal{H}$-matrix can be constructed and applied to a vector in $\mathcal{O}(k_{\max} n \log n)$ operations, where $k_{\max}$ is independent of κ and n.

Since in the high frequency regime κ is proportional to n, the complexity in this case reduces to $\mathcal{O}(n \log^2 n)$. However, the explicit dependence on κ is interesting since for a satisfactory accuracy the number of elements n needs to be chosen much larger than κ; in our numerical experiments with piecewise constant basis functions $n \approx 32\kappa$. The detailed complexity estimates serve better to explain and predict the results of numerical experiments.

To illustrate our methods, in the section on numerical results we solve an acoustic scattering problem, where the scatterer is either the unit disk or a non-convex object: the inverted ellipse. The numerical results are satisfactory up to very high frequencies, and also for low and intermediate frequencies. The sharpness of the complexity estimates is supported by the numerical results.

The paper is divided into five sections, first of which is this introduction, and an appendix. In Section 2 we state the Helmholtz problem we wish to solve and the corresponding Brakhage-Werner integral formulation. Next, in Section 3 we give a brief introduction to $\mathcal{H}$- and $\mathcal{H}^2$-matrices. Section 4 contains the main part of the paper. First the analytical tools for the construction of the matrices are developed. We then discuss the numerical instability issues, recompression, preconditioning, and give the algorithm for the construction of a stable, data-sparse approximation to the Galerkin matrix. We conclude the section with a proof of the complexity estimates. In the final section, we give the results of numerical experiments. The appendix contains proofs of the technical lemmata needed in Section 4.

2 Statement of the problem

Let $\Omega \in \mathbb{R}^d$, $d = 2, 3$, be a bounded domain with a smooth boundary Γ and exterior Ω^c. We are interested in the numerical solution of the exterior Dirichlet problem,

$$\begin{aligned} \Delta u + \kappa^2 u &= 0, \qquad x \in \Omega^c, \\ u(x) &= F(x), \quad x \in \Gamma, \\ \lim_{r\to\infty} r^{(d-1)/2} \left(\frac{\partial u}{\partial r} - i\kappa u \right) &= 0, \end{aligned} \tag{2.1}$$

where the wave number κ is a positive real parameter (see [38]).

The fundamental solution of the Helmholtz elliptic operator, which respects the condition at infinity, in 2D is the Hankel function of the first kind of order 0,

$$G_\kappa(x, y) = \frac{i}{4} H_0(\kappa \|x - y\|), \tag{2.2}$$

and in 3D the zero order spherical Hankel function of the first kind,

$$G_\kappa(x, y) = \frac{1}{4\pi} \frac{e^{i\kappa\|x-y\|}}{\|x - y\|}. \tag{2.3}$$

To solve this problem numerically using BEM, the elliptic partial differential equation is formulated as a boundary integral equation. In this paper we use the Brakhage-Werner formulation [13]. In this formulation the solution is represented as a combination of the single layer and the double layer operators applied to an unknown density φ:

$$u(x) = \int_\Gamma \frac{\partial}{\partial n_y} G_\kappa(x, y)\varphi(y) \mathrm{d}\Gamma_y - i\alpha \int_\Gamma G_\kappa(x, y)\varphi(y)\mathrm{d}\Gamma_y \qquad (x \in \Omega^c), \tag{2.4}$$

where n_y is the unit normal to Γ at $y \in \Gamma$, and $\alpha > 0$ is an arbitrary coupling parameter. Allowing x to tend to the boundary Γ and using the boundary condition we obtain the following boundary integral equation for the unknown density φ:

$$\frac{1}{2}\varphi(x) + \int_\Gamma \frac{\partial}{\partial n_y} G_\kappa(x, y)\varphi(y)\mathrm{d}\Gamma_y - i\alpha \int_\Gamma G_\kappa(x, y)\varphi(y)\mathrm{d}\Gamma_y = F(x) \qquad (x \in \Gamma). \tag{2.5}$$

The reason for using a combination of double and single layer potentials is the well-known fact that the single layer, double layer, and hypersingular operators are not invertible for certain special values of the wave number κ (see [38], [29, Lemma 8.5.3]). It can be shown that for $F \in L^2(\Gamma)$, the variational formulation of (2.5) has a unique solution in $L^2(\Gamma)$ (see [6]).

To discretise the integral operators occurring in the Brakhage-Werner formulation we apply the Galerkin method. If we use $\{\phi_1, \ldots, \phi_n\}$ as both the test and trial basis, then the discrete counterpart of (2.5) becomes

$$(\mathcal{I}/2 + K - i\alpha V)\mathbf{v} = \mathbf{b}, \tag{2.6}$$

where $\mathcal{I}, K, V \in \mathbb{C}^{n\times n}$ are the matrices defined by

$$(\mathcal{I})_{lk} = \int_\Gamma \int_\Gamma \phi_l(x)\phi_k(y)\mathrm{d}\Gamma_y\mathrm{d}\Gamma_x, \tag{2.7}$$

$$(V)_{lk} = \int_\Gamma \int_\Gamma G_\kappa(x,y)\phi_l(x)\phi_k(y)\mathrm{d}\Gamma_y\mathrm{d}\Gamma_x, \tag{2.8}$$

$$(K)_{lk} = \int_\Gamma \int_\Gamma \frac{\partial}{\partial n_y} G_\kappa(x,y)\phi_l(x)\phi_k(y)\mathrm{d}\Gamma_y\mathrm{d}\Gamma_x, \tag{2.9}$$

and the right-hand side $\mathbf{b} = (b_l) \in \mathbb{C}^n$ is defined by

$$b_l = \int_\Gamma F(x)\phi_l(x)\mathrm{d}\Gamma_x.$$

If $\mathbf{v} = (v_l) \in \mathbb{C}^n$ is the solution of (2.6) then an approximation $\hat{\varphi}(y)$ to the density $\varphi(y)$, at $y \in \Gamma$, is given by

$$\varphi(y) \approx \hat{\varphi}(y) := \sum_{l=1}^{n} v_l\phi_l(y),$$

which is then substituted into (2.4) to obtain the corresponding approximation to the solution u. Stability and convergence estimates for standard piecewise polynomial basis functions ϕ_l can be found in [6, 14, 23]. The main aim of this paper is to develop efficient methods for the construction and storage of the matrix $B = \mathcal{I}/2 + K - i\alpha V$ and for the solution of the linear problem (2.6). The matrix B is dense, hence we have $\mathcal{O}(n^2)$ complexity for storage and matrix-vector multiplication. In this paper we show that a much lower complexity is sufficient if we are satisfied with only an *approximation* of the Galerkin matrix B. Since for piecewise constant basis functions, the matrix $\mathcal{I}$ is a diagonal matrix and hence sparse, for most of the paper we only deal with the dense matrix $A := K - i\alpha V$.

$\mathcal{H}$-matrix techniques have already been successfully applied to integral equations with asymptotically smooth kernel functions $s(\cdot,\cdot)$ (see [7, 11, 32]). A function $s(\cdot,\cdot)$ is said to be *asymptotically smooth* if there exist constants c_1 and c_2 and a singularity degree $\sigma \in \mathbb{N}$ such that for any $z \in \{x_j, y_j\}$ and $n \in \mathbb{N}$ the inequality

$$|\partial_z^n s(x,y)| \leq n!\, c_1(c_2\|x-y\|)^{-n-\sigma} \tag{2.10}$$

holds. For the Helmholtz kernel G_κ, however, the inequality

$$|\partial_z^n G_\kappa(x,y)| \leq n!\, c_1(1+\kappa\|x-y\|)^n(c_2\|x-y\|)^{-n-\sigma} \tag{2.11}$$

holds. Hence, if $\kappa\,\mathrm{diam}(\Omega)$ is a small constant, i.e. if we are in the low frequency regime, the methods developed for general asymptotically smooth kernels, for example the interpolation method described in [11], should still be efficient. In the high frequency regime, this is no longer the case, and the $\mathcal{H}$-matrix techniques cannot be efficiently applied without a more involved structure of the $\mathcal{H}^2$-matrices.

For the rest of the paper, we restrict the discussion to two dimensions, $d = 2$. Further, the test and trial basis will be the usual piecewise constant finite element basis lifted to Γ. We proceed by giving a brief description of $\mathcal{H}$- and $\mathcal{H}^2$-matrices. For details we refer the reader to [9, 26, 31].

3 $\mathcal{H}$- and $\mathcal{H}^2$-matrices: The basics

Let the boundary Γ be subdivided into n disjoint panels π_j, $j \in \mathcal{J} := \{1, \ldots, n\}$. We consider piecewise constant basis functions ϕ_j, such that $supp\, \phi_j = \pi_j$, $j \in \mathcal{J}$.

Definition 3.1 *Given a constant $C_{\text{leaf}} > 0$, a labeled tree $\mathcal{T}_{\mathcal{J}}$, is said to be a* **cluster tree** *for $\mathcal{J}$ if the following conditions hold:*

- *For each $\tau \in \mathcal{T}_{\mathcal{J}}$, the label denoted by $\hat{\tau}$ is a subset of $\mathcal{J}$. In particular, the label of the root of the tree is the cluster $\mathcal{J}$ containing all the indices.*
- *If $\tau \in \mathcal{T}_{\mathcal{J}}$ has sons, then the sons form a partition of τ, i.e., $\hat{\tau} = \dot{\bigcup}\{\hat{\tau'} : \tau' \in sons(\tau)\}$.*
- *For every $\tau \in \mathcal{T}_{\mathcal{J}}$, $\#sons(\tau) \in \{0, 2\}$, and $\#\hat{\tau} > 0$.*
- *For each leaf τ, $\#\hat{\tau} \leq C_{\text{leaf}}$.*

We say that the root of the tree is at level 0, and that if a parent is at level l then its children are at level $l + 1$. We introduce the notation,

$$\Omega_\tau := \cup_{i \in \hat{\tau}} \pi_i \subseteq \Gamma,$$

for the subset of Γ corresponding to a cluster $\tau \in \mathcal{T}_{\mathcal{J}}$. The set of clusters which are at the same level are denoted by

$$\mathcal{T}_{\mathcal{J}}^{(l)} := \{\tau \in \mathcal{T}_{\mathcal{J}} : \tau \text{ at level } l\}.$$

Remark 3.2 *A couple of simple properties of the cluster tree will be useful for later analysis:*

- *the total number of clusters is bounded by $2n - 1$,*
- *at the lowest level p, there are at most n clusters, i.e. $\#\mathcal{T}_{\mathcal{J}}^{(p)} \leq n$.*

Introduce a restriction operator $\chi_\tau : \mathbb{R}^{n \times n}$ for each $\tau \in \mathcal{T}_{\mathcal{J}}$ by

$$(\chi_\tau)_{kj} = \begin{cases} 1, & \text{if } k = j \in \hat{\tau}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

We call a pair of clusters (τ, σ) a *block*. The corresponding block of the matrix A is then $\chi_\tau A \chi_\sigma$. Note that,

$$(\chi_\tau A \chi_\sigma)_{kj} = \begin{cases} (A)_{kj} = \int_{\Omega_\tau} \int_{\Omega_\sigma} (\frac{\partial}{\partial n_y} - i\alpha) G_\kappa(x, y) \phi_k(x) \phi_j(y) \mathrm{d}\Gamma_y \mathrm{d}\Gamma_x, & \text{if } k \in \hat{\tau} \text{ and } j \in \hat{\sigma}, \\ 0, & \text{otherwise.} \end{cases}$$

Let us briefly explain the importance of such a block. If $\Omega_\tau \cap \Omega_\sigma = \emptyset$, then the singular kernel restricted to these domains is smooth:

$$s(x, y) := (\frac{\partial}{\partial n_y} - i\alpha) G_\kappa(x, y) \in C^\infty, \quad x \in \Omega_\tau, \quad y \in \Omega_\sigma, \tag{3.2}$$

and the kernel can be approximated by a *separable expansion*:

$$s(x,y) \approx \sum_{l=0}^{M}\sum_{m=0}^{M} s_{l,m}u_l(x)v_m(y), \quad x \in \Omega_\tau, y \in \Omega_\sigma. \tag{3.3}$$

This can, for example, be achieved by using Taylor expansions (see [33]) or interpolation (see [11]). Such an expansion allows us to approximate the block $\chi_\tau A \chi_\sigma$ of the matrix by a low rank matrix:

$$\chi_\tau A \chi_\sigma \approx USV^\top, \tag{3.4}$$

where

$$(U)_{kl} := \begin{cases} \int_{\Omega_\tau} u_l(x)\phi_k(x)d\Gamma_x, & \text{if } k \in \hat{\tau}, l = 1,\ldots,M, \\ 0, & \text{otherwise,} \end{cases} \tag{3.5}$$

$$(V)_{jl} := \begin{cases} \int_{\Omega_\sigma} v_l(y)\phi_j(y)d\Gamma_y, & \text{if } j \in \hat{\sigma}, l = 1,\ldots,M, \\ 0, & \text{otherwise,} \end{cases} \tag{3.6}$$

and $(S)_{lm} := s_{lm}$. Note that for $\chi_\tau A \chi_\sigma$ we need $\mathcal{O}(|\tau||\sigma|)$ amount of storage, whereas for $USV^\top$ $\mathcal{O}(|\tau|M + |\sigma|M)$. If $M \ll \max\{|\tau|,|\sigma|\}$, it can be significantly advantageous to us the low rank approximation of the block.

The blocks for which we expect to be able to obtain a low rank approximation we call the *admissible blocks*. These blocks must be disjoint, otherwise the kernel is singular restricted to this block and we cannot expect to have a good approximation by a separable expansion. The admissibility property we control by a fixed parameter $\eta < 1$. In the following definition and throughout the paper, $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^2$.

