# Fast, Parallel Techniques for Time-Domain Boundary Integral Equations

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr.rer.nat.)

im Fachgebiet

Mathematik

vorgelegt

von M.Sc. Maryna Kachanovska
geboren am 21.02.1987 in Borowa, Ukraine

Die Annahme der Dissertation wurde empfohlen von:

1. Professor Dr. Dr. h.c. Wolfgang Hackbusch (Leipzig)
2. Professor Dr. Achim Schädle (Düsseldorf)

Die Verleihung des akademischen Grades erfolgt mit Bestehen
der Verteidigung am 15.01.2014 mit dem Gesamtprädikat summa cum laude

**Bibliographische Daten**

---

Fast, Parallel Techniques for Time-Domain Boundary Integral Equations
(Schnelle, parallele Methoden für Randintegralgleichungen im Zeitbereich)
Kachanovska, Maryna
Universität Leipzig, Dissertation, 2013
164 Seiten, 24 Abbildungen, 191 Referenzen

**Acknowledgements**

First of all, I would like to express my deepest gratitude to my PhD advisor, Dr. Lehel Banjai, for his guidance, support and patience. I am indebted to him for excellent scientific advice, constructive criticism, as well as a great deal of time he dedicated for correcting my scientific writing and presentation.

It is my pleasure to thank Prof. Dr. Dr. h.c. Wolfgang Hackbusch for providing me with an opportunity to work at the Max Planck Institute, as well as the International Max Planck Research School for the financial support.

I am grateful to the staff of the Max Planck Institute for ensuring fantastic working conditions. My special thanks goes to Mrs. Hünniger, Mrs. Herrmann, Mr. Achilles, Mrs. Petsch, library and computer groups, and, of course, to Mrs. Rackwitz, who was able to deal with all the difficult visa issues with almost no involvement from my side.

I am very thankful to Dr. Ronald Kriemann for patiently answering my numerous questions on $\mathcal{H}$LibPro library.

My time at the Max Planck Institute wouldn't be that pleasant without an informal support of my friends, in particular Liliya Avdiyenko and Mahboubeh Nadaf. It was nice to have discussions with Stefan Handschuh, Philipp Wähnert, Peter Meszmer, Jonas Ballani, Volker Gruhne and Ya Zhang.

My sincere thanks goes to the scientific and teaching staff of the Institute of Physics and Technology of Kyiv Polytechnic Institute, who, during my studies there, created an atmosphere of curiosity and enthusiasm for science.

Finally, I wish to thank my family and Filippo for their love and constant support without which this work wouldn't be possible.

# Abstract

This work addresses the question of the efficient numerical solution of time-domain boundary integral equations with retarded potentials arising in the problems of acoustic and electromagnetic scattering. The convolutional form of the time-domain boundary operators allows to discretize them with the help of Runge-Kutta convolution quadrature. This method combines Laplace-transform and time-stepping approaches and requires the explicit form of the fundamental solution only in the Laplace domain to be known. Recent numerical and analytical studies revealed excellent properties of Runge-Kutta convolution quadrature, e.g. high convergence order, stability, low dissipation and dispersion.

As a model problem, we consider the wave scattering in three dimensions. The convolution quadrature discretization of the indirect formulation for the three-dimensional wave equation leads to the lower triangular Toeplitz system of equations. Each entry of this system is a boundary integral operator with a kernel defined by convolution quadrature. In this work we develop an efficient method of almost linear complexity for the solution of this system based on the existing recursive algorithm. The latter requires the construction of many discretizations of the Helmholtz boundary single layer operator for a wide range of complex wavenumbers. This leads to two main problems: the need to construct many dense matrices and to evaluate many singular and near-singular integrals.

The first problem is overcome by the use of data-sparse techniques, namely, the high-frequency fast multipole method (HF FMM) and $\mathcal{H}$-matrices. The applicability of both techniques for the discretization of the Helmholtz boundary single-layer operators with complex wavenumbers is analyzed. It is shown that the presence of decay can favorably affect the length of the fast multipole expansions and thus reduce the matrix-vector multiplication times. The performance of $\mathcal{H}$-matrices and the HF FMM is compared for a range of complex wavenumbers, and the strategy to choose between two techniques is suggested.

The second problem, namely, the assembly of many singular and nearly-singular integrals, is solved by the use of the Huygens principle. In this work we prove that kernels of the boundary integral operators $w_n^h(d)$ ($h$ is the time step and $t_n = nh$ is the time) exhibit exponential decay outside of the neighborhood of $d \approx nh$ (this is the consequence of the Huygens principle). The size of the support of these kernels for fixed $h$ increases with $n$ as $n^\alpha$, $\alpha < 1$, where $\alpha$ depends on the order of the Runge-Kutta method and is (typically) smaller for Runge-Kutta methods of higher order. Numerical experiments demonstrate that theoretically predicted values of $\alpha$ are quite close to optimal.

In the work it is shown how this property can be used in the recursive algorithm to construct only a few matrices with the near-field, while for the rest of the matrices the far-field only is assembled. The resulting method allows to solve the three-dimensional wave scattering problem with asymptotically almost linear complexity. The efficiency of the approach is confirmed by extensive numerical experiments.

# Contents

# Introduction

Many physical and engineering applications require the solution of initial boundary value problems posed outside of a bounded obstacle. A non-exhaustive list of those includes scattering (both direct and inverse) problems for the wave and Maxwell equations, the wave propagation in poroelastic half-space, heat transfer and some problems of fluid dynamics. The main challenge of the design of efficient numerical methods for such problems lies in the unboundedness of the domain. Several methods were developed in the last three decades to overcome such difficulties. Such approaches can be divided into several types.

Absorbing boundary conditions (ABCs) and perfectly matched layer (PML) techniques allow to solve the problem inside an artificially bounded domain. For an overview of these methods see [96]. Absorbing boundary conditions method [31,62,80,106] introduces an artificial boundary that encloses the bounded domain and auxiliary relations on this boundary. These relations are chosen to damp the amplitudes of the reflected waves. The resulting formulations are typically solved with the help of the finite element method (FEM) in spatial variables and finite differences in time (or via the transition to the frequency domain). However, high-order absorbing boundary conditions often require approximating of derivatives of high order and special treatment at the corners, which complicates their implementation. For the review of ABCs see works [95, 119, 181] and references therein. The perfectly matched layer method was developed by J.-P. Berenger [39, 40] and can be viewed as a generalization of absorbing boundary conditions. Similarly, the problem is solved inside an artificially bounded domain. However, instead of the artificial boundary, an absorbing (perfectly matched) layer is introduced. Inside this layer the wave equation is modified so that the wave decays exponentially and no reflection occurs at the boundary of the artificial domain. Again, in spatial variables the problem is typically discretized with the help of FEM, while in time finite difference schemes are employed. The PML techniques are easy to implement, however, their analysis and development are often non-trivial and are a subject of the past and present research [37, 38, 59, 63, 76, 92, 127]. The comparison of PML and ABCs can be found in works [96, 134, 151].

Another class of methods is based on the infinite element formulations [10], which allow to solve the problem posed in the infinite domain with the help of the finite element method, using a finite element space supplemented with infinite elements. A detailed introduction to this approach can be found in [11] and references therein.

The main idea of the method we consider in the thesis, namely time-domain boundary integral equations (TDBIE), is to express the solution of the initial boundary problem in terms of time-dependent boundary integrals. Important features of this approach include:

1. Reduction of the problem posed originally in $d$-dimensional unbounded domain to a problem posed on $(d-1)$-dimensional manifold (the boundary of this domain). The resulting integral equations are typically discretized in space with the help of

the boundary element method. The implication of the dimensionality reduction is that instead of $O(n^d)$ degrees of freedom needed to discretize the initial boundary value problem in space with the FEM, only $O(n^{d-1})$ boundary elements are required. However, the resulting matrices are often densely populated (though in the case when the strong Huygens principle holds they can exhibit sparsity). Another computational difficulty intrinsic to the boundary element method (BEM) is the necessity to evaluate many weakly-singular and nearly singular integrals, see also [118, 168, 169].

2. The method is applicable only to the problems for which the fundamental solution (or its Laplace transform) is known. This excludes equations with non-constant coefficients and some non-linear problems.

In this work we develop the TDBIE-based method for the solution of problems for which the strong Huygens principle (or its analogue) holds, e.g. acoustic and electromagnetic scattering in three dimensions, see also [30]. As a model problem we consider the wave scattering in $\mathbb{R}^3$.

TDBIE are a ubiquitous tool for solving problems of acoustics and electromagnetics. Much effort is presently dedicated to the design of new methods of the discretization of TDBIE [74, 166] and efficient algorithms for the solution of the discretized problems [143], as well as the concise analysis of the well-posedness of discretized and continuous integral formulations [133]. A comprehensive at the time of publishing review of the methods of the discretization of the TDBIE in time can be found in [65]. Whilst the Galerkin boundary element method is commonly used for treating the dependence on spatial variables, methods employed for the time discretization are far more diverse.

The simplest and probably the earliest approach is the transition to the frequency (Laplace) domain (see [66, 67] for the elastodynamics). The Laplace transform of hyperbolic problems leads to elliptic boundary value problems, and the fast methods for the solution of the latter are very well developed (for a review see [177]; a non-exhaustive list includes the use of wavelet basis functions, $\mathcal{H}$-matrices, $\mathcal{H}^2$-matrices, fast multipole methods). However, such approach is advantageous only if the solution is localized in the frequency domain. Moreover, the space-time locality due to the Huygens principle is lost during the transition to the Laplace domain, and its application to this method remains nontrivial.

Another method is collocation in time [72, 73, 141] used extensively by the engineering community. There exists a wide range of fast algorithms for the solution of such settings. We dedicate the second part of Chapter 1 to the review of these methods. The major disadvantage of collocation in time is its instability over long times, which was recently solved by the use of special spatial quadrature [174].

Galerkin formulations were developed and analyzed in [18, 19, 109]. They are used in the commercial code SONATE [2]. However, the application of the Galerkin method in time requires very precise spatial quadratures, see [166, 178] and references therein. This difficulty was recently overcome by the use of specially designed basis functions [166]. We discuss some of the features of Galerkin methods in more detail in Chapter 1.

The convolution quadrature (CQ) method was developed by Ch. Lubich in [136, 137] for the solution of Volterra integral equations. Initially, it was based on multistep methods for the discretization of ordinary differential equations. In [139] Runge-Kutta convolution quadrature was introduced for the solution of abstract parabolic equations. Multistep CQ was first employed to discretize a time-domain boundary integral formulation of the wave equation in [138]. Further developments include the design of fast convolution quadrature

for parabolic problems [171] and application of convolution quadrature to boundary integral equations stemming from the poro- and viscoelasticity theory [172, 173]. The interest in the application of convolution quadrature to the transient wave scattering was revived by works [115,116,131], where the authors considered the BDF2-based convolution quadrature discretization. It was demonstrated how the use of the Huygens principle combined with data-sparse techniques can improve the complexity of this algorithm. In [29] the authors propose a fast method for the solution of the wave equation in the unbounded domain and present a concise stability and convergence analysis. One of the techniques developed in [29], decoupled convolution quadrature, serves as a basis for the fast Runge-Kutta CQ method of [144]. The recent work [54] is dedicated to the analysis of multistep convolution quadrature combined with the Galerkin discretization for the scattering by a sound-hard obstacle, while in [55] a procedure for the reduced convolution weight computation was suggested. In [17] the convolution quadrature formulation for Maxwell equations was studied analytically and numerically.

The analysis supported by numerical experiments in [21] demonstrates advantages of convolution quadrature compared to other methods of the discretization of TDBIE. A non-exhaustive list of these includes:

- excellent stability, see [21];
- the use of Runge-Kutta CQ allows to achieve arbitrary high convergence rates, see [27, 28];
- the method does not require sophisticated spatial quadratures, and hence can be straightforwardly applied when boundary elements are curvilinear;
- only the fundamental solution in the Laplace domain needs to be known. This is of particular importance for problems in the visco- and elastodynamics, see [129,172,173].

Additionally, Runge-Kutta CQ has low dissipation and dispersion [22]. Particularly, for the acoustic scattering, this is true as well for multistep convolution quadrature applied to the discretization of the direct boundary integral formulation on convex domains. In case when the domain is not convex, or the indirect boundary formulation is used, Runge-Kutta convolution quadrature is likely to outperform multistep CQ, see the related experiments in [21].

Despite these attractive features of convolution quadrature, the field of fast CQ based methods for the transient acoustic scattering is still in the stage of infancy. To our knowledge, there exist very few fast convolution quadrature methods. Particularly, the method of [115,116,131] though offering a great improvement both in the asymptotic complexity and in constants in complexity estimates, does not allow to compute the solution in linear time. We analyze the applicability of the related ideas to Runge-Kutta CQ in Chapter 3.2. Another fast algorithm, directional FMM-accelerated convolution quadrature of [144], requires solution of many Helmholtz integral formulations with wavenumbers that have large real and small imaginary part, see also [21]. Currently there exist, to our knowledge, no efficient preconditioners for this kind of problems. In [159] the authors developed a multistep CQ method based on the fast multipole accelerated BEM of [84, 191]. Though this algorithm is of better complexity compared to conventional convolution quadrature methods, the total solution time still does not scale linearly. Hence we address this method only very briefly.

The present work is dedicated to the development of an efficient Runge-Kutta convolution quadrature algorithm of almost linear complexity. This approach is based on the

recursive algorithm of [121], which is of almost linear complexity in time. Adapted to Runge-Kutta convolution quadrature, it requires the construction of discretized Helmholtz boundary potentials for a wide range of complex frequencies. We solve this problem by the use of data-sparse techniques, namely the high-frequency fast multipole method (HF FMM) of [57, 155] and $\mathcal{H}$-matrices. This allows to create an algorithm of almost linear complexity both in time and space. Though such approach is indeed more efficient than the conventional Runge-Kutta CQ algorithm, it suffers from a serious drawback, namely the need to evaluate many singular and weakly singular integrals (the near-field). In this work we prove a sparsity property of convolution weights of Runge-Kutta CQ that allows to avoid the evaluation of the expensive near-field part for all the discretizations but a few. Based on this property, we design an algorithm of almost linear complexity whose efficiency is supported by numerical experiments.

This work can be subdivided into the following parts.

Chapter 1 is dedicated to a review of the theory of integral equations with retarded potentials, as well as the existing methods for their solution. There we also describe conventional Runge-Kutta convolution quadrature algorithms.

In Chapter 2 we review data-sparse approximations, namely $\mathcal{H}$-matrices and fast multipole methods. We describe the HF FMM in the framework of $\mathcal{H}^2$-matrices. In the end of the section we analyze the error of the fast multipole method applied to the approximation of the boundary potentials of the Helmholtz equation with decay.

Chapter 3 can be divided into three main parts. In the first part we show that Runge-Kutta convolution weights $w_n^h(d)$ decay exponentially away from $d \approx nh$. Our estimates reflect the dependence of the speed of decay on the order of a Runge-Kutta method. In the second part of the section we discuss the applicability of main ideas of sparse convolution quadrature [115, 116, 131] to Runge-Kutta based CQ, as well as demonstrate principal difficulties associated with the use of sparse Runge-Kutta CQ. In the third part of this section we present a recursive convolution quadrature algorithm that allows to construct only a small number of matrices with the near-field.

Finally, Chapter 4 is devoted to numerical experiments, which highlight the efficiency of the suggested technique.

# Chapter 1

# Time-Domain Boundary Integral Formulations for the Wave Equation in Three Dimensions

This chapter is dedicated to the review of the questions of existence and uniqueness of the solutions of time-domain boundary integral equations (TDBIE), as well as summary of the existing fast methods for numerical solution of the TDBIE. As a model problem, we consider wave scattering in three dimensions.

## 1.1 Introduction into the Theory of Time-Domain Boundary Integral Equations

The theory of time-domain boundary integral equations for boundary value problems for the wave equation was first developed in [18,19]. It is heavily based on the apparatus of the Laplace transform and existence and uniqueness results for integral formulations of the wave equation in the Laplace domain (i.e. the Helmholtz equation with a complex wavenumber). Recently, improved continuity estimates on the underlying integral operators were derived solely in the time domain [77]. A detailed introduction into the theory of retarded potentials can be found in [170]. The works [64,142,177] are dedicated to the exposition of the theory of boundary integral equations.

As a model problem, we consider wave scattering by a sound-soft obstacle. Let $\Omega \subset \mathbb{R}^3$ be a bounded Lipschitz domain and let $\Gamma$ be its boundary.

An incident wave $u^{inc}$ satisfies the wave equation in the free space:

$$\begin{aligned} \frac{\partial^2 u^{inc}}{\partial t^2}(t,x) - \Delta u^{inc} &= f(t,x), \qquad (t,x) \in [0,\ T] \times \mathbb{R}^3, \\ u^{inc}(0,x) &= u_0(x), \qquad x \in \mathbb{R}^3, \\ \left.\frac{\partial u^{inc}(t,x)}{\partial t}\right|_{t=0} &= u_1(x), \qquad x \in \mathbb{R}^3. \end{aligned}$$

The source term and initial conditions satisfy

$$\begin{aligned} \operatorname{supp} f(t,x) &\Subset \Omega^c, \qquad t \geq 0, \\ \operatorname{supp} u_0(x) &\Subset \Omega^c, \\ \operatorname{supp} u_1(x) &\Subset \Omega^c, \\ f &\in C(\mathbb{R}_{\geq 0}, L^2(\mathbb{R}^3)), \\ u_0 &\in H_0^1\left(\mathbb{R}^3\right), \\ u_1 &\in L_2\left(\mathbb{R}^3\right). \end{aligned}$$

Then the total field $u^{tot}$ solves the wave equation in the exterior of the domain $\Omega$:

$$\begin{aligned} \frac{\partial^2 u^{tot}}{\partial t^2}(t,x) - \Delta u^{tot}(t,x) &= f(t,x), && (t,x) \in [0,\ T] \times \Omega^c, \\ u^{tot}(t,x) &= 0, && (t,x) \in [0,\ T] \times \Gamma, \\ u^{tot}(0,x) &= u_0(x), && x \in \Omega^c, \\ \left.\frac{\partial u^{tot}(t,x)}{\partial t}\right|_{t=0} &= u_1(x), && x \in \Omega^c. \end{aligned}$$

Defining the scattered field as

$$u(t,x) = u^{tot}(t,x) - u^{inc}(t,x),$$

we obtain the following boundary-value problem:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2}(t,x) - \Delta u(t,x) &= 0, && (t,x) \in [0,\ T] \times \Omega^c, \\ u(t,x) = g(t,x) &\equiv -u^{inc}(t,x), && (t,x) \in [0,\ T] \times \Gamma, \\ u(0,x) = \left.\frac{\partial u(t,x)}{\partial t}\right|_{t=0} &= 0, && x \in \Omega^c. \end{aligned} \tag{1.1}$$

The well-posedness of this boundary-value problem has been proved in [18].

The solution to the scattering problem can be represented as the single-layer potential of an unknown density $\lambda$ (this is an indirect formulation):

$$\begin{aligned} u(t,x) = \left(S\lambda\right)(t,x) &= \int\limits_{\Gamma} \frac{\lambda(t - \|x-y\|, y)}{4\pi\|x-y\|} d\Gamma_y d\tau \\ &= \int\limits_0^t \int\limits_{\Gamma} \frac{\delta(t - \tau - \|x-y\|)}{4\pi\|x-y\|} \lambda(\tau, y) d\Gamma_y d\tau, \qquad (t,x) \in [0,\ T] \times \Omega^c \end{aligned} \tag{1.2}$$

where $\delta(t)$ is the Dirac delta function. It is possible to show that the single layer potential is continuous across $\Gamma$, hence, letting $x \to \Gamma$,

$$g(t,x) = \left(V\lambda\right)(t,x) = \int\limits_0^t \int\limits_{\Gamma} \frac{\delta(t - \tau - \|x-y\|)}{4\pi\|x-y\|} \lambda(\tau, y) d\Gamma_y d\tau, \qquad (t,x) \in [0,\ T] \times \Gamma. \tag{1.3}$$

To justify the well-posedness of (1.3), we perform the transition to the Laplace domain. Let us denote the Laplace transform of $u(t,x),\ g(t,x)$ by $U(x),\ G(x)$:

$$U(x) = \int_0^{+\infty} \mathrm{e}^{-st} u(t,x)dt,$$
$$G(x) = \int_0^{+\infty} \mathrm{e}^{-st} u(t,x)dt, \qquad \mathrm{Re}\, s > 0.$$

Note that $U(x),\ G(x)$ depend on $s \in \mathbb{C}$, though this dependence is not stated explicitly here.

In the Laplace domain the initial boundary-value problem (1.1) becomes the Dirichlet problem for the Helmholtz equation with the complex wavenumber:

$$\begin{aligned} -\Delta U + s^2 U &= 0, \qquad x \in \Omega^c, \\ U(x) &= G(x), \qquad x \in \Gamma. \end{aligned} \tag{1.4}$$

The solution to the above problem can be written as a single layer potential of the unknown density $\Lambda$ (which implicitly depends on $s \in \mathbb{C}$, too):

$$U(x) = \left(\mathcal{S}(s)\Lambda\right)(x) = \frac{1}{4\pi} \int_\Gamma \frac{\mathrm{e}^{-s\|x-y\|}}{\|x-y\|} \Lambda(y) d\Gamma_y, \qquad x \in \Omega^c.$$

Let $\gamma$ denote the boundary trace operator. The continuity of the single layer potential through $\Gamma$ implies

$$G(x) = \left(\mathcal{V}(s)\Lambda\right)(x) := \gamma\left(\mathcal{S}(s)\Lambda\right)(x) = \frac{1}{4\pi} \int_\Gamma \frac{\mathrm{e}^{-s\|x-y\|}}{\|x-y\|} \Lambda(y) d\Gamma_y,\ x \in \Gamma. \tag{1.5}$$

The following arguments from [138] show how the transition back to the time domain can be made. Let

$$\Sigma_\sigma = \left\{ s \in \mathbb{C} :\ \mathrm{Re}\, s > \sigma \right\}.$$

Let $X, Y$ be two Hilbert spaces and let $\sigma_0 \in \mathbb{R}$. By $L(X,Y)$ the space of bounded linear operators from $X$ to $Y$ is denoted. Let $K(s) :\ \Sigma_{\sigma_0} \to L(X,Y)$ be an analytic function of $s$ bounded as

$$\|K(s)\|_{X \to Y} \leq M|s|^\mu, \qquad \mu \in \mathbb{R},\ \mathrm{Re}\, s > \sigma_0. \tag{1.6}$$

Then, for $K_m(s) = s^{-m}K(s)$, $m > \mu + 1$, we can define the inverse Laplace transform as the $m$-th derivative (in the sense of distributions) of the causal operator-valued function

$$k_m(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} \mathrm{e}^{st} K_m(s) ds, \qquad t \in \mathbb{R},\ \sigma > \sigma_0. \tag{1.7}$$

We define the convolution operator as

$$\left(K(\partial_t)g\right)(t) = \left(\frac{d}{dt}\right)^m \int_{-\infty}^{t} k_m(t-\tau) g(\tau) d\tau.$$

Before continuing, we will need some auxiliary definitions due to [138] and [18].

Given a Hilbert space $X$, let $g(t)$ be a smooth mapping $\mathbb{R} \to X$. Denoting the Fourier transform of $g(t)$ by $\mathcal{F}g$, we can define for $r \in \mathbb{R}$ the following spaces

$$H^r(\mathbb{R}; X) = \left\{ g : \int_{-\infty}^{+\infty} (1 + |\xi|)^{2r} \| (\mathcal{F}g)(\xi) \|_X^2 < +\infty \right\},$$
$$H_0^r(0, T; X) = \left\{ g|_{(0,T)} : \; g \in H^r(\mathbb{R}; X), \; g(t) = 0, t < 0 \right\}.$$

The space $H_0^r(0, T; X)$ contains causal functions $g$ whose first $r - 1$ derivatives are zero at $t = 0$, i.e.

$$g'(0) = \ldots = g^{(r-1)}(0) = 0.$$

The distributional derivative $g^{(r)}$ is square-integrable on $(0, \; T)$.

The following lemma was proved in [138].

**Lemma 1.1.1.** *If $K(s)$ satisfies (1.6), $K(\partial_t)$ can be extended by density to a bounded linear operator*

$$K(\partial_t) : \; H_0^{r+\mu}(0, T; X) \to H_0^r(0, T; Y), \qquad r \in \mathbb{R}.$$

Now we would like to apply these results to (1.3). Note that this equation can be formally written as the following convolution:

$$\mathcal{V}(\partial_t)\lambda = g, \tag{1.8}$$

where

$$\mathcal{V}(s)\Lambda = \int_\Gamma \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|} \Lambda(y) d\Gamma_y.$$

To justify (1.8) we have to show that (1.6) holds for $K(s) = \mathcal{V}(s)$. The following proposition from [18] provides us with the required bounds.

We denote by $\langle ., . \rangle$ the sesquilinear duality pairing that extends the inner product on $\Gamma$, i.e.

$$\langle \phi, \psi \rangle = \int_\Gamma \phi(x) \overline{\psi(x)} d\Gamma_x.$$

**Proposition 1.1.2.** *For $\operatorname{Re} s > 0$, the boundary single layer operator $\mathcal{V}(s)$ is an isomorphism*

$$\mathcal{V}(s) : H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma).$$

*If $\operatorname{Re} s > \sigma_0$, for some $\sigma_0 > 0$, then*

$$\|\mathcal{V}(s)\|_{H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma)} \leq \frac{C_1}{\sigma_0} \max\left(\frac{1}{\sigma_0^2}, 1\right) |s|,$$
$$\|\mathcal{V}^{-1}(s)\|_{H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma)} \leq \frac{C_2}{\sigma_0} \max\left(\frac{1}{\sigma_0}, 1\right) |s|^2,$$

*for some $C_1, C_2 > 0$ that depend on $\Gamma$ only. For all $\phi \in H^{-\frac{1}{2}}(\Gamma)$, the following coercivity estimate holds:*

$$\mathrm{Re}\langle \phi, s\mathcal{V}(s)\phi \rangle \geq C_3 \min(\sigma_0, 1)|s|^{-1}\|\phi\|^2_{H^{-\frac{1}{2}}(\Gamma)},$$

*where $C_3 > 0$ and does not depend on $s, \phi$.*

Then, using Lemma 1.1.1 we can derive the following proposition (see [138] and [18]).