Definition 3.3 *For each $\tau \in \mathcal{T}_\mathcal{J}$ let a centre $c_\tau \in \mathbb{R}^2$ and a radius $\rho_\tau > 0$ be given such that $\Omega_\tau \subseteq D(c_\tau,\rho_\tau) = \{y \in \mathbb{R}^2 \mid \|y - c_\tau\| < \rho_\tau\}$. Then we say that a block $b = (\tau,\sigma) \in \mathcal{T}_\mathcal{J} \times \mathcal{T}_\mathcal{J}$ is admissible if*

$$\rho_\tau + \rho_\sigma \leq \eta\|c_\tau - c_\sigma\|. \tag{3.7}$$

To easily access such blocks we construct a block cluster tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$. The tree is constructed by induction.

Definition 3.4 *The root of the* **block cluster tree** $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ *is the node $\mathcal{J}\times\mathcal{J}$. For each $b = (\tau,\sigma) \in \mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ proceed as:*

- *If b is admissible add it to the set of admissible leaves $\mathcal{L}^+$ of $T_{\mathcal{J}\times\mathcal{J}}$.*
- *If τ and σ are leaves of $\mathcal{T}_\mathcal{J}$, add b to the set of inadmissible leaves $\mathcal{L}^-$.*
- *Otherwise, repeat the procedure for all pairs formed by the sons of τ and σ (if one of the clusters has no sons use the cluster instead), which are then the sons of b in the tree $T_{\mathcal{J}\times\mathcal{J}}$.*

Note that the set of leaves of the block cluster tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ is partitioned into the set of *admissible leaves* $\mathcal{L}^+$ and the set of *inadmissible leaves* $\mathcal{L}^-$. The levels of the block cluster tree can be defined analogously to the case of the cluster tree. A property of the block cluster tree that is useful for complexity estimates is the *sparsity constant*.

Definition 3.5 *Define the sparsity constant of $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ by*

$$C_{\mathrm{sp}} := \max\left\{\max_{\tau\in\mathcal{T}_\mathcal{J}} \#\{\sigma \in \mathcal{T}_\mathcal{J} \mid (\tau,\sigma) \in \mathcal{T}_{\mathcal{J}\times\mathcal{J}}\}, \max_{\sigma\in\mathcal{T}_\mathcal{J}} \#\{\tau \in \mathcal{T}_\mathcal{J} \mid (\tau,\sigma) \in \mathcal{T}_{\mathcal{J}\times\mathcal{J}}\}\right\}.$$

When dealing with sparse matrices, the cost of storage and matrix-vector multiplication is governed by the maximum number of non-zero entries in a row or column. The sparsity constant C_{sp} is roughly the analogous measure for data sparse $\mathcal{H}$ matrices. In [26] it is shown that $T_{\mathcal{J}}$ and $T_{\mathcal{J}\times\mathcal{J}}$ can be constructed so that C_{sp} is bounded independently of the size of $\#\mathcal{J}$.

3.1 $\mathcal{H}$-matrices

Definition 3.6 *Let $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ be a block cluster tree and let $k : \mathcal{L}^+ \to \mathbb{N}_0$ be a rank distribution. We define the set of $\mathcal{H}$-matrices as*

$$\mathcal{H}\left(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, k(\cdot)\right) := \{M \in \mathbb{C}^{n\times n} |\ \textit{rank}(\chi_\tau M \chi_\sigma) \leq k(b) \textit{ for all admissible leaves } b = (\tau,\sigma) \in \mathcal{L}^+\}.$$

If $k(b) \leq k_{\max}$ for all $b \in \mathcal{L}^+$, it can be shown that the cost of storage and the cost of the matrix-vector multiplication of an $\mathcal{H}$-matrix is $\mathcal{O}(nk_{\max}p)$, where $p > 1$ is the depth of the block cluster tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$.

Lemma 3.7 *Let $M \in \mathcal{H}(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, k(\cdot))$, $k_{\max} := \max\{k(b) : b \in \mathcal{L}^+\}$, and let p be the depth of $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$. Then*

$$N_{\text{st}} \leq 2C_{\text{sp}}(p+1)\max\{k_{\max}, C_{\text{leaf}}\}n \textit{ and } N_{\mathcal{H}\cdot v} \leq 2N_{\text{st}},$$

where N_{st} is the storage requirement and $N_{\mathcal{H}\cdot v}$ the complexity of the matrix-vector multiplication.

We recall that C_{leaf} is an upper bound for the number of indices in a leaf cluster (see Definition 3.1). The proof of the lemma can be found in [26].

Instead of using separable expansions, an optimal approximation to the Galerkin matrix A from the set of $\mathcal{H}$-matrices, can be obtained by applying the SVD to each admissible block $\chi_\tau A \chi_\sigma$. Let $\chi_\tau A \chi_\sigma = U\Sigma V^\top$ be a singular value decomposition with singular values ordered so that $\Sigma_{11} \geq \Sigma_{22} \cdots \geq \Sigma_{nn} \geq 0$. As an approximation of the block, we can use the rank k *reduced singular value decomposition* $U_k\Sigma_k V_k^\top$, where $\Sigma_k := \text{diag}(\Sigma_{11}, \Sigma_{22}, \ldots, \Sigma_{kk})$, and U_k and V_k consist of the first k columns of matrices U and V respectively. The error of the approximation in the spectral norm is bounded by $\Sigma_{k+1,k+1}$:

$$\|\chi_\tau A \chi_\sigma - U_k\Sigma_k V_k^\top\|_2 \leq \Sigma_{k+1,k+1},$$

which is optimal, in this norm, for a rank k approximation. For a proof of this standard result see for example [44].

In Figure 3.1 we display the results of the following experiment: For a fixed accuracy $\epsilon = 1\times10^{-5}$ and a range of values of the wave number κ, compute the minimum rank k such that a rank k matrix $A_kB_k^\top$ exists with $\|\chi_\tau A\chi_\sigma - A_kB_k^\top\|_2 < \epsilon$. Figure 3.1 indicates that the necessary rank k is proportional to the wave number κ. Therefore in the high frequency regime, where we increase κ and require $\kappa h \approx \kappa/n = const$, complexity according to Lemma 3.7 is still $\mathcal{O}(n^2)$. Since the SVD gives us the optimal results, this experiment indicates that computing an $\mathcal{H}$-matrix approximation to the whole Galerkin matrix must be prohibitively costly in the high frequency regime.

3.2 $\mathcal{H}^2$-matrices

The structure of $\mathcal{H}^2$-matrices is considerably more involved than that of the $\mathcal{H}$-matrices; here we adopt the description given in [10]. Just as we have used the notion of a separable expansion to describe $\mathcal{H}$-matrices, we use it here to introduce the $\mathcal{H}^2$-matrices. In particular, we describe how a separable expansion can be used to construct an $\mathcal{H}^2$-matrix M, so that M is an approximation to the Galerkin matrix A.

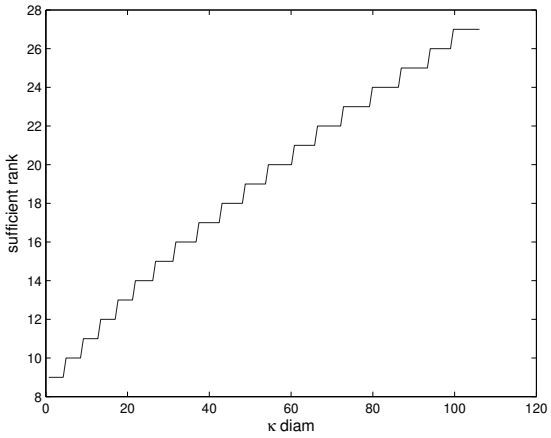


Figure 3.1: We compute the optimal low rank approximations $A_k B_k^\top$ to the matrix $\chi_\tau A \chi_\sigma$, where $(\tau, \sigma) \in \mathcal{L}^+$, by SVD. For a range of values of κ we plot the minimum rank necessary so that $\|\chi_\tau A \chi_\sigma - A_k B_k^\top\|_2 < 10^{-5}$.

Let $b = \tau \times \sigma$ be an admissible block and let us assume that we have an approximate separable expansion:

$$s(x,y) \approx \sum_{l=0}^{L_\tau} \sum_{m=0}^{L_\sigma} s_{l,m}^b u_l^\tau(x) v_m^\sigma(y), \quad x \in \Omega_\tau, y \in \Omega_\sigma. \tag{3.8}$$

Here, as we have indicated by the notation, we require that the basis functions $u_l^\tau(\cdot)$ (respectively $v_m^\sigma(\cdot)$) depend only on the cluster τ (respectively σ), and that the coefficients $s_{l,m}^b$ depend only on the block cluster $b = \tau \times \sigma$. Therefore the corresponding matrices U_τ and V_σ, see (3.5) and (3.6), can be reused whenever τ or σ appear in a different admissible cluster, e.g. $b' = \tau \times \sigma' \in \mathcal{L}^+$ for $\sigma' \neq \sigma$. This is clearly advantageous in terms of storage requirements. The matrices U_τ we call the *cluster basis* and give the following definition for a general cluster tree.

Definition 3.8 (cluster basis) *Let $\mathcal{T}_\mathcal{J}$ be a cluster tree and let a rank distribution $k : \tau \mapsto L_\tau \in \mathbb{N}_0$, $\tau \in \mathcal{T}_\mathcal{J}$, be given. Then a family $U = (U_\tau)_{\tau \in \mathcal{T}_\mathcal{J}}$ is called a cluster basis for $\mathcal{T}_\mathcal{J}$ with rank distribution k, if $U_\tau \in \mathbb{C}^{n \times L_\tau}$ and $\chi_\tau U_\tau = U_\tau$ for all $\tau \in \mathcal{T}_\mathcal{J}$.*

The condition $\chi_\tau U_\tau = U_\tau$ simply means that $(U_\tau)_{jl} = 0$ if $j \notin \hat{\tau}$, see (3.5). Further, note that the rank distribution is defined on the clusters, not on the block clusters.

We require additional structure, in particular we require that each function $u_l^\tau(\cdot)$ is a linear combination of basis functions $u_l^{\tau'}(\cdot)$ and $u_l^{\tau''}(\cdot)$ of its child clusters τ' and τ''. Namely, we require that

$$u_l^\tau(x) = \sum_{j=1}^{L_{\tau'}} t_{jl}^{\tau'} u_j^{\tau'}(x), \quad u_l^\tau(y) = \sum_{j=1}^{L_{\tau''}} t_{jl}^{\tau''} u_j^{\tau''}(y), \tag{3.9}$$

for $x \in \Omega_{\tau'}, y \in \Omega_{\tau''}$, $l = 1, 2, \ldots, L_\tau$. In matrix notation, this implies that

$$U_\tau = U_{\tau'} T_{\tau'} + U_{\tau''} T_{\tau''}, \tag{3.10}$$

where $(T_{\tau'})_{lj} = t_{lj}^{\tau'}$ and $(T_{\tau''})_{lj} = t_{lj}^{\tau''}$. Therefore we only need to store the cluster bases for the leaves and the *transfer matrices* T_τ for all clusters. As we will see later, this is advantageous both in terms of storage and the cost of performing a matrix-vector product.

Definition 3.9 (nested cluster basis) *Let $\mathcal{T}_\mathcal{J}$ be a cluster tree and let U be a corresponding cluster basis with rank distribution k. Let $T = (T_\tau)_{\tau\in\mathcal{T}_\mathcal{J}}$ be a family of matrices such that $T_{\tau'} \in \mathbb{C}^{L_{\tau'}\times L_\tau}$ for each $\tau' \in \mathcal{T}_\mathcal{J}$ that has a parent cluster τ. The cluster basis U is said to be nested with transfer matrices T if*

$$U_\tau = U_{\tau'}T_{\tau'} + U_{\tau''}T_{\tau''}, \tag{3.11}$$

for each parent cluster τ with son clusters τ' and τ''.

We are now in the position to define the class of $\mathcal{H}^2$-matrices.

Definition 3.10 ($\mathcal{H}^2$-matrix) *Let $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ be a block cluster tree, $k : \tau \to L_\tau$ a rank distribution, and U and V two nested cluster bases with transfer matrices T^U and T^V, respectively. Let $M \in \mathbb{C}^{n\times n}$. If for each $b = (\tau, \sigma) \in \mathcal{L}^+$ there exists $S_b \in \mathbb{C}^{L_\tau\times L_\sigma}$ such that*

$$\chi_\tau M \chi_\sigma = U_\tau S_b V_\sigma^\top,$$

the matrix M is said to be an $\mathcal{H}^2$-matrix with **row cluster basis** *U and* **column cluster basis** *V. The collection of such matrices is denoted by $\mathcal{H}^2(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, U, V, k(\cdot))$. The family $S = (S_b)_{b\in\mathcal{L}^+}$ is called the family of* **coefficient matrices**.

Note that we have not yet explicitly said what should be done with the inadmissible blocks, that is with the Galerkin matrix blocks $\chi_\tau A\chi_\sigma$, $b = \tau\times\sigma \in \mathcal{L}^-$. These blocks should simply be stored as dense matrices. The final part in approximating the Galerkin matrix A by an $\mathcal{H}^2$-matrix is to copy these blocks, i.e. require that

$$\chi_\tau M \chi_\sigma = \chi_\tau A \chi_\sigma, \quad b = \tau\times\sigma \in \mathcal{L}^-.$$

3.2.1 Fast matrix-vector multiplication

Let $\mathcal{T}_\mathcal{J}$ be a cluster tree and $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ a corresponding block cluster tree with the set of admissible leaves $\mathcal{L}^+$ and the set of inadmissible leaves $\mathcal{L}^-$ (see Definitions 3.1 and 3.4). For an arbitrary vector $u \in \mathbb{C}^n$ and $M \in \mathcal{H}^2(\mathcal{T}_{\mathcal{J}\times\mathcal{J}}, U, V, k(\cdot))$ we consider the computation of the matrix-vector product $v = Mu$. To do this as efficiently as possible, the structure of $\mathcal{H}^2$-matrices is used to the full extent. The computation is described in the following four step algorithm:

1. Upward pass from level p to level 0 of the tree $\mathcal{T}_\mathcal{J}$:
   - for all leaves $\sigma \in \mathcal{T}_\mathcal{J}$ compute $u_\sigma = V_\sigma^\top u$,
   - for all parents σ on the current level, set $u_\sigma = \left(T^{V_{\sigma'}}\right)^\top u_{\sigma'} + \left(T^{V_{\sigma''}}\right)^\top u_{\sigma''}$.
2. Far field interaction:
   - for all $\tau \in \mathcal{T}_\mathcal{J}$ compute $v_\tau = \sum_{(\tau,\sigma)\in\mathcal{L}^+} S_{\tau,\sigma} u_\sigma$.
3. Downward pass from level 0 to level p of tree $\mathcal{T}_\mathcal{J}$:
   - initialize the output vector v by zero,
   - for each child cluster τ' set $v_{\tau'} = v_{\tau'} + T^U_{\tau'} v_\tau$,
   - for every leaf $\tau \in \mathcal{T}_\mathcal{J}$ set $v = v + U_\tau v_\tau$.

4. Near field interaction:

   - $v = v + \sum_{(\tau,\sigma)\in\mathcal{L}^-} \chi_\tau M \chi_\sigma u.$

It is not immediately clear if $\mathcal{H}^2$-matrices offer any real advantage for the case of high frequency scattering. Indeed, since the SVD obtains optimal results, we know that the rank of a block $U_\tau S_b V_\sigma^\top \approx \chi_\tau A \chi_\sigma$ must increase at least linearly with κ. Therefore, if S_b is a dense matrix the complexity would again be at least $\mathcal{O}(\kappa^2) = \mathcal{O}(n^2)$. The complexity can only be reduced if the coefficient matrices S_b have some structure, e.g. if they are sparse or Toeplitz. In the next section we show that a separable expansion exists such that the coefficient and transfer matrices are either diagonal or Toeplitz.

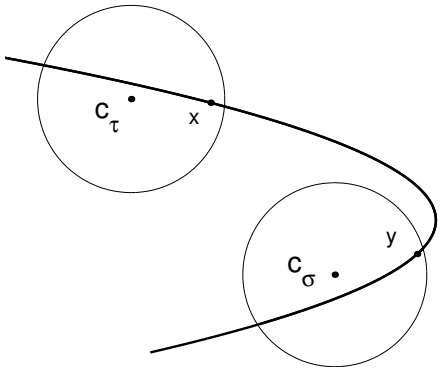
4 Construction of the $\mathcal{H}^2$-matrix

In this section we describe a separable expansion that has the properties (3.8) and (3.9) for the kernel function of the Brakhage-Werner integral operator. As described in the previous section we will then be able to construct an $\mathcal{H}^2$-matrix approximation to the Galerkin matrix. We make use of a separable expansion of the fundamental solution to the Helmholtz problem. This expansion has been developed and made well-known in the fast multipole community (see for example [40] and [5]). We will not give all the details but refer to results from the fast multipole literature. However, we give some convergence proofs, since in the literature we could not find the results exactly appropriate to our needs.

4.1 Separable expansions

For ease of notation, for a vector $x \in \mathbb{R}^2$ we denote its polar coordinates by (ρ_x, θ_x). In the following, $J_n(\cdot)$ denotes the Bessel function of the first kind of order n and $H_n(\cdot)$ the Hankel function of the first kind of order n.

Let $b = (\tau, \sigma) \in \mathcal{L}^+$, Ω_τ and Ω_σ be contained in disks with centres c_τ and c_σ and radii ρ_τ and ρ_σ, and let $x, y \in \mathbb{R}^2$ be such that $x \in \Omega_\tau$ and $y \in \Omega_\sigma$. The situation is depicted here:



where the bold line depicts a segment of the boundary Γ and the intersection of the disk centred at c_τ (respectively c_σ) with the boundary Γ is the region Ω_τ (resp. Ω_σ).

We will use the following notation in this section:

$$c_\tau - c_\sigma = \rho^{(\tau,\sigma)}(\cos\theta^{(\tau,\sigma)},\ \sin\theta^{(\tau,\sigma)})^\top, \tag{4.1}$$
$$z := (y - c_\sigma) - (x - c_\tau).$$

Since $b = (\tau, \sigma)$ is an admissible cluster,

$$\rho_\tau + \rho_\sigma \leq \eta \rho^{(\tau,\sigma)}. \tag{4.2}$$

Also, since $x \in \Omega_\tau$ and $y \in \Omega_\sigma$,

$$\|x - c_\tau\| < \rho_\tau, \quad \|y - c_\sigma\| < \rho_\sigma. \tag{4.3}$$

We refer to a result by Amini and Profit [5], which gives a separable approximation to the fundamental solution G_κ and a remainder convenient for finding error bounds.

Theorem 4.1 *Let L be an odd integer, $L = 2M + 1$. Then, using notation (4.1) and under the conditions (4.2) and (4.3),*

$$G_\kappa(x,y) = \frac{i}{4} H_0(\kappa\|y-x\|) = \sum_{l=1}^{L} \bar{f}_l(\kappa(x-c_\tau)) s_l(\kappa(c_\tau - c_\sigma)) f_l(\kappa(y-c_\sigma)) + \\ \frac{i}{4} \sum_{|m|>M} J_m(\kappa\rho_z) e^{-im\theta_z} \left(H_m(\kappa\rho^{(\tau,\sigma)}) e^{im\theta^{(\tau,\sigma)}} + i^{m-a} H_a(\kappa\rho^{(\tau,\sigma)}) e^{ia\theta^{(\tau,\sigma)}} \right),$$

where $a(m)$ is the unique integer such that $a \equiv m \pmod{L}$ and $a \in [-M, M]$. The functions f_l and s_l are defined by

$$f_l(\zeta) = e^{i\rho_\zeta \cos(2\pi l/L - \theta_\zeta)}, \quad s_l(\zeta) = \frac{i}{4} \sum_{m=-M}^{M} \frac{(-i)^m}{L} H_m(\rho_\zeta) e^{im(\theta_\zeta - 2\pi l/L)},$$

and $\bar{f}_l$ is the complex conjugate of f_l.

The above form of the separable expansion is the most commonly used diagonal form in fast multipole methods. For a detailed derivation see [15]. The next step is to give a bound on the number of terms needed to obtain a fixed accuracy $\epsilon > 0$. The result is not difficult to derive, once the following lemma has been proved. A similar result is proved in [5], but with some further restrictions on η and the length of expansion M.

Lemma 4.2 *Let $\rho > 0$, $0 < \epsilon < 1/2$, and $0 < \eta < 1$ be given. Then there exists a constant $C(\eta)$ such that for any $0 < r \leq r_{\max} = \eta\rho$ and $M \geq C(\eta)(r + \log \frac{1}{\epsilon})$,*

$$\sum_{n=M}^{\infty} |J_n(r)| < \epsilon \quad \textit{and} \quad \sum_{n=M}^{\infty} |H_n(\rho) J_n(r)| \leq \sum_{n=M}^{\infty} |H_{n+1}(\rho) J_n(r)| < \epsilon.$$

Proof. The proof is given in the Appendix. ∎

Theorem 4.3 *Let the conditions of Theorem 4.1 hold and let $0 < \epsilon < 1/2$ and $\kappa > 0$ be given. Then there exists a constant $C(\eta) > 0$ depending only on η, such that*

$$\left| G_\kappa(x,y) - \sum_{l=1}^{2M+1} \bar{f}_l(\kappa(x-c_\tau)) s_l(\kappa(c_\tau - c_\sigma)) f_l(\kappa(y-c_\sigma)) \right| < \epsilon,$$

for any $M \geq C(\eta)(\kappa(\rho_\tau + \rho_\sigma) + \log(\frac{1}{\epsilon}))$.

Proof. We express the remainder as in Theorem 4.1:

$$\begin{aligned} R_M &:= G_\kappa(x,y) - \sum_{l=1}^{2M+1} \bar{f}_l(\kappa(x-c_\tau)) s_l(\kappa(c_\tau - c_\sigma)) f_l(\kappa(y-c_\sigma)) \\ &= \frac{i}{4} \sum_{|m|>M} J_m(\kappa\rho_z) e^{-im\theta_z} \left(H_m(\kappa\rho^{(\tau,\sigma)}) e^{im\theta^{(\tau,\sigma)}} + i^{m-a} H_a(\kappa\rho^{(\tau,\sigma)}) e^{ia\theta^{(\tau,\sigma)}} \right), \end{aligned}$$

where $|a| \leq M$. Since for a fixed argument $x > 0$, $|H_m(x)|$ is an increasing function of $m \geq 0$, see [3], we have that

$$|R_M| \leq \frac{1}{2} \sum_{|m|>M} |J_m(\kappa\rho_z) H_m(\kappa\rho^{(\tau,\sigma)})|.$$

The result now follows immediately from Lemma 4.2, since according to (4.1) and (4.2) $\rho_z = \|(y-c_\sigma) - (x-c_\tau)\| < \rho_\tau + \rho_\sigma \leq \eta\rho^{(\tau,\sigma)}$. ∎

In the following corollary we give an expression for a separable expansion of the singular kernel of the Brakhage-Werner formulation.

Corollary 4.4 *Under the conditions of Theorem 4.3, and with $\alpha \leq \kappa$, there exists a constant $C(\eta)$ such that*

$$\left| \left(\frac{\partial}{\partial n_y} - i\alpha \right) G_\kappa(x,y) - \sum_{l=1}^{2M+1} \bar{f}_l(\kappa(x-c_\tau)) s_l(\kappa(c_\tau - c_\sigma)) \left(\frac{\partial}{\partial n_y} - i\alpha \right) f_l(\kappa(y-c_\sigma)) \right| < \epsilon$$

for any $M \geq C(\eta)(\kappa(\rho_\tau + \rho_\sigma) + \log(\kappa) + \log(\frac{1}{\epsilon}))$.

Proof. For the proof we need the estimate

$$\left| \frac{\partial}{\partial n_y} J_m(\kappa\rho_z) e^{-im\theta_z} \right| \leq \frac{3\kappa}{2} J_{m-1}(\kappa\rho_z), \tag{4.4}$$

which holds under the condition $|m| > \kappa\rho_z + 2$ (see [5]). Note that $C(\eta)$ can be chosen so that any m, with $|m| > M$, satisfies such a condition. By Lemma 4.2, the remainder in Theorem 4.1 converges absolutely. The series obtained by formally differentiating each term in this remainder is due to (4.4) and Lemma 4.2 also absolutely convergent and hence we are allowed to differentiate term by term:

$$\begin{aligned} R_M^{(1)} &:= \left(\frac{\partial}{\partial n_y} - i\alpha \right) G_\kappa(x,y) - \sum_{l=1}^{2M+1} \bar{f}_l(\kappa(x-c_\tau)) s_l(\kappa(c_\tau - c_\sigma)) \left(\frac{\partial}{\partial n_y} - i\alpha \right) f_l(\kappa(y-c_\sigma)) \\ &= \frac{i}{4} \sum_{|m|>M} \left(\frac{\partial}{\partial n_y} - i\alpha \right) J_m(\kappa\rho_z) e^{-im\theta_z} \left(H_m(\kappa\rho^{(\tau,\sigma)}) e^{im\theta^{(\tau,\sigma)}} + i^{m-a} H_a(\kappa\rho^{(\tau,\sigma)}) e^{ia\theta^{(\tau,\sigma)}} \right), \end{aligned}$$

and bound the new remainder by

$$|R_M^{(1)}| \leq \left(\frac{3\kappa}{2} + \alpha \right) \sum_{|m|>M-1} \left| J_m(\kappa\rho_z) H_{m+1}(\kappa\rho^{(\tau,\sigma)}) \right|.$$

The proof now follows from an application of Lemma 4.2. ∎

Note that the separable expansion given by the above corollary is not exactly of the form required by (3.8). The basis functions $u_l^\tau(\cdot)$ in (3.8) were required to depend only on the cluster

τ. This is not the case for the functions $f_l(\cdot)$ since they explicitly depend on the length of the expansion $2M+1$ which in turn depends on ρ_τ *and* ρ_σ. In fast multipole methods this difficulty is avoided by only considering admissible blocks $b=(\tau,\sigma)$ for which $\rho_\tau=\rho_\sigma$. Not to be restricted by this kind of a condition, we introduce a different separable expansion.

To do this, we find it helpful to recall that the Bessel functions are the Fourier coefficients of plane waves $\{f_l\}$:

$$J_n(r)=\frac{1}{\pi i^n}\int_0^\pi e^{ir\cos\theta}\cos(n\theta)\mathrm{d}\theta=\frac{1}{2\pi i^n}\int_0^{2\pi}e^{ir\cos\theta}e^{in\theta}\mathrm{d}\theta,\quad n=0,1,\ldots, \tag{4.5}$$

see [1, 45]. Note also that $J_{-n}=(-1)^nJ_n$. The relationship between Bessel functions and plane waves is of crucial importance for all the results in this section.

We will not only want to transform the plane wave functions to the Bessel functions, but also change the number of functions in the expansion. To do this we will make use of a simple operator P_{M_1,M_2} which either truncates a vector or appends zeros to it depending on the sign of M_1-M_2. For example

$$P_{3,2}\begin{pmatrix}a_1\\a_2\\a_3\end{pmatrix}=\begin{pmatrix}0&0&0\\1&0&0\\0&1&0\\0&0&1\\0&0&0\end{pmatrix}\begin{pmatrix}a_1\\a_2\\a_3\end{pmatrix}=\begin{pmatrix}0\\a_1\\a_2\\a_3\\0\end{pmatrix};\ P_{2,3}\begin{pmatrix}b_1\\b_2\\b_3\\b_4\\b_5\end{pmatrix}=P_{3,2}^\top\begin{pmatrix}b_1\\b_2\\b_3\\b_4\\b_5\end{pmatrix}=\begin{pmatrix}b_2\\b_3\\b_4\end{pmatrix}.$$

The definition for general M_1 and M_2 is given next.

Definition 4.5 *Let $L_1=2M_1+1$ and $L_2=2M_2+1$ be two positive odd integers. If $M_1\geq M_2$, define the operator P_{M_1,M_2} by induction on M_1-M_2:*

1. *The matrix $P_{M,M}:=I\in\mathbb{R}^{2M+1\times 2M+1}$ is the identity matrix.*
2. *Define $P_{M+j+1,M}\in\mathbb{R}^{2(M+j+1)+1\times 2M+1}$ by $P_{M+j+1,M}:=\begin{pmatrix}0\cdots 0\\P_{M+j,M}\\0\cdots 0\end{pmatrix}$.*

If $M_2>M_1$, then $P_{M_1,M_2}:=(P_{M_2,M_1})^\top$.