**Proposition 1.1.3.** *The boundary single layer operator $\mathcal{V}(\partial_t)$ maps*

$$\mathcal{V}(\partial_t):\ H^r_0(0,T;H^{-\frac{1}{2}}(\Gamma)) \to H^{r-1}_0(0,T;H^{\frac{1}{2}}(\Gamma)),\ r \in \mathbb{R},$$

*and its (convolutional) inverse (in a sense that $\mathcal{V}^{-1}(s)\mathcal{V}(s) = I$):*

$$\mathcal{V}^{-1}(\partial_t):\ H^r_0(0,T;H^{\frac{1}{2}}(\Gamma)) \to H^{r-2}_0(0,T;H^{-\frac{1}{2}}(\Gamma)),\ r \in \mathbb{R}.$$

These results show the well-posedness of the problem (1.8). To obtain the solution to the scattering problem $u(t,x),\ x \in \Omega^c$, we can use

$$u(t,x) = \mathcal{S}(\partial_t)\mathcal{V}^{-1}(\partial_t)g(t,x).$$

Bounds on the norm of the operator $\mathcal{S}(\partial_t)\mathcal{V}^{-1}(\partial_t)$ in the Laplace domain are given by the following lemma [18] (see also [170, Chapter 3.3]).

**Lemma 1.1.4.** *Given $s \in \mathbb{C}:\ \mathrm{Re}\, s > \sigma_0$, for some $\sigma_0 > 0$,*

$$\|\mathcal{S}(s)\mathcal{V}^{-1}(s)\|_{H^{\frac{1}{2}}(\Gamma)\to H^1(\Omega^c)} \leq \frac{C}{\sigma_0}\max\left(\frac{1}{\sigma_0^{\frac{3}{2}}}, 1\right)|s|^{\frac{3}{2}}.$$

We have demonstrated that the indirect boundary integral formulation (1.3) for the scattering problem is well-posed. The solution of this boundary integral equation, namely $\lambda(t,x)$, is a non-physical quantity:

$$\lambda(t,x) = \partial^-_\nu u(t,x) - \partial^+_\nu u(t,x),$$

where $\partial^-_\nu,\ \partial^+_\nu$ are the interior and exterior normal derivatives on the boundary (see [169]). The function $u(t,x)$ solves the scattering problem (1.1) and satisfies the Dirichlet problem for the wave equation with the same Dirichlet data $g(t,x)$ inside the domain:

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2}(t,x) - \Delta u(t,x) &= 0, && (t,x) \in [0,\ T] \times \Omega, \\ u(t,x) &= g(t,x), && (t,x) \in [0,\ T] \times \Gamma, \\ u(0,x) = \left.\frac{\partial u(t,x)}{\partial t}\right|_{t=0} &= 0, && x \in \Omega. \end{aligned}$$

Therefore, the indirect formulation (1.3) allows to obtain the solution to both scattering and the above problem (or the transmission problem, see [170]).

Another option to solve the scattering problem is to employ the direct formulation based on the Kirchoff formula:

$$u = \mathcal{D}(\partial_t)g - \mathcal{S}(\partial_t)\phi,\ x \in \Omega^c \tag{1.9}$$

where $\mathcal{S}(\partial_t)$ is the single-layer potential, see (1.2), $\phi$ is an unknown density and $\mathcal{D}(\partial_t)$ is the double-layer potential. Given the normal vector $\nu_y$, $y \in \Gamma$, pointing outwards of the domain $\Omega$, the double-layer potential is defined as:

$$(\mathcal{D}(\partial_t)g)(t,x) = \int_\Gamma \frac{\partial g}{\partial t}(t - \|x-y\|, y)\frac{(x-y,\nu_y)}{4\pi\|x-y\|^2}d\Gamma_y + \int_\Gamma g(t-\|x-y\|,y)\frac{(x-y,\nu_y)}{4\pi\|x-y\|^3}d\Gamma_y, \qquad x \in \Omega^c.$$

The unknown $\phi(t,x)$ in (1.9) is the exterior normal derivative of $u(t,x)$:

$$\phi(t,x) = \partial_\nu^+ u(t,x).$$

Importantly, the evaluation of (1.9) inside the domain $\Omega$ gives

$$\mathcal{D}(\partial_t)g(t,x) - \mathcal{S}(\partial_t)\phi(t,x) = 0, \qquad x \in \Omega.$$

To obtain the corresponding boundary integral equation, we introduce

$$\mathcal{K} = \frac{\gamma^+ + \gamma^-}{2}\mathcal{D},$$

where $\gamma^+, \gamma^-$ are the exterior and interior trace operators. Using the jump properties of the $\gamma_+\mathcal{D}$, $\gamma_-\mathcal{D}$ (see, e.g. [142]), we arrive at the direct boundary integral formulation:

$$\mathcal{V}(\partial_t)\phi = -\frac{g}{2} + \mathcal{K}(\partial_t)g. \tag{1.10}$$

As before, to show the well-posedness of the direct formulation, we need the bounds on the operators $\mathcal{D}$, $\mathcal{K}$ in the Laplace domain. The double layer potential for the Helmholtz equation with decay (1.4) as an operator

$$\mathcal{D}(s) : H^{\frac{1}{2}}(\Gamma) \to H^1(\Omega^c),$$
$$\mathcal{D}(s)\Phi = \int_\Gamma \frac{d}{d\nu_y}\left(\frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\right)\Phi(y)d\Gamma_y, \; x \in \Omega_+.$$

The corresponding double layer boundary integral operator is defined as:

$$\mathcal{K}(s) : H^{\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma),$$
$$\mathcal{K}(s) = \frac{\gamma^+ + \gamma^-}{2}\mathcal{D}(s),$$

The following lemma can be found in [19] and [133].

**Lemma 1.1.5.** *For all $s : \operatorname{Re} s > 0$, the operator*

$$\mathcal{D}(s) : \; H^{\frac{1}{2}}(\Gamma) \to H^1(\Omega^c)$$

*is bounded. For $\operatorname{Re} s > \sigma_0 > 0$, it satisfies:*

$$\|\mathcal{D}(s)\|_{H^{\frac{1}{2}}(\Gamma)\to H^1(\Omega^c)} \leq C\frac{|s|^{\frac{3}{2}}}{\sigma_0}\max\left(1, \frac{1}{\sigma_0^{\frac{3}{2}}}\right).$$

*Similarly, for all* $s : \operatorname{Re} s > 0$*, the operator*

$$\mathcal{K}(s) : H^{\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma)$$

*is bounded. For* $\operatorname{Re} s > \sigma_0 > 0$*,*

$$\|\mathcal{K}(s)\|_{H^{\frac{1}{2}}(\Gamma)\to H^{\frac{1}{2}}(\Gamma)} \leq \tilde{C}\frac{|s|^{\frac{3}{2}}}{\sigma_0}\max\left(1,\ \frac{1}{\sigma_0^{\frac{3}{2}}}\right).$$

*The constants* $C$*,* $\tilde{C}$ *depend on* $\Gamma$ *only.*

The direct boundary integral formulation for the exterior problem for the Helmholtz equation with decay (1.4) can be then defined as follows. Find $\Phi \in H^{-\frac{1}{2}}(\Gamma)$ satisfying

$$\mathcal{V}(s)\Phi = -\frac{G}{2} + \mathcal{K}(s)G. \tag{1.11}$$

The operator $\mathrm{DtN}^+(s) = \mathcal{V}^{-1}(s)\left(-\frac{I}{2} + \mathcal{K}(s)\right)$ is the exterior Dirichlet-to-Neumann map. The well-posedness of the formulation (1.11) can be seen from the following lemma (see [18, 19] and [133]).

**Lemma 1.1.6.** *For* $\operatorname{Re} s > \sigma_0 > 0$*, the operator*

$$\mathrm{DtN}^+(s) :\ H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma)$$

*is bounded and satisfies:*

$$\begin{aligned}\|\,\mathrm{DtN}^+(s)\|_{H^{\frac{1}{2}}(\Gamma)\to H^{-\frac{1}{2}}(\Gamma)} &\leq C_1\frac{1}{\sigma_0}\max\left(1,\frac{1}{\sigma_0}\right)|s|^2,\\ -\operatorname{Re}\langle \mathrm{DtN}^+(s)\phi, s\phi\rangle &\geq C_2\sigma_0\min\left(1,\sigma_0^2\right)\|\phi\|_{H^{\frac{1}{2}}(\Gamma)},\end{aligned}$$

*where* $C_1, C_2 > 0$ *are independent of* $s$ *and* $\phi$*.*

In most cases for the numerical solution of the TDBIE the indirect formulation is employed. In the present work we use this formulation as well, since, as explained in [21], for most interesting cases, i.e. trapping obstacles, the convolution quadrature discretizations of the direct and the indirect formulations behave similarly. For convex and star-shaped obstacles the application of convolution quadrature to the direct formulation allows to solve the wave scattering problem with less computation effort, see [21], as well as the related discussion in Section 1.2.11.3. For the comparison of direct and indirect boundary integral formulations for time-independent problems see [169].

## 1.2 Fast Methods of the Solution of Time-Domain Boundary Integral Equations

In this section we concentrate on the numerical solution of the indirect formulation

$$g(t,x) = \int_0^t \int_\Gamma \frac{\delta(t-\tau-\|x-y\|)}{4\pi\|x-y\|}\lambda(\tau,y)d\Gamma_y, \qquad (t,x)\in[0,T]\times\Gamma, \tag{1.12}$$

though methods presented here can be extended to (1.10). A detailed survey of methods for the discretization of this equation can be found in [65].

In the first part of this section we review existing methods for the solution of the time-domain integral equation (1.12), including the conventional marching-on-in-time method, the plane-wave time-domain algorithm, the time-domain adaptive integral method as well as the non-uniform Cartesian grid time-domain algorithm. These methods were originally developed for collocation in time but can be easily applied to speed up Galerkin-based algorithms. We also briefly mention fast Galerkin methods.

The second part of this section is dedicated to convolution quadrature and existing improvements of this method, namely, sparse multistep convolution quadrature, decoupled convolution quadrature, multipole accelerated convolution quadrature of [159] and recursive CQ algorithm.

Sparse multistep convolution quadrature was developed in a series of works [115,116,131] to speed up back substitution. In this section we sketch main ideas of this algorithm, postponing the discussion of its applicability to Runge-Kutta CQ to Section 3.2. Next, we briefly review convolution quadrature of [159] which is also based on back substitution.

Unlike back substitution, the conventional recursive convolution quadrature algorithm is of almost linear complexity in the number of time steps. One of its components, decoupled convolution quadrature, serves as a basis for another fast CQ, namely decoupled convolution improved by the use of the directional fast multipole method, see [144]. We finish the overview of existing fast TDBIE methods with a brief description of this algorithm.

### 1.2.1 Marching-on-in-Time

The conventional marching-on-in-time (MOT) method is based on temporal collocation and spatial Galerkin discretizations. Let the time interval $[0,\, T]$ be subdivided into $N$ subintervals of size $h$. The set of the basis functions in time $(T_n(t))_{n=0}^{N}$ is chosen so that

$$T_j(t) = T(t - t_j), \qquad t_j = jh, \tag{1.13}$$

where $T : \mathbb{R} \to \mathbb{R}$ is compactly supported. For example, one can use hat functions [153], namely,

$$T(t) = \begin{cases} 1 - \frac{|t|}{h}, & -h \le t \le h, \\ 0, & \text{else}, \end{cases} \tag{1.14}$$

or continuous piecewise quadratic functions [140]. For simplicity we assume in this section that $T(t)$ is chosen as in (1.14).

Let $(\phi_j(x))_{j=1}^{M}$ be a set of both trial and test basis functions. These functions are assumed to be locally supported, so that $\operatorname{diam}(\operatorname{supp}\phi_j) = C_j h$, where $0 < c < C_j < C$, $j = 1, \dots, M$. If $g(t, x)$ is temporally bandlimited to $\omega_m$, the number of basis functions is chosen as $M \approx C_s \omega_m^2$ (or $O(\omega_m^2)$) and $N \approx C_t \omega_m$ (i.e. $O(\omega_m)$), where $C_s, C_t > 0$ do not depend on $\omega_m$, see [85].

We look for $\lambda(t, x)$ in the form

$$\lambda(t, x) \approx \sum_{n=0}^{N} \sum_{m=1}^{M} \lambda_n^m T_n(t) \phi_m(x). \tag{1.15}$$

Substituting the expression (1.15) into (1.12) and testing it at $t = t_j$ with each of the spatial basis functions, we obtain the system of equations with respect to the vector of coefficients $\mathbf{\Lambda} = (\lambda_n^m),\ m = 1 \dots M,\ n = 0 \dots N$:

$$\int\limits_{\Gamma} g(t_j, x)\phi_l(x)d\Gamma_x = \sum_{n=0}^{N}\sum_{m=1}^{M}\lambda_n^m \int\limits_{\Gamma}\int\limits_{\Gamma}\int\limits_{0}^{t_j} \frac{\delta(t_j - \tau - \|x-y\|)}{4\pi\|x-y\|} \times \phi_m(y)\phi_l(x)T_n(\tau)d\tau d\Gamma_y d\Gamma_x, \qquad (1.16)$$
$$l = 1, \dots, M, \qquad j = 0, \dots, N.$$

Our goal is to demonstrate that the system of equations (1.16) can be written as a lower triangular Toeplitz system. First, we define

$$\tilde{Z}_{n,j} := \int\limits_{0}^{t_j} \frac{\delta(t_j - \tau - \|x-y\|)}{4\pi\|x-y\|} T_n(\tau)d\tau = \int\limits_{0}^{t_j} \frac{\delta(t_j - \tau - \|x-y\|)}{4\pi\|x-y\|} T(\tau - t_n)d\tau,$$

where the last expression follows from (1.13). Next, let us show that

$$\tilde{Z}_{n,j} = Z_{n-j}.$$

From (1.14) it follows that $\tilde{Z}_{n,j} = 0$ for $n > j$. For $n \le j$, with the help of a change of variables, and using (1.14), we obtain the following expression:

$$\tilde{Z}_{n,j} = \int\limits_{-h}^{t_{j-n}} \frac{\delta(t_{j-n} - \tau - \|x-y\|)}{4\pi\|x-y\|} T(\tau)d\tau = \int\limits_{0}^{t_{j-n+1}} \frac{\delta(t_{j-n+1} - \tau - \|x-y\|)}{4\pi\|x-y\|} T_1(\tau)d\tau. \qquad (1.17)$$

For $j, n = 0, \dots, N$

$$\mathbf{\Lambda}_j := \left[\lambda_j^1, \dots, \lambda_j^M\right]^T,$$
$$\boldsymbol{G}_j := \left[g_j^1, \dots, g_j^M\right]^T,$$

and

$$(Z_n)_{lm} = \iint\limits_{\Gamma\times\Gamma} \phi_l(x)\phi_m(y) \int\limits_{0}^{t_{n+1}} \frac{\delta(t_{n+1} - \tau - \|x-y\|)}{4\pi\|x-y\|} T_1(\tau)d\tau d\Gamma_x d\Gamma_y, \qquad l, m = 1, \dots, M.$$

Then, with the help of (1.17), the system of equations (1.16) can be rewritten as

$$\sum_{n=0}^{j} Z_n \mathbf{\Lambda}_{j-n} = \boldsymbol{G}_j.$$

Hence, to find the solution on the time interval $[0,\ T]$, we need to solve the lower-triangular system of equations

$$\begin{pmatrix} Z_0 & 0 & 0 & \dots & 0 \\ Z_1 & Z_0 & 0 & \dots & 0 \\ \vdots & & & & \\ Z_N & Z_{N-1} & Z_{N-2} & \dots & Z_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_0 \\ \boldsymbol{\Lambda}_1 \\ \vdots \\ \boldsymbol{\Lambda}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_0 \\ \boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{G}_N \end{pmatrix}. \tag{1.18}$$

In the next two sections we introduce two algorithms for the solution of the system (1.18) that serve as a basis for some of the fast techniques for the solution of time-domain integral equations.

### 1.2.2 Back Substitution

One of the methods for solving (1.18) is back substitution, see [153]. At each step $j$, $j = 1, \dots, N$, one solves the equation

$$Z_0 \boldsymbol{\Lambda}_j = \boldsymbol{G}_j - \sum_{n=1}^{j} Z_n \boldsymbol{\Lambda}_{j-n}. \tag{1.19}$$

Only the inversion of the matrix $Z_0$ is required. This matrix contains only $O(M)$ non-zero elements, and iterative methods for the solution of this system are known to converge in a few iterations [14]. It can be shown, with the methods similar to that in [116], that under the assumptions on $N$ and $M$ made in Section 1.2.1, each of the matrices $Z_n$, $n = 1, \dots, N$, has $O(nM)$ non-zero entries. This results in the $O(M^2N)$ complexity of the solution of the system (1.19), see [58]. The dominant costs of the algorithm are in the evaluation of the summation on the right hand side.

### 1.2.3 Recursive Algorithm

Here we describe the recursive algorithm for the solution of the lower triangular Toeplitz system (1.18) according to [121], where it was derived to solve Volterra convolution equations.

The structure of the matrix in (1.18) is shown in Figure 1.1. Identical subblocks are marked with the same letters. The main idea of the algorithm is to substitute the solution of the full system by solving many small triangular systems with the matrix $T_0$ and the computation of the matrix-vector product with matrices $T_1$, $T_2$. The small triangular system can be solved, for example, by back substitution.

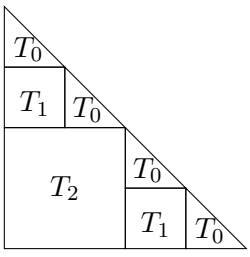


*Figure 1.1: Structure of the matrix in the system (1.18).*

**Remark 1.2.1.** *The levels of the recursive algorithm are enumerated starting from the base case. Namely, matrix-vector products with matrices $T_1$ are computed on the first stage or level of the algorithm and matrix-vector products with $T_2$ on the second level.*

Let us introduce several basic procedures of the algorithm.
`Solve` ($n_0$, $n_1$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$) - solves recursively the system of equations

$$\begin{pmatrix} Z_0 & 0 & \cdots & 0 \\ Z_1 & Z_0 & \cdots & 0 \\ \vdots & & & \\ Z_{n_1} & Z_{n_1-1} & \cdots & Z_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_{n_0} \\ \boldsymbol{\Lambda}_{n_0+1} \\ \vdots \\ \boldsymbol{\Lambda}_{n_0+n_1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_{n_0} \\ \boldsymbol{G}_{n_0+1} \\ \vdots \\ \boldsymbol{G}_{n_0+n_1} \end{pmatrix}.$$

`Multiply` ($m$, $n$, $p$, $l$, $\boldsymbol{\Lambda}$, $\boldsymbol{H}$) - performs the matrix-vector multiplication

$$\begin{pmatrix} \boldsymbol{H}_l \\ \boldsymbol{H}_{l+1} \\ \vdots \\ \boldsymbol{H}_{l+n-m} \end{pmatrix} = \begin{pmatrix} Z_m & Z_{m-1} & \cdots & Z_1 \\ Z_{m+1} & Z_m & \cdots & Z_2 \\ \vdots & & & \\ Z_n & Z_{n-1} & \cdots & Z_{n-m+1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_p \\ \boldsymbol{\Lambda}_{p+1} \\ \vdots \\ \boldsymbol{\Lambda}_{p+m-1} \end{pmatrix}. \tag{1.20}$$

`SolveTri` ($n_0$, $n_1$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$) - solves the (small) triangular system of equations directly

$$\begin{pmatrix} Z_0 & 0 & \cdots & 0 \\ Z_1 & Z_0 & \cdots & 0 \\ \vdots & & & \\ Z_{n_1-n_0} & Z_{n_1-n_0-1} & \cdots & Z_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_{n_0} \\ \boldsymbol{\Lambda}_{n_0+1} \\ \vdots \\ \boldsymbol{\Lambda}_{n_1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{G}_{n_0} \\ \boldsymbol{G}_{n_0+1} \\ \vdots \\ \boldsymbol{G}_{n_1} \end{pmatrix}.$$

Let us fix $J > 0$: every system of size smaller or equal to $J + 1$ is to be solved directly. By procedure `Solve`, larger systems will be split in two and solved recursively, until their size reaches $J + 1$, when they are solved by `SolveTri`. A pseudocode of this procedure is given below.

---

**function** `Solve` ($n_0$, $n_1$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$)
**if** ($n_1 - n_0 \leq J$) **then**
  `SolveTri`($n_0$, $n_1$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$);
**else**
  $n_{\frac{1}{2}} = \left\lfloor \frac{n_0+n_1}{2} \right\rfloor$;
  `Solve`($n_0$, $n_{\frac{1}{2}}$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$);
  `Multiply` ($n_{\frac{1}{2}} - n_0 + 1$, $n_1 - n_0$, $n_0$, $n_{\frac{1}{2}} + 1$, $\boldsymbol{\Lambda}$, $\boldsymbol{H}$);
  $\boldsymbol{G}|_{n_{\frac{1}{2}}+1,\ldots,n_1} = \boldsymbol{G}|_{n_{\frac{1}{2}}+1,\ldots,n_1} - \boldsymbol{H}|_{n_{\frac{1}{2}}+1,\ldots,n_1}$;
  `Solve`($n_{\frac{1}{2}} + 1$, $n_1$, $\boldsymbol{G}$, $\boldsymbol{\Lambda}$);
**end if**
**endFunction**

---

The matrix in (1.20) is Toeplitz, and the fast matrix-vector multiplication can be done by embedding this matrix into a twice larger circulant matrix that in turn can be diagonalized with FFT. Let us describe this procedure in more detail.

#### 1.2.3.1 Fast Matrix-Vector Multiplication

Let us consider the procedure for the fast computation of the matrix-vector product (1.20), namely

$$\begin{pmatrix} \boldsymbol{H}_m \\ \boldsymbol{H}_{m+1} \\ \vdots \\ \boldsymbol{H}_n \end{pmatrix} = \begin{pmatrix} Z_m & Z_{m-1} & \cdots & Z_1 \\ Z_{m+1} & Z_m & \cdots & Z_2 \\ \vdots & & & \\ Z_n & Z_{n-1} & \cdots & Z_{n-m+1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_0 \\ \boldsymbol{\Lambda}_1 \\ \vdots \\ \boldsymbol{\Lambda}_m \end{pmatrix}.$$

We define

$$\tilde{\boldsymbol{\Lambda}} := [\boldsymbol{\Lambda}_0,\ \ldots,\ \boldsymbol{\Lambda}_m]^T,$$
$$\tilde{\boldsymbol{H}} := [\boldsymbol{H}_m,\ \ldots,\ \boldsymbol{H}_n]^T.$$

First, the vector $\tilde{\boldsymbol{\Lambda}}$ is extended with $n-m$ zeros at the end and the vector $\tilde{\boldsymbol{H}}$ with extra $m$ elements at the beginning. Then the product (1.20) can be rewritten as:

$$\begin{pmatrix} \boldsymbol{H}_0 \\ \boldsymbol{H}_1 \\ \vdots \\ \boldsymbol{H}_{m-1} \\ \boldsymbol{H}_m \\ \boldsymbol{H}_{m+1} \\ \vdots \\ \boldsymbol{H}_n \end{pmatrix} = \begin{pmatrix} Z_0 & Z_n & \cdots & Z_1 \\ Z_1 & Z_0 & \cdots & Z_2 \\ \vdots & & & \\ Z_{m-1} & Z_{m-2} & \cdots & Z_m \\ Z_m & Z_{m-1} & \cdots & Z_{m+1} \\ Z_{m+1} & Z_m & \cdots & Z_{m+2} \\ \vdots & & & \\ Z_n & Z_{n-1} & \cdots & Z_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\Lambda}_0 \\ \boldsymbol{\Lambda}_1 \\ \vdots \\ \boldsymbol{\Lambda}_m \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Setting

$$\boldsymbol{H} = [\boldsymbol{H}_0,\ \ldots,\ \boldsymbol{H}_n]^T,$$
$$\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_0, \ldots, \boldsymbol{\Lambda}_m,\ 0,\ \ldots,\ 0]^T$$

and using the fact that the circulant matrix can be diagonalized by the discrete Fourier transform, we can rewrite the above matrix-vector product as

$$\boldsymbol{H} = \mathcal{F}_{n+1}^{-1} D_{n+1} \mathcal{F}_{n+1} \boldsymbol{\Lambda}. \tag{1.21}$$

Here $\mathcal{F}_{n+1}$ is the discrete Fourier matrix of size $(n+1) \times (n+1)$ and $D_{n+1}$ is a diagonal matrix whose elements $\boldsymbol{d}_{jj}$ are matrices

$$\boldsymbol{d}_{jj} = \sum_{k=0}^{n} Z_k e^{-i\frac{2\pi}{n+1}kj},\ j = 0, \ldots, n. \tag{1.22}$$

Note that matrices $\boldsymbol{d}_{jj}$ are no longer sparse, therefore the matrix-vector multiplication with each of these matrices requires $O(M^2)$ steps. Excluding the time needed to construct $\boldsymbol{d}_{jj}$, such matrix-vector multiplication can be done in $O(n \log nM + nM^2)$ steps with the help of the FFT.

The complexity of the full algorithm scales as $O(N \log^2 NM + N \log NM^2)$, or

$$O(N \log NM^2),$$

which is slightly worse than the complexity of back substitution. However, as will be shown later, an efficient approach can be designed based on this algorithm.

### 1.2.4 Plane-Wave Time-Domain Algorithm

As it was remarked before, the bottleneck of the conventional MOT method is the computation of the sum in (1.19). The plane-wave time-domain (PWTD) algorithm was developed to speed-up the evaluation of such sums exploiting separability properties of transient fields [12, 58, 85, 86]. The complexity of the PWTD algorithm varies from $O(NM^{\frac{3}{2}}\log M)$ for the one-level scheme to $O(NM\log^2 M)$ for the multi-level scheme.

Here we will only show the expansion the PWTD algorithm is based on. The method resembles closely the structure of fast multipole methods [103, 154, 155], aiming at a fast evaluation of the (discretized) expression:

$$\begin{aligned} f(t,x) &= \int_0^t \int_\Gamma \frac{\delta(t-\tau-\|x-y\|)}{4\pi\|x-y\|} q(\tau,y)\, d\Gamma_y d\tau \\ &= \int_\Gamma \frac{q(t-\|x-y\|,y)}{4\pi\|x-y\|} d\Gamma_y, \end{aligned} \tag{1.23}$$

where $q(\tau, y)$ is a given density.

The following lemma serves as a basis for the derivation of one of the separable expansions for the kernel of the integral operator (1.23), see [85] and references therein.

**Lemma 1.2.2.** *Let $x, y \in \Gamma$. Then,*

$$Q(t,x,y) = \frac{q(t-\|x-y\|,y)}{4\pi\|x-y\|} \tag{1.24}$$

*can be decomposed into the sum:*

$$Q(t,x,y) = Q^{pw}(t,x,y) + Q^{g}(t,x,y), \tag{1.25}$$

*where*

$$Q^{pw}(t,x,y) = -\frac{1}{8\pi^2}\frac{\partial}{\partial t}\int_{\mathbb{S}^2} q\left(t-\hat{s}\cdot(x-y),y\right) d\hat{s}, \tag{1.26}$$

$$Q^{g}(t,x,y) = \frac{q(t+\|x-y\|,y)}{4\pi\|x-y\|}. \tag{1.27}$$

*Proof.* Consider the expression for $Q^{pw}$ and rewrite the integral over the sphere in spherical coordinates. Without loss of generality, we assume that $x-y = (0,\ 0,\ \|x-y\|)$:

$$\begin{aligned} Q^{pw}(t,x,y) &= -\frac{1}{8\pi^2}\frac{\partial}{\partial t}\int_0^{2\pi}\int_0^{\pi} q(t-\|x-y\|\cos\theta,y)\sin\theta d\theta d\phi \\ &= -\frac{1}{4\pi}\frac{\partial}{\partial t}\int_{-1}^{1} q(t-\|x-y\|\xi,y) d\xi \\ &= \frac{q(t-\|x-y\|,y)}{4\pi\|x-y\|} - \frac{q(t+\|x-y\|,y)}{4\pi\|x-y\|}. \end{aligned}$$

□

In the frequency domain $Q^{pw}(t,x,y)$ can be represented in a plane-wave basis. To deal with the term $Q^g(t,x,y)$ ('ghost signal'), the signal $Q(t,x,y)$ is split into a sum of compactly supported and approximately temporally bandlimited functions $Q_\ell(t,x,y)$, $\ell = 1,\dots L$. This should be done so that for each $\ell = 1,\dots,L$ the 'ghost signal' $Q^g_\ell(t,x,y)$, defined as in (1.27), vanishes in a sufficiently large subdomain $D_\ell$ of $\Gamma\times\Gamma$.

Let $\sigma_x,\sigma_y \subset \Gamma$ be sufficiently distant domains with $\sigma_x\times\sigma_y \in D_\ell$. Let points $x_0\in\sigma_x$, $y_0\in\sigma_y$ be fixed. Then, for all $x\in\sigma_x$, $y\in\sigma_y$, the following formally holds [85]:

$$Q^{pw}_\ell(t,x,y) = \int\limits_{\mathbb{S}^2}\ \int\limits_0^{t-\hat{s}\cdot(x-x_0)} \delta(t-t_1-\hat{s}\cdot(x-x_0))V_\ell(t_1,x_0,y,\hat{s})dt_1 d\hat{s},$$

where

$$V_\ell(t_1,x_0,y,\hat{s}) = \int\limits_0^{t_1-\hat{s}\cdot(x_0-y_0)} T(t_1-t_2-\hat{s}\cdot(x_0-y_0))P_\ell(t_2,y,\hat{s})dt_2,$$

$$T(t) = -\frac{1}{8\pi^2}\frac{\partial}{\partial t}\delta(t)$$

and

$$P_\ell(t_2,y,\hat{s}) = \int\limits_0^{t_2+\hat{s}\cdot(y-y_0)} \delta(t_2-\tau+\hat{s}\cdot(y-y_0))Q_\ell(\tau,y)d\tau, \qquad \ell = 1,\dots,L.$$

This expression serves as a basis for derivation of the plane-wave time-domain multipole expansions.

The algorithm in [85] is based on Whittacket-type plane-wave representation of transient fields. It is also possible to use other plane-wave representations [86], though the corresponding methods can appear to be more difficult to implement, see also [135].

In spite of being asymptotically of almost optimal complexity, the PWTD algorithm is known to outperform other fast solvers, e.g. time-domain adaptive integral method (TDAIM), only for problems characterized by not less than tenths of thousands spatial unknowns, and in practice can appear to be less efficient than TDAIM for quasiplanar structures [14]. We briefly describe the latter method in the next section.

### 1.2.5 Time-Domain Adaptive Integral Method

The TDAIM is based on the recursive algorithm of Section 1.2.3. The key observation is that for rectangular domains the matrix-vector products with matrices (1.22) are essentially convolutions in space and can be evaluated with the help of further FFTs [189]. Hence, 4D-FFT can be employed for efficient computation of (1.20).

The algorithm works for an arbitrary domain, however, to make use of the FFT-accelerated spatial convolutions on uniform meshes, it employs the ideas from [42]. The method described in this work, frequency-domain adaptive integral method, deals with the fast evaluation of the Galerkin matrix-vector product

$$b_\alpha = \sum_{\beta=1}^{M}\iint\limits_{\Gamma\times\Gamma}\frac{\mathrm{e}^{i\kappa\|x-y\|}}{4\pi\|x-y\|}a_\beta\phi_\beta(x)\phi_\alpha(y)d\Gamma_x d\Gamma_y, \qquad \alpha = 1,\dots,M,\ \kappa\in\mathbb{R}.$$

The main idea of this algorithm is to substitute the integral over $\Gamma \times \Gamma$ by the integral of a modified integrand over $B \times B$, where $B$ is a three-dimensional box enclosing the domain. The kernel function remains unaffected, while the Galerkin basis functions have to be changed. The resulting integral can be efficiently evaluated with the help of FFT techniques. The time-domain adaptive integral method adapts this idea to compute matrix-vector products (1.22).

The complexity of the approach scales as

$$O(NM^{\frac{3}{2}} \log^2 M)$$

for 3D surfaces and

$$O(NM \log^2 M)$$

for quasi-planar surfaces (e.g. surfaces of very small height). The required memory scales as $O(M^2)$ for 3D surfaces and $O(M^{\frac{3}{2}})$ for quasi-planar surfaces. For more details on the complexity analysis see [14]. This algorithm relies on the FFT solely and does not require the use of non-standard techniques, while being easily parallelizable. However, its memory requirements are not improved compared to conventional MOT solvers.

In the next section we describe one of the latest, to our knowledge, algorithms for the efficient evaluation of (1.23).

### 1.2.6 Multilevel (Cartesian) Non-Uniform Grid Time-Domain Algorithm

This method was developed to overcome the main difficulties associated with the fast methods described previously. Namely, the PWTD algorithm is difficult to implement and adapt to the mixed frequency regime (i.e. when the spectrum of the incident wave has both high and low frequencies), while the TDAIM is not applicable in the mixed frequency regime at all, according to [143].

The detailed description of the algorithm can be found in [43, 143]. The method aims at the fast evaluation of

$$f(t,x) = \int_{\Gamma} \frac{q(t - \|x-y\|, y)}{4\pi\|x-y\|} d\Gamma_y, \tag{1.28}$$

where $q(t,x)$ is a bandlimited (with the bandlimit $\omega$) causal function.

Let us describe a two-level version of this algorithm, due to [43]. The surface of the scatterer is subdivided into $P$ non-overlapping domains

$$\Gamma = \bigcup_{p=1}^{P} \Gamma_p$$

of approximately equal size. The center of the smallest sphere surrounding the domain $\Gamma_p$ we denote by $x_p$ and its radius by $r_p$. Then,

$$\begin{aligned} f(t,x) &= \int_{\Gamma} \frac{q(t - \|x-y\|, y)}{4\pi\|x-y\|} d\Gamma_y = \sum_{p=1}^{P} f_p(t,x), \\ f_p(t,x) &= \int_{\Gamma_p} \frac{q(t - \|x-y\|, y)}{4\pi\|x-y\|} d\Gamma_y, \qquad p = 1, \ldots, P. \end{aligned} \tag{1.29}$$

For each of the domains $\Gamma_p$ one can define the near-field zone $\Gamma_p^N$ (points $x \in \Gamma$ lying close to or inside this domain) and the far-field zone $\Gamma_p^F = \Gamma \setminus \Gamma_p^N$. The evaluation (1.29) for each $p = 1, \ldots, P$ is done directly if $x \in \Gamma_p^N$. For $x$ in the far-field zone, one defines an auxiliary quantity ('compensated field')

$$\tilde{f}_p(\tau, x) = \|x - x_p\| \int_{\Gamma_p} \frac{q(\tau + \tilde{x}_p(x) - \|x - y\|, y)}{4\pi\|x - y\|} d\Gamma_y, \tag{1.30}$$

where $\tilde{x}_p = \sqrt{\|x - x_p\|^2 + \frac{r_p^2}{2}}$ and $\tau = t - \tilde{x}_p(x)$. Then the actual field is

$$f_p(t, x) = \frac{1}{\|x - x_p\|} \tilde{f}_p(t - \tilde{x}_p, x). \tag{1.31}$$

Importantly, $\tilde{f}_p(t, x)$ and $f_p(t, x)$ for fixed $t$ are smooth in $x \in \Gamma_p^F$. Introducing a grid (e.g. Cartesian) surrounding the far-field zone of $\Gamma_p$, $\tilde{f}_p(\tau, x)$ is evaluated using (1.30) in points of this grid $(x_k)_{k=1}^{K_p}$. The choice of $K_p$ depends on the size of $\Gamma_p$, the bandwidth of the function $q$ and the desired accuracy. Typically, given a function $q(t, x)$ (approximately) temporally bandlimited to maximum frequency $\omega$, it is chosen as $K_p = O(\omega^2 r_p^2) + O(1)$. Next, $\tilde{f}_p$ is interpolated to the surface, namely

$$\tilde{f}(\tau, x) = \sum_{k: x_k \in \sigma(x)} w_k^p(x) \tilde{f}_p(\tau, x_k),$$

where $\sigma(x)$ is a neighborhood of $x$ and $w_k^p$ are interpolation weights. Finally, $f_p(t, x)$ is restored using (1.31). Such procedure is shown to be significantly more efficient than the direct computation of (1.29).

The multilevel version of this algorithm has a butterfly-like structure, see [148]. The complexity of the evaluation (1.28) in the discretized problem using the multilevel algorithm is as low as $O(N_t N_s \log^\alpha N_s)$, with $\alpha = 1, 2$, where $N_s$ is size of the spatial discretization and $N_t$ is the number of the time steps.

### 1.2.7 Fast Galerkin Methods

The use of MOT methods may present two difficulties: instabilities on long times and instabilities for non-convex obstacles [156]. These problems can be successfully overcome with the use of specially suited temporal basis functions, see e.g. [185], and averaging/filtering techniques [158]. Recently, in [174] the authors introduced a new spatial quadrature that allows the stable and accurate implementation of MOT based methods.

Another way to deal with this kind of issues is to employ instead of the collocation in time the Galerkin discretization method, as it was originally done in the seminal works [18, 19]. The theory of Galerkin methods for time-domain boundary integral equations is well-developed, see [108]. Unlike the MOT method, the Galerkin method is based on the integral formulation

$$\int_0^T \int_\Gamma \int_\Gamma \frac{\dot{\lambda}(t - \|x - y\|, y)}{4\pi\|x - y\|} \xi(t, x) d\Gamma_x d\Gamma_y dt = \int_0^T \int_\Gamma \dot{g}(t, x) \xi(t, x) d\Gamma_x dt, \tag{1.32}$$

for all functions $\xi$. Here, $T$ is the length of the time interval. Some justification of the well-posedness of this formulation is given in [108]. Let us remark that in [18] the coercivity and the stability of a slightly different variational formulation was rigorously proved.

Given the boundary element basis

$$\Big\{\phi_i(x)\psi_j(t),\ i=1,\dots,M,\ j=1,\dots,N\Big\},$$

the solution of (1.32) is represented in the form

$$\lambda(t,x)=\sum_{i=1}^{M}\sum_{j=1}^{N}\lambda_j^i\phi_i(x)\psi_j(t),$$
$$\lambda_j^i=\int_0^T\int_\Gamma\lambda(t,x)\phi_i(x)\psi_j(t)d\Gamma_x dt.$$

This leads to the system of equations

$$\sum_{m=1}^{M}\sum_{n=1}^{N}\lambda_n^m\int_0^T\iint_{\Gamma\times\Gamma}\frac{\dot\psi_n(t)}{4\pi\|x-y\|}\phi_m(x)\phi_i(y)\psi_k(t)d\Gamma_y d\Gamma_x dt=\int_0^T\int_\Gamma\dot g(t,x)\phi_i(y)\psi_k(t)d\Gamma_y dt,$$
$$i=1,\dots,M,\ k=1,\dots,N.$$

Typically one chooses hat functions as a temporal basis, see [18, 109]. In the same works the authors describe how to deal with rather difficult spatial quadratures inherent to this kind of basis sets. Such choice has other major advantages, namely, the system of equations is lower triangular Toeplitz and matrix blocks

$$\begin{aligned}\mathbb{A}^{nm}&=\left(a_{ij}^{nm}\right)_{i,j=1}^{M},\\ a_{ij}^{nm}&=\int_0^T\iint_{\Gamma\times\Gamma}\frac{\dot\psi_n(t)\psi_m(t)}{4\pi\|x-y\|}\phi_i(y)\phi_j(x)d\Gamma_y d\Gamma_x dt\end{aligned}\tag{1.33}$$

are sparse, similarly to that of collocation (MOT) methods. As we already mentioned, many fast methods developed initially for marching-on-in-time, are applicable to Galerkin methods of this class. The commercial code SONATE, see [2], is based on the Galerkin time-domain method.

Recently, in [128, 165, 166] a new Galerkin method was developed. It employs infinitely smooth basis functions (based on the PUM (partition of unity method) of [13]) in time. This allows to overcome difficulties connected to the spatial integration and preserves sparsity of the matrix blocks (1.33). Efficient evaluation of elements of the system matrix can be done using tensor decomposition techniques, see [128]. However, the system in this case is no longer lower triangular Toeplitz.

There exist other variational formulations, e.g. energetic boundary integral formulation [5, 6]. Recently, it was applied to the Neumann exterior problem in three dimensions in [4]; in the same reference computational aspects of this method are discussed in detail.

### 1.2.8 Convolution Quadrature

The convolution quadrature (CQ) algorithm dates back to 1988 [136, 137], where it was applied to approximate numerically convolution integrals

$$g(t) = \int_0^t k(t-\tau)v(\tau)d\tau. \tag{1.34}$$

This method requires only the Laplace transform of $k(t)$ to be known and was originally based on the use of multistep methods. In 1993, convolution quadrature was re-formulated for Runge-Kutta methods and successfully applied to semilinear parabolic evolution equations [139]. Though there exist several approaches to introduce convolution quadrature, we choose the one from these seminal works. Before introducing Runge-Kutta convolution quadrature, we need a few basic definitions from the theory of numerical solutions of ordinary differential equations.

#### 1.2.8.1 Preliminaries: Runge-Kutta Methods

The material of this section is well-known and can be found in classical monographes [49, 122, 123].

We consider the initial-value problem on $[0,\ T]$

$$\begin{aligned} y' &= f(t,y), \\ y(0) &= y^0. \end{aligned} \tag{1.35}$$

Let $f$ be Lipschitz continuous in $y$ and continuous in $t$, so that by Picard-Lindelöf theorem the problem (1.35) has a unique solution. The interval $[0,\ T]$ is subdivided into $N+1$ subintervals of size $h = \frac{T}{N}$. We denote by

$$y_n \approx y(nh)$$

a Runge-Kutta approximation to $y(t)$ at time $t_n = nh$.

Let an $m$-stage Runge-Kutta method be given by the Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}, \tag{1.36}$$

where $b, c \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times m}$.

The Runge-Kutta approximation of (1.35) is given by

$$\begin{aligned} Y_{nj} &= y_n + h\sum_{i=1}^{m} a_{ji} f(t_n + c_i h, Y_{ni}), \qquad j = 1,\dots,m, \\ y_{n+1} &= y_n + h\sum_{j=1}^{m} b_j Y_{nj}, \\ y_0 &= y(0), \qquad n = 0,\dots,N-1. \end{aligned} \tag{1.37}$$

The values $Y_{nj}$ in (1.37) can be viewed as an approximation to $y(t)$ at time steps $t_{nj} = nh + c_j h$ ('internal stages').

**Definition 1.2.3.** *A Runge-Kutta method has order $p$ if for a sufficiently smooth problem (1.35)*

$$\|y(h) - y_1\| \leq Kh^{p+1},\ K > 0.$$

**Definition 1.2.4.** *The stage order of a Runge-Kutta method equals to $q$ if for a sufficiently smooth problem (1.35) for all $1 \leq j \leq m$,*

$$\|Y_{nj} - y_n\| \leq Ch^{q+1},\ C > 0.$$

Another important property of a Runge-Kutta method is its stability.

A Runge-Kutta method (1.36) is called *A-stable* if the numerical solution $y_n$ of the Dahlquist equation

$$\begin{aligned} y' &= \lambda y, \qquad \operatorname{Re}\lambda < 0, \\ y(0) &= 1, \end{aligned} \tag{1.38}$$

remains bounded for an arbitrary fixed timestep $h > 0$ and as $n \to +\infty$.

The numerical solution of the Dahlquist equation (1.38) can be alternatively written as:

$$y_n = R(h\lambda)^n,$$

where $R(z) = 1 + zb^T(I - Az)^{-1}\mathbb{1}$.

**Definition 1.2.5.** *The function $R(z) = 1 + zb^T(I - Az)^{-1}\mathbb{1}$, $\mathbb{1} = [1,\ \ldots,\ 1]^T$, is called the stability function of the Runge-Kutta method (1.36).*

**Definition 1.2.6.** *The set $S = \{z \in \mathbb{C} : |R(z)| \leq 1\}$ is called the stability domain of the Runge-Kutta method (1.36).*

**Definition 1.2.7.** *A Runge-Kutta method is A-stable if*

$$\mathbb{C}_- = \{z \in \mathbb{C} : \operatorname{Re} z \leq 0\} \subset S.$$

The following theorem is a direct consequence of definitions of the stability function and order of a Runge-Kutta method and can be found in [122, Theorem 2.2].

**Theorem 1.2.8.** *The stability function of a Runge-Kutta method of order $p$ is a rational approximation to the exponential of order $p$:*

$$R(z) = \mathrm{e}^z + Cz^{p+1} + O(z^{p+2}), \qquad C \neq 0,\ z \to 0. \tag{1.39}$$

Rational functions that, for a given degree of the numerator and denominator, have the highest order of approximation are called Padé approximations. The following theorem can be found in [122, Theorem 3.11].

**Theorem 1.2.9.** *The $(k, j)$-Padé approximation to $\mathrm{e}^z$ given by*

$$R_{kj}(z) = \frac{N_{kj}(z)}{Q_{kj}(z)},$$

*where $N_{kj}(z)$ is a polynomial of order $k$ and $Q_{kj}(z)$ is a polynomial of order $j$, is the unique rational approximation to $\mathrm{e}^z$ of the order $j + k$ s.t. the degrees of the numerator and the denominator are $k$ and $j$, respectively.*

In view of Theorem 1.2.8, Ehle suggested [78] that $A$-stable Runge-Kutta methods can be constructed employing Padé approximations to the exponential as $R(z)$ . In [79] stability properties of such approximations were discussed. The following result was conjectured and partially proved in the same reference, while the rest of the proof was done in the seminal paper [183].

**Theorem 1.2.10** (Ehle's Conjecture)**.** *Any Padé approximation* $R(z) = \frac{P(z)}{Q(z)}$, $\deg P = k$, $\deg Q = n$ *is A-stable iff* $n - 2 \leq k \leq n$.

The next result will be of use later and can be found in [79].

**Theorem 1.2.11.** *All zeros of Padé approximants satisfying conditions of Theorem 1.2.10 lie in the open left-half plane.*

Another important concept is $L$-stability.

**Definition 1.2.12.** *A Runge-Kutta method with the stability function* $R(z)$ *is called L-stable if it is A-stable and additionally*

$$R(\infty) = 0.$$

We will assume that the matrix $A$ is nonsingular.

**Definition 1.2.13.** *The Runge-Kutta method with the nonsingular matrix A is called stiffly accurate if* $c_m = 1$ *and*

$$a_{mj} = b_j,\ j = 1, \ldots, m.$$

For stiffly accurate Runge-Kutta methods,

$$b^T A^{-1} = [0,\ \ldots,\ 1]\,. \tag{1.40}$$

It is well known that the stability function for a stiffly accurate Runge-Kutta method can be written in a simpler form:

$$\begin{aligned} R(z) &= 1 + z b^T (I - Az)^{-1} \mathbb{1} = b^T A^{-1}(I - Az)(I - Az)^{-1}\mathbb{1} + z b^T (I - Az)^{-1}\mathbb{1} \\ &= b^T (A^{-1} - z + z)(I - Az)^{-1}\mathbb{1} = b^T A^{-1}(I - Az)^{-1}\mathbb{1}. \end{aligned} \tag{1.41}$$

The next proposition [122, Proposition 3.8] connects $A$-stability, $L$-stability and stiff accuracy.

**Proposition 1.2.14.** *Stiffly accurate A-stable methods are L-stable.*

#### 1.2.8.2 Derivation of Runge-Kutta Convolution Quadrature

Let us consider the convolution equation

$$g(t) = K(\partial_t)\lambda(t) = \int_0^t k(t-\tau)\lambda(\tau)d\tau, \qquad 0 < t < \infty,$$

where $K(\partial_t)$ is defined in Section 1. Additionally, we assume that $\mu < 0$ in (1.6). The results can be extended to the case $\mu \geq 0$ using (1.7). As before, $\lambda$ and $g$ are causal.

We can rewrite the above equation substituting $k(t)$ by the Bromwich integral of its Laplace transform $K(s)$, $\operatorname{Re} s > 0$:

$$g(t) = \frac{1}{2\pi i} \int\limits_{\sigma+i\mathbb{R}} K(s) \int\limits_0^t \mathrm{e}^{s(t-\tau)} \lambda(\tau) d\tau ds, \tag{1.42}$$

where $\sigma > \sigma_0$, see Section 1. The integral $v(t) = \int\limits_0^t \mathrm{e}^{s(t-\tau)} \lambda(\tau) d\tau$ solves the following ODE:

$$\begin{aligned} v'(\tau) &= sv(\tau) + \lambda(\tau), \\ v(0) &= 0. \end{aligned} \tag{1.43}$$

Therefore, we can substitute $v(t)$ with the numerical approximation of this ODE obtained with the help of a linear multistep or Runge-Kutta method. The latter will be used in the current work.

The time interval $[0,\ T]$ is subdivided into $N$ equal time steps of size $h$. By $g_n$ and $\boldsymbol{g}_n$ we denote:

$$g_n = g(nh), \qquad \boldsymbol{g}_n = \begin{pmatrix} g(nh + c_1 h) \\ \vdots \\ g(nh + c_m h) \end{pmatrix}.$$

Similarly $v_n$, $\boldsymbol{v}_n$, $\lambda_n$, $\boldsymbol{\lambda}_n$ are defined. We will use Runge-Kutta methods with the nonsingular matrix $A$ that satisfy the following assumptions.

**Assumption 1.2.15.** *(a) A-stability;*

*(b) stiff accuracy;*

*(c) for all $y \neq 0$, $|R(iy)| < 1$.*

These assumptions originate from the theory of Runge-Kutta convolution quadrature, see [28]. It is possible to weaken them, see [139] for the Runge-Kutta convolution quadrature derivation for parabolic problems, or [21] for the use of trapezoidal rule for the scattering problem.

The Runge-Kutta discretization of (1.43) is then given by

$$\begin{aligned} \boldsymbol{v}_n &= v_n \mathbb{1} + hA \left( s\boldsymbol{v}_n + \boldsymbol{\lambda}_n \right), \\ v_{n+1} &= v_n + hb^T \left( s\boldsymbol{v}_n + \boldsymbol{\lambda}_n \right), \\ v_0 &= 0. \end{aligned} \tag{1.44}$$

For $\boldsymbol{v}(\xi) = \sum\limits_{n=0}^{\infty} \boldsymbol{v}_n \xi^n$, $\boldsymbol{\lambda}(\xi) = \sum\limits_{n=0}^{\infty} \boldsymbol{\lambda}_n \xi^n$, $|\xi| < 1$, (1.44) gives

$$\boldsymbol{v}(\xi) = \left( \frac{\Delta(\xi)}{h} - s \right)^{-1} \boldsymbol{\lambda}(\xi), \tag{1.45}$$

where the matrix-valued function $\Delta(\xi)$ for Runge-Kutta methods under consideration is defined as (see [21, 139])

$$\Delta(\xi) = \left( A + \frac{\xi}{1-\xi} \mathbb{1} b^T \right)^{-1} = A^{-1} - \xi A^{-1} \mathbb{1} b^T A^{-1},\ |\xi| < 1. \tag{1.46}$$

Substituting $v(t) = \int_0^t \mathrm{e}^{s(t-\tau)}\lambda(\tau)d\tau$ in (1.42) with its numerical approximation (1.45) allows to obtain the following semi-discretized equation:

$$\begin{aligned}\boldsymbol{g}(\xi) := \sum_{n=0}^{\infty} \boldsymbol{g}_n(x)\xi^n &= \frac{1}{2\pi i}\int_{\sigma+i\mathbb{R}} K(s)\boldsymbol{\lambda}(\xi)\left(\frac{\Delta(\xi)}{h} - s\right)^{-1} ds \\ &= K\left(\frac{\Delta(\xi)}{h}\right)\boldsymbol{\lambda}(\xi),\ |\xi| < 1, \end{aligned} \tag{1.47}$$

where the last formula was obtained from the application of Cauchy's integral theorem, using the bound (1.6) with $\mu < 0$ for $K(s)$. Next, the convolution kernel $K\left(\frac{\delta(\xi)}{h}\right)$ is expanded into a series in $\xi$:

$$K(\xi) = \sum_{n=0}^{\infty} W_n^h(K)\xi^n,\ |\xi| < 1. \tag{1.48}$$

The coefficients of this expansion, $W_n^h(K)$, are called convolution weights. Inserting (1.48) into (1.47) and matching the powers of $\xi$ in the obtained expression gives the following equation for $\boldsymbol{g}_n$:

$$\boldsymbol{g}_n = \sum_{k=0}^{n} W_{n-k}^h(K)\boldsymbol{\lambda}_k, \qquad n = 0, \ldots, N. \tag{1.49}$$

When an $m$-stage Runge-Kutta method is used, the convolution weights $W_j^h(\mathcal{K})$, $j = 0, \ldots, N$, are matrices of size $m \times m$.