Next we give the details of the transformation from a plane wave basis to a Bessel basis. For a pictorial explanation see Figure 4.1.

Proposition 4.6 *Let $M_1,M_2\in\mathbb{N}$ with $M_1\geq M_2$ and let $L_1=2M_1+1, L_2=2M_2+1$. For $x\in\mathbb{R}^2$, let $\mathbf{f}_{M_1}(x)$ and $\mathbf{g}_{M_2}(x)$ be defined by*

$$(\mathbf{f}_{M_1}(x))_l:=f_l(x)=e^{i\rho_x\cos(2\pi l/L_1-\theta_x)}\ \text{and}\ (\mathbf{g}_{M_2}(x))_j:=g_j(x):=i^{j-M_2-1}J_{j-M_2-1}(\rho_x)e^{i(j-M_2-1)\theta_x},$$

$l=1,\ldots,L_1$, $j=1,\ldots,L_2$. Further, let the shifted Fourier matrix $F_{M_1}\in\mathbb{C}^{L_1\times L_1}$ be defined by

$$(F_{M_1})_{ml}=\frac{1}{L_1}e^{i(m-M_1-1)\frac{2\pi l}{L_1}},\quad l,m=1,2,\ldots,L_1.$$

There exists a constant $C>0$ such that for any $\epsilon>0$, if $M_2>C(\rho_x+\log(\frac{1}{\epsilon}))$ then

$$r_{M_2}:=\|\mathbf{f}_{M_1}(x)-F_{M_1}^{-1}P_{M_1,M_2}\mathbf{g}_{M_2}(x)\|_\infty\leq\epsilon.$$

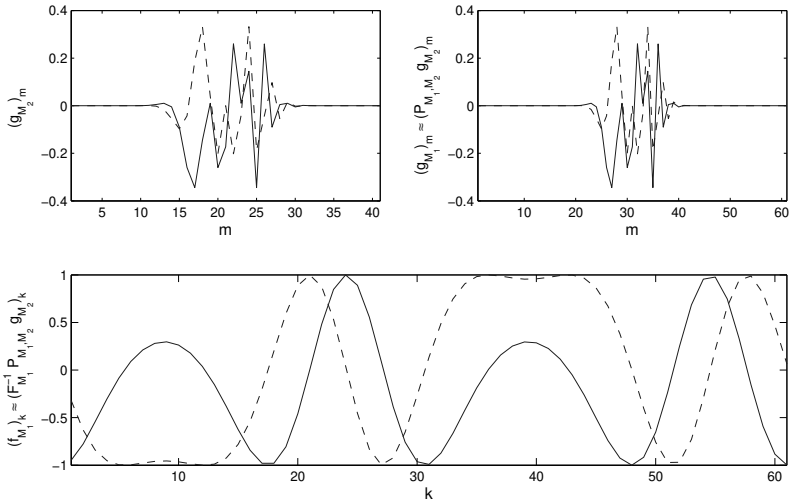


Figure 4.1: In this figure we show the transformation from a Bessel basis to a plane wave basis. Here $M_1 = 30$, $M_2 = 20$, and $|x| = 5$. Top left we plot the coefficients of Bessel function basis $\mathbf{g}_{M_2}(x)$; the real parts are connected by a solid line and the imaginary by a dashed line. We append zeros to $\mathbf{g}_{M_2}(x)$ to obtain an approximation $P_{M_1,M_2}\mathbf{g}_{M_2}(x) \approx \mathbf{g}_{M_1}(x)$; shown in the top right plot. Next we apply the matrix $F_{M_1}^{-1}$ to obtain an approximation to the plane wave basis $\mathbf{f}_{M_1}(x)$ shown in the last plot.

Proof. The proof is given in the Appendix. ∎

We now finish the subsection by giving the separable expansion applicable to clusters of different size.

Theorem 4.7 *Let $b = (\tau, \sigma)$ be an admissible cluster. Then define the coefficient matrix $S_b = (s_{l,k}^b)$ by*

$$S_b = P_{M_\tau, M_{\tau,\sigma}} F_{M_{\tau,\sigma}} diag\left(s_l(\kappa(c_\tau - c_\sigma))\right) F_{M_{\tau,\sigma}}^{-1} P_{M_{\tau,\sigma}, M_\sigma}, \tag{4.6}$$

where $M_\tau, M_\sigma, M_{\tau,\sigma} \in \mathbb{N}$ and $s_l(\cdot)$ are defined in Theorem 4.1, $l = 1, \ldots, 2M_{\tau,\sigma} + 1$. Under the conditions of Corollary 4.4 there exist constant C and $C(\eta)$ such that

$$\left| \left(\frac{\partial}{\partial n_y} - i\alpha \right) G_\kappa(x,y) - \sum_{l=1}^{2M_\tau+1} \sum_{l=1}^{2M_\sigma+1} s_{l,k}^{(\tau,\sigma)} \bar{g}_l(\kappa(x - c_\tau)) \left(\frac{\partial}{\partial n_y} - i\alpha \right) g_l(\kappa(y - c_\sigma)) \right| < \epsilon$$

for any $M_\tau \geq C(\kappa\rho_\tau + \log(\frac{1}{\epsilon}))$, $M_\sigma \geq C(\kappa\rho_\sigma + \log\kappa + \log(\frac{1}{\epsilon}))$, and $M_{\tau,\sigma} \geq C(\eta)(M_\tau + M_\sigma)$.

Proof. The main fact to notice is that $\|D_M\|_\infty = \|P_{M_1,M_2}\|_\infty = \|F_M\|_\infty = 1$, so the errors are not amplified by these matrices. Since $F_M^{-1} = (2M+1)F_M^*$, where F_M^* is the conjugate transpose of F_M, we have that $\|F_M^{-1}\|_\infty = 2M + 1$. Since this term also does not have a significant effect on the exponential convergence, the error estimate follows directly from Corollary 4.4 and Proposition 4.6. ∎

4.2 Transfer operators

To be able to construct the $\mathcal{H}^2$-matrix, we need also the nestedness condition to be fulfilled; see (3.9). Rewriting (3.9) in terms of our basis functions g_l, if τ' is a child cluster of τ we need to find

a transfer matrix $T_{\tau'} = (t_{lj}^{\tau'})$ such that

$$g_l(\kappa(x - c_\tau)) = \sum_{j=1}^{L_{\tau'}} t_{jl}^{\tau'} g_j(\kappa(x - c_{\tau'})), \quad \text{for } x \in \Omega_\tau, \quad l = 1, \ldots, L_\tau. \tag{4.7}$$

Here we see that the transfer matrix needs to do two things: change the centre of the expansion from $c_{\tau'}$ to c_τ and change the length of the expansion from $L_{\tau'}$ to L_τ; the latter procedure is often called, and performed by, interpolation. In our case we will be able to guarantee (4.7) only approximately.

The connection between Bessel functions and the plane waves, see Proposition 4.6, is useful here as well. One part of the transfer, translation of the centre of expansion, is easy for the plane waves, and the other, the interpolation, is easy for the Bessel functions. The translation for the plane wave functions is given by,

$$f_l(\kappa(x - c_\tau)) = f_l(\kappa(c_{\tau'} - c_\tau)) f_l(\kappa(x - c_{\tau'})), \tag{4.8}$$

where f_l are defined as in Theorem 4.1. This property of plane waves is not difficult to check (for a proof see [4]). For the Bessel functions, the change of the centre is not as simple but the interpolation, i.e. the change of the length of expansion, is trivial. It consists simply of truncation or padding by zeros of the basis vectors; see Figure 4.1. We give now the definition of the translation operator.

Definition 4.8 *Let $L = 2M + 1$ be an odd positive integer and let τ and τ' be two clusters. Define the diagonal matrix $D_M^{\tau,\tau'} \in \mathbb{C}^{L \times L}$ that translates the centre of expansion from $c_{\tau'}$ to c_τ by*

$$(D_M^{\tau,\tau'})_{ll} := f_l(\kappa(c_{\tau'} - c_\tau)), \quad l = 1, \ldots, 2M + 1.$$

To simplify the notation we will leave out the various subscripts and superscripts if they are clear from the context. Combining the change of the centre of the plane wave expansion and the interpolation of the Bessel function expansion with Proposition 4.6, allows us to easily construct the transform operator. The details are given in the next theorem.

Theorem 4.9 *Let $x, c_\tau, c_{\tau'} \in \mathbb{R}^2$ be fixed and let $L_\tau = 2M_\tau + 1$ and $L_{\tau'} = 2M_{\tau'} + 1$ for some $M_\tau, M_{\tau'} \in \mathbb{N}$. Define $\mathbf{g}_{M_\tau}(x) \in \mathbb{C}^{L_\tau}$ and $\mathbf{g}_{M_{\tau'}}(x) \in \mathbb{C}^{L_{\tau'}}$ by*

$$(\mathbf{g}_{M_\tau}(x))_l := g_l(\kappa(x - c_\tau)) \quad \text{and} \quad (\mathbf{g}_{M_{\tau'}}(x))_j := g_j(\kappa(x - c_{\tau'})),$$

where g_l are defined in Proposition 4.6. There exists a constant $C > 0$ such that for any $\epsilon > 0$, if $M_\tau > C(\kappa\|x - c_\tau\| + \log(\frac{1}{\epsilon}))$ and $M_{\tau'} > C(\kappa\|x - c_{\tau'}\| + \log(\frac{1}{\epsilon}))$

$$\|\mathbf{g}_{M_\tau}(x) - F_{M_\tau} D_{M_\tau} F_{M_\tau}^{-1} P_{M_\tau, M_{\tau'}} \mathbf{g}_{M_{\tau'}}(x)\|_\infty < \epsilon,$$

holds.

Proof. The proof is very similar to the proof of Theorem 4.7. ∎

Therefore the transfer matrix is given by: $T_{\tau'} = \left(F_{M_\tau} D_{M_\tau} F_{M_\tau}^{-1} P_{M_\tau, M_{\tau'}}\right)^\top$. Since the operator $\left(\frac{\partial}{\partial n_y} - i\alpha\right)$ is linear, the same transfer matrix can be used for the basis functions $\left(\frac{\partial}{\partial n_y} - i\alpha\right) g_l(\kappa(y - c_\sigma))$.

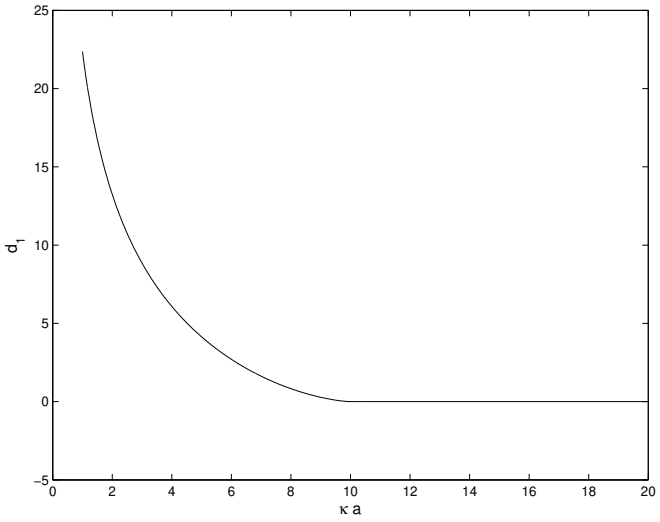


Figure 4.2: For a fixed expansion length $L = 10$ and $\eta = 1/2$, the number of digits lost due to numerical instability is plotted against κa, where a is the size of the clusters.

4.3 Numerical stability

An important fact is hidden by error estimates of the type given in Theorem 4.3. Due to numerical stability problems, not every accuracy $\epsilon > 0$ can be reached when working in finite precision. Numerical stability problems of the separable expansion are due to the exponential increase of Hankel functions $H_l(x)$ for fixed x and $l > x$ (see [1]). A careful analysis of the numerical stability issues has been performed by Ohnuki and Chew [39], whose results we will make use of.

Let us return to the setting of Theorem 4.1 and let us assume that the radii of the clusters are $\rho_\tau = \rho_\sigma = a/2 > 0$ (see also (4.2)). We recall that L is the length of the expansion used to approximate the Hankel function. Then define,

$$d_1 := \begin{cases} 0, & L < \frac{1}{\eta}\kappa a, \\ \left\{ \left(L - \frac{1}{\eta}\kappa a \right) / \left(1.8[\frac{1}{\eta}\kappa a]^{1/3} \right) \right\}^{3/2}, & \text{otherwise.} \end{cases} \tag{4.9}$$

In [39] it is argued that d_1 is a good approximation to the number of digits lost due to numerical stability problems. For example, this means that, if the required accuracy is $\epsilon = 10^{-5}$ and the other parameters are such that $d_1 = 10$, in double precision the stability problems should not be visible. However, a considerably higher accuracy could not be obtained. It is clear by inspecting (4.9) that fewer digits are lost if κa is large, that means if the wave number times the size of the cluster is large (see also Figure 4.2). This suggests that the separable expansion should be used only for admissible block clusters that are formed of clusters large enough for numerical stability problems not to be visible. In our setting the clusters need not have equal radii. In practice, we have found that the following definition is suitable.