Now we apply the Runge-Kutta convolution quadrature discretization to (1.12):

$$g(t,x) = \int_0^t \int_\Gamma \frac{\delta(t-\tau-\|x-y\|)}{4\pi\|x-y\|}\lambda(\tau,y)d\Gamma_y d\tau. \tag{1.50}$$

In our case $K(s)$ is the boundary Helmholtz single-layer operator

$$\left(\mathcal{V}(s)\phi\right)(x) = \int_\Gamma \frac{\mathrm{e}^{-\operatorname{Re} s\|x-y\|}}{4\pi\|x-y\|}\phi(y)d\Gamma_y, \qquad x \in \Gamma.$$

Hence, convolution weights $W_n^h$ are boundary integral operators

$$\left(W_n^h\left(\mathcal{V}\right)\boldsymbol{\lambda}\right)(x) = \int_\Gamma w_n^h(\|x-y\|)\boldsymbol{\lambda}(y)d\Gamma_y, \qquad x \in \Gamma, \tag{1.51}$$

whose kernels $w_n^h(\|x-y\|)$ are coefficients of the expansion of $\mathcal{K}_d(\xi) = \frac{\exp\left(-\frac{\Delta(\xi)}{h}d\right)}{4\pi d}$ into the Taylor series in $\xi$, i.e.

$$\mathcal{K}_d(\xi) = \frac{\exp\left(-\frac{\Delta(\xi)}{h}d\right)}{4\pi d} = \sum_{n=0}^{\infty} w_n^h\left(d\right)\xi^n. \tag{1.52}$$

If for the discretization an $m$-stage Runge-Kutta method is used, the convolution weights are continuous operators

$$W_j^h : \left(H^{-\frac{1}{2}}(\Gamma)\right)^m \to \left(H^{\frac{1}{2}}(\Gamma)\right)^m,$$

where by $X^m$ we denote the space of $m$-dimensional vectors of elements of $X$, namely

$$\begin{aligned} X^m &= \left\{ f = [f_1, \dots, f_m]^T, \qquad f_j \in X, \qquad j = 1, \dots m \right\}, \\ \|f\|_{X^m}^2 &= \sum_{j=1}^m \|f_j\|_X^2. \end{aligned}$$

**Remark 1.2.16.** *In the convolution quadrature theory both operators $W_j^h$ and their kernels $w_j^h(d)$ are called convolution weights. In the course of the work we try to make sure that it is always clear which convolution weights, $W_j^h$ or $w_j^h(d)$, are in question.*

The Runge-Kutta convolution quadrature discretization of (1.50) is written as

$$\begin{aligned} \boldsymbol{g}_n(x) &= \sum_{i=0}^n \left( W_{n-i}^h(\mathcal{V})\, \boldsymbol{\lambda}_i \right)(x), \\ g_{n+1}(x) &= b^T A^{-1} \sum_{i=0}^n \left( W_{n-i}^h(\mathcal{V})\, \boldsymbol{\lambda}_i \right)(x), \end{aligned} \tag{1.53}$$

where the last expression was obtained from Definition 1.2.13 and (1.40).

Equations (1.53) for $m = 0, \dots, N$ form the lower triangular block Toeplitz system

$$\begin{pmatrix} W_0^h & 0 & \cdots & 0 \\ W_1^h & W_0^h & \cdots & 0 \\ \cdots & & & \\ W_N^h & W_{N-1}^h & \cdots & W_0^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_N \end{pmatrix}. \tag{1.54}$$

#### 1.2.8.3 Convolution Weights

Let $K(s)$ satisfy the assumptions of Section 1.1. Namely, given $\sigma_0 \in \mathbb{R}$, $K(s)$ is an analytic mapping from $\{s \in \mathbb{C} : \operatorname{Re} s > \sigma_0\}$ to the space $L(X, Y)$ of continuous operators from $X$ to $Y$. Additionally, it is bounded by (see (1.6)):

$$\|K(s)\|_{X \to Y} < M|s|^\mu, \ \mu \in \mathbb{R}. \tag{1.55}$$

W.l.o.g. we assume $\sigma_0 > 0$.

Weights $W_n^h$ of convolution quadrature based on Runge-Kutta methods satisfying Assumption 1.2.15 are well-defined [28]. Lemma 3.1.7 shows that the eigenvalues of the matrix $\Delta(\xi)$, $|\xi| < 1$, lie on the right half of the complex plane, and hence, for sufficiently small $h > 0$, the map $\xi \to K\left(\frac{\Delta(\xi)}{h}\right)$ is analytic. Our goal is to find bounds on norms of convolution weights $W_n^h$ in terms of $h$.

The following lemma is due to [27]. It shows that the norm $\left\| K\left(\frac{\Delta(\xi)}{h}\right) \right\|_{X^m \to Y^m}$ can be bounded by $O(h^{-\mu})$.

**Lemma 1.2.17.** *Let $\mu \geq 0$ in (1.55). Let an $m$-stage Runge-Kutta method satisfy Assumption 1.2.15. Then for every $\tilde{\sigma}_0 > \sigma_0$ there exists $h_0$, s.t. for all $h < h_0$ the eigenvalues of $\Delta(\xi)$ for $|\xi| \leq \mathrm{e}^{-h\tilde{\sigma}_0}$ lie in the half-plane $\operatorname{Re} z \geq h\sigma_0$ and*

$$\sup_{|\xi| \leq \mathrm{e}^{-h\tilde{\sigma}_0}} \left\| K\left(\frac{\Delta(\xi)}{h}\right) \right\|_{X^m \to Y^m} \leq CMh^{-\mu},$$

*where $C > 0$ depends only on the Runge-Kutta method.*

The following lemma is a corollary of the above statement.

**Lemma 1.2.18.** *Let $\mu \geq 0$ in (1.55). Let an $m$-stage Runge-Kutta method satisfy Assumption 1.2.15. For all $\tilde{\sigma}_0 > \sigma_0$ there exists $\bar{h} > 0$, s.t. for all $0 < h < \bar{h}$ and $n \in \mathbb{N}_0 : \ 0 \leq n \leq \frac{1}{h\tilde{\sigma}_0}$,*

$$\left\| W_n^h(K) \right\|_{X^m \to Y^m} \leq CMh^{-\mu},$$

*where $C$ depends on the Runge-Kutta method only.*

*Proof.* The Cauchy's integral theorem gives an explicit expression for the convolution weight, see (1.48),

$$W_n^h(K) = \frac{1}{2\pi i} \oint_{\mathcal{C}} \frac{K\left(\frac{\Delta(\zeta)}{h}\right)}{\zeta^{n+1}} d\zeta,$$

where $\mathcal{C}$ is a contour lying within the domain of analyticity of $K\left(\frac{\Delta(\zeta)}{h}\right)$ and enclosing 0. Let us fix $\tilde{\sigma}_0 > \sigma_0$. The previous lemma shows that there exists $h_0$, s.t. for all $0 < h < h_0$ this contour can be chosen as the circle $|\zeta| = \mathrm{e}^{-h\tilde{\sigma}_0}$. Therefore,

$$\begin{aligned} \left\| W_n^h \right\|_{X^m \to Y^m} &\leq \frac{1}{2\pi} \oint_{|\zeta|=1} \left\| K\left(\frac{\Delta(\zeta)}{h}\right) \right\|_{X^m \to Y^m} \mathrm{e}^{nh\tilde{\sigma}_0} \, |d\zeta| \\ &\leq CM\mathrm{e}^{nh\tilde{\sigma}_0} h^{-\mu}, \end{aligned}$$

for some $C > 0$ depending on the Runge-Kutta method only. For $n \leq \frac{1}{h\tilde{\sigma}_0}$

$$\mathrm{e}^{nh\tilde{\sigma}_0} \leq \mathrm{e},$$

from which the statement of the lemma follows. □

This lemma can be applied to bound the convolution weights of the boundary single-layer operator and its inverse for the wave equation on the finite interval $[0, \ T]$, $T > 0$. The proof of the following corollary is similar to the proof of Lemma 5.3 in [29].

**Corollary 1.2.19.** *Let an $m$-stage Runge-Kutta method satisfy Assumption 1.2.15. Then the convolution weights $W_n^h(\mathcal{V})$ and $W_n^h\left(\mathcal{V}^{-1}\right)$ are bounded linear operators from $\left(H^{-\frac{1}{2}}(\Gamma)\right)^m$ to $\left(H^{\frac{1}{2}}(\Gamma)\right)^m$ and from $\left(H^{\frac{1}{2}}(\Gamma)\right)^m$ to $\left(H^{-\frac{1}{2}}(\Gamma)\right)^m$ correspondingly.*

*Let $T > 0$. Then there exists $\bar{h}$, s.t. for all $0 < h < \bar{h}$ and $N \in \mathbb{N}_0 : \ 0 \le N \le \frac{T}{h}$, the norms of convolution weights can be bounded as follows:*

$$\|W_n^h(\mathcal{V})\|_{\left(H^{-\frac{1}{2}}(\Gamma)\right)^m \to \left(H^{\frac{1}{2}}(\Gamma)\right)^m} \le c_1 \max(T^3, T) h^{-1},$$
$$\|W_n^h\left(\mathcal{V}^{-1}\right)\|_{\left(H^{\frac{1}{2}}(\Gamma)\right)^m \to \left(H^{-\frac{1}{2}}(\Gamma)\right)^m} \le c_2 \max\left(T^2, T\right) h^{-2},$$

*for some $c_1,\ c_2 > 0$ that do not depend on $h, n, T$.*

*Proof.* According to Proposition 1.1.2, for all $\sigma_0 > 0$ and $s \in \mathbb{C} : \operatorname{Re} s > \sigma_0$,

$$\|\mathcal{V}(s)\|_{H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma)} \le \frac{C_1}{\sigma_0} \max\left(\frac{1}{\sigma_0^2}, 1\right) |s|,$$
$$\|\mathcal{V}^{-1}(s)\|_{H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma)} \le \frac{C_2}{\sigma_0} \max\left(\frac{1}{\sigma_0}, 1\right) |s|^2.$$

Let us first derive the bound for $W_n^h(\mathcal{V})$, $n \ge 0$. Since the choice of $\sigma_0 > 0$ in the above bounds can be done arbitrarily, we can set $\sigma_0 = \frac{1}{T}$, $T > 0$. Then, for all $s : \operatorname{Re} s > \sigma_0$,

$$\|\mathcal{V}(s)\|_{H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma)} \le C_1 T \max(T^2, 1)|s|.$$

This, combined with the statement of the previous lemma gives the desired bounds. Similarly, the bounds for convolution weights $W_n^h\left(\mathcal{V}^{-1}\right)$ can be derived. □

#### 1.2.8.4 Convergence of Runge-Kutta Convolution Quadrature

The questions of convergence of Runge-Kutta convolution quadrature were studied in [21, 27, 28].

The following theorem provides sharp bounds on the order of convergence of Runge-Kutta convolution quadrature for (1.50). As before, $p$ denotes the classical order of Runge-Kutta methods, and $q$ is the stage order.

**Theorem 1.2.20.** *[28, Theorem 5] Let $\ell > q + 3$. If $g(0) = \ldots = \partial_t^{\ell-1} g(0) = 0$, there exists $\bar{h}$ s.t. for all $0 < h < \bar{h}$ and $t_n = nh$, $0 < n \le N = \left\lceil \frac{T}{h} \right\rceil$, it holds:*

$$\|\lambda_n(.) - \lambda(t_n, .)\|_{H^{-\frac{1}{2}}(\Gamma)} \le C h^q \left( \|\partial_t^\ell g(0, .)\|_{H^{\frac{1}{2}}(\Gamma)} + \int_0^{t_n} \|\partial_t^{\ell+1} g(\tau, .)\|_{H^{\frac{1}{2}}(\Gamma)} d\tau \right),$$

*where $C$ depends on $\bar{h}$ and $T$.*

This theorem shows that the Runge-Kutta convolution quadrature semidiscretization of (1.50) converges with the reduced order $q$ rather than the classical order $p$. In the same work [28] it was shown that this is not the case if the solution is computed away from the boundary: the order of convergence of Runge-Kutta convolution quadrature coincides with the classical order of the Runge-Kutta method.

**Theorem 1.2.21.** *Let $u = \mathcal{S}(\partial_t) \mathcal{V}^{-1}(\partial_t) g$. Then the Runge-Kutta convolution quadrature discretization of this equation is identical to the Runge-Kutta semi-discretization of the scattering problem (1.1). If, additionally, for some $\ell > p + 4$,*

$$g(0) = \ldots = \partial_t^{\ell-1} g(0) = 0,$$

*then there exists* $\bar{h}$*, s.t. for all* $0 < h < \bar{h}$ *and* $x \in \Omega^c$ : $\operatorname{dist}(x, \Gamma) \geq \delta > 0$*,* $t_n = nh$*,* $0 < n \leq N = \lceil \frac{T}{h} \rceil$*, it holds:*

$$|u(nh, x) - u_n(x)| \leq C_\delta h^p \left( \|\partial_t^\ell g(0, .)\|_{H^{\frac{1}{2}}(\Gamma)} + \int_0^{t_n} \|\partial_t^{\ell+1} g(\tau, .)\|_{H^{\frac{1}{2}}(\Gamma)} d\tau \right),$$

*with* $C_\delta$ *depending on* $\bar{h}$*,* $T$ *and* $\delta$*.*

### 1.2.9 Sparse Multistep Convolution Quadrature

One of the ways to deal with (1.54) is to employ the back substitution algorithm, constructing the discretizations of integral operators $W_n^h$. This was done in [115, 116, 131] for the BDF2 (backward differentiation formula of the second order) method. In this case kernels of integral operators $W_n^h$ are known explicitly:

$$w_n^h(d) = \frac{1}{n!} \frac{1}{4\pi d} \left( \frac{d}{2h} \right)^{\frac{n}{2}} \mathrm{e}^{-\frac{3}{2}\frac{d}{h}} H_n \left( \sqrt{\frac{2d}{h}} \right),$$

where $H_n$ are Hermite polynomials. The approach is heavily based on the following property of the BDF2 convolution weights.

**Lemma 1.2.22.** *Let* $n \geq 1$ *and* $c_{n,\epsilon}^h = 3h\sqrt{n} \log \frac{1}{\epsilon}$*. For* $d \in I_{n,\epsilon}^h = \left[ nh - c_{n,\epsilon}^h, nh + c_{n,\epsilon}^h \right]$*, it holds*

$$|w_n^h(d)| \leq \frac{\epsilon}{4\pi d}.$$

The use of this lemma allows to skip evaluating some of the entries of Galerkin discretizations of boundary integral operators $W_n^h$. The resulting matrices, besides being sparse, are also blockwise low-rank, see [116, 131], hence $\mathcal{H}$- and $\mathcal{H}^2$-matrix techniques or the panel clustering method can be successfully employed.

Unlike the BDF2 case, the kernels of operators $W_n^h$ of Runge-Kutta convolution quadrature do not possess a simple structure. The following lemma provides a way to evaluate them. Let us consider the modified kernels

$$\omega_n^h(d) = 4\pi d w_n^h(d).$$

**Lemma 1.2.23.** *[124, 137] Given* $\epsilon > 0$*, for all* $n > 0$*, the choice* $\rho = (\sqrt{\epsilon})^{\frac{1}{n+1}}$ *ensures, for all* $k \leq n$*,* $h, d > 0$*,*

$$\left| \omega_k^h(d) - \frac{\rho^{-m}}{n+1} \sum_{\ell=0}^{n} \mathrm{e}^{-\Delta\left(\rho \mathrm{e}^{i\ell\frac{2\pi}{n+1}}\right)\frac{d}{h}} \mathrm{e}^{-i\frac{2\pi}{n+1}\ell k} \right| < C\sqrt{\epsilon}, \tag{1.56}$$

*where* $C$ *is a constant that can be bounded independently of* $n, k, \epsilon, d$*.*

As explained in Section 3.1.2, this method allows to compute convolution weights up to the accuracy $\sqrt{\epsilon_m}$, where $\epsilon_m$ is the machine precision. There we show that for a range of $d$ it is possible to achieve the accuracy $\epsilon_m$.

The construction of the Galerkin discretization of a convolution weight $W_k^h$ for a fixed $k$ requires the evaluation of the kernel $w_k^h(d)$ at some $d \geq 0$. Lemma 1.2.23 can be applied

in two ways. First, it provides means to evaluate $w_k^h(d)$ with the complexity $O(k)$, if the sum in (1.56) is computed directly. For large $k$ this approach is likely to be inefficient. On the other hand, if the sum in (1.56) is computed with the help of the FFT, one obtains the values $w_k^h(d)$, $k = 0, \ldots, n$ simultaneously in $O(n \log n)$ steps. To make use of this advantage, however, we need to construct several Galerkin discretizations of convolution weights $W_k^h$, $k = 0, \ldots, n$, also simultaneously. The efficient assembly of such matrices does not seem to be easily implementable.

We postpone the further discussion of the applicability of some of the ideas of this method to Runge-Kutta convolution quadrature to Section 3.2.

### 1.2.10 Fast Multipole BEM in Time-Domain

Similarly to sparse multistep convolution quadrature, the algorithm suggested in [159–161], or fast multipole BEM in time-domain, improves convolution quadrature based on back substitution. This method was implemented both for 2- and 3-dimensional problems. Here we just demonstrate how it can be used to solve the equation (1.12).

The application of back substitution to convolution quadrature requires the solution of $O(N)$ equations

$$W_0^h \boldsymbol{\lambda}_n = \boldsymbol{g}_n - \sum_{k=0}^{n-1} W_{n-k} \boldsymbol{\lambda}_k, \qquad n = 0, \ldots, N. \tag{1.57}$$

To evaluate the sums on the right-hand side of the above expression, the authors of [159–161] make use of (1.56), namely

$$W_k^h u \approx \frac{\rho^{-m}}{L+1} \sum_{\ell=0}^{L} \int_\Gamma \frac{\mathrm{e}^{-\Delta\left(\rho \mathrm{e}^{i\ell\frac{2\pi}{L+1}}\right)\frac{\|x-y\|}{h}}}{4\pi\|x-y\|} \mathrm{e}^{-i\frac{2\pi}{L+1}\ell k} u(y) d\Gamma_y, \qquad k = 0, \ldots, L,$$

where $L \geq N$. The substitution of the above into the sum on the right-hand side of (1.57) makes it possible to evaluate each of $O(N)$ such sums with the help of the FFT of size $N$ and $O(N)$ matrix-vector multiplications with discretized Helmholtz boundary single layer operators. Such matrix-vector products are accelerated with the help of the fast multipole method. In [159] the FMM of [107, 191] is employed.

Totally, this algorithm requires $O(N^2)$ matrix-vector multiplications to be computed. Under the assumption that the number of boundary elements $M = O(N^2)$ and the size of a boundary element

$$ch \leq \Delta x \leq C'h, \qquad C', c > 0,$$

see the related discussion in Section 1.2.11.3, the complexity of the FMM accelerated matrix-vector product is not better than $O(M^{\frac{3}{2}})$ (this is due to the use of the fast multipole method of [107,191]). This implies that the complexity of fast multipole BEM in time domain scales not better than $O\left(N^2 M^{\frac{3}{2}}\right)$.

### 1.2.11 Recursive Convolution Quadrature

The matrix of the system (1.54) is lower triangular Toeplitz. Hence, the recursive FFT-based algorithm described in detail in Section 1.2.3 can be applied to solve the above system of equations. It is based on two main procedures, `SolveBasic` and `Multiply`. Hence, we

only need to show how they can be performed efficiently in the context of convolution quadrature. In our description of these procedures we closely follow [29, 30, 121].

#### 1.2.11.1 Matrix-Vector Multiplication

We first describe the algorithm for the fast matrix-vector multiplication

$$\begin{pmatrix} \boldsymbol{r}_0 \\ \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_{n-\ell} \end{pmatrix} = \begin{pmatrix} W_\ell^h & W_{\ell-1}^h & \cdots & W_1^h \\ W_{\ell+1}^h & W_\ell^h & \cdots & W_2^h \\ \vdots & & & \\ W_n^h & W_{n-1}^h & \cdots & W_{n-\ell+1}^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_\ell \end{pmatrix} \tag{1.58}$$

in accordance to [30, 124, 137]. Here we present a rather informal description of this algorithm.

The vector $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_0, \ldots, \boldsymbol{\lambda}_\ell]^T$ is extended with $n - \ell$ zeros at the end, and the vector $\boldsymbol{r} = [\boldsymbol{r}_0, \ldots, \boldsymbol{r}_{n-\ell}]^T$ with extra $\ell$ elements at the beginning. Let

$$\boldsymbol{h} := [\boldsymbol{h}_0, \ldots, \boldsymbol{h}_n]^T, \qquad \boldsymbol{h}_k = \boldsymbol{r}_{k-\ell}, \qquad k = \ell, \ldots, n.$$

Then, for some $0 < \rho \leq 1$ (the need of which will be substantiated further) the matrix-vector product (1.58) can be rewritten in the form:

$$\begin{pmatrix} \boldsymbol{h}_0 \\ \boldsymbol{h}_1\rho \\ \vdots \\ \boldsymbol{h}_{\ell-1}\rho^{\ell-1} \\ \boldsymbol{h}_\ell\rho^\ell \\ \boldsymbol{h}_{\ell+1}\rho^{\ell+1} \\ \vdots \\ \boldsymbol{h}_n\rho^n \end{pmatrix} = \begin{pmatrix} W_0^h & W_n^h\rho^n & ... & W_1^h\rho \\ W_1^h\rho & W_0^h & ... & W_2^h\rho^2 \\ \vdots & & & \\ W_{\ell-1}^h\rho^{\ell-1} & W_{\ell-2}^h\rho^{\ell-2} & ... & W_\ell^h\rho^\ell \\ W_\ell^h\rho^\ell & W_{\ell-1}^h\rho^{\ell-1} & ... & W_{\ell+1}^h\rho^{\ell+1} \\ W_{\ell+1}^h\rho^{\ell+1} & W_\ell^h\rho^\ell & ... & W_{\ell+2}^h\rho^{\ell+2} \\ \vdots & & & \\ W_n^h\rho^n & W_{n-1}^h\rho^{n-1} & ... & W_0^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1\rho \\ \vdots \\ \boldsymbol{\lambda}_\ell\rho^\ell \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{1.59}$$

We denote

$$\boldsymbol{h}_\rho = \left[\boldsymbol{h}_0,\ \boldsymbol{h}_1\rho,\ \ldots,\ \boldsymbol{h}_{\ell-1}\rho^{\ell-1},\ \boldsymbol{h}_\ell\rho^\ell,\ \boldsymbol{h}_{\ell+1}\rho^{\ell+1},\ \ldots,\ \boldsymbol{h}_n\rho^n\right]^T,$$
$$\boldsymbol{p}_\rho = \left[\boldsymbol{\lambda}_0,\ \boldsymbol{\lambda}_1\rho,\ \ldots,\ \boldsymbol{\lambda}_\ell\rho^\ell,\ 0,\ \ldots,\ 0\right]^T.$$

The above matrix is circulant and can be diagonalized with the help of the Fourier transform $\mathcal{F}_{n+1}$ of size $n + 1$:

$$\boldsymbol{h}_\rho = \mathcal{F}_{n+1}^{-1} D_{n+1,\rho} \mathcal{F}_{n+1} \boldsymbol{p}_\rho. \tag{1.60}$$

Elements $\hat{d}_{jj}$ of the diagonal matrix $D_{n+1,\rho}$ are given by the following expression:

$$\hat{d}_{jj} = \sum_{k=0}^{n} W_k^h \rho^k e^{-i\frac{2\pi}{n+1}kj}, \qquad j = 0, \ldots, n.$$

With the help of (1.48), the above expression can be rewritten, for all $j = 0, \ldots, n$,

$$\hat{d}_{jj} = \mathcal{V}\left(-\frac{\Delta(\rho e^{-i\frac{2\pi}{n+1}j})}{h}\right) - \sum_{k=n+1}^{\infty} W_k^h \rho^k e^{-i\frac{2\pi}{n+1}kj} \tag{1.61}$$

Our task now is to show that it is possible to choose a parameter $\rho$ so that instead of a diagonal matrix $D_{n+1,\rho}$ we can use a diagonal matrix $\mathcal{D}_{n+1,\rho}$ with elements $d_{jj}$ given by:

$$d_{jj} = \mathcal{V}\left(-\frac{\Delta(\rho e^{-i\frac{2\pi}{n+1}j})}{h}\right), \qquad j = 0, \ldots, n.$$

Let us define $\boldsymbol{h}'_\rho$, $\boldsymbol{h}'$ as

$$\boldsymbol{h}'_\rho = \mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho, \tag{1.62}$$

$$\boldsymbol{h}'_j = \rho^{-j}\left(\boldsymbol{h}'_\rho\right)_j, \qquad j = 0,\ \ldots,\ n, \tag{1.63}$$

and examine the difference $\boldsymbol{h}' - \boldsymbol{h}$:

$$\boldsymbol{h}'_j - \boldsymbol{h}_j = -\frac{\rho^{-j}}{n+1}\sum_{r=0}^{n} \mathrm{e}^{i\frac{2\pi}{n+1}jr} \sum_{k=n+1}^{\infty} W_k^h \rho^k e^{-i\frac{2\pi}{n+1}kr} \sum_{q=0}^{\ell} \rho^q \boldsymbol{\lambda}_q \mathrm{e}^{-i\frac{2\pi}{n+1}rq} =$$

$$= \sum_{k=n+1}^{\infty} W_k^h \sum_{q=0}^{\ell} \boldsymbol{\lambda}_q \rho^{k+q-j} \delta_{(k+q-j) \bmod (n+1),0}, \qquad j = 0,\ \ldots,\ n,$$

where $\delta_{k,q}$ is Kronecker delta. Convolution weights $W_k^h$ are bounded operators

$$W_k^h :\ \left(H^{-\frac{1}{2}}(\Gamma)\right)^m \to \left(H^{\frac{1}{2}}(\Gamma)\right)^m.$$

This implies that there exists $C' > 0$ (that depends on the time step $h$), s.t.:

$$\|\boldsymbol{h}'_j - \boldsymbol{h}_j\|_{\left(H^{\frac{1}{2}}(\Gamma)\right)^m} \le C'\rho^{n+1} \sum_{k=0}^{\ell} \|\boldsymbol{\lambda}_k\|_{\left(H^{-\frac{1}{2}}(\Gamma)\right)^m}, \qquad j = 0, \ldots, n. \tag{1.64}$$

Choosing $\rho$ small enough one can ensure that (1.60) is well approximated by (1.62).

However, the numerical evaluation of (1.63) can be done only up to a certain accuracy (limited by the machine precision). Let numerically computed $\left(\mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho\right)$ be denoted by $\boldsymbol{v}$. Let the error of the evaluation of $\left(\mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho\right)$ be equal to $\epsilon_0$, i.e.

$$\left\|\left(\mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho\right)_k - \boldsymbol{v}_k\right\|_{\left(H^{\frac{1}{2}}(\Gamma)\right)^m} \le \epsilon_0 \sum_{j=0}^{n} \|\left(\boldsymbol{p}_\rho\right)_j\|_{\left(H^{-\frac{1}{2}}(\Gamma)\right)^m}, \qquad k = 0, \ldots, n.$$

Indeed, for $\rho < 1$,

$$\left\|\left(\mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho\right)_k - \boldsymbol{v}_k\right\|_{\left(H^{\frac{1}{2}}(\Gamma)\right)^m} \le \epsilon_0 \sum_{j=0}^{\ell} \|\boldsymbol{\lambda}_j\|_{\left(H^{-\frac{1}{2}}(\Gamma)\right)^m}, \qquad k = 0, \ldots, n.$$

Then (1.63) is computed with the precision $\rho^{-j}\epsilon_0$, $j = 0,\ \ldots, n$.

Hence there exists $C > 0$ (depending on the time step $h$), s.t.

$$\left\|\boldsymbol{h}_j - \rho^{-j}\left(\mathcal{F}_{n+1}^{-1}\mathcal{D}_{n+1,\rho}\mathcal{F}_{n+1}\boldsymbol{p}_\rho\right)_j\right\|_{\left(H^{\frac{1}{2}}(\Gamma)\right)^m} \le CE_j(\rho) \sum_{k=0}^{\ell} \|\boldsymbol{\lambda}_k\|_{\left(H^{-\frac{1}{2}}(\Gamma)\right)^m},$$

$$E_j(\rho) = \rho^{n+1} + \rho^{-j}\epsilon_0, \qquad j = 0 \ldots n.$$

The above expression attains its maximum for $j = n$; the minimum of $E_n(\rho)$, $\rho < 1$, is achieved when $\rho = \epsilon_0^{\frac{1}{2n+1}}$ and equals to $\sqrt{\epsilon_0^{1+\frac{1}{2n+1}}}$. Thus, we can compute the matrix-vector product (1.58) with the precision $\sqrt{\epsilon_0}$ in $O(n \log n)$ steps.

A more rigorous error analysis can be done as described in [29].

**Remark 1.2.24.** *The actual construction of the Galerkin discretization of the operator* $\mathcal{V}\left(\frac{\Delta(z)}{h}\right)$, $|z| \le 1$, *has to be done in two steps:*

1. *diagonalize the matrix* $\Delta(z)$ *by the eigenvalue decomposition:*

$$\Delta(z) = Q \operatorname{diag}[\lambda_1, \ldots, \lambda_m] Q^{-1}, \qquad Q \in \mathbb{C}^{m\times m},\ \lambda_j \in \mathbb{C},\ j = 1, \ldots, m.$$

   *In [21] it was demonstrated that for 2- and 3-stage Radau IIA methods there exists only a few values of* $z : |z| \le 1$ *s.t.* $\Delta(z)$ *is not diagonalizable (see also Figure 3.1). However, they are highly unlikely to be hit during the computation, see also Section 3.1. If nevertheless* $|z| = \rho$ *appears to be close to one of such values, one can slightly perturb the parameter* $\rho$*.*

   *For* $|z| = 1$ *and 2- and 3-stage Radau IIA methods the matrix* $\Delta(z)$ *is diagonalizable, see Remark 3.1.8.*

2. *compute the Galerkin discretization of* $\mathcal{V}\left(\frac{\lambda_j}{h}\right)$, $j = 1, \ldots, m$.

**Remark 1.2.25.** *Since in the time domain all the values are real, after the discrete Fourier transform because of symmetry only a half of matrix-vector multiplications need to be performed, the other half are obtained by complex conjugation. Therefore, it is sufficient to construct discretizations of the boundary integral operators* $\mathcal{V}\left(\frac{\Delta(\rho e^{-i\frac{2\pi}{n+1}j})}{h}\right)$ *for* $j = 0, \ldots, \lfloor\frac{n+1}{2}\rfloor$ *and compute matrix-vector products only with these matrices.*

#### 1.2.11.2 Solution of a Small System

A method for the solution of the small system

$$\begin{pmatrix} W_0^h & 0 & \cdots & 0 \\ W_1^h & W_0^h & \cdots & 0 \\ \vdots & & & \\ W_J^h & W_{J-1}^h & \cdots & W_0^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_J \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_J \end{pmatrix}. \tag{1.65}$$

presented below is due to [29] and bears similarities with [137], as well as with the method described in the previous section. Note that

$$\boldsymbol{g}_n = \sum_{j=0}^{n} W_{n-j}^h(\mathcal{V})\boldsymbol{\lambda}_j \Rightarrow \boldsymbol{\lambda}_n = \sum_{j=0}^{n} W_{n-j}^h(\mathcal{V}^{-1})\boldsymbol{g}_i.$$

Hence, with the help of the scaled Fourier transform approach described in the previous section, we can rewrite (1.65):

$$\boldsymbol{\lambda}_j \approx \rho^{-j}\left(\mathcal{F}_{m+1}^{-1}\mathcal{D}_{m+1}\left(\mathcal{V}^{-1}\right)\mathcal{F}_{m+1}\boldsymbol{g}_\rho\right)_j, \qquad j = 0, \ldots, J,$$
$$\boldsymbol{g}_\rho = \left[\boldsymbol{g}_0,\ \boldsymbol{g}_1\rho,\ \ldots,\ \boldsymbol{g}_J\rho^J\right]^T.$$

Again, given $\epsilon_0$ as in the previous section, we choose $\rho = \epsilon_0^{\frac{1}{2(J+1)}}$ to ensure the optimal accuracy. A concise analysis of the accuracy of this procedure can be found in [29].

Note that in practice we do not construct the discretizations of $\mathcal{V}^{-1}$ but rather assemble Galerkin discretizations of the boundary operators $\mathcal{V}$ (we denote them by $\boldsymbol{V}$), and then solve the corresponding systems of equations. In [21] it was shown that if the size of the small system remains constant, the range of frequencies $s_k$ for which we need to construct $\boldsymbol{V}(s_k)$ obey

$$\left|\frac{\operatorname{Re} s_k}{\operatorname{Im} s_k}\right| < \text{const},$$

and hence the $\mathcal{H}$-matrix approximation is of almost linear complexity in this case. The condition number of matrices $\boldsymbol{V}(s)$ increases as the boundary element meshwidth $\Delta x$ decreases,

$$\operatorname{cond}(\boldsymbol{V}(s)) \leq C(\Delta x)^{-1},$$

see e.g. the proof of Lemma 4.5.1 in [169]. To solve this problem, we employ an $\mathcal{H}$-matrix based $\mathcal{LU}$-preconditioner of almost linear complexity ($O(M \log^2 M)$) suggested in [33].

#### 1.2.11.3 Complexity of the Approach

Let $N$ be the number of time steps and $M$ be the size of the spatial discretization. Altogether, $O(N)$ Galerkin discretizations of single layer operators need to be constructed. This implies that total matrix assembly and storage costs scale as $O(NC(M))$, where $C(M)$ is the complexity of the assembly/storage of a single matrix. In the course of the algorithm $O(N \log N)$ matrix-vector multiplications need to be done, thus resulting in $O(N \log NT(M))$ complexity, where $T(M)$ is the complexity of a single matrix-vector product. The solution of a small lower triangular system can be done in $O(T(M))$ time. Application of all FFTs requires $O(N \log^2 NM)$ operations. Hence, the complexity of the solution of the system (1.54) after all matrices have been constructed scales as $O(N \log NT(M) + N \log^2 NM)$.

Let us assume that the incident wave $u^{inc}$ is temporally and spatially (approximately) bandlimited to a frequency $f_m$. For many applications the case of a large bandwidth $f_m \gg 1$ is of importance. Then the values of $N$, $M$ have to be chosen so that $M = O(N^2)$, see also Section 1.2.1. The primary reason for such a choice is the sampling condition: the meshwidth has to satisfy $\Delta x \approx C_1 f_m^{-1}$ and the time step $h \approx C_2 f_m^{-1}$, for some constants $C_1, C_2 > 0$. Additionally, we assume that $M = O\left(\frac{1}{\Delta x^2}\right)$.

In [22,56] the authors analyzed dissipation and dispersion errors associated with the time discretization, for both multistep and Runge-Kutta convolution quadrature. The results of these studies can be summarized as follows:

1. if the direct integral formulation is used, and the domain is convex or star-shaped, the time step has to be chosen as $h \approx C_2 f_m^{-1}$;

2. otherwise $h \approx C_2 f_m^{-1-\frac{1}{p}}$, where $p$ is the classical order of the Runge-Kutta method (or the order of the multistep method).

The errors due to the spatial discretization (also for Maxwell equations) were analyzed in [17, 26, 29, 138]. In the thesis we assume $\Delta x \approx h$, similarly to MOT methods. We did not encounter significant pollution effects with such a choice, at least for the range of discretizations of interest.

### 1.2.12 Decoupled Convolution and Directional Fast Multipole Method

The algorithm used in [144] is based on the method described in Section 1.2.11.2 applied to the full system of equations. In this work the mixed initial boundary-value problem was solved using the Calderón projector. We describe the main idea of this algorithm applied to the acoustic scattering. There are two main difficulties associated with the original algorithm of Section 1.2.11.2:

- it is necessary $O(N)$ discretizations of boundary integral operators $V(s)$ for a range of frequencies $s \in \mathbb{C}$;
- $O(N)$ systems of equations that involve these boundary integral operators need to be solved.

The algorithm aims at overcoming the first difficulty by the use of the directional fast multipole method, see [81, 83, 145]. However, it still requires to invert the Helmholtz boundary single layer operator for many frequencies, that, for small time steps, lie close to the imaginary axis, as we will demonstrate in further sections. To our knowledge, this problem for non-convex domains is difficult to be overcome by the use of other integral formulations, as analysis in [41] recently revealed.

# Chapter 2

# Data-Sparse Techniques for the Helmholtz Equation with Decay

This section is dedicated to the analysis of $\mathcal{H}$- and $\mathcal{H}^2$-matrix techniques applied to the Galerkin discretization of the single-layer boundary integral operator for $-\Delta + s^2$, $s \in \mathbb{C}$, $\operatorname{Re} s \geq 0$,

$$\boldsymbol{V}_{ij} = \iint_{\Gamma\times\Gamma} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y, \qquad i,j = 1,\ldots,M. \tag{2.1}$$

Here we assumed that $(\phi_i)_{i=1}^M$ are both test and trial basis functions. For simplicity, throughout this section we use piecewise constant basis functions. All the arguments can be extended to a more general case under the assumption that the test and trial functions are compactly supported (more precisely, supported on a constant number of mesh elements).

## 2.1 $\mathcal{H}$- and $\mathcal{H}^2$-matrices

The notion of $\mathcal{H}$-matrices was introduced in [110]. The questions of the efficient construction of $\mathcal{H}$-matrices and the complexity of algebraic operations are addressed in [100]. Recent monographs [34, 111] are dedicated to $\mathcal{H}$-matrix theory and provide both theoretical and numerical evidence of the efficiency of $\mathcal{H}$-matrix techniques. The non-exhaustive list of applications include the approximation of boundary integral operators [114], of the inverse of FEM matrices [36], and efficient $\mathcal{LU}$-preconditioners in BEM and FEM [33, 101].

In this section we review the main notions of the theory of $\mathcal{H}$-matrices, following [111].

### 2.1.1 Asymptotically Smooth Functions

**Definition 2.1.1.** *Given $X, Y \subset \mathbb{R}^d$, a function $k : X \times Y \to \mathbb{C}$ is called separable if it can be written in the following form:*

$$k(x,y) = \sum_{\nu=1}^{r} a_\nu(x)b_\nu(y),\ (x,y) \in X \times Y.$$

*The right hand side of*

$$k(x,y) = \sum_{\nu=1}^{r} a_\nu^{(r)}(x)b_\nu^{(r)}(y) + R_r(x,y),\ (x,y) \in X \times Y.$$

*is called an $r$-term separable expansion of $k(x,y)$ with the remainder $R_r$.*

One of the methods to obtain a separable expansion is polynomial interpolation. Namely, let $X$ be a box

$$X = [a_1,\ b_1] \times \dots [a_d,\ b_d]. \tag{2.2}$$

Let a set of interpolation points $(x_k^{(j)})_{j=1}^{N_k}$ be defined on intervals $[a_k,\ b_k]$, $k = 1, \dots, d$. Let

$$L_{k,\nu}(x) = \prod_{\mu=1\dots N_k,\ \mu\neq\nu} \frac{x - x_k^{(\mu)}}{x_k^{(\nu)} - x_k^{(\mu)}}$$

denote the $\nu^{th}$ Lagrange polynomial. Given $x = (x_1, \dots, x_d)$,

$$k^{(r)}(x,y) = \sum_{\nu_1=1}^{N_1} L_{1,\nu_1}(x_1) \sum_{\nu_2=1}^{N_2} L_{2,\nu_2}(x_2) \dots \sum_{\nu_d=1}^{N_d} L_{1,\nu_d}(x_d) k\left(\left(x_1^{(\nu_1)}, \dots x_d^{(\nu_d)}\right), y\right) \tag{2.3}$$

constitutes an $r = N_1 N_2 \dots N_d$-term separable expansion of $k(x,y)$. Let us for simplicity assume that $N_1 = N_2 = \dots N_d = m$. By $\Lambda_i$ we denote the Lebesgue constant for the set of interpolation points on $[a_i, b_i]$; if the interpolation points are chosen as Chebyshev nodes

$$\Lambda_i = \Lambda < 1 + \frac{2}{\pi}\log(m+1).$$

The interpolation error, i.e. the remainder $R_r$, can be bounded by, see [111, Lemma B.3.4],

$$|k(x,y) - k^{(r)}(x,y)| \leq \frac{1}{m!}\Lambda^{d-1} \sum_{i=1}^{d} \|\omega_i\|_\infty \|\partial_{x_i}^m k(\cdot, y)\|_{\infty,X},$$

where $\omega_i = \omega_i(x_i) = \prod_{j=1}^{m}(x_i - x_j^{(i)})$. If all the partial derivatives of $k(x,y)$ do not grow too fast, his error can be easily controlled. This motivates the introduction of another important concept, namely asymptotic smoothness.

**Definition 2.1.2.** *Let $X, Y \subseteq \mathbb{R}^d$ and let $k : \{(x,y) \in X \times Y, x \neq y\} \to \mathbb{C}$ be smooth. Then $k$ is called asymptotically smooth if there exist $C, \kappa, \gamma, s$, s.t.*

$$|\partial_x^\alpha \partial_y^\beta k(x,y)| \leq C(\alpha+\beta)!|\alpha+\beta|^\kappa \gamma^{|\alpha+\beta|} \|x-y\|^{-|\alpha|-|\beta|-s}, \tag{2.4}$$

*for all $x \in X$, $y \in Y$, $x \neq y$, $\alpha, \beta \in \mathbb{N}_0^d$, $\alpha + \beta \neq 0$.*

The following lemma shows that under additional geometrical assumptions on the domains $X$, $Y$, tensor-product interpolation (2.3) of the asymptotically smooth $k(x,y)$ converges to $k(x,y)$ exponentially.

**Lemma 2.1.3.** *[111, Proposition 4.2.13, p.69] Let $k(x,y)$ be asymptotically smooth in $X \times Y \subset \mathbb{R}^d \times \mathbb{R}^d$. Let additionally (2.2) hold. Let the number of interpolation points in each of the directions equal $m-1$ and the Lebesgue constant $\Lambda_j = O(c^m)$, for some $c > 1$*

*and for all $j = 1, \ldots, d$. Given $m + s \geq 0$ (where $s$ is defined by (2.4)), tensor-product interpolation (2.3) approximates $k(x, y)$ with the error*

$$\|k(., y) - k^{(r)}(., y)\|_{\infty,X} \leq c_1 \left( \frac{c_2 \operatorname{diam}_\infty(X)}{\operatorname{dist}(y, X)} \right)^m, \; y \in Y \setminus X, \tag{2.5}$$

*where $c_1, c_2$ do not depend on $m$ and*

$$\operatorname{diam}_\infty(X) = \max\{b_i - a_i : \; 1 \leq i \leq d\}.$$

Hence there exists a separable expansion for an asymptotically smooth $k(x, y)$ in $X, Y$ if

$$\eta \operatorname{diam}_\infty(X) < \operatorname{dist}(Y, X),$$

for some $\eta > 1$.

### 2.1.2 Cluster Trees and Block Cluster Trees

Let the boundary $\Gamma$ be subdivided into $M$ panels $\pi_i$, and let the corresponding index set be defined as $\mathcal{I} = \{1, \ldots, M\}$. Note that when piecewise-constant basis functions are employed, $\operatorname{supp} \phi_i = \pi_i$, $i = 1, \ldots, M$.

**Definition 2.1.4.** *Given a constant $C$, a tree $\mathcal{T}_\mathcal{I}$ is called a cluster tree corresponding to an index set $\mathcal{I}$ if $\mathcal{T}_\mathcal{I}$ is a binary labeled tree with the following properties:*

- *the label $\hat{\tau}$ of a vertex $\tau$ of $\mathcal{T}_\mathcal{I}$ is a subset of $\mathcal{I}$;*
- *the label of the root of the tree is $\mathcal{I}$;*
- *the label of a vertex $\tau$ is a disjoint union of labels of its sons;*
- *for every leaf $\tau$, $\#\hat{\tau} \leq C$.*

The leaves of the cluster tree $\mathcal{T}_\mathcal{I}$ are denoted by $\mathcal{L}(\mathcal{T}_\mathcal{I})$. All the vertices located at the level $\ell$ of the cluster tree $\mathcal{T}_\mathcal{I}$ are denoted by $\mathcal{T}_\mathcal{I}^\ell$; the root is located at the level $\ell = 0$.

The structure of the cluster tree introduces a hierarchical subdivision of $\Gamma$ into sets of panels. A set of panels corresponding to a cluster $\tau$ is denoted by $\Omega_\tau$:

$$\Omega_\tau = \bigcup_{i \in \hat{\tau}} \pi_i.$$

The bounding box of a cluster $\tau$ is the (axis-parallel) box containing the set $\Omega_\tau$; the center of the box we denote by $c_\tau$ and its diameter by $d_\tau$. The next definition can be found in [44, Def. 3.16].

**Definition 2.1.5.** *A predicate $\mathcal{A}$: $\mathcal{T}_\mathcal{I} \times \mathcal{T}_\mathcal{I} \to \{\text{true}, \text{false}\}$ is an admissibility condition for $\mathcal{T}_\mathcal{I} \times \mathcal{T}_{\mathcal{T}_I}$, if $\mathcal{A}(\tau, \upsilon) = \text{true}$ implies that for all $\tau' \in \operatorname{sons}(\tau)$, $\mathcal{A}(\tau', \upsilon) = \text{true}$ and for all $\upsilon' \in \operatorname{sons}(\upsilon)$, $\mathcal{A}(\tau, \upsilon') = \text{true}$.*

Now we have all the ingredients to introduce the concept of the admissible block-cluster tree. We adopt here a slightly modified definition, similar to the one used in the high-frequency fast multipole method [57]. In the $\mathcal{H}$-matrix theory it corresponds to the level-consistent admissible block-cluster tree.

**Definition 2.1.6.** *Let $\mathcal{T}_\mathcal{I}$ be a cluster tree. We will call an admissible block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ a subtree of a labeled tree $\mathcal{T}_\mathcal{I} \times \mathcal{T}_\mathcal{I}$ that satisfies the following conditions:*

1. *The root of the tree is* $(\mathrm{root}(\mathcal{T}_\mathcal{I}), \mathrm{root}(\mathcal{T}_\mathcal{I}))$.
2. *The son clusters of each block-cluster $b = (\tau, \sigma)$ are defined by*

$$\mathrm{sons}(b) = \begin{cases} \{(\tau', \sigma'),\ \tau' \in \mathrm{sons}(\tau),\ \sigma' \in \mathrm{sons}(\sigma)\}, & \mathrm{sons}(\tau) \neq \emptyset,\ \mathrm{sons}(\sigma) \neq \emptyset, \\ \emptyset, & \mathrm{sons}(\tau) = \emptyset \text{ or } \mathrm{sons}(\sigma) = \emptyset; \end{cases}$$

3. *A block-cluster $(\tau, \sigma)$ is a leaf if and only if one of the following holds true:*
   (a) *$(\tau, \sigma)$ is admissible;*
   (b) *$(\tau, \sigma)$ is not admissible, and $\tau \in \mathcal{L}(\mathcal{T}_\mathcal{I})$ or $\sigma \in \mathcal{L}(\mathcal{T}_\mathcal{I})$;*

Let us note that the actual choice of the admissibility condition depends on the integration kernel. For asymptotically smooth kernels the natural choice is, see (2.5),

$$\eta\, \mathrm{dist}(\tau, \sigma) \geq \max\{d_\tau, d_\sigma\}, \tag{2.6}$$

for some $\eta > 0$.

In the literature on the fast multipole methods it is quite common to use an admissibility condition of the form (2.6), or a similar one: only the neighboring clusters are not admissible. We use a slightly different admissibility condition, see also [162].

**Definition 2.1.7.** *We will call a pair of clusters $(\tau, \sigma)$ admissible if for some fixed $\eta > 1$ the following holds true:*

$$\|c_\tau - c_\sigma\| \geq \frac{\eta}{2}(d_\tau + d_\sigma).$$

Thus, all the leaves of the admissible block-cluster tree can be split into two sets, namely $\mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ of admissible block-clusters and $\mathcal{L}_-(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ of non-admissible block-clusters. The first set is called the **far-field**, while the second one is referred to as the **near-field**.

### 2.1.3 $\mathcal{H}$-matrices

We wish to approximate a Galerkin matrix of a (boundary) integral operator

$$(\mathcal{M})_{ij} = \int\limits_{\Gamma\times\Gamma} k(x, y)\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y,\ i, j = 1, \ldots, M, \tag{2.7}$$

in the $\mathcal{H}$-matrix format. The main idea that lies behind $\mathcal{H}$-matrix techniques is the following.

Let us assume that the kernel of the integral operator $k(x, y)$ is an asymptotically smooth function. The admissibility condition has to be chosen so that $k(x, y)$ has a separable expansion inside all the admissible clusters, see Section 2.1.1. Given $(\tau, \upsilon) \in \mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$, for all $i \in \hat{\tau}$, $j \in \hat{\upsilon}$, it should hold that

$$\iint\limits_{\Omega_\tau\times\Omega_\upsilon} k(x, y)\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y \approx \sum_{k=1}^{r} \left( \int\limits_{\Omega_\tau} a_k^{(r)}(x)\phi_i(x)d\Gamma_x \right) \left( \int\limits_{\Omega_\upsilon} b_k^{(r)}(y)\phi_j(y)d\Gamma_y \right). \tag{2.8}$$

Let $\#\hat{\tau} = n$, $\#\hat{\upsilon} = m$. We denote by $\mathcal{M}|_{\hat{\tau}\times\hat{\upsilon}}$ the following matrix block:

$$(\mathcal{M}|_{\hat{\tau}\times\hat{\upsilon}})_{k_i\ell_j} = \mathcal{M}_{ij}, \qquad k_i \in \{1,\dots,n\},\ \ell_j \in \{1,\dots,m\},\ i \in \hat{\tau},\ j \in \hat{\upsilon}.$$

The expansion (2.8) shows that the matrix block $\mathcal{M}|_{\hat{\tau}\times\hat{\upsilon}}$ can be approximated by a rank $r$-matrix. Hence instead of storing all matrix entries it is possible to keep in the memory only $r$ $n$-dimensional vectors $A^{(\alpha)}$, $\alpha = 1,\dots r$,

$$A^{(\alpha)}_{k_i} = \int_{\Omega_\tau} a^{(r)}_\alpha(x)\phi_i(x)d\Gamma_x, \qquad i \in \hat{\tau},$$

and $r$ $m$-dimensional vectors $B^{(k)}$, $\alpha = 1,\dots r$,

$$B^{(\alpha)}_{\ell_j} = \int_{\Omega_\upsilon} b^{(r)}_\alpha(x)\phi_j(x)d\Gamma_x, \qquad j \in \hat{\upsilon}.$$

This allows to reduce both storage costs and improve the time of the matrix-vector multiplication from $O(nm)$ to $O(r(n+m))$.