Definition 4.10 *Let $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ be a block cluster tree and let $a > 0$ be given. Divide the set of admissible leaves $\mathcal{L}^+$ into two disjoint subsets by*

$$\mathcal{L}_1^+ := \left\{ (\tau, \sigma) \in \mathcal{L}^+ : \max\{\operatorname{diam}(\Omega_\tau), \operatorname{diam}(\Omega_\sigma)\} \geq a \right\} \quad \textit{and} \quad \mathcal{L}_2^+ := \mathcal{L}^+ \setminus \mathcal{L}_1^+.$$

We will use the separable expansion only in admissible blocks belonging to $\mathcal{L}_1^+$. Note that a should be chosen proportional to $1/\kappa$, i.e. $a\kappa = const.$

4.4 Definition of the $\mathcal{H}$- and $\mathcal{H}^2$-matrix approximant

We are now in a position to construct an accurate hierarchical matrix approximation to the Galerkin matrix. Let $b = (\tau, \sigma) \in \mathcal{L}_1^+$. We recall that

$$(\chi_\tau A \chi_\sigma)_{kl} = \int_{\Omega_\tau} \int_{\Omega_\sigma} \left(\frac{\partial}{\partial n_y} - i\alpha \right) G_\kappa(x,y) \phi_l(x) \phi_k(y) \mathrm{d}\Gamma_y \mathrm{d}\Gamma_x, \quad \text{if } k \in \hat{\tau} \text{ and } l \in \hat{\sigma}.$$

Using the separable expansion given in Theorem 4.7 we can now, following the description given in Section 3.2, construct the $\mathcal{H}^2$-matrix approximant.

Definition 4.11 *If $\tau \in \mathcal{T}_\mathcal{J}$ is a leaf cluster, given an odd number $L_\tau \in \mathbb{N}$, define the corresponding row cluster basis U_τ and column cluster basis V_τ by*

$$(U_\tau)_{kj} = \begin{cases} \int_{\Omega_\tau} \bar{g}_j(\kappa(x - c_\tau)) \phi_k(x) \mathrm{d}\Gamma_x, & \text{if } k \in \tau \text{ and } j = 1, \dots L_\tau, \\ 0, & \text{if } k \notin \tau, \end{cases} \tag{4.10}$$

and

$$(V_\tau)_{kj} = \begin{cases} \int_{\Omega_\tau} (\frac{\partial}{\partial n_x} - i\alpha) g_j(\kappa(x - c_\tau)) \phi_k(x) \mathrm{d}\Gamma_x, & \text{if } k \in \tau \text{ and } j = 1, \dots L_\tau, \\ 0, & \text{if } k \notin \tau. \end{cases} \tag{4.11}$$

Note that we have only made the definition applicable to leaf clusters. The reason behind this is that if we had used the same definition for the parent clusters, the nestedness condition (see Definition 3.9) could only be satisfied approximately. Instead, we first define the transfer matrices using Theorem 4.9 and then use (3.11) as a definition of cluster bases for parent clusters.

Definition 4.12 *Let $\tau' \in \mathcal{T}_\mathcal{J}$ be a child cluster with parent cluster τ and let odd numbers $L_\tau = 2M_\tau + 1$ and $L_{\tau'} = 2M_{\tau'} + 1$ be given. Then define the corresponding transfer matrix $T_{\tau'}^V$ for the column cluster basis by*

$$T_{\tau'}^V := \left(F_{M_\tau} D_{M_\tau} F_{M_\tau}^{-1} P_{M_\tau, M_{\tau'}} \right)^\top.$$

The transfer matrices for U are the conjugates of the transfer matrices for V:

$$T_{\tau'}^U := \overline{T_{\tau'}^V}.$$

Now we can recursively define the cluster bases for parent nodes.

Definition 4.13 *If $\tau \in \mathcal{T}_\mathcal{J}$ is a parent cluster with child clusters τ and τ', define the corresponding row U_τ and column V_τ cluster basis matrices by*

$$U_\tau := U_{\tau'} T_{\tau'}^U + U_{\tau''} T_{\tau''}^U, \quad V_\tau := V_{\tau'} T_{\tau'}^V + V_{\tau''} T_{\tau''}^V.$$

Finally we define the coefficient matrices S.

Remark 4.14 *For a parent cluster τ, let $\tilde{U}_\tau$ be the matrix defined by (4.10). Then, $U_\tau \approx \tilde{U}_\tau$ where U_τ is defined in Definition 4.13. The error can be controlled using Theorem 4.9.*

Definition 4.15 *Let $b = (\tau, \sigma) \in \mathcal{L}_1^+$ and let $L_\tau = 2M_\tau + 1$, $L_\sigma = 2M_\sigma + 1$, and $L_{\tau,\sigma} = 2M_{\tau,\sigma} + 1$ be given. Then define the corresponding coefficient matrix $S_{\tau,\sigma} \in \mathbb{C}^{L_\tau \times L_\sigma}$ by*

$$S_{\tau,\sigma} := P_{M_\tau, M_{\tau,\sigma}} F_{M_{\tau,\sigma}} \tilde{S}_{\tau,\sigma} F_{M_{\tau,\sigma}}^{-1} P_{M_{\tau,\sigma}, M_\sigma},$$

where the auxiliary coefficient matrix $\tilde{S}_{\tau,\sigma} \in \mathbb{C}^{L_{\tau,\sigma} \times L_{\tau,\sigma}}$ *is a diagonal matrix with*

$$(\tilde{S}_{\tau,\sigma})_{ll} = s_l(\kappa(c_\tau - c_\sigma))$$

and $s_l(\cdot)$ is given in Theorem 4.1; see Theorem 4.7.

Remark 4.16 *The cost of constructing $\tilde{S}_{\tau,\sigma}$, using the definition of s_l directly, requires $\mathcal{O}(L^2)$ operations. However, since the diagonal of $\tilde{S}_{\tau,\sigma}$ is the discrete Fourier transform of the vector $\left(\frac{(-i)^{-M}}{L} H_{-M}(\kappa\rho^{(\tau,\sigma)}) e^{-M\theta^{(\tau,\sigma)}}, \frac{(-i)^{-M+1}}{L} H_{-M+1}(\kappa\rho^{(\tau,\sigma)}) e^{(-M+1)\theta^{(\tau,\sigma)}}, \ldots, \frac{(-i)^{M}}{L} H_{M}(\kappa\rho^{(\tau,\sigma)}) e^{M\theta^{(\tau,\sigma)}}\right)^\top$, it can be computed in $\mathcal{O}(L \log L)$ operations using FFT (see [4]).*

Remark 4.17 *Note that we are allowed to choose M_τ, M_σ, and $M_{\tau,\sigma}$ independently of each other. If we had used $\tilde{S}_{\tau,\sigma}$ as the coefficient matrices, such freedom would not have been available. In practice, we have found that the freedom to choose different lengths of expansion for the cluster bases and for the separable expansions, reduces the computational and storage requirements significantly.*

Remark 4.18 *We have only given local estimates of approximation errors. The global error estimate depends on the norms of the transfer and coefficient matrices. The entries in the coefficient matrices, as discussed in Section 4.3, can be large. The subclass of admissible blocks $\mathcal{L}_1^+$ has been constructed to control this negative effect.*

4.4.1 ACA for small admissible blocks

We have yet to say what should be done with admissible blocks in $\mathcal{L}_2^+$ for which the separable expansion becomes unstable. The simplest way of dealing with the numerical instability would be to regard block clusters in $\mathcal{L}_2^+$ in the same way as the elements of $\mathcal{L}^-$: the corresponding parts of the Galerkin matrix would not be approximated by a data sparse format but just copied as dense blocks. This would be very costly for domains with small detail, where many panels would be needed to resolve the small detail geometry and the part of the Galerkin matrix due to these panels would be large (see Remark 4.21). A simple alternative is to approximate these blocks by low rank matrices obtained using the adaptive cross approximation (ACA) algorithm.

ACA, regarded as a black-box algorithm, performs as follows: Given a function $f(l, j)$, defined for $l, j = 1, \ldots, m$, and a desired accuracy $\epsilon > 0$, it returns rank k matrices $A_k, B_k \in \mathbb{C}^{m \times k}$ such that $\|A_k B_k^\top - X\|_2 \lesssim \epsilon$, where $(X)_{lj} = f(l, j)$, i.e. it computes a rank k approximation to a matrix. To do this, the ACA evaluates the function $f(\cdot, \cdot)$ at $\mathcal{O}(mk)$ arguments and overall requires $\mathcal{O}(mk)$ storage and computational time. We have used the symbol $\lesssim$ above to indicate that the ACA does not guarantee an exact spectral error estimate, but rather a good estimate of this error.

For the case $f(l, j) = s(x_l, y_j)$, where s is an asymptotically smooth kernel, and x_l and y_j are restricted to two cluster that satisfy an admissibility condition, the ACA algorithm has been investigated theoretically in [7, 12]. For the case of the Helmholtz kernel no theory exists at the moment, however good numerical results have already been reported in [43]. Our experience is also positive, and we illustrate the ACA here with a single experiment. In fact, we repeat the experiment on the performance of the SVD, see Figure 3.1, but this time using the ACA algorithm. The results, and a comparison with the optimal SVD, are given in Figure 4.3.

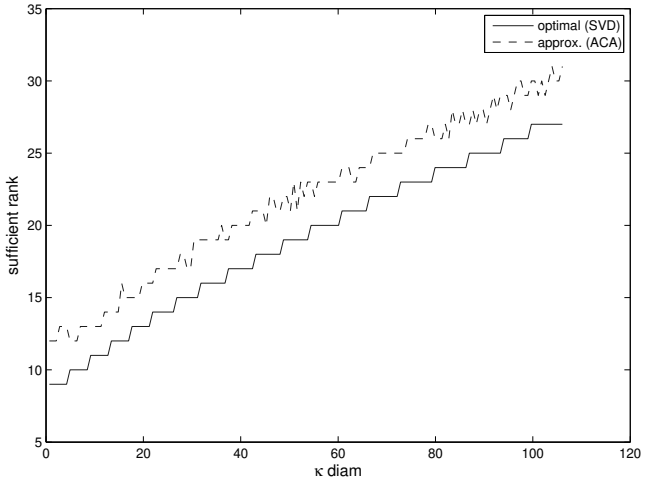


Figure 4.3: We compute low rank approximations $A_k B_k^\top$ to the matrix $\chi_\tau A \chi_\sigma$, where $(\tau, \sigma) \in \mathcal{L}^+$, by ACA and by SVD. For a range of values of κ we plot the minimum rank necessary for the two methods so that $\|\chi_\tau A \chi_\sigma - A_k B_k^\top\|_2 < 10^{-5}$.

Figure 4.3 suggests that the ACA seems to perform very well even for large frequencies. For this reason, in our implementation, we favour the use of ACA to a theoretically more sound algorithm, e.g. a low rank approximation obtained by interpolation or Taylor expansions. Error estimates for the interpolation or Taylor expansions could be obtained using the bound (2.11) on partial derivatives of the fundamental solution. Another reason for using ACA is its ease of use and implementation.

Finally we note that, as we have seen in Figure 4.3, ACA does not encounter stability problems for small clusters. This is no surprise, since this kind of instability is not a property of the Helmholtz problem, but an artifact of the multipole expansion.

4.5 Construction of a data-sparse approximation to the Galerkin matrix

We now describe the steps required to build a data sparse approximation $\hat{B}$ to the complete Galerkin matrix $B = \mathcal{I}/2 + A$. Assuming double precision computations, the new, numerically stable, algorithm is given next.

- The parameter η controlling the admissibility condition needs first to be fixed. We find that the choice $\eta = 2/3$ works well in practice.
- Given $\epsilon > 0$, choose $a > 0$ using (4.9) such that $16 - d_1 > \log_{10} \frac{1}{\epsilon}$. Note that this implies that $a\kappa = C(\eta, \epsilon)$, a constant depending on η and ϵ.
- Construct the cluster tree $\mathcal{T}_\mathcal{J}$ and the block cluster tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$.
- For each cluster $\tau \in \mathcal{T}_\mathcal{J}$ set $M_\tau = \lfloor C_1 \kappa \rho_\tau + C_1 \log \frac{1}{\epsilon} \rfloor$, for some constant $C_1 > 0$.
- For each admissible block cluster $b = (\tau, \sigma) \in \mathcal{L}_1^+$, set[1] $M_{\tau,\sigma} = \lfloor C_2 \kappa (\rho_\tau + \rho_\sigma) + C_2 \log \frac{1}{\epsilon} \rfloor$, for some constant $C_2 > 0$.

[1]For simplicity we have ignored the term $\log \kappa$ required by Theorem 4.7. In fact, in numerical experiments we always have $\log \frac{1}{\epsilon} > \log \kappa$.

- For each leaf τ construct the row and cluster bases U_τ and V_τ.
- For each child cluster τ' construct $T_{\tau'}$.
- For each $b = (\tau, \sigma) \in \mathcal{L}_1^+$ construct the auxiliary diagonal coefficient matrix $\tilde{S}_{\tau,\sigma}$. Also define, but do not compute, $\chi_\tau \hat{B} \chi_\sigma := U_\tau S_{\tau,\sigma} V_\sigma^\top$.
- For each $b = (\tau, \sigma) \in \mathcal{L}_2^+$ construct a low rank approximation $\chi_\tau \hat{B} \chi_\sigma$ to $\chi_\tau B \chi_\sigma$ using ACA.
- For each inadmissible leaf $b = (\tau, \sigma) \in \mathcal{L}^-$ leave the data unperturbed: $\chi_\tau \hat{B} \chi_\sigma = \chi_\tau B \chi_\sigma$.

The numerical instability issues have been investigated by a number of authors. In [46], the authors use an alternative separable expansion that can be stabilized by scaling. To do that, however, one must sacrifice the Toeplitz structure initially present. Also the rank obtained using this expansion is much larger than the one obtained using ACA, since ACA produces results close to the optimal result of the SVD. An altogether different approach using so called "exponential expansions" has been developed by Greengard et al. [27] and Darve [20, 21]. Here an integral representation of the fundamental solution is used,

$$\frac{i}{4} H_0(\kappa \sqrt{x^2 + y^2}) = \frac{i}{4\pi} \int_{-\infty}^{\infty} \frac{e^{i\lambda x} e^{-\sqrt{\lambda^2 - \kappa^2} y}}{\sqrt{\kappa^2 - \lambda^2}} d\lambda,$$

valid for $y > 0$ (see [27]). An equivalent expression can be given in 3D as well which is the only case covered by [20, 21, 27]. In fact this approach is most useful in 3D, where it helps to speed-up the cost of the translation operators from $189p^4$ to $40p^2 + 6p^3$, where p is the length of expansion used in the low frequency (see [27]). In two dimensions the advantages are likely to be more modest; for the Laplace case a reduction from $27p^2/2$ to $8p^2 + 27p$ is obtained (see [36]).