**Definition 2.1.8.** *Let $\mathcal{I}$ be an index set and $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ be an admissible block-cluster tree. Let also $k : \mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}) \to \mathbb{N}_+$. A matrix $M \in \mathbb{C}^{\mathcal{I}\times\mathcal{I}}$ is called an $\mathcal{H}$-matrix (or hierarchical matrix) if for each $b = (\tau,\sigma) \in \mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ the matrix $M|_b$ is a $k(b)$-rank matrix, i.e.*

$$\operatorname{rank} M|_b \le k(b)$$

*and is represented in the form*

$$M|_b = A_b B_b^T,$$

*where $A_b \in \mathbb{R}^{\tau\times\{1,\dots,k(b)\}}$, $B_b \in \mathbb{R}^{\sigma\times\{1,\dots,k(b)\}}$.*

An important notion for analyzing the complexity of $\mathcal{H}$-matrix arithmetic is the sparsity constant. We provide here a definition adapted to our needs; for more general definitions see [111].

**Definition 2.1.9.** *The sparsity constant for $\mathcal{L}_0 \subset \mathcal{L}(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ is defined as*

$$C_{sp}(\mathcal{L}_0) = \max\left\{\max_{\tau\in\mathcal{T}_\mathcal{I}}\{\#\sigma \in \mathcal{T}_\mathcal{I} : (\tau,\sigma)\in\mathcal{L}_0\},\ \max_{\sigma\in\mathcal{T}_\mathcal{I}}\{\#\tau \in \mathcal{T}_\mathcal{I} : (\tau,\sigma)\in\mathcal{L}_0\}\right\}.$$

In [117] it was demonstrated that under some mild assumptions on $\Gamma$, the sparsity constant can be bounded by a constant that depends on the admissibility condition and the space dimension.

The following lemma can be found in [100, Lemma 2.5] and [111, Lemma 6.3.6].

**Lemma 2.1.10.** *Let $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ be a given admissible block-cluster tree with the sparsity constant $C^+_{sp}$ for $\mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ and $C^-_{sp}$ for $\mathcal{L}_-(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$. Let $M$ be an $\mathcal{H}$-matrix, and $k > 0$ be s.t. for all $b \in \mathcal{L}_+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$*

$$\operatorname{rank}(M|_b) \le k.$$

*Additionally, let $n_{\min} \in \mathbb{N}_+$ be s.t. for all $(\tau,\sigma) \in \mathcal{L}_-(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$:*

$$\#\tau \le n_{\min} \ \text{ and } \ \#\sigma \le n_{\min}.$$

*Then the following complexity estimates hold:*

1. *The storage costs scale as*

$$S_{\mathcal{H}} \leq 2C_{sp}^{+} \max(n_{\min}, k)(\operatorname{depth}(\mathcal{T}_{\mathcal{I}}) + 1)\#\mathcal{I}.$$

2. *The complexity of the matrix vector product can be bounded by*

$$M \leq 2S_{\mathcal{H}}.$$

From now we assume that $C_{sp}$ does not depend on size of the discretization. The cluster tree is constructed so that $\operatorname{depth}(\mathcal{T}_{\mathcal{I}}) = O(\log \#\mathcal{I})$. Under these suppositions the storage costs, as well as the complexity of the matrix-vector product scale almost linearly, i.e. $O(M \log M)$, with respect to the size of the index set $\mathcal{I}$, and hence the number of Galerkin basis functions.

In practice the construction of $\mathcal{H}$-matrices is usually done using techniques based on the ideas from [98], e.g. ACA [32], ACA+ [99] or HCA [45] (although formula (2.5) is available). Such methods, besides being computationally efficient, possess major advantages over the polynomial expansion (2.5):

- no a priori information on ranks is needed, only evaluations of the integral kernel are used;
- low-rank approximations constructed with the help of such techniques can be close (and in practice are close) to optimal [35, 98].

The optimal low-rank approximation can be constructed with the help of the singular value decomposition, see Appendix C. Due to high computation costs, the SVD is employed in the $\mathcal{H}$-matrix theory only rarely (e.g. for coarsening, see [99]).

### 2.1.4 $\mathcal{H}$-matrices for Helmholtz Boundary Integral Operators

Questions of the applicability of $\mathcal{H}$-matrices to the Helmholtz equation have been studied in various works [21, 23, 97], see also [34] and references therein. Let us address the simplest case, namely, the use of $\mathcal{H}$-matrices for the approximation of the Galerkin discretization of the boundary integral operator

$$\begin{aligned} &\mathcal{V}(s) : \ H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma), \\ &\left(\mathcal{V}(s)\phi\right)(x) = \int_{\Gamma} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\phi(y)d\Gamma_y, \qquad s \in \mathbb{C}, \end{aligned}$$

namely

$$(\mathcal{M})_{ij} = \int_{\Gamma\times\Gamma} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y, \ i,j = 1,\ldots,M,$$

where $(\phi_i(x))_{i=1}^{M}$ are piecewise-constant test (and trial) basis functions.

In a nutshell, the results of these studies are the following.

1. If $s = i\kappa$, $\kappa \in \mathbb{R}$, the complexity (storage and matrix-vector multiplication) is bounded by $O(M|\kappa| \log M)$; the hidden constant depends on the accuracy. A typical assumption on the number of Galerkin basis functions is

$$M = O(|\kappa|^2), \qquad \text{as } |\kappa| \to +\infty.$$

   Consequently, the complexity scales as

$$O(M^{\frac{3}{2}} \log M).$$

2. For complex $s : \operatorname{Re} s > 0$, in [21] it was shown that if $\frac{|\operatorname{Im} s|}{|\operatorname{Re} s|} < c$, for some $c > 0$, the Helmholtz kernel is asymptotically smooth and the complexity of the $\mathcal{H}$-matrix approximation is almost linear. Based on the results of [47], in [34, p.114, Theorem 3.18, p.157] it was demonstrated that the matrix-vector multiplication and storage costs depend on $M$ as

$$O\left(M\left(C + \left|\frac{\operatorname{Im} s}{\operatorname{Re} s}\right|\right)\log M\right),$$

   for some $C > 0$. Under the assumption $M = O(|s|^2)$, the $\mathcal{H}$-matrix approximation is of almost linear complexity, namely $O(M \log M)$.

In general, the $\mathcal{H}$-matrix assembly complexity includes additional logarithmic factors related to the evaluation of the 4-dimensional BEM integrals with the accuracy sufficient to preserve the stability of the Galerkin method, see [87, 118, 167–169]. Particularly, due to the use of tensor-Gauss quadratures with coordinate transformations (see the above references), the asymptotic complexity of the matrix construction is larger than the storage and matrix-vector multiplication complexity by the factor up to $O\left(\log^k M\right)$, $k \leq 4$.

#### 2.1.4.1 Efficient Construction of $\mathcal{H}$-Matrices for the Boundary Single Layer Operator of the Helmholtz Equation with Decay

In [91] the error of the fast multipole method for the Helmholtz equation with decay has been studied. It was suggested that the relative error in the case $\operatorname{Re} s > 0$ does not serve any more as a good error estimator. Clearly, the elements of the Galerkin matrix of the Helmholtz boundary single layer operator satisfy the inequality

$$\left|\int_{\pi_i}\int_{\pi_j} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y\right| \leq \mathrm{e}^{-\operatorname{Re} s\,\operatorname{dist}(\pi_i,\pi_j)}\int_{\pi_i}\int_{\pi_j} \frac{1}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y,$$

and hence become exponentially small when $\operatorname{dist}(\pi_i, \pi_j)$ gets larger. Therefore, the contribution of the blocks corresponding to the parts of the boundary $\Gamma$ that are distant from each other, can be neglected up to a certain tolerance when computing the matrix-vector product.

Accordingly, it is possible to skip constructing some blocks in the $\mathcal{H}$-matrix approximation. Let $(\tau, \sigma)$ denote an admissible block-cluster, and let the distance between the bounding boxes of the clusters $\tau$ and $\sigma$ equal $d > 0$. If

$$\frac{\exp(-d \operatorname{Re} s)}{4\pi d} < \epsilon, \tag{2.9}$$

for a fixed accuracy $\epsilon > 0$, the corresponding block can be approximated by a zero matrix. The actual choice of $\epsilon$ has to be made based on extensive numerical experiments.

The accuracy of the approximation of other blocks within the ACA+ algorithm may be reduced as well. Since in the ACA/ACA+ algorithm it is easier to control the relative accuracy, we proceed as follows. If the distance between two admissible clusters is $d$, the relative accuracy of the approximation may be scaled by $\mathrm{e}^{\mathrm{Re}\, sd}$, what we also do. This is in correspondence with the definition of the scaled error in [91].

### 2.1.5 $\mathcal{H}^2$-Matrices

The notion of $\mathcal{H}^2$-matrices was introduced in [113]. In [46] the authors developed a black-box algorithm that compresses a given matrix in the $\mathcal{H}^2$-matrix format. For the cases when the construction of a dense matrix is too expensive (e.g. discretizations of integral operators), an efficient method of the construction of $\mathcal{H}^2$-matrix based approximations was suggested in [112].

Another way to assemble an $\mathcal{H}^2$-matrix is based on the use of known explicitly separable expansions of an integral kernel, e.g. those coming from fast multipole methods. This was done for the discretization of the boundary single-layer operator for the Helmholtz equation in two dimensions in [23], as well as implicitly in [8].

In this section we review the main definitions of the $\mathcal{H}^2$-matrix theory based on recent monographs [44, 111] and lecture notes [157].

Let us fix a cluster tree $\mathcal{T}_{\mathcal{I}}$ and an admissible block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$.

**Definition 2.1.11.** *A family of matrices $\left(V^t\right)_{t\in\mathcal{T}_{\mathcal{I}}}$, s.t. for all $t \in \mathcal{T}_{\mathcal{I}}$ the matrix $V^t \in \mathbb{C}^{\hat{t}\times K_t}$ for some finite index set $K_t$, is called a cluster basis.*

**Definition 2.1.12.** *(Uniform $\mathcal{H}$-matrix) Let $\left(V^t\right)_{t\in\mathcal{T}_{\mathcal{I}}}$, $\left(W^t\right)_{t\in\mathcal{T}_{\mathcal{I}}}$. A matrix $M \in \mathbb{C}^{\mathcal{I}\times\mathcal{I}}$ is called a uniform $\mathcal{H}$-matrix if for all admissible $(t,s) \in \mathcal{L}\left(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}\right)$ there exists $S^{t,s} \in \mathbb{C}^{K^t\times K^s}$ s.t.*

$$M|_{\hat{t}\times\hat{s}} = V^t S^{t,s} \left(W^s\right)^T.$$

*Matrices $S^{t,s}$ are called coupling matrices.*

A uniform $\mathcal{H}$-matrix is an $\mathcal{H}$-matrix, since the ranks of all its subblocks corresponding to admissible clusters are bounded:

$$\operatorname{rank}\left(V^t S^{t,s}\left(W^s\right)^T\right) \leq \operatorname{rank} S^{t,s} \leq \min\left(\#K^t, \#K^s\right).$$

**Definition 2.1.13.** *A cluster basis $\left(V^t\right)_{t\in\mathcal{T}_{\mathcal{I}}}$ is called nested if for every non-leaf cluster $t$ and for all $t' \in \operatorname{sons}(t)$ there exists a matrix $T^{t'} \in \mathbb{C}^{K^{t'}\times K^t}$ ('transfer matrix'), such that*

$$V^t = V^{t'} T^{t'}.$$

**Definition 2.1.14.** *A uniform $\mathcal{H}$-matrix whose column and row cluster bases are nested is called an $\mathcal{H}^2$-matrix.*

The use of $\mathcal{H}^2$-matrices is motivated by a possible reduction of storage and computation costs when dealing with the nested cluster basis compared to the cluster basis. Namely, if $V^t$ are dense matrices, storing them for all $t \in \mathcal{T}_{\mathcal{I}}$ may be costly. In the case when the nested cluster basis is used, one only needs to store the cluster basis for leaves and (possibly)

transfer matrices. If these are of a special structure (e.g. are sparse), storage costs and time for the computation of the matrix-vector product may be reduced significantly.

The algorithm for the efficient matrix-vector multiplication

$$y = Mx,$$

with $M$ being an $\mathcal{H}^2$-matrix, is performed in three stages.

1. Forward transformation. During the forward transformation vectors $x^s$, for all $s \in \mathcal{T}_\mathcal{I}$, are computed:

$$x^s = (W^s)^T \, x|_{\hat{s}} \,. \tag{2.10}$$

   If the cluster basis is nested, this computation can be performed recursively:

$$x^s = \begin{cases} (W^s)^T \, x|_{\hat{s}}, & \text{if } s \in \mathcal{L}_{\mathcal{T}_\mathcal{I}}, \\ \sum\limits_{t \in \operatorname{sons}(s)} \left(T_W^t\right)^T x^t, & \text{otherwise,} \end{cases} \tag{2.11}$$

   where $T_W^s$ are the transfer matrices of the cluster basis $(W^s)_{s \in \mathcal{T}_\mathcal{I}}$.

2. Multiplication. Let $R_t = \{s \in \mathcal{T}_\mathcal{I} : \, (t,s) \in \mathcal{L}_+ (\mathcal{T}_{\mathcal{I}\times\mathcal{I}})\}$, $t \in \mathcal{T}_\mathcal{I}$. The result of the multiplication is

$$y^t = \sum_{s \in R_t} S^{t,s} x^s, \tag{2.12}$$

   for all clusters $t \in \mathcal{T}_\mathcal{I}$.

3. Backward transformation. The result of the backward transformation is the vector $(y_j)_{j \in \mathcal{I}}$, given by

$$y_j = \sum_{t \in \mathcal{T}_\mathcal{I} :\, j \in \hat{t}} \left(V^t y^t\right)_j .$$

   If the cluster basis is nested, this computation is performed recursively, similarly to the forward transformation. For $s \in \mathcal{T}_\mathcal{I}$ we first recursively compute:

$$y \,|_{\hat{s}} = y^s + T_V^s y^t, \qquad s \in \operatorname{sons}(t), \tag{2.13}$$

   where $T_V^s$ are the transfer matrices of the cluster basis $(V^s)_{s \in \mathcal{T}_\mathcal{I}}$.

   Next, for all $i \in \mathcal{I}$

$$y_i = \left(V^s y^{\hat{s}}\right)_i , \; s \in \mathcal{T}_\mathcal{I}, \; i \in \hat{s}. \tag{2.14}$$

4. The non-admissible blocks are treated as in the case of $\mathcal{H}$-matrices.

**Remark 2.1.15.** *If, for a given $\ell > 1$, there are no admissible clusters at the levels $k$ : $1 \le k < \ell$ of the block-cluster tree, there is no need to perform the forward and backward transformation for the levels $k < \ell$: for all such clusters $t$, $R_t = \emptyset$, hence they do not contribute to the whole matrix-vector product.*

## 2.2 High-Frequency Fast Multipole Method

The history of fast multipole methods starts with the seminal works [103, 154], where an algorithm for the fast evaluation of the sums

$$f_j = \sum_{n=1}^{N} q_n \frac{1}{\|x_n - x_j\|},\ x_j \in \mathbb{R}^d,\ j, \dots, N,\ d = 2, 3$$

was developed. The one-level fast multipole method for the Helmholtz potential had been introduced in [155]. An excellent algorithm-oriented description of this method can be found in [61]. A wide range of works is dedicated to the various improvements and the efficient implementation of the high-frequency fast multipole algorithm: see [53, 58, 70, 107, 164, 175, 176, 179] and references therein. In [68] the author developed the fast multipole algorithm coupled with the microlocal discretization, particularly efficient for high frequencies. A stable for all frequencies fast multipole method for Maxwell equations was introduced in [71].

The Helmholtz equation with decay had been considered, to our knowledge, only in several works. Namely, in [57] the authors mentioned that the choice of the lengths of the underlying expansions can be performed ignoring the complex part of the wavenumber, though for large decays more savings are possible. The fast multipole method for the Yukawa potential $\frac{\mathrm{e}^{-\lambda r}}{r}$, $\lambda > 0$, was developed in [104]. In [93, 102] it was shown that to achieve a fixed relative accuracy, the length of the fast multipole expansion in the presence of decay has to be chosen slightly larger than the length of the expansion in the no-decay case. In [102] the authors suggested a close to optimal empirical formula to determine the length of the multipole expansion. The work [188] is dedicated to the numerical studies of the applicability of the high-frequency fast multipole method to the Helmholtz equation with decay; the authors demonstrated that if the decay is sufficiently large, cancellation errors can occur, and proposed a strategy to avoid these errors. The same kind of issue was studied in [20].

In [91] a numerical study of the error of the truncation of the multipole expansion for complex wavenumbers has been performed, as well as the notion of the scaled error, see Section 2.1.4.1, was introduced. In the same work the authors numerically examined the effect of decay on the length of the fast multipole expansion and suggested empirical formulas well-suited for the scaled error control.

We refer to [91] for the review of other works on the fast multipole method for the Helmholtz equation with a complex wavenumber, as well as the list of possible applications.

Let us also mention that recently several novel fast multipole schemes have been developed; a non-exhaustive list of those includes the black-box fast multipole method for non-oscillatory kernels [190], the family of directional fast multipole algorithms for oscillatory kernels [81–83] and the fast butterfly algorithm [50, 148, 150].

This section is organized as follows. We start with the review of definitions and properties of special functions. Then, we describe the high-frequency fast multipole method (HF FMM) of [57, 155] in the framework of $\mathcal{H}^2$-matrices concentrating on technical and algorithmic questions. Finally, we analyze the error of the fast multipole method for a general complex wavenumber.

### 2.2.1 Special Functions

The Legendre polynomials are defined as

$$P_n(z) = \frac{1}{2^n n!} \frac{d^n}{dz^n}(z^2 - 1)^n.$$

They are orthogonal with respect to the $L_2$-product on $[-1,\ 1]$:

$$\int_{-1}^{1} P_n(x) P_m(x) dx = \frac{2}{2n+1} \delta_{nm},$$

where $\delta_{nm}$ is the Kronecker delta. The following classical theorem (see [75]) describes the convergence rate of the Legendre approximation to analytic functions.

**Theorem 2.2.1.** *Let $f(z)$ be analytic in the interior of a Bernstein ellipse*

$$E_\rho = \left\{ \frac{\rho \mathrm{e}^{i\phi} + \rho^{-1} \mathrm{e}^{-i\phi}}{2},\ \phi \in [0,\ 2\pi) \right\}$$

*for some $\rho > 1$, but not in the interior of any $E_{\rho'}$, with $\rho' > \rho$. Then*

$$f(z) = \sum_{n=0}^{\infty} a_n P_n(z)$$

*with*

$$a_n = \frac{2n+1}{2} \int_{-1}^{1} f(x) P_n(x) dx.$$

*The series converges absolutely and uniformly on any closed set in the interior of $E_\rho$ and diverges in the exterior of $E_\rho$. Moreover,*

$$\limsup_{n \to \infty} |a_n|^{\frac{1}{n}} = \frac{1}{\rho}.$$

The associated Legendre functions are defined with the help of the Legendre polynomials:

$$\begin{aligned} P_n^m(x) &= (-1)^m (1 - x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x), \qquad 0 \le m \le n, \\ P_n^{-m}(x) &\equiv (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(x), \\ P_n^0(x) &\equiv P_n(x), \\ P_n^m(x) &\equiv 0, \qquad |m| > n. \end{aligned}$$

The normalized associated Legendre functions

$$\begin{aligned} \bar{P}_n^m &= \sqrt{\left(n + \frac{1}{2}\right) \frac{(n-m)!}{(n+m)!}} (1 - x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x), \\ \bar{P}_n^m &\equiv \bar{P}_n^{-m}, \qquad m < 0. \end{aligned} \tag{2.15}$$

By $\hat{r}, \hat{s}, ...$ we denote unit vectors in $\mathbb{R}^3$, namely, given a vector $x \in \mathbb{R}^3$,

$$\hat{x} = \frac{x}{\|x\|}.$$

The spherical coordinates of a vector in $\mathbb{R}^3$ are given by $(\rho, \phi, \theta)$, with $\phi$ being the azimuth and $\theta$ the inclination. Then the Cartesian coordinates of a vector $\hat{s}$ on the unit sphere are read as

$$\hat{s} = (\cos\phi \sin\theta,\ \sin\phi \sin\theta,\ \cos\theta), \qquad \phi \in [0,\ 2\pi]\,,\ \theta \in [0,\ \pi]\,. \tag{2.16}$$

A spherical harmonic of degree $n$ and order $m$ is a function

$$\begin{aligned} &Y_n^m : \mathbb{S}^2 \to \mathbb{C}, \\ &Y_n^m(\hat{s}) \equiv Y_n^m(\theta, \phi) = \sqrt{\frac{(2n+1)(n-m)!}{4\pi(n+m)!}} e^{im\phi} P_n^m(\cos\theta), \qquad |m| \le n, \\ &Y_n^{m*}(\hat{s}) \equiv (-1)^m Y_n^{-m}(\hat{s}). \end{aligned} \tag{2.17}$$

These functions constitute an orthonormal basis of $L_2(\mathbb{S}^2)$ and

$$\int_{\mathbb{S}^2} Y_n^m(\hat{s}) Y_p^{l*}(\hat{s}) d\hat{s} = \delta_{np}\delta_{ml}. \tag{2.18}$$

They are connected to the Legendre polynomials via the addition theorem. Given two unit vectors $\hat{x}$, $\hat{y}$,

$$P_n(\hat{x} \cdot \hat{y}) = \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^m(\hat{x}) Y_n^{m*}(\hat{y}). \tag{2.19}$$

Spherical Bessel functions of the first kind $j_n(x)$ and spherical Bessel functions of the third kind $h_n^{(1)}(x)$, $h_n^{(2)}(x)$ are defined as in [3, (10.1.1)]. We denote

$$h_n(x) \equiv h_n^{(1)}(x).$$

The analytic expressions for these functions are given by Rayleigh's formulas:

$$j_n(z) = z^n \left(-\frac{1}{z}\frac{d}{dz}\right)^n \frac{\sin z}{z}, \tag{2.20}$$

$$h_n(z) = j_n(z) - iz^n \left(-\frac{1}{z}\frac{d}{dz}\right)^n \frac{\cos z}{z}. \tag{2.21}$$

Spherical Bessel functions are the coefficients of the expansion of the plane-wave function in the Legendre polynomial basis [3, (10.1.47)]:

$$e^{iz\cos\theta} = \sum_{n=0}^{\infty} (2n+1) i^n j_n(z) P_n(\cos\theta). \tag{2.22}$$

**Remark 2.2.2.** *Theorem 2.2.1 allows to conclude that the series (2.22) converges super-geometrically, since the function* $\mathrm{e}^{izt}$ *is entire in* $t$.

The following is a particular case of the Funk-Hecke theorem [105, Theorem 3.4.1].

**Theorem 2.2.3** (Funk-Hecke theorem)**.** *Let $f$ be a bounded integrable function on $[-1,1]$. Then $f_{\hat{\alpha}}(\hat{s}) = f(\hat{\alpha}\cdot\hat{s})$, $\hat{\alpha}\in\mathbb{S}^2$, is integrable on $\mathbb{S}^2$ and, for all $n\in\mathbb{N}$,*

$$\int_{\mathbb{S}^2} f(\hat{\alpha}\cdot\hat{s})P_n(\hat{q}\cdot\hat{s})d\hat{s} = a_n(f)P_n\left(\hat{q}\cdot\hat{\alpha}\right),$$

*where*

$$a_n(f) = 2\pi\int_{-1}^{1} f(t)P_n(t)dt.$$

The next identity can be immediately derived from the above theorem combined with (2.22) and Theorem 2.2.1:

$$\int_{\mathbb{S}^2} \mathrm{e}^{i\lambda\hat{y}\cdot\hat{s}}P_k(\hat{s}\cdot\hat{x})d\hat{s} = 4\pi i^k j_k(\lambda)P_k(\hat{y}\cdot\hat{x}),\ k\in\mathbb{N},\ \lambda\in\mathbb{C}. \tag{2.23}$$

The following expression serves as the basis for the fast multipole method and is known under the name 'addition theorem' (or 'Gegenbauer's addition theorem'), see [3, (10.1.45), (10.1.46)]:

$$h_0(\kappa\|x-y\|) = \sum_{n=0}^{\infty}(2n+1)h_n(\kappa\|x\|)j_n(\kappa\|y\|)P_n(\hat{x}\cdot\hat{y}), \tag{2.24}$$
$$x,y\in\mathbb{R}^3 : \|x\| > \|y\|.$$

Another component of the fast multipole method is numerical integration over the unit sphere. In the fast multipole method literature it is often performed with the help of the quadrature rule introduced in the following lemma from [155].