The advantage of our method is in its simplicity and its effectiveness as will be demonstrated by numerical examples. Furthermore, the $\mathcal{H}$-matrix part $\hat{B}_\mathcal{H}$ of the matrix $\hat{B}$, defined by

$$\hat{B}_\mathcal{H} := \sum_{(\tau,\sigma) \in \mathcal{L}^- \cup \mathcal{L}_2^+} \chi_\tau \hat{B} \chi_\sigma, \tag{4.12}$$

can be coarsened and compressed and also used as a preconditioner. We elaborate on these issues in the next section. Let us just note that the matrix $\hat{B}_{\mathcal{H}^2} := \hat{B} - \hat{B}_\mathcal{H}$ is an $\mathcal{H}^2$-matrix. Hence our approximation really is a sum of an $\mathcal{H}$-matrix and an $\mathcal{H}^2$-matrix.

4.6 Recompression and preconditioning

The storage requirements of the coefficient matrices and transfer matrices are, due to their simple structure, low. Since the Fourier matrices are never constructed, but their action computed by FFT, for each coefficient or transfer matrix only one or two diagonal matrices need to be stored. The storage cost for the cluster bases U and V is also not large since they only need to be stored for leaf clusters. The main storage cost is due to the $\mathcal{H}$-matrix $\hat{B}_\mathcal{H}$, see (4.12). The recompression techniques developed in [25] can be applied to this matrix. We give here a brief description, but for details refer the reader to [25].

The recompression consists of two steps. As mentioned before, the ACA does not compute the optimal low rank matrix. To close this gap, the SVD is applied to each admissible block of the matrix $\hat{B}_\mathcal{H}$. This can be done efficiently since the SVD of a rank k matrix already given in a factorized form $M = A_k B_k^\top \in \mathbb{C}^{m \times n}$, $A_k \in \mathbb{C}^{m \times k}$ and $B_k \in \mathbb{C}^{n \times k}$, can be computed in $\mathcal{O}(k^2(m+n))$ operations (see [26]). The second recompression optimizes the block structure making it coarser. In this

n	κ	Total time (s)	Recomp. time (s)	Mem. (MB)	Mem. recomp. (MB)
2^{10}	2^5	2.72	0.35	4	2
2^{11}	2^6	6.39	0.88	9	6
2^{12}	2^7	9.29	0.90	16	9
2^{13}	2^8	22.77	2.29	35	21
2^{14}	2^9	45.21	3.97	72	42
2^{15}	2^{10}	91.75	8.39	152	92
2^{16}	2^{11}	192.6	17.4	318	188

Table 4.1: In this table we display the total time for the construction of $\hat{B}_{\mathcal{H}^2}$ and $\hat{B}_{\mathcal{H}}$, the time for the recompression and coarsening, and the memory consumption before and after the recompression.

second step the storage is also reduced, but perhaps more importantly the coarser block structure allows for faster arithmetical operations. In particular, for preconditioning we are interested in the hierarchical LU-decomposition (see [8]). The effect of recompression on the storage costs of the Galerkin matrix $\hat{B}$ is shown in Table 4.1.

Ultimately, we wish to efficiently solve linear systems of the type $\mathbf{b} = \hat{B}\mathbf{v}$. To do this we will use iterative methods that make use only of matrix-vector products. To improve the convergence of such methods, preconditioning can be used. In [2] and [34] it is recommended to use a splitting

$$\hat{B} = \hat{B}_1 + C_1,$$

where $\hat{B}_1$ is a sparse matrix and solve the following preconditioned system instead,

$$\hat{B}_1^{-1}\hat{B}\mathbf{v} = (I + \hat{B}_1^{-1}C_1)\mathbf{v} = \hat{B}_1^{-1}\mathbf{b}.$$

In [2], $\hat{B}_1$ is chosen to be the tridiagonal band of $\hat{B}$ together with the extreme anti-diagonal corner elements $(\hat{B})_{1n}$ and $(\hat{B})_{n1}$.

We employ a similar approach, but the $\mathcal{H}$-matrix $\hat{B}_{\mathcal{H}}^{-1}$ (cf. (4.12)) will be the basis of the preconditioner. We will not compute $\hat{B}_{\mathcal{H}}^{-1}$ directly, but rather compute an $\mathcal{H}$-matrix LU-decomposition of $\hat{B}_{\mathcal{H}}$. Two triangular $\mathcal{H}$-matrices $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ can be computed efficiently such that $L_{\mathcal{H}}U_{\mathcal{H}} \approx \hat{B}_{\mathcal{H}}$. The accuracy of the LU-decomposition can be varied. Lower accuracy will allow for faster computational times (see [8, 25]). Since the LU-decomposition will only be used for preconditioning, high accuracy is not essential. The preconditioned linear system now reads:

$$(L_{\mathcal{H}}U_{\mathcal{H}})^{-1}\hat{B}\mathbf{v} = (L_{\mathcal{H}}U_{\mathcal{H}})^{-1}\mathbf{b}. \tag{4.13}$$

This system will be solved using an iterative process that at each iteration requires a multiplication of $\hat{B}$ and a vector, and the solution of two triangular systems given in $\mathcal{H}$-matrix format. The latter can be done in $\mathcal{O}(n \log n)$ time by $\mathcal{H}$-matrix equivalents of forward and backward substitutions, as described in [8].

4.7 Complexity analysis

Before we estimate the computational complexity of the construction of the matrix and the cost of matrix-vector multiplications, we make a couple of assumptions that hold in standard situations. First of all, without loss of generality, we assume that $\operatorname{diam}(\Omega) \leq 1$ and that C_{sp} is a constant.

The final assumption, pertinent to the two dimensional problem, is that there exists a constant $C_{\rm ct}$, such that for any level l

$$\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} 2\rho_\tau = \sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} \operatorname{diam}(\Omega_\tau) \le C_{\rm ct}. \tag{4.14}$$

This condition simply prevents pathological cases, such as the case where each child cluster has the same diameter as its parent cluster. A standard algorithm for the construction of the cluster tree, as described in [26], would prevent such a case from happening. In the best case, when the diameter of each child cluster is exactly half the diameter of its parent, (4.14) holds with $C_{\rm ct} = \operatorname{diam}(\Omega)$. The condition is useful since it gives the following inequality:

$$\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} M_\tau \le C_1(C_{\rm ct}\kappa + \#\mathcal{T}_{\mathcal{J}}^{(l)} \log\frac{1}{\epsilon}).$$

Also, we recall that there are at most $2n-1$ clusters in the cluster tree $\mathcal{T}_{\mathcal{J}}$. Hence for any level L,

$$\sum_{l=0}^{L} \#\mathcal{T}_{\mathcal{J}}^{(l)} \le 2n-1.$$

Now we are in a position to give estimates for the storage and the cost of construction and matrix-vector multiplication for the $\mathcal{H}^2$-matrix $\hat{B}_{\mathcal{H}^2}$.

Lemma 4.19 (storage) *If p is the depth of $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ and (4.14) holds, then there exists a constant C depending only on C_1, C_2, $C_{\rm ct}$, and $C_{\rm sp}$ such that for large enough κ ($\kappa > \max\{1, \log\frac{1}{\epsilon}\}$ is sufficient)*

$$N_{\rm st} \le C(p\kappa + n\log\frac{1}{\epsilon}) \quad \textit{and} \quad N_{\rm con} \le C(p\kappa\log\kappa + n\log\kappa\log\frac{1}{\epsilon}),$$

where $N_{\rm st}$ is the storage requirement and $N_{\rm con}$ the cost of constructing the $\mathcal{H}^2$-matrix $\hat{B}_{\mathcal{H}^2}$.

Proof. The cost of storing and constructing the row and column cluster bases for the leaf clusters is the same. It can be estimated as follows (recall Remark 3.2):

$$\begin{aligned}\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(p)}} \#\tau M_\tau &\le C_{\rm leaf} \sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(p)}} M_\tau \\ &\le C_{\rm leaf}C_1(C_{\rm ct}\ \kappa + \#\mathcal{T}_{\mathcal{J}}^{(p)}\log\frac{1}{\epsilon}) \\ &\le C_{\rm leaf}C_1(C_{\rm ct}\ \kappa + n\log\frac{1}{\epsilon}).\end{aligned}$$

The cost of storing the coefficient matrices is proportional to

$$\begin{aligned}\sum_{b=(\tau,\sigma)\in\mathcal{L}_1^+} M_{\tau,\sigma} &\le \sum_{b=(\tau,\sigma)\in\mathcal{L}_1^+} C_2\kappa(\rho_\tau+\rho_\sigma) + C_2\log(\frac{1}{\epsilon}) \\ &\le \sum_{l=0}^{p}\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} \#\{\sigma : (\tau,\sigma)\in\mathcal{L}_1^+ \text{ or } (\sigma,\tau)\in\mathcal{L}_1^+\}(C_2\kappa\rho_\tau + C_2\log(\frac{1}{\epsilon})) \\ &\le \sum_{l=0}^{p}\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} C_{\rm sp}C_2(\kappa\rho_\tau + \log(\frac{1}{\epsilon})) \le C_{\rm sp}C_2\sum_{l=0}^{p}(C_{\rm ct}\ \kappa + \#\mathcal{T}_{\mathcal{J}}^{(l)}\log\frac{1}{\epsilon}) \\ &\le C_{\rm sp}C_2(C_{\rm ct}\kappa(p+1) + (2n-1)\log\frac{1}{\epsilon}).\end{aligned}$$

Since for each coefficient matrix we require a single application of FFT, the cost of the construction is larger than the storage cost by a logarithmic factor:

$$\begin{aligned}\log M_{\tau,\sigma} &\leq \log C_2 + \log(\kappa(\rho_\tau + \rho_\sigma) + \log\frac{1}{\epsilon}) \\ &\leq \log C_2 + \log(\kappa(\rho_\tau + \rho_\sigma + 1)) \leq \log C_2 + \log 2\kappa.\end{aligned}$$

So the total cost is increased by a multiplicative factor of $\mathcal{O}(\log\kappa)$:

$$C_{\text{sp}}C_2(C_{\text{ct}}\ \kappa(p+1) + (2n-1)\log\frac{1}{\epsilon})\log(2C_2\kappa).$$

The cost of the construction and the storage of transfer matrices can be estimated as follows:

$$\sum_{l=0}^{p-1}\sum_{\tau\in\mathcal{T}_{\mathcal{J}}^{(l)}} M_\tau \leq C_1(C_{\text{ct}}\ \kappa(p+1) + (2n-1)\log\frac{1}{\epsilon}).$$

■

Lemma 4.20 (multiplication) *Under the same conditions as in the previous lemma there exists a constant C such that*

$$N_{\mathcal{H}\cdot v} \leq C N_{\text{con}},$$

where $N_{\mathcal{H}\cdot v}$ is the cost of matrix-vector multiplication for $\hat{B}_{\mathcal{H}^2}$.

Proof. We compute the cost of matrix-vector multiplication following the steps of the fast algorithm explained in Section 3.2.1. The reasoning is the same as in the proof of the previous lemma.

1. *Upward pass:*
   (a) The cost of applying the cluster bases to a vector for the leaves is of the same order as the cost of constructing them. Hence by the proof of Lemma 4.19 the total cost for all leaf clusters is $\mathcal{O}(\kappa + n\log\frac{1}{\epsilon})$.
   (b) The cost of applying the transform matrices to a vector is larger than the cost of constructing them since applications of FFT are necessary. The further logarithmic factor gives the complexity $\mathcal{O}(p\kappa\log\kappa + n\log\kappa\log\frac{1}{\epsilon})$.
2. *Far field interaction:* The cost of multiplication is the same as the cost of constructing the coefficient matrices since in both cases FFT is used. Hence the cost is $\mathcal{O}(p\kappa\log\kappa + n\log\kappa\log\frac{1}{\epsilon})$
3. *Downward pass:* Same cost as in 1b.
4. *Near field interaction:* The near field of the $\mathcal{H}^2$-matrix $\hat{B}_{\mathcal{H}^2}$ is in fact zero. So there is no cost.

Combining the above estimates gives the result. ■

Since we are particularly interested in the high frequency regime, i.e., $\kappa \propto n$, assuming $p = \mathcal{O}(\log n)$ and ϵ a constant, we have that the cost of storage is $\mathcal{O}(n\log n)$ and the cost of construction and matrix-vector multiplication is $\mathcal{O}(n\log^2 n)$. However, in practical situations κ is considerably smaller than n so that we expect the costs to behave closer to $\mathcal{O}(n)$ and $\mathcal{O}(n\log n)$ for the storage and matrix-vector complexity respectively. We complete this section with remarks about the costs associated with the $\mathcal{H}$-matrix $\hat{B}_{\mathcal{H}}$.

Remark 4.21 *Note that by definition, for* $b = (\tau, \sigma) \in \mathcal{L}_2^+$, $\kappa(\rho_\tau + \rho_\sigma) \leq a\kappa = C(\eta, \epsilon)$. *Hence, the length of expansion required by Theorem 4.3, is proportional to* $C(\eta, \epsilon)$ *and independent of* κ. *Assuming that the ACA recovers this behaviour (in fact, in practice, ACA gives a much lower rank than the separable expansion would produce) we have that*

$$k_{\max} := \max_{b=(\tau,\sigma)\in\mathcal{L}_2^+} rank(\chi_\tau \hat{B} \chi_\sigma) \leq C(\eta, \epsilon),$$

where $C(\eta, \epsilon)$ *is a generic constant depending only on* η *and* ϵ. *Hence, using Lemma 3.7, we have that the cost of construction and matrix vector multiplication is* $\mathcal{O}(npk_{\max})$, *where* $k_{\max}$ *is independent of* κ. *Therefore the costs associated with the* $\mathcal{H}$*-matrix part are not asymptotically larger than the costs associated with the* $\mathcal{H}^2$*-matrix. Note that such an estimate is not possible without the use of some data sparse representation for these blocks, since a small size of* $\operatorname{diam}(\Omega_\tau)$ *does not imply a small cardinality* $\#\hat{\tau}$.

5 Numerical results

In this section we demonstrate how our algorithm behaves in practice through numerical examples. We do this by considering the exterior Helmholtz problem (2.1) with the boundary data $F(x) = -e^{i\kappa x.d}$, where $d = (\cos\pi/4, \sin\pi/4)^\top$. This problem describes the time harmonic acoustic scattering problem, where a plane wave coming from infinity at an angle $\pi/4$, is being scattered by a sound-soft obstacle $\Omega \subset \mathbb{R}^2$ (see [17, 38]). The solution we seek is the scattered wave. We give results for two different obstacles. First of all, we solve the problem for the case of the unit disk for which an analytic solution can be obtained through the Mie series. The second scatterer we investigate is the inverted ellipse, which is the smooth, non-convex shape shown in Figure 5.1 and defined by the following mapping:

$$\gamma(t) = \sqrt{1 - .99\cos(t)^2}(-\sin(t), \cos(t))^\top : [0, 2\pi) \to \Gamma. \tag{5.1}$$

We give results of experiments for both the low frequency and the high frequency regimes. We have used the iterative solver GMRES to solve the arising linear systems. To speed up the convergence of the solver, we have used the preconditioner described in Section 4.6. All the computations were done on a 2.8 GHz Pentium IV processor. In all of the computations we have chosen the coupling parameter $\alpha = \kappa$ as suggested by [2] and [23].

5.1 The low frequency regime

For the low frequency regime we fix $\kappa = 64$ and increase the number of panels n. To approximate the Galerkin matrix we use the $\mathcal{H}$-matrix obtained by ACA. We have used a low-accuracy LU-decomposition of the whole Galerkin matrix, as the preconditioner.

The results for the case of scattering by the unit disk are shown in Table 5.1. For this problem the exact solution u and the boundary density φ are known. Apart from the L^2-error on the boundary: $\|\varphi - \hat{\varphi}\|_{L^2(\Gamma)}$, we also consider a measure of the error outside the domain. This error is estimated by computing the approximate solution u_j at points $x_j \in \Omega^c$, $j = 1, 2, \ldots, 100$. The points x_j are chosen to be equally spaced on the disk of radius 1.2. As the measure of the error we use the average:

$$\text{error} = \sum_{j=1}^{100} |u(x_j) - u_j|/100.$$

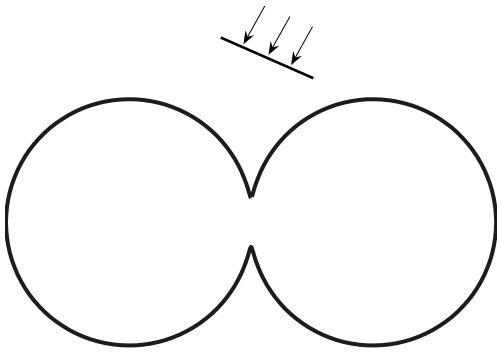


Figure 5.1: A non-convex (but smooth) obstacle and a plane wave coming from infinity.

n	Setup(s)	Solve(s)	Mem.(MB)	Mem./n(kB)	iter.	$\|\varphi - \hat{\varphi}\|_{L^2(\Gamma)}$	err.
2^9	1.79	.14	1.9	3.9	5/17	2.1×10^{-1}	5.8×10^{-3}
2^{10}	3.9	.26	3.8	3.8	6/21	1.1×10^{-1}	1.4×10^{-3}
2^{11}	8.5	.52	7.2	3.6	7/24	5.4×10^{-2}	3.5×10^{-4}
2^{12}	22.4	1.76	15.6	3.9	9/28	2.7×10^{-2}	8.8×10^{-5}
2^{13}	51.5	4.02	34.4	4.3	10/31	1.3×10^{-2}	2.4×10^{-5}
2^{14}	98.5	6.5	76.8	4.8	11/33	6.7×10^{-3}	7.6×10^{-6}

Table 5.1: CPU times and memory consumption in the low frequency regime with $\kappa = 64$. Columns 2 to 7 give the following information: time to construct the matrices (including coarsening), time to construct the preconditioner and solve the linear system, total memory requirement, total memory per degree of freedom, the number of iterations with and without the preconditioner, and the error.

By inspecting Table 5.1, we can see that the convergence is of $\mathcal{O}(n^{-1})$ for the error on the boundary and $\mathcal{O}(n^{-2})$ for the error outside the obstacle. The higher order convergence outside the boundary can be explained by the Aubin-Nitsche duality technique and the higher regularity of the solution in the exterior (see [14, §5.12] [42, §4.2.5]). Note however that going from $n = 2^{13}$ to $n = 2^{14}$ the ratio of the error outside the boundary is not exactly 4, which is what one would expect for $\mathcal{O}(n^{-2})$ convergence. The reason behind this goes deep in to the implementation issues. Namely, for quadrature we use spectrally accurate Gaussian quadrature, so that for all these examples we use $q = 2$ quadrature points per element in one dimension, therefore for the double integrals we use q^2 quadrature points. At the final stage, $n = 2^{14}$ the errors in the quadrature are starting to be seen. To see a perfect $\mathcal{O}(n^{-2})$ convergence we would have to increase q to 3. This would increase the computational time for the construction of the matrix at the stage $n = 2^{14}$, $(3/2)^2 \approx 2.3$ times. Since the convergence of the quadrature routines we use are exponential, the choice $q = 3$ would suffice for much larger n than 2^{14}. To illustrate this issue we perform a further computation with $n = 2^{14}$ and $q = 3$ and obtain the following results:

n	Setup(s)	Solve(s)	Mem.(MB)	Mem./n(kB)	iter.	$\|\varphi - \hat{\varphi}\|_{L^2(\Gamma)}$	err.
2^{14}	186.2	8.2	76.8	4.8	11/33	6.7×10^{-3}	5.6×10^{-6}

n	κ	Setup(s)	Solve(s)	Mem. (MB)	Mem./n(kB)	iter.	$\|\varphi - \hat{\varphi}\|_{L^2(\Gamma)}$	err.
2^{10}	2^5	2.72	.19	2.8	2.8	5/18	5.5×10^{-2}	4.1×10^{-4}
2^{11}	2^6	6.44	.46	6.6	3.3	6/22	5.4×10^{-2}	3.5×10^{-4}
2^{12}	2^7	9.27	1.33	10.0	2.5	8/27	5.3×10^{-2}	3.3×10^{-4}
2^{13}	2^8	22.6	3.2	21.6	2.7	10/32	5.2×10^{-2}	3.4×10^{-4}
2^{14}	2^9	44.8	9.6	43.2	2.7	14/38	5.2×10^{-2}	3.4×10^{-4}
2^{15}	2^{10}	91.0	25.5	92.8	2.9	18/46	5.2×10^{-2}	3.8×10^{-4}
2^{16}	2^{11}	196.2	62.0	192.0	3.0	19/56	5.2×10^{-2}	3.7×10^{-4}

Table 5.2: CPU times and memory consumption in the high frequency regime for scattering by the unit disk.

As expected, both the computational times and memory consumption scale almost linearly. Preconditioning reduces the number of iterations significantly. The number of iterations does increase with n, however only slowly.

5.2 The high frequency regime

For the high frequency regime we increase both n and κ, keeping $n/\kappa = const$. We apply the mixed format of an $\mathcal{H}^2$-matrix with low-rank matrices obtained by ACA as described in Section 4.4. The results for the case of the unit disk obstacle are shown in Table 5.2. The error is measured as for the low-frequency case. Note also, that the error stays approximately constant. Again, the preconditioner reduces the number of iterations significantly. Still, a slow increase of the number of iterations, as κ is increased, is noticeable.

We perform the same experiment, but this time with the inverted ellipse as the obstacle. The inverted ellipse is scaled so as to be contained just inside the unit disk; see (5.1). Since for this problem the analytical solution is not known, to estimate the error we compute a more accurate approximation (with n approximately doubled) and use it as the exact solution. The results are shown in Table 5.3. We see that the more complicated domain has no significant adverse effect. The cost of constructing the matrices has increased by a small amount, as well as the memory consumption. The number of iterations for the solution of the linear systems has not shown a clear increase, compared to the case of the unit disk. This suggests that the preconditioner has accounted for the more difficult geometry. Note that the number of iterations needed when no preconditioning is used, is considerably higher than in the case of the unit disk. In the last two computations, we have interrupted the solver at the 80th iteration.

A Proofs of lemmata

Lemma A.1 *Let $r, \rho > 0$ and let $n, m \in \mathbb{Z}$ with $|m| + 1 > \rho$. Then*

$$|J_n(r)| \le e^{r \sinh a - a|n|} \qquad \text{for any } a > 0, \tag{A.1a}$$

$$\text{and } |H_m(\rho)| \le \sqrt{3/2} + \frac{2}{\pi} e^{-\rho \sinh \delta + \delta(|m|+1)}, \quad \delta = \operatorname{arcosh}\left((|m|+1)/\rho\right). \tag{A.1b}$$

Also,

$$\left| J_n(r) - \frac{(-i)^n}{L} \sum_{l=1}^{L} e^{ir\cos(\frac{2\pi l}{L})} e^{\frac{2\pi i l n}{L}} \right| \le 4\pi \frac{e^{r\sinh a - (L-n)a}}{1 - e^{-La}} \qquad \text{for any } L \in \mathbb{N} \text{ and } a > 0. \tag{A.2}$$

n	κ	Setup (s)	Solve (s)	Mem.(MB)	Mem./n(kB)	iter.	error
2^{10}	2^5	3.88	0.23	2.8	2.8	12/30	6.9×10^{-5}
2^{11}	2^6	9.35	0.59	6.8	3.4	14/37	4.8×10^{-5}
2^{12}	2^7	20.1	1.35	17.2	4.3	14/48	3.9×10^{-5}
2^{13}	2^8	37.7	4.31	34.4	4.3	13/59	6.1×10^{-5}
2^{14}	2^9	77.7	7.1	68.8	4.3	13/78	7.4×10^{-5}
2^{15}	2^{10}	134.9	16.1	118.4	3.7	13/80+	6.7×10^{-5}
2^{16}	2^{11}	248.0	50.3	211.2	3.3	20/80+	6.4×10^{-5}

Table 5.3: CPU times and memory consumption in the high frequency regime for scattering by the inverted ellipse.

Proof. Since $|J_n(r)| = |J_{-n}(r)|$ and $|H_m(\rho)| = |H_{-m}(\rho)|$, without loss of generality we can assume that $m, n \geq 0$.

For a fixed r, $c_n := i^n J_n(r)$ is the nth Fourier coefficient of the complex analytic function $f(z) := e^{ir\cos z}$, see (4.5). For any $a > 0$, f is analytic in the horizontal strip $|\mathrm{Im}\, z| < a$ and hence $g(w) := f(\frac{1}{i}\log w)$ is analytic in the annulus $e^{-a} < |w| < e^a$. The Fourier coefficients of f are just the Laurent coefficients of g. These can be bounded by Cauchy's estimate, see [35], giving

$$|c_n| \leq \max_{e^{-a}<|w|<e^a} |g(w)|e^{-an} = \max_{|\mathrm{Im}|\, z<a} |f(z)|e^{-an}.$$

Since $\max_{|\mathrm{Im}\, z|<a} |f(z)| = \max_{|\mathrm{Im}\, z|=a} |f(z)| \leq e^{r\sinh a}$,

$$|c_n| \leq e^{r\sinh a - an} \quad \text{for any } a > 0.$$

This finishes the proof of (A.1a).

To obtain the bound in (A.1b), we use the integral representation of $H_m(\cdot)$,

$$H_m(\rho) = J_m(\rho) + \frac{i}{\pi}\int_0^\pi \sin(\rho\sin\theta - m\theta)\mathrm{d}\theta - \frac{i}{\pi}\int_0^\infty (e^{mt} + (-1)^m e^{-mt})e^{-\rho\sinh t}\mathrm{d}t,$$

which can be found in [24]. Since, according to [1, 9.1.60] $J_m(\rho) \leq \sqrt{1/2}$, and $|\frac{1}{\pi}\int_0^\pi \sin(\rho\sin\theta - m\theta)\mathrm{d}\theta| \leq \frac{1}{\pi}\int_0^\pi \mathrm{d}\theta = 1$, we have that

$$|H_m(\rho)| \leq \sqrt{3/2} + \frac{2}{\pi}\int_0^\infty e^{mt-\rho\sinh t}\mathrm{d}t.$$

By inspecting the derivative with respect to t of the function $e^{(m+1)t-\rho\sinh t}$, we find that $e^{(m+1)t-\rho\sinh t} \leq e^{(m+1)\delta-\rho\sinh\delta}$ for $\delta = \mathrm{arcosh}((m+1)/\rho)$ and any $t > 0$. Hence,

$$\frac{2}{\pi}\int_0^\infty e^{mt-\rho\sinh t}\mathrm{d}t \leq \frac{2}{\pi}\, e^{(m+1)\delta-\rho\sinh\delta}\int_0^\infty e^{-t}\mathrm{d}t = \frac{2}{\pi}\, e^{(m+1)\delta-\rho\sinh\delta}.$$

With this, the proof of the second inequality (A.1b) is finished.