**Lemma 2.2.4.** *Let $f$ be a spherical harmonic of degree $n_1$, and $g$ be a spherical harmonic of degree $n_2$, $f = f(\hat{s})$, $g = g(\hat{s})$, where $\hat{s}$ is given by (2.16). For any $n_\theta \geq \left\lceil\frac{n_1+n_2+1}{2}\right\rceil$, $n_\phi \geq n_1+n_2+1$ the quadrature rule on the unit sphere given by the nodes and weights*

$$\begin{aligned}(\phi_k,\theta_j) &= \left((k-1)\frac{2\pi}{n_\phi}, \arccos x_j\right),\\ w_{kj} &= \frac{2\pi}{n_\phi}\omega_j, \qquad k=1,..,n_\phi,\ j=1,..,n_\theta,\end{aligned} \tag{2.25}$$

*with $(x_j)_{j=1}^{n_\theta}$, $(\omega_j)_{j=1}^{n_\phi}$ being Gaussian quadrature nodes and weights on the interval $[-1,1]$, integrates the product of $f$ and $g$ exactly.*

*Proof.* Integration of the product of $f$ and $g$ requires the evaluation of the integrals of the type, see (2.17),

$$\int_0^{2\pi}\mathrm{e}^{im\phi}\mathrm{e}^{im'\phi}d\phi\int_{-1}^{1}P_{n_1}^{m}(x)P_{n_2}^{m'}(x)dx,$$

where $m, m' \in \mathbb{Z}$, $|m| \leq n_1$, $|m'| \leq n_2$. The first integral can be evaluated exactly with the help of the trapezoidal rule with at least $n = n_1 + n_2 + 2$ points, more precisely, the quadrature with nodes $\tilde{\phi}_k$ and weights $\tilde{\omega}_k$, $k = 1, \ldots, n$:

$$\tilde{\phi}_k = \frac{2\pi}{n-1}(k-1),$$
$$\tilde{\omega}_1 = \tilde{\omega}_n = \frac{2\pi}{2(n-1)},\ \tilde{\omega}_j = \frac{2\pi}{n-1}, \qquad 2 \leq j \leq n-1.$$

The integrand of

$$\int_{-1}^{1} P_{n_1}^{m}(x) P_{n_2}^{-m}(x) dx$$

is a polynomial of the degree not larger than $n_1 + n_2$, hence this integral can be integrated exactly with any Gaussian quadrature rule with $\left\lceil \frac{n_1+n_2+1}{2} \right\rceil$ points. □

Here we employ the Gauss-Legendre quadrature. The abscissas of the quadrature of the order $n$ are given by the zeros of the Legendre polynomial $P_n$, and the weights by

$$w_\ell = \frac{2}{(1-x_\ell)^2 \left(P_n'(x_\ell)\right)^2},\ \ell = 1, \ldots, n.$$

**Remark 2.2.5.** *In what follows we use the quadrature rule with $n_\theta = \left\lceil \frac{n_1+n_2+1}{2} \right\rceil$ and $n_\phi = 2n_\theta$.*

**Remark 2.2.6.** *We adopt a short notation for the quadrature rule defined in Lemma (2.2.4):*

$$(w_\ell, \hat{s}_\ell)_{\ell=1}^{L},\ (w_\ell, \hat{r}_\ell)_{\ell=1}^{L}, \ldots \tag{2.26}$$

*stands for a quadrature rule with $L = 2n_\theta^2$, and $\hat{s}_\ell$ ($\hat{r}_\ell, \ldots$) is a vector (2.16) with $\phi$, $\theta$ given by (2.25).*

**Remark 2.2.7.** *We will denote the integral $\int_{\mathbb{S}^2} f(\hat{s}) d\hat{s}$ computed with the help of the quadrature $(w_\ell, \hat{s}_\ell)_{\ell=1}^{L}$ by $Q_L\left[f(\hat{s})\right]$. When necessary, the variable of the integration is stated explicitly in the upper index:*

$$\int_{\mathbb{S}^2} f(\hat{s} \cdot \hat{r}) d\hat{s} \approx Q_L^{\hat{s}}\left[f(\hat{s} \cdot \hat{r})\right]. \tag{2.27}$$

Additionally, we will use the following lemma which is a straightforward corollary of Lemma 2.2.4 and (2.19).

**Lemma 2.2.8.** *Given $M \in \mathbb{N}_+$ and $m, n \in \mathbb{N}_0 :\ \left\lceil \frac{m+n+1}{2} \right\rceil \leq M$,*

$$Q_M^{\hat{s}}\left[P_m\left(\hat{q} \cdot \hat{s}\right) P_n\left(\hat{r} \cdot \hat{s}\right)\right] = \begin{cases} \frac{4\pi}{2n+1} P_n\left(\hat{q} \cdot \hat{r}\right), & \text{if } n = m, \\ 0, & \text{otherwise.} \end{cases}$$

*for all $\hat{q}, \hat{r} \in \mathbb{S}_2$.*

*Proof.* We use the addition theorem (2.19) for Legendre functions to rewrite

$$Q_M^{\hat{s}}\left[P_m\left(\hat{q}\cdot\hat{s}\right)P_n\left(\hat{r}\cdot\hat{s}\right)\right] = \frac{4\pi}{2m+1}\frac{4\pi}{2n+1}\sum_{\ell=-m}^{m} Y_m^{\ell}(\hat{q})\sum_{k=-n}^{n} Y_n^{k*}(\hat{r})Q_M^{\hat{s}}\left[Y_m^{\ell*}(\hat{s})Y_n^{k}(\hat{s})\right].$$

The quadrature rule of order $M$ as in the statement of the lemma integrates the product of these spherical harmonics exactly. Hence for $m \neq n$ the result follows from orthogonality of spherical harmonics, see (2.18). For $m = n$, we again employ (2.18) to get

$$\begin{aligned} Q_M^{\hat{s}}\left[P_m\left(\hat{q}\cdot\hat{s}\right)P_n\left(\hat{r}\cdot\hat{s}\right)\right] &= \left(\frac{4\pi}{2n+1}\right)^2 \sum_{\ell=-m}^{m} Y_m^{\ell}(\hat{q})Y_m^{\ell*}(\hat{r}) \\ &= \frac{4\pi}{2n+1}P_n\left(\hat{q}\cdot\hat{r}\right). \end{aligned}$$

□

### 2.2.2 High-Frequency Fast Multipole Algorithm

The high-frequency fast multipole method is based on the expansion (2.24). Namely, given $s \in \mathbb{C}$, $x, y, x_\beta, y_\alpha \in \mathbb{R}^3$, it holds:

$$\frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|} = -\frac{s}{4\pi}\sum_{n=0}^{\infty}(2n+1)h_n(is\|c_{\alpha\beta}\|)j_n(is\|y-x+c_{\alpha\beta}\|)P_n(\hat{c}_{\alpha\beta}\cdot\hat{r}_{\alpha\beta}),$$
$$\text{for } \|c_{\alpha\beta}\| > \|r_{\alpha\beta}\|,$$

where $c_{\alpha\beta} = y_\alpha - x_\beta$ and $r_{\alpha\beta} = x - y + c_{\alpha\beta}$.

Truncating the above series at $N+1$ terms, employing (2.23) and interchanging the limits of integration, gives

$$\begin{aligned} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|} = -\frac{s}{16\pi^2}\sum_{n=0}^{N}(2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|)\int_{\mathbb{S}^2}\mathrm{e}^{-s(r_{\alpha\beta},\hat{r})} \\ \times P_n(\hat{c}_{\alpha\beta}\cdot\hat{r})d\hat{r} + E_{tr}(N), \end{aligned} \tag{2.28}$$

where $E_{tr}(N)$ is the truncation error.

The next step is the discretization. The addition theorem (2.19) combined with (2.22) shows that the integrand of (2.28) is a sum of products of spherical harmonics, hence the quadrature rule of Lemma 2.2.4 can be employed. In [69] it was suggested that $L$ should be chosen so that $L \geq 2(N+1)^2$. This gives the following separable expansion of the Helmholtz kernel:

$$\begin{aligned} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|} = \sum_{k=0}^{L} w_k \mathrm{e}^{s(y-y_\alpha,\hat{r}_k)}\left(-\frac{s}{(4\pi)^2}\sum_{n=0}^{N}(2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|)P_n(\hat{c}_{\alpha\beta}\cdot\hat{r}_k)\right) \\ \times \mathrm{e}^{-s(x-x_\beta,\hat{r}_k)} + E_{tr}(N) + E_I(L,N), \end{aligned}$$

where $E_I$ is the integration error.

Another way to discretize (2.28) based on the modification of the integrand and the use of the trapezoidal quadrature rule was recently suggested in [53, 164].

The fast multipole method can be cast into the framework of $\mathcal{H}^2$-matrices. Before, this was done in [8, 23]. Namely, the matrix-vector multiplication described in Section 2.1.5 can be viewed as a generalized description of the fast multipole algorithm, with properly defined cluster basis, transfer and coupling matrices.

Let us consider the Galerkin discretization of the Helmholtz boundary single layer operator

$$(\boldsymbol{V}(s))_{ij} = \iint\limits_{\Gamma\times\Gamma} \phi_i(x)\phi_j(y)\frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}d\Gamma_x d\Gamma_y, \qquad i,j = 1,\ldots,M.$$

In this section we show how this matrix can be approximated with the help of the fast multipole method. We will comment on the choice of the parameters and the error control in the further sections.

We fix the block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$. We use the uniform partition of the domain, i.e. all the bounding boxes of the clusters located at the same level of the block-cluster tree are of the same size.

With each level $\ell$ of the cluster tree we associate a set of quadrature nodes on the unit sphere

$$(w_n, \hat{r}_n)_{n=1}^{L_\ell},\ L_\ell \in \mathbb{N}, \tag{2.29}$$

defined as in Lemma 2.2.4.

The version of the algorithm described here is the high-frequency FMM of [57] with minor modifications.

#### 2.2.2.1 Cluster Basis

Given a cluster $\tau_\alpha$ located at the level $\ell$ of the cluster tree and $c_\alpha$ being the center of its bounding box, we define the column cluster basis as a matrix

$$\left(W^{\tau_\alpha}\right)_{k_j n}(s) = \int\limits_{\tau_\alpha} \mathrm{e}^{-s(x-c_\alpha,\hat{r}_n)}\phi_j(x)d\Gamma_x, \tag{2.30}$$
$$k_j \in \{1,\ldots,\#\hat{\tau}_\alpha\},\ j\in\hat{\tau}_\alpha,\ n = 1,\cdots,L_\ell.$$

The row cluster basis for the cluster $\tau_\alpha$ has a different form:

$$\left(V^{\tau_\alpha}\right)_{k_j n}(s) = w_n \int\limits_{\tau_\alpha} \mathrm{e}^{s(x-c_\alpha,\hat{r}_n)}\phi_j(x)d\Gamma_x, \qquad k_j \in \{1,\ldots,\#\hat{\tau}_\alpha\},\ j\in\hat{\tau}_\alpha,\ n = 1,\ldots,L_\ell.$$

**Efficient Computation and Storage of the Cluster Basis** It is sufficient to compute the column cluster basis only, whereas the row cluster basis can be constructed based on the symmetry of the quadrature points (2.25) on the unit sphere. Let $p := x - c_\alpha = (p_1,\ p_2,\ p_3) \in \mathbb{R}^3$. For quadrature nodes $\hat{r} = (\cos\phi_k \sin\theta_j,\ \sin\phi_k\sin\theta_j,\ \cos\theta_j)^T$, it holds that

$$\mathrm{e}^{s(x-c_\alpha,\hat{r})} = \mathrm{e}^{s(p_1\cos\phi_k\sin\theta_j + p_2\sin\phi_k\sin\theta_j + p_3\cos\theta_j)} = \mathrm{e}^{-s\left(p,\hat{q}_{kj}\right)},$$
$$\hat{q}_{kj} = (\cos(\pi+\phi_k)\sin(\pi-\theta_j),\ \sin(\pi+\phi_k)\sin(\pi-\theta_j),\ \cos(\pi-\theta_j)).$$

Let $L_\ell = 2n_\theta^2$, $n_\theta \in \mathbb{N}$, see Remark 2.2.5. Since the nodes of the Gauss-Legendre quadrature are symmetric about 0,

$$\pi - \theta_j = \theta_{n_\theta - j + 1},$$

and also

$$\pi + \phi_k = \pi + \frac{\pi}{n_\theta} k = \frac{2\pi}{2n_\theta} \left( (n_\theta + 1)k \bmod 2n_\theta \right),$$

the vector $\hat{q}_{kj}$ indeed belongs to the set $(\hat{r}_n)_{n=1}^{L_\ell}$. Hence, only the column leaf cluster basis need to be computed and stored.

For some applications it is necessary to compute the cluster basis for many values of $s \in \mathbb{C}$. In this case storing all of the dense matrices (2.30) may be expensive. Alternatively, the function $f_\alpha(x, \hat{r}) = \mathrm{e}^{-s(x - c_\alpha, \hat{r})}$ can be interpolated in $x$ with the help of multivariate interpolation (as done when evaluating the boundary integrals with the help of a quadrature rule). Then the entries of the column leaf cluster basis are

$$\begin{aligned} \left(W^{\tau_\alpha}\right)_{jn}(s) &= \int_{\tau_\alpha} \mathrm{e}^{-s(x - c_\alpha, \hat{r}_n)} \phi_j(x) d\Gamma_x \\ &\approx \sum_{k=1}^{K} \mathrm{e}^{-s(x_{k,\alpha} - c_\alpha, \hat{r}_n)} \omega_{j,k}^{\alpha}, \end{aligned} \tag{2.31}$$

where $x_{k,\alpha} \in \tau_\alpha$ are quadrature nodes and $\omega_{j,k}^{\alpha}$, $k = 1, \ldots, K$ are weights. Hence for all $s \in \mathbb{C}$ we can store only the weights $\omega_{j,k}^{\alpha}$, the interpolation points $x_{k,\alpha}$ and the centers of the clusters $c_\alpha$, and then compute $W^{\tau_\alpha}$ on the fly when reading the data from the memory to perform the matrix-vector multiplication.

#### 2.2.2.2 Transfer Matrices

In the fast multipole method transfer matrices are represented by translation operators. Namely, for the column cluster basis transfer matrices correspond to the multipole-to-multipole (M2M) translations, and for the row cluster basis they are equivalent to local-to-local (L2L) translations.

Before defining transfer matrices, let us provide some information on one of the ingredients of these operators, namely the fast spherical harmonic transform as described in detail in [126].

**Fast Spherical Harmonic Transform** Let

$$f : \mathbb{S}^2 \to \mathbb{C}.$$

Let us set $\hat{s}(\theta, \phi) = (\cos\phi \sin\theta,\ \sin\phi \sin\theta,\ \cos\theta)$. We assume that for some $K \in \mathbb{N}$,

$$f(\hat{s}(\theta, \phi)) = \sum_{k=0}^{K} \sum_{m=-k}^{k} f_k^m Y_k^m(\theta, \phi). \tag{2.32}$$

Given the values of the function $f$ on the grid $(\phi_\ell,\ \theta_n)_{\ell,n=1}^{n_\phi, n_\theta}$ defined by (2.25)

$$f_{\ell,n} = f(\hat{s}(\theta_n, \phi_\ell)),\ \ell = 1, \ldots, n_\phi,\ n = 1, \ldots, n_\theta,$$

we need to compute the values of the function

$$F(\hat{s}(\theta,\phi)) = \sum_{k=0}^{N} \sum_{m=-k}^{k} f_k^m Y_k^m(\theta,\phi), \tag{2.33}$$

on the grid of different size, namely $(\phi_\ell,\ \theta_n)_{\ell,n=1}^{n'_\phi,n'_\theta}$ defined by (2.25). We assume $n_\phi = 2n_\theta$ and $n'_\phi = 2n'_\theta$, set

$$L = 2n_\theta^2, \qquad L' = 2(n'_\theta)^2,$$

and define $R^{L',L}$ as an operator

$$F_{\ell,n} = \left(R^{L',L} f\right)_{\ell,n} = F\left(\left(\cos\phi'_\ell \sin\theta'_n,\ \sin\phi'_\ell \sin\theta'_n,\ \cos\theta'_n\right)\right),$$
$$\ell = 1,\dots,n'_\phi,\ n = 1,\dots,n'_\theta.$$

To perform the truncation exactly, $n_\phi$ has to be chosen so that

$$n_\phi \geq K + N + 1. \tag{2.34}$$

A trivial algorithm for the spherical harmonic transform can be described in two steps.

1. Evaluate $f_m^n$ using the quadrature rule from Lemma 2.2.4:

$$f_n^m = \int_{\mathbb{S}^2} f(\hat{s}) Y_n^{-m}(\hat{s}) d\hat{s} = \sum_{\ell=1}^{n_\phi} \sum_{k=1}^{n_\theta} f_{\ell,k} w_{k,l}. \tag{2.35}$$

2. Define $F$ as in (2.33) and evaluate $F$ on the corresponding grid:

$$F_{\ell,n} = \sum_{k=0}^{N} \sum_{m=-k}^{k} f_k^m Y_m^k(\theta'_n,\phi'_\ell), \qquad \ell = 1,\dots,n'_\phi,\ n = 1,\dots,n'_\theta. \tag{2.36}$$

The fast spherical harmonic transform makes use of the structure of the sums (2.35,2.36) exploiting the fast Fourier and Legendre transforms. This algorithm proceeds as follows.

1. For every $n = 1,\dots,n_\theta$, compute

$$\hat{f}_n^m = \frac{2\pi}{n_\phi} \sum_{\ell=1}^{n_\phi} f_{\ell,n} \mathrm{e}^{i\frac{2\pi}{n_\phi}(\ell-1)m}, \qquad m = -n_\theta + 1,\dots,n_\theta - 1.$$

   with the help of the inverse fast Fourier transform. Particularly, for $m < 0$

$$\hat{f}_n^m = \hat{f}_n^{\left(m+n_\phi\right) \bmod n_\phi}.$$

   This operation is of the complexity $O(n_\theta n_\phi \log n_\phi)$.

2. For every $m = -N,\dots,N$, $k = 1,\dots,n'_\theta$, evaluate

$$\hat{F}_k^m = \epsilon_{N+1}^m \sum_{n=1}^{n_\theta} \hat{f}_n^m \tilde{w}_n \frac{\bar{P}_{N+1}^m(\cos\theta'_k)\bar{P}_N^m(\cos\theta_n) - \bar{P}_N^m(\cos\theta'_k)\bar{P}_{N+1}^m(\cos\theta_n)}{\cos\theta'_k - \cos\theta_n},$$

where $\bar{P}_n^m$ are the normalized associated Legendre functions, see (2.15), $\tilde{w}_n$ are the weights of the Gauss-Legendre quadrature of the order $n_\theta$ and

$$\epsilon_n^m = \sqrt{\frac{n^2 - m^2}{4n^2 - 1}}.$$

If $\cos\theta_n = \cos\theta'_n$, the quotient can be evaluated using l'Hôpital's rule. For $m = -N, \ldots, N$ the matrices

$$\begin{aligned}
(\mathcal{P}_1^m)_{kn} &= \frac{\bar{P}_{N+1}^m(\cos\theta'_k)\bar{P}_N^m(\cos\theta_n)}{\cos\theta'_k - \cos\theta_n}, \\
(\mathcal{P}_2^m)_{kn} &= \frac{\bar{P}_N^m(\cos\theta'_k)\bar{P}_{N+1}^m(\cos\theta_n)}{\cos\theta'_k - \cos\theta_n}, \qquad k = 1, \ldots, n'_\theta,\ n = 1, \ldots, n_\theta,
\end{aligned}$$

can be efficiently represented in the $\mathcal{H}$-matrix format (as Nyström discretizations of the asymptotically smooth kernels, see Section 2.1.3) or with the help of the one-dimensional fast multipole method, see [187].

This operation can be performed with the asymptotic complexity $O(Nn'_\theta \log n'_\theta)$.

3. Compute the quantities

$$F_{m,n} = \sum_{\ell=-N}^{N} \hat{F}_n^\ell \mathrm{e}^{-i\frac{2\pi}{n'_\phi}\ell(m-1)}, \qquad m = 1, \ldots n'_\phi,\ n = 1, \ldots, n'_\theta,$$

with the help of the fast Fourier transform.

Note that from the description of the spherical harmonics transform it follows that, for all $N \in \mathbb{N}_+$,

$$R^{N,N} = \mathrm{Id}\,.$$

**Remark 2.2.9.** *In our implementation of the fast multipole algorithm, we use*

$$\begin{aligned}
n_\theta &= K + 1, \qquad & n_\phi &= 2n_\theta, \\
n'_\theta &= N + 1, \qquad & n'_\phi &= 2n'_\theta.
\end{aligned}$$

*In the course of the fast multipole algorithm it is also necessary to evaluate the function given on the 'new' grid* $\left(\theta'_j, \phi'_k\right)$, $j = 1, \ldots, n'_\theta, k = 1, \ldots, n'_\phi$ *on the 'old' grid* $(\theta_j, \phi_k)$, $j = 1, \ldots, n_\theta, k = 1, \ldots, n_\phi$. *The algorithm proceeds as in 1-3, interchanging in the description* $n_\phi$ *and* $n'_\phi$ *and* $n_\theta$ *and* $n'_\theta$. *The matrix-vector multiplication in Step 2 has to be substituted by the matrix-vector multiplication with transposed matrices* $(\mathcal{P}_1^m)^T$, $(\mathcal{P}_2^m)^T$.

The transpose of the spherical harmonics transform is

$$\left(R^{N,M}\right)^T = R^{M,N}.$$

**Transfer Matrices (M2M and L2L Translation Operators)** Let clusters $\tau_\alpha \notin \mathcal{L}_{\mathcal{T}_\mathcal{I}}$ and $\tau_\beta \in \text{sons}(\tau_\alpha)$ be located correspondingly at the levels $k$ and $k+1$ of the cluster tree. Let the centers of their bounding boxes be $c_\alpha,\ c_\beta$. Then the translation operators for the column cluster basis are defined as:

$$T_c^{\tau_\beta}(s) = R^{L_{k+1},L_k} D^{\tau_\alpha,\tau_\beta}(-s),$$

where $R^{L_{k+1},L_k}$ is the fast spherical harmonics transform and $D^{\tau_\alpha,\tau_\beta}(s)$ is a diagonal translation operator. Its entries are explicitly given by

$$D_{\ell\ell}^{\tau_\alpha,\tau_\beta}(s) = \exp\left(s(c_\beta - c_\alpha, \hat{r}_\ell)\right), \qquad \ell = 1,\dots,L_k, \tag{2.37}$$

where $(\hat{r}_\ell)_{\ell=1}^{L_k}$ are as in (2.29).

The translation operators for the row cluster basis are defined similarly:

$$T_r^{\tau_\beta}(s) = R^{L_{k+1},L_k} D^{\tau_\alpha,\tau_\beta}(s). \tag{2.38}$$

**Efficient Computation and Storage of Translation Operators** Let us consider a cluster $\tau_\beta$ with the bounding box centered at $c_\beta$ and its parent cluster $\tau_\alpha$ (whose bounding box is centered at $c_\alpha$). The cluster $\tau_\beta$ is located at the level $\ell$ of the cluster tree, and the cluster $\tau_\alpha$ at the level $\ell - 1$. Then the multipole-to-multipole (local-to-local) translation operator $T_c^{\tau_\beta}$ depend only on the cluster basis rank $L_\ell$ on the level $\ell$, cluster basis rank $L_{\ell-1}$ on the level $\ell - 1$ and on the $c_{\alpha\beta} = c_\beta - c_\alpha$. If the uniform partition of the domain is used, there exists only a fixed number of different $c_{\alpha\beta}$ per level, see Figure 2.1. Hence only a few translation operators need to be constructed and stored (and this is the reason to use the uniform partition of the domain).

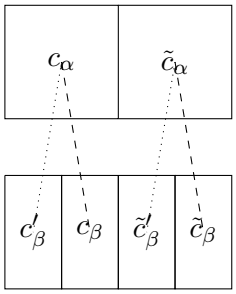


*Figure 2.1: Bounding boxes on two levels of a uniform binary cluster tree. In this case only two translation matrices per level are needed.*

#### 2.2.2.3 Multipole-to-Local Operators, or Coupling Matrices

Given an admissible block-cluster $b = (\tau_\alpha, \tau_\beta)$ located at the level $\ell$ of the block-cluster tree, the corresponding multipole-to-local translation operator is defined as, see [48, 70],

$$S^b = D^b, \tag{2.39}$$

where $D^b$ is a diagonal matrix with elements

$$D_{kk}^b = -\frac{s}{16\pi^2} \sum_{n=0}^{n_b-1} (2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|) P_n(\hat{c}_{\alpha\beta} \cdot \hat{r}_k), \qquad k = 1,\dots,L_\ell, \tag{2.40}$$

where $c_{\alpha\beta} = c_\alpha - c_\beta$. Recall that $(\hat{r}_k)_{k=1}^{L_\ell}$ are the nodes of the quadrature on the unit sphere, see also (2.29).

**Remark 2.2.10.** *In the work [57], the multipole-to-local operator is defined slightly differently, namely,*

$$S^b = R^{L_\ell, N_\ell} \tilde{D}^b R^{N_\ell, L_\ell}, \tag{2.41}$$

*where $\tilde{D}^b$ is a diagonal matrix*

$$\tilde{D}^b_{kk} = -\frac{s}{16\pi^2} \sum_{n=0}^{n_b-1} (2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|) P_n(\hat{c}_{\alpha\beta} \cdot \hat{s}_k), \qquad k = 1, \ldots, N_\ell. \tag{2.42}$$

*Here $(\hat{s}_k)_{k=1}^{N_\ell}$ is the set of quadrature points on the unit sphere, see Lemma 2.2.4. Given $L_\ell = 2n_\ell^2$, an accurate choice of $N_\ell$ is*

$$N_\ell = 2n_{max}^2, \qquad n_{max} \geq \left\lceil \frac{2n_\ell + n_b + 1}{2} \right\rceil, \tag{2.43}$$

*for all admissible $b$ located at the level $\ell$, see also (2.34) and [57]. However, in practice, a slightly more efficient $N_\ell = 2 \max\limits_{b \in \mathcal{L}_+^\ell} n_b^2$ does not deteriorate the accuracy (this value also coincides with the heuristic suggested in [57] for the non-decay case).*

*In the present work we use the coupling matrices defined by (2.39), rather than (2.41): our numerical experiments did not encounter a significant deterioration of accuracy when a simpler and more efficient (2.39) is used.*

**Efficient Construction and Storage of Multipole-to-Local Operators** A straightforward computation of the diagonal translation matrix (2.40) would require $O(n_b L_\ell)$ operations, which, for $s = -i\kappa$, $\kappa \in \mathbb{R}$, scales as $O(\kappa^3)$, see Section 2.2.3.2. Although this operation, as we show in the second part of this section, is repeated only a constant number of times per level, it can potentially destroy asymptotic complexity estimates of the fast multipole algorithm (see also [57]). There are several ways to deal with this problem, namely, the use of the Clenshaw summation algorithm [60] or the local interpolation approach, briefly described in [57]. We used the method that bears similarities with the latter one. More specifically, the function (see the expression (2.40))

$$f(t) = -\frac{s}{(4\pi)^2} \sum_{n=0}^{n_b-1} (2n+1) i^n h_n(is\|c_{\alpha\beta}\|) P_n(t),$$

is a polynomial in $t \in [-1,\ 1]$ of degree $n_b - 1$, hence can be represented by its values in $n_b$ Chebyshev points $\{t_j\}_{j=1}^{n_b}$ of the second kind. The evaluation at any other point $p \in [-1,\ 1]$ can be done with the help of the barycentric Lagrange interpolation [163]:

$$f(p) = \begin{cases} \dfrac{\sum\limits_{j=1}^{n_b}{}' (-1)^j \frac{f(t_j)}{p - t_j}}{\sum\limits_{j=1}^{n_b}{}' \frac{(-1)^j}{p - t_j}}, & p \neq t_j, \\ f(t_j), & p = t_j, \end{cases} \tag{2.44}$$

where the prime indicates that the terms $j = 1$ and $j = n_b$ are multiplied by $\frac{1}{2}$. Our task is to evaluate this fraction for $O(k^2)$ points $p = p_1, \ldots, p_{L_\ell}$. Clearly, summations in the numerator and the denominator can be viewed as the multiplication of the matrix

$$M_{ij} = \begin{cases} \frac{1}{p_i - t_j}, & p_i \neq t_j, \\ 0, & \text{else}, \end{cases} \qquad i = 1, \ldots, L_b,$$

by the corresponding vectors. This matrix, in turn, for large $n_b,\ L_\ell$ can be efficiently approximated with the help of $\mathcal{H}$-matrix techniques, and the evaluation of (2.44) for $L_b$ points will require at most $O(L_\ell \log L_\ell)$ operations. The cases $p_j = t_i$ should be treated explicitly. The disadvantage of this method is that it needs the $\mathcal{H}$-matrix approximation to be quite accurate and hence is efficient only for rather big values of $n_b,\ L_\ell$.