The quantity that we want to bound in (A.2) is the remainder of the composite trapezoidal rule for 2π-periodic functions. The periodic integrand is $f_n(\theta) := (-i)^n \exp(ir\cos\theta)\exp(-in\theta)$. Since $f_n(\cdot)$ is an entire function, the remainder is bounded by the expression

$$4\pi \max_{|\mathrm{Im}\ z|<a} |f_n(z)| \frac{e^{-La}}{1-e^{-La}} \qquad \text{for any } a > 0,$$

see [22, §4.6.5]. The proof is finished by bounding $f_n(\cdot)$:

$$\max_{|\mathrm{Im}\ z|<a} |f_n(z)| \leq e^{r\sinh a + na}.$$

■

Next we give the proof of Lemma 4.2:

Proof. Let us first prove the easier, first inequality. From (A.1a) we have that

$$\sum_{n=M}^{\infty} |J_n(r)| \leq e^{r\sinh a} \sum_{n=M}^{\infty} e^{-an} = \frac{e^{r\sinh a - aM}}{1-e^{-a}} \qquad \text{for arbitrary } a>0.$$

From this expression the required result is easily deduced. For example choose $a=1$; then

$$\sum_{n=M}^{\infty} |J_n(r)| \leq \frac{e^{r\sinh 1 - M}}{1-e^{-1}} \leq \epsilon \qquad \text{for } M > r\sinh 1 + \log(\frac{1}{\epsilon}) - \log(1-e^{-1}).$$

Since, using the assumption $\epsilon < 1/2$, $r\sinh 1 + 2\log(\frac{1}{\epsilon}) > r\sinh 1 + \log(\frac{1}{\epsilon}) - \log(1-e^{-1})$, we have that for the first inequality it suffices to choose $C(\eta) \geq 2$.

Let us turn to the second inequality. For n such that $\rho < n+2 \leq 2\rho$ we can employ (A.1b) to obtain that $|H_{n+1}(\rho)| \leq \sqrt{3/2} + \exp\{-\rho\sinh(\mathrm{arcosh}((n+2)/\rho)) + \mathrm{arcosh}((n+2)/\rho)(n+2)\}$. Since the functions sinh and arcosh are increasing we have the following bound,

$$|H_{n+1}(\rho)| \leq \sqrt{\tfrac{3}{2}} + e^{\rho(-\sinh(\mathrm{arcosh}(1)) + 2\,\mathrm{arcosh}(2))} \leq \sqrt{\tfrac{3}{2}} + e^{3\rho} \leq \sqrt{\tfrac{3}{2}} + e^{3(n+2)}, \quad \text{for } \rho < n+1 \leq 2\rho. \tag{A.3}$$

Since for $1 \leq \nu \leq x$, $|H_\nu(x)| \leq 1$, the above bound is valid for all n such that $2 \leq n+2 \leq 2\rho$. Let us first, consider the case $M+2 \leq 2\rho$ and define $M_1 := 2\lfloor \rho - 2 \rfloor$. Then, making use of the inequalities (A.3) and (A.1a), we have that

$$\begin{aligned}
\sum_{n=M}^{\infty} |H_{n+1}(\rho)J_n(r)| &\leq \sum_{n=M}^{M_1} |H_{n+1}(\rho)J_n(r)| + \sum_{n=M_1+1}^{\infty} |H_{n+1}(\rho)J_n(r)| \\
&\leq \sqrt{3/2} \sum_{n=M}^{M_1} |J_n(r)| + \sum_{n=M}^{M_1} e^{r\sinh a - (a-3)n + 6} + \sum_{n=M_1+1}^{\infty} |H_{n+1}(\rho)J_n(r)| \\
&\leq \sqrt{3/2} \sum_{n=M}^{\infty} |J_n(r)| + e^6 \sum_{n=M}^{\infty} e^{r\sinh a - (a-3)n} + \sum_{n=M_1+1}^{\infty} |H_{n+1}(\rho)J_n(r)|.
\end{aligned}$$

We already know how to deal with the first sum. The second can be dealt with in a similar way by choosing $a > 3$. Hence, without loss of generality in the remainder of the proof we will assume that $M+2 > 2\rho$.

From (A.1a) and (A.1b) we have that, for an arbitrary $a > 0$,

$$|H_{n+1}(\rho)J_n(r)| \leq \sqrt{\frac{3}{2}}|J_n(r)| + e^{r_{\max}\ \sinh a - an - \rho\sinh\delta_n + \delta_n(n+2)}, \quad \delta_n = \mathrm{arcosh}((n+2)/\rho).$$

With the choice $a = \gamma_n := \mathrm{arsinh}(\frac{\rho}{r_{\max}}\sinh\delta_n)$, the above expression becomes

$$|H_{n+1}(\rho)J_n(r)| \leq \sqrt{\frac{3}{2}}|J_n(r)| + e^{-n(\gamma_n - \delta_n) + 2\delta_n}.$$

We recall that $\operatorname{arsinh}(x) = \log(x + \sqrt{x^2+1})$ and $\operatorname{arcosh}(x) = \log(x + \sqrt{x^2-1})$, for $x > 1$, and are hence increasing functions of x. Therefore δ_n is an increasing sequence. Together with $\rho/r_{\max} = \frac{1}{\eta} > 1$, this implies that $\gamma_n > \delta_n$ for all n. Further, the function $h(x) := \operatorname{arsinh}(\frac{\rho}{r_{\max}} \sinh x) - x$ is an increasing function since

$$h'(x) = \frac{\frac{\rho}{r_{\max}} \cosh x}{\sqrt{1 + \frac{\rho^2}{r_{\max}^2} \sinh^2 x}} - 1 > 0, \qquad \text{for } x > 1.$$

Therefore, $\gamma_n - \delta_n = h(\delta_n)$ is a positive, monotonically increasing sequence. Since $\frac{\rho}{r_{\max}} = 1/\eta$ and $(n+2)/\rho \geq (M+2)/\rho > 2$, we have that $\gamma_n - \delta_n \geq \beta$, where $\beta := \operatorname{arsinh}(\frac{1}{\eta} \sinh(\operatorname{arcosh}(2))) - \operatorname{arcosh}(2) > 0$. Hence, we obtain the following estimate:

$$\sum_{n=M}^{\infty} |H_{n+1}(\rho) J_n(r)| \leq \sqrt{\frac{3}{2}} \sum_{n=M}^{\infty} |J_n(r)| + \sum_{n=M}^{\infty} e^{-\beta n + 2\delta_n}.$$

The first sum we have already dealt with. We concentrate now on the second sum. Note that,

$$e^{2\delta_n} = e^{2 \operatorname{arcosh}((n+2)/\rho)} = \left(\frac{n+2}{\rho}\right)^2 \left(1 + \sqrt{1 - \frac{\rho^2}{(n+2)^2}}\right)^2 \leq 4 \left(\frac{n+2}{\rho}\right)^2.$$

Hence, for some constant $C > 0$,

$$\begin{aligned}
\sum_{n=M}^{\infty} e^{-\beta n + 2\delta_n} &\leq \frac{4}{\rho^2} \sum_{n=M}^{\infty} (n+2)^2 e^{-\beta n} \\
&= \frac{4}{\rho^2} \frac{e^{-\beta M}}{(1-e^{-\beta})^3} (4 + M^2(e^{-\beta} - 1)^2 + 2M(2 - 3e^{-\beta} + e^{-2\beta}) - 3e^{-\beta} + e^{-2\beta}) \\
&\leq C \left(\frac{M}{\rho}\right)^2 \frac{e^{-\beta M}}{(1-e^{-\beta})^3} \leq C \left(\frac{M\eta}{r}\right)^2 \frac{e^{-\beta M}}{(1-e^{-\beta})^3}.
\end{aligned}$$

Since the bound depends exponentially on M and further only mildly on η and r, the proof is finished. ∎

We conclude with the proof of Proposition 4.6.

Proof. Let us first consider the case $M_1 = M_2$. Then what we need to prove reduces to showing that a $C > 0$ exists such that

$$\|\mathbf{f}_{M_2}(x) - F_{M_2}^{-1} \mathbf{g}_{M_2}(x)\|_\infty < \epsilon,$$

for all $M_2 \geq C(\rho_x + \log \frac{1}{\epsilon})$. Since $\|F_{M_2}^{-1}\|_\infty = 2M_2 + 1$, the above inequality is implied by the following:

$$\|\mathbf{g}_{M_2}(x) - F_{M_2} \mathbf{f}_{M_2}(x)\|_\infty < (2M_2 + 1)\epsilon.$$

Now recall that,

$$i^n J_n(\rho_x) e^{in\theta_x} = \frac{1}{2\pi} \int_0^{2\pi} e^{i\rho_x \cos\theta} e^{in(\theta+\theta_x)} \mathrm{d}\theta = \frac{1}{2\pi} \int_0^{2\pi} e^{i\rho_x \cos(\theta-\theta_x)} e^{in\theta} \mathrm{d}\theta.$$

We can therefore proceed by approximating the integral with the composite trapezoidal rule and use (A.2) to bound the error. The rest of the proof is very similar to the proof of the first inequality in Lemma 4.2.

If $M_1 > M_2$, zeros first need to be appended to the vector $\mathbf{g}_{M_2}(x)$ to get an approximation to the vector $\mathbf{g}_{M_1}(x)$. The error in this approximation also decreases exponentially with $M_2 > \rho_x$, since Bessel functions $J_n(r)$ decrease exponentially for $n > r$ (see (A.1a)). Therefore, the case $M_1 > M_2$ can be dealt with by a triangle inequality. ∎

References

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. Dover Publications Inc., New York, 1992.

[2] S. Amini and N. D. Maines. Preconditioned Krylov subspace methods for boundary element solution of the Helmholtz equation. *Internat. J. Numer. Methods Engrg.*, 41(5):875–898, 1998.

[3] S. Amini and A. Profit. Analysis of the truncation errors in the fast multipole method for scattering problems. In *Proceedings of the 8th International Congress on Computational and Applied Mathematics, ICCAM-98 (Leuven)*, volume 115, pages 23–33, 2000.

[4] S. Amini and A. Profit. Multi-level fast multipole solution of the scattering problem. *Engineering Analysis with Boundary Elements*, 27(5):547–564, 2003.

[5] S. Amini and A. T. J. Profit. Analysis of a diagonal form of the fast multipole algorithm for scattering theory. *BIT*, 39(4):585–602, 1999.

[6] L. Banjai and S. Sauter. A refined Galerkin error and stability analysis for highly indefinite variational problems. *to appear in SIAM J. Numer. Anal.*

[7] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[8] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74(3):225–247, 2005.

[9] S. Börm. $\mathcal{H}^2$-matrices—multilevel methods for the approximation of integral operators. *Comput. Vis. Sci.*, 7(3-4):173–181, 2004.

[10] S. Börm. $\mathcal{H}^2$-matrix arithmetics in linear complexity. *Computing*, 77(1):1–28, 2006.

[11] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72(3-4):325–332, 2004.

[12] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Numer. Math.*, 101(2):221–249, 2005.

[13] H. Brakhage and P. Werner. Über das Dirichletsche Außenraumproblem für die Helmholtzsche Schwingungsgleichung. *Arch. der Math.*, 16:325–329, 1965.

[14] G. Chen and J. Zhou. *Boundary element methods*. Computational Mathematics and Applications. Academic Press Ltd., London, 1992.

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

[16] D. Colton and R. Kress. *Integral equation methods in scattering theory*. John Wiley & Sons Inc., New York, 1983.

[17] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*. Springer-Verlag, Berlin, second edition, 1998.

[18] W. Dahmen. Wavelet and multiscale methods for operator equations. In *Acta numerica, 1997*, volume 6, pages 55–228. Cambridge Univ. Press, Cambridge, 1997.

[19] E. Darve. The fast multipole method: numerical implementation. *J. Comput. Phys.*, 160(1):195–240, 2000.

[20] E. Darve and P. Havé. Efficient fast multipole method for low-frequency scattering. *J. Comput. Phys.*, 197(1):341–363, 2004.

[21] E. Darve and P. Havé. A fast multipole method for Maxwell equations stable at all frequencies. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 362(1816):603–628, 2004.

[22] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press Inc., Orlando, FL, second edition, 1984.

[23] K. Giebermann. *Schnelle Summationsverfahren zur numerischen Lösung von Integralgleichungen für Streuprobleme im* $\mathbb{R}^3$. PhD thesis, Universität Karlsruhe, 1997.

[24] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Academic Press Inc., San Diego, CA, 2000.

[25] L. Grasedyck. Adaptive recompression of $\mathcal{H}$-matrices for BEM. *Computing*, 74(3):205–223, 2005.

[26] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[27] L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura. Accelerating fast multipole methods for the Helmholtz equation at low frequencies. *IEEE Comp. Sci. Eng.*, 32(5):32–38, 1998.

[28] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

[29] W. Hackbusch. *Integral equations*. Birkhäuser Verlag, Basel, 1995.

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

[31] W. Hackbusch and S. Börm. Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices. *Computing*, 69(1):1–35, 2002.

[32] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. II. Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

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

[34] P. J. Harris and K. Chen. On efficient preconditioners for iterative solution of a Galerkin boundary element equation for the three-dimensional exterior Helmholtz problem. *J. Comput. Appl. Math.*, 156(2):303–318, 2003.

[35] P. Henrici. *Applied and computational complex analysis. Vol. 3*. John Wiley & Sons Inc., New York, 1986.

[36] T. Hrycak and V. Rokhlin. An improved fast multipole algorithm for potential fields. *SIAM J. Sci. Comput.*, 19(6):1804–1826, 1998.

[37] C. Lu and W. C. Chew. A multilevel algorithm for solving a boundary integral equation of wave scattering. *Microwave Opt. Technol. Lett.*, 7(10):466–470, 1994.

[38] J.-C. Nédélec. *Acoustic and electromagnetic equations*. Springer-Verlag, New York, 2001.

[39] S. Ohnuki and W. C. Chew. Truncation error analysis of multipole expansion. *SIAM J. Sci. Comput.*, 25(4):1293–1306, 2003/04.

[40] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comput. Phys.*, 86(2):414–439, 1990.

[41] V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Appl. Comput. Harmon. Anal.*, 1(1):82–93, 1993.

[42] S. Sauter and C. Schwab. *Randelementmethoden*. Teubner, Leipzig, 2004.

[43] M. Stolper. Computing and compression of the boundary element matrices for the Helmholtz equation. *J. Numer. Math.*, 12(1):55–75, 2004.

[44] L. N. Trefethen and D. Bau, III. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[45] G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, Cambridge, England, 1944.

[46] J.-S. Zhao and C. C. Chew. MLFMA for solving integral equations of 2-D electromagnetic problems from static to electrodynamic. *Microwave Opt. Technol. Lett.*, 20(5):306–311, 1999.