As before, the symmetry of the quadrature on the unit sphere, as well as the uniformity of the block-cluster tree allow us to construct and store per level only a small number of multipole-to-local translations (see also [48]). This is due to the fact that the elements of the matrix $D^b$

$$D^b_{kk} = -\frac{s}{16\pi^2} \sum_{n=0}^{n_b - 1} (2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|) P_n(\hat{c}_{\alpha\beta} \cdot \hat{r}_k), \qquad k = 1, \ldots, N_\ell,$$

depend only on the direction $\hat{c}_{\alpha\beta}$ and on the distance $d_b = \|c_{\alpha\beta}\|$. The value $n_b$, as we show later, depends only on $d_b$ and the size of a cluster at the level $\ell$. Hence the elements of the matrix $D^b$ can be obtained by permuting the diagonal entries of the matrix

$$\tilde{D}^b_{kk} = -\frac{s}{16\pi^2} \sum_{n=0}^{n_b - 1} (2n+1)(-i)^n h_n(is\|c_{\alpha\beta}\|) P_n(\hat{\lambda} \cdot \hat{r}_k), \qquad k = 1, \ldots, N_\ell,$$

where $\hat{\lambda} = [\eta \hat{c}_{\alpha\beta,1},\ \mu \hat{c}_{\alpha\beta,2},\ \nu \hat{c}_{\alpha\beta,3}]$, with $\eta, \mu, \nu \in \{-1, +1\}$.

A more efficient realization of (2.41) in (2.12) reads as

$$y^t = R^{L_\ell, N_\ell} \sum_{v \in R_t} D^b R^{L_\ell, N_\ell} x^v.$$

**Remark 2.2.11.** *Although sections on efficient construction and storage of the cluster basis and translation matrices may seem to provide unnecessary technical details, they are* ***crucial*** *for the implementation of the fast multipole algorithm. An algorithm implemented without them appears to be unpractical even for quite large problems (about $10^5$ unknowns).*

### 2.2.3 Error Control of the High-Frequency FMM

The question of the proper choice of the lengths of expansions in the fast multipole method had been intensively studied in [51, 52, 61, 69, 70, 130, 152]. Since Bessel functions are rather difficult to analyze, in most cases the error analysis is based on asymptotic expansions or explicit bounds on these functions; a very precise, near-optimal error analysis was recently made in [51, 52]. To our knowledge, these works use the fact that the wavenumber is purely real, and hence cannot be in a straightforward way adapted to the complex wavenumber case. In the recent works [91, 102] the authors numerically investigate the error of the

truncation of the Gegenbauer's series in the case when the wavenumber has a decaying part, as well as provide empirical formulas for the choice of the length of the expansion. In this section we study analytically the error of the truncation of the Gegenbauer's series for the complex wavenumber, as well as analyze other sources of errors of the fast multipole method.

#### 2.2.3.1 Behavior of Spherical Bessel and Hankel Functions

First we examine the behavior of spherical Bessel and Hankel functions of a complex argument. There exists a wide range of literature on these functions, see e.g. classical monographs [149, 184], and their asymptotic behavior is to a large extent known. We summarize these results here. First we consider spherical Bessel functions $j_n(z)$, $\arg z \in (0,\ \pi)$ in different regimes.

1. **Fixed order $n$, small argument $z$.** As $z \to 0$ [3, (9.1.7)]:

$$j_n(z) \to \frac{\sqrt{\pi} z^n}{2^{n+1}} \frac{1}{\Gamma(n+\frac{3}{2})}.$$

.

2. **Order $n$ : $n < |z|$, $|z| \to +\infty$.** We are interested in the range $\arg z \in (0,\ \pi)$. The expression [3, (10.1.14)]

$$j_n(z) = \frac{1}{2}(-i)^n \int_{-1}^{1} \mathrm{e}^{izt} P_n(t) dt \tag{2.45}$$

shows that

$$|j_n(-\bar{z})| = |j_n(z)|\,.$$

For $z \in \left(0, \frac{\pi}{2}\right)$, the following asymptotic expansion holds for $j_n(z)$, see expressions (10.17.13, 10.17.14, 10.17.15) combined with (10.4.4) in [1]:

$$\begin{aligned} j_n(z) = \frac{1}{2z} \Bigg( & (-i)^{n+1}\mathrm{e}^{iz} + i^{n+1}\mathrm{e}^{-iz} - b_n \frac{(-i)^{n+2}\mathrm{e}^{iz}}{z} \\ & - b_n \frac{i^{n+2}\mathrm{e}^{-iz}}{z} + (-i)^{n+1}\mathrm{e}^{iz} R_n^+(z) + i^{n+1}\mathrm{e}^{-iz} R_n^-(z) \Bigg), \end{aligned}$$

where

$$b_n = \frac{(2n+1)^2 - 1}{8},$$

and

$$|R_n^+| \le 2 \frac{((2n+1)^2-1)((2n+1)^2-3^2)}{2 \cdot 8^2} \frac{\mathrm{e}^{\frac{n^2+n}{|z|^2}}}{|z|^2},$$

$$|R_n^-| \le 4 \frac{((2n+1)^2-1)((2n+1)^2-3^2)}{2 \cdot 8^2} \frac{\mathrm{e}^{2\frac{n^2+n}{|z|^2}}}{|z|^2}.$$

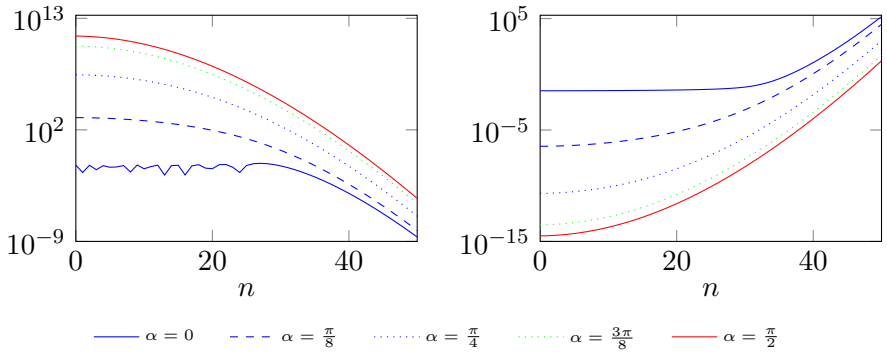


*Figure 2.2: In the left plot $|j_n(re^{i\alpha})|$ for different values of $\alpha$ and fixed $r = 30$ is depicted. The magnitude $|h_n(re^{i\alpha})|$ for the same values of $\alpha$ and $r$ is plotted on the right.*

From the above we can see that

$$|j_n(z)| \sim \frac{e^{\operatorname{Im} z}}{2|z|} \left| \left(1 - i\frac{b_n}{z}\right) + (-1)^{n+1} e^{2iz} \left(1 + i\frac{b_n}{z}\right) + f(z,n) \right|,$$

where $f(z,n) = O\left(\frac{n^4}{|z|^2}\right)$. Alternatively,

$$|j_n(z)| \sim \frac{e^{\operatorname{Im} z}}{2|z|} \left| \left(1 + (-1)^{n+1} e^{2iz}\right) + \delta(z,n) \right|, \tag{2.46}$$

where $\delta(z,n) = O\left(\frac{n^2}{|z|} + \frac{n^4}{|z|^2}\right)$.

3. **Regime $n \approx |z|$, $n \to +\infty$.** Let $z = (n + \frac{1}{2})t$. We are interested in the case $\arg t \in (0,\ \pi)$.

   The asymptotic expansion for this regime can be found in [3, 9.3.35,10.4.59]:

$$j_n\left(\left(n + \frac{1}{2}\right)t\right) \sim \frac{1}{(2n+1)\sqrt{t}} \frac{e^{(n+\frac{1}{2})\eta(t)}}{(1-t^2)^{\frac{1}{4}}} \tag{2.47}$$

   where $\eta(t) = \sqrt{1-t^2} - \log\left(\frac{1+\sqrt{1-t^2}}{t}\right)$.

   In the case $t \in \mathbb{R}$, spherical Bessel functions $j_n(t(n + \frac{1}{2}))$ oscillate, however, remain bounded as $n \to +\infty$.

4. **The order is much larger than the argument $n \gg |z|$, $n \to +\infty$.**

   In this regime $j_n(z)$ decays super-exponentially, see [3, (9.3.1)]:

$$j_n(z) \sim \sqrt{\frac{e}{2}} \frac{(ez)^n}{(2n+1)^{n+1}}. \tag{2.48}$$

Another bound on spherical Bessel functions of complex argument valid for all $n \in \mathbb{N}$ is given by [3, 9.1.62]

$$|j_n(z)| \leq \frac{|z|^n}{(2n+1)!!} e^{\operatorname{Im} z} = \frac{|z|^n (2n)!}{2^n n!} e^{\operatorname{Im} z} \tag{2.49}$$

Using Stirling's approximation

$$1 \leq \frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} \leq \frac{\mathrm{e}}{\sqrt{2\pi}},$$

this can be rewritten as

$$|j_n(z)| \leq \mathrm{e}^{\operatorname{Im} z} \frac{\mathrm{e}}{2|z|\sqrt{\pi}} \left(\frac{|z|\mathrm{e}}{2(n+1)}\right)^{n+1}. \tag{2.50}$$

The behavior of spherical Hankel functions $h_n(z)$ is in some sense opposite to that of spherical Bessel functions: they decay exponentially with $\operatorname{Re} z$ in the regime $n < |z|$, see Figure 2.2.

1. **Fixed order $n$, small argument $z$.** The behavior of the function $h_n(z)$, $z \to 0$, is given by [3, 9.1.9]:

   $$h_n(z) \to -i\frac{\Gamma(n+\frac{1}{2})}{\sqrt{\pi}} \frac{2^n}{z^{n+1}} \tag{2.51}$$

   as $z \to 0$.

2. **Order $n < |z|$.**

   According to [1, 10.17.13,10.17.14,10.17.15],

   $$|h_n(z)| = \frac{\mathrm{e}^{-\operatorname{Im} z}}{|z|} \left|1 + i\frac{b_n}{z} + R_1(z)\right|,$$

   where $b_n = \frac{(2n+1)^2-1}{8}$ and, for $\arg z \in [0,\ \pi]$ (which is the case of interest for us),

   $$|R_1(z)| \leq \frac{\left((2n+1)^2-1\right)\left((2n+1)^2-3^2\right)}{2 \cdot 8^2|z|^2} \mathrm{e}^{\frac{n^2+n}{|z|^2}}.$$

   This implies that in the regime $n < |z|$

   $$|h_n(z)| \sim \left(1 + \gamma\left(|z|, n\right)\right) \frac{\mathrm{e}^{-\operatorname{Im} z}}{|z|}, \tag{2.52}$$

   where $\gamma(|z|, n) = O\left(\frac{n^2}{|z|} + \frac{n^4}{|z|^2}\right)$.

3. **Regime $n \approx |z|$, $n \to +\infty$.** Let $z = (n+\frac{1}{2})t$. We are interested in the case $\arg t \in (0,\ \pi)$.

   The behavior of the spherical Hankel function $h_n(z)$ is defined by the asymptotic expansion given in [3, 9.3.37, 10.4.59]:

   $$h_n\left(\left(n+\frac{1}{2}\right)t\right) \sim -\frac{2i}{(2n+1)\sqrt{t}(1-t^2)^{\frac{1}{4}}} \mathrm{e}^{-(n+\frac{1}{2})\eta(t)}, \tag{2.53}$$

   where $\eta(t) = \sqrt{1-t^2} - \log\left(\frac{1+\sqrt{1-t^2}}{t}\right)$. In our case $\arg t \in (0,\ \pi)$.

   If $t$ is purely real, the function oscillates but remains bounded with $n \to +\infty$.

4. **The order is much larger than the argument $n \gg |z|$, $n \to +\infty$.** In this regime $h_n(z)$ experiences superexponential growth, see [3, 9.3.1]:

$$h_n(z) \sim -\frac{i}{z}\sqrt{\frac{2}{\mathrm{e}}}\left(\frac{2n+1}{\mathrm{e}z}\right)^n. \tag{2.54}$$

Additionally, magnitudes of spherical Hankel functions are strictly monotonically increasing in their order, see [1, 10.37.1]:

$$|h_n(z)| < |h_m(z)|,\ m > n, \tag{2.55}$$

when $\operatorname{Re} z \geq 0$. The proof of this result can be found in [88].

#### 2.2.3.2 Truncation of the Fast Multipole Expansion

In this section we study the dependence of the truncation parameter $N$ in (2.28) on the complex wavenumber $s \in \mathbb{C}$, $\operatorname{Re} s > 0$.

Let $\|x\|$ and $\|y\|$ be fixed, and let also $\|x\| > \|y\|$. We are looking for $N$ s.t.

$$\left| h_0(is\|x-y\|) - \sum_{\ell=0}^{N-1} (2\ell+1) h_\ell(is\|x\|) j_\ell(is\|y\|) P_\ell\left(\hat{x}\cdot\hat{y}\right) \right| < \epsilon, \tag{2.56}$$

for a fixed $\epsilon > 0$. Crucially, to truncate the Gegenbauer's series we use the criteria based on the absolute error rather than the relative one, similarly to [57, 91].

Let $t = \hat{x}\cdot\hat{y}$, $t \in [-1,\ 1]$. Then the addition theorem for the spherical Bessel functions (2.24) is the Legendre polynomial expansion of the function

$$f(t) = h_0\left(is\left(\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}\right) = -\frac{\mathrm{e}^{-s\left(\|x\|^2+\|y\|^2-2\|x\|\|y\|t\right)^{\frac{1}{2}}}}{s\left(\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}}$$

in $t$. For $\operatorname{Re} t < t_{max} = \frac{\|x\|^2+\|y\|^2}{2\|x\|\|y\|}$ the function $f(t)$ is analytic, hence Theorem 2.2.1 can be applied. The parameter $\rho$ for the corresponding Bernstein ellipse is defined as

$$\rho = t_{max} + \sqrt{t_{max}^2 - 1} = \frac{\|x\|}{\|y\|}.$$

This rate of convergence coincides with the one deduced in [152, 155].

**Remark 2.2.12.** *If $\operatorname{Re} s = 0$ (no-decay case), the length of the Gegenbauer's series (2.56) can be estimated by a semi-empirical formula, see [61],*

$$N = |s|\|y\| + C\log(\pi + |s|\|y\|),$$

*where $C$ is the constant that depends on accuracy. The convergence of the Gegenbauer's series for large $|s|\|y\|$ starts when $j_n(is\|y\|)$ starts decaying superexponentially [69, 70], see (2.48).*

Let us first summarize the results of this section. The criteria based on (2.56) is often used to choose the length of the multipole expansion, see [57, 70]. Under the condition

$$\left|\frac{\operatorname{Im} s}{\operatorname{Re} s}\right| < C, \qquad \operatorname{Re} s > \sigma > 0, \tag{2.57}$$

for some $C, \sigma > 0$, the length of the fast multipole expansion can be bounded by a constant independent of $\operatorname{Im} s$ (that depends though on $C$, $\sigma$, $\epsilon$, $\rho$). This behavior is similar to that of $\mathcal{H}$-matrices, see Section 2.1.4. The result can be seen by noticing that there exists $r \in \mathbb{R}$, $r > \sigma$, s.t. for all $s \in \mathbb{C}$ with $|s| > r$:

$$|h_0(is\|x-y\|)| = \left| \frac{\mathrm{e}^{-\operatorname{Re} s\|x-y\|}}{|s|\|x-y\|} \right| < \epsilon.$$

Hence the length of the expansion is bounded by the maximal of the lengths of the expansions over all $s \in \mathbb{C}$ satisfying (2.57) and $|s| < r$. This justifies the use of the empirical formulas for the length of the expansion derived in [91]: it indeed can be bounded by a constant when decay is significantly large. Another implication of this is the complexity of the fast multipole approximation to the Galerkin discretization of the Helmholtz single layer boundary operator: for multilevel fast multipole methods based on the expansion (2.56), for $s$ satisfying (2.57) and $M = O(|s|^2)$, it scales as $O(M)$.

For $s = |s|\mathrm{e}^{i\alpha}$ with $|\alpha|$ close to $\frac{\pi}{2}$ this constant bound is far from optimal. This can be seen in Figure 2.3 ($\|y\| = 2$, $\|x\| = 4$, $\hat{x}\cdot\hat{y} = 1$): the length of the expansion needed to achieve a given accuracy increases with increasing $|s|$ on the whole interval under consideration. Notably, for $|\alpha| \le \frac{\pi}{4}$, the length of the expansion does not seem to increase with $|s|$. Our goal is to provide some theoretical justification for this phenomenon. Here we present more refined bounds on (2.56), motivating as well the error analysis in the subsequent sections.

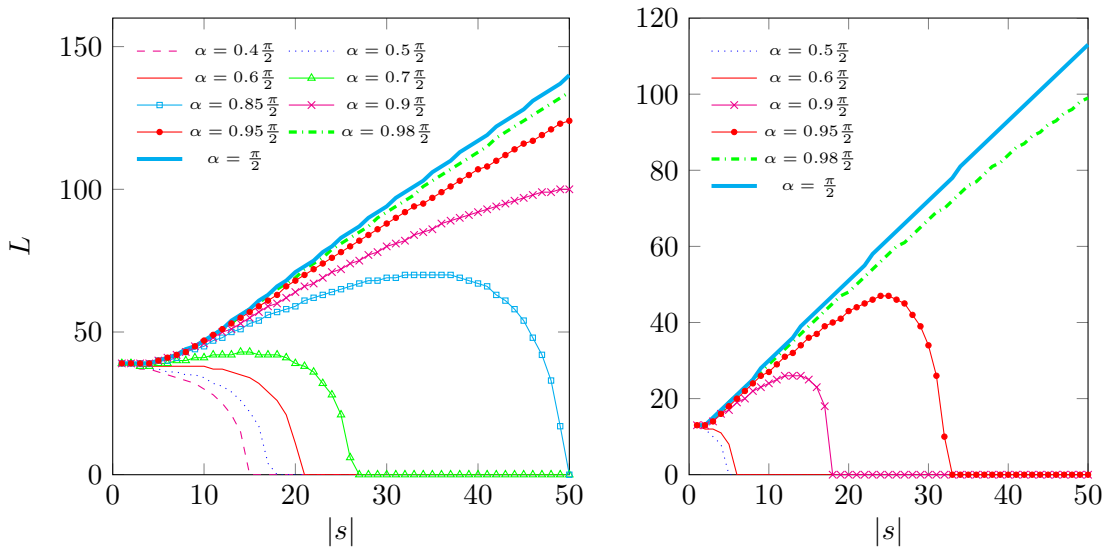


*Figure 2.3: Dependence of the length of the truncated expansion for accuracies $\epsilon = 10^{-12}$ (the left plot) and $\epsilon = 10^{-4}$ (the right plot) on $|s|$, for $s = |s|\mathrm{e}^{i\alpha}$ and different values of $\alpha$.*

The following result is due to [182].

**Theorem 2.2.13.** *Let the function $f$ be analytic inside and on a Bernstein ellipse $E_\rho$, $\rho > 1$. Let $\{a_n\}$ be the coefficients of the Legendre series expansion of $f$. Then the following bound holds true for all $n \ge 0$:*

$$|a_n| \le (2n+1)\rho^{-n-1}\mathcal{M}\pi^{-1}l(E_\rho)(1-\rho^{-2})^{-1},$$

*where $\mathcal{M} = \max_{z\in E_\rho} |f(z)|$ and $l(E_\rho)$ is the circumference of the ellipse $E_\rho$.*

The next lemma bounds the values of the function $f(t)$ on the Bernstein ellipse.

**Lemma 2.2.14.** *Given $s = |s|e^{i\alpha}$, $\alpha \in \left[-\frac{\pi}{2},\ \frac{\pi}{2}\right]$, the function*

$$f(t) = h_0\left(is\left(x^2 + y^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}\right),$$

*inside the Bernstein ellipse $E_\xi$, $\xi < \rho = \frac{\|x\|}{\|y\|}$, is bounded by*

$$|f(t)| \le \max\left(1, \mathrm{e}^{\|y\||s|(|\sin\alpha| - \cos\alpha)\lambda(\rho)}\right)\left(|s|\|y\|\sqrt{\rho}\sqrt{\rho - \xi + \frac{1}{\rho} - \frac{1}{\xi}}\right)^{-1},$$

*where*

$$\lambda(\rho) = \frac{1}{2}\left(\rho - \frac{1}{\rho}\right). \tag{2.58}$$

*Proof.* Let us bound the numerator and the denominator of

$$f(t) = -\frac{\exp\left(-s\left(\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}\right)}{s\left(\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}}$$

on the boundary of the Bernstein ellipse

$$E_\xi = \left\{z : z = \frac{\xi \mathrm{e}^{i\phi} + \xi^{-1}\mathrm{e}^{-i\phi}}{2},\ \phi \in [0,\ 2\pi)\right\}, \qquad \xi < \rho = \frac{\|x\|}{\|y\|}.$$

The absolute value of the denominator

$$\begin{aligned} d(\phi) :&= \left|\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|\frac{(\xi e^{i\phi} + \xi^{-1}e^{-i\phi})}{2}\right|^{\frac{1}{2}} \\ &= \sqrt{\|x\|\|y\|}\left|\rho + \rho^{-1} - \xi e^{i\phi} - \xi^{-1}e^{-i\phi}\right|^{\frac{1}{2}}. \end{aligned}$$

The minimum of this expression is achieved when $\phi = 0$:

$$d(\phi) \ge \sqrt{\rho}\|y\|\sqrt{\rho - \xi + \frac{1}{\rho} - \frac{1}{\xi}}.$$

Let, for a given $\xi < \rho$,

$$G(t) := \exp\left(-s\left(\|x\|^2 + \|y\|^2 - 2\|x\|\|y\|t\right)^{\frac{1}{2}}\right),\ t \in E_\xi.$$

From the maximum principle it follows that for all $t$ in the interior of the Bernstein ellipse $E_\rho$ (and hence $E_\xi$, $1 < \xi < \rho$),

$$|G(t)| \le \max_{t' \in E_\rho} |G(t')|.$$

Hence, we are looking for $\max\limits_{t' \in E_\rho} |G(t')|$. Given $s = |s|\mathrm{e}^{i\alpha}$, for all $t \in E_\rho$,

$$\begin{aligned} |G(t)| &\le \left|\exp\left(-s(2\|x\|\|y\|)^{1/2}\sqrt{\frac{\rho + \rho^{-1}}{2} - \frac{\rho e^{i\phi} + \rho^{-1}e^{-i\phi}}{2}}\right)\right| \\ &= \exp\left(-(2\|x\|\|y\|)^{1/2}|s|\sqrt{|z|}\cos\left(\alpha + \frac{\beta}{2}\right)\right), \end{aligned} \tag{2.59}$$

where

$$z = \frac{\rho + \rho^{-1}}{2} - \frac{\rho e^{i\phi} + \rho^{-1} e^{-i\phi}}{2},\ \beta = \arg z. \tag{2.60}$$

We find $|z| = Z(\beta)$ using the geometric meaning of (2.60).

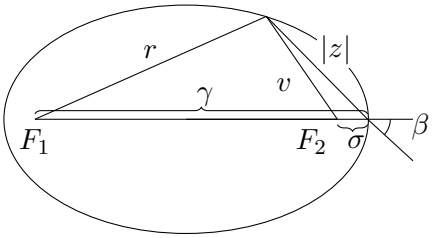


*Figure 2.4: Bernstein ellipse*

The foci of the Bernstein ellipse lie in the points

$$F_1 = (-1,\ 0),\ F_2 = (1,\ 0).$$

From the properties of the Bernstein ellipse, using Figure 2.4,

$$\gamma = \frac{\rho + \rho^{-1}}{2} + 1,\ \sigma = \frac{\rho + \rho^{-1}}{2} - 1, \tag{2.61}$$

$$r + v = \gamma + \sigma = \rho + \rho^{-1}, \tag{2.62}$$

$$r^2 = |z|^2 + \gamma^2 - 2|z|\gamma\cos\beta, \tag{2.63}$$

$$v^2 = |z|^2 + \sigma^2 - 2|z|\sigma\cos\beta, \tag{2.64}$$

where

$$\beta \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right).$$

From equations (2.62) and (2.64), we obtain the following expression for $r,\ |z|$:

$$\begin{aligned} |z|^2 + \sigma^2 - 2|z|\sigma\cos\beta &= (\gamma + \sigma - r)^2 \\ &\overset{(2.63)}{=} \gamma^2 + \sigma^2 + (|z|^2 + \gamma^2 - 2|z|\gamma\cos\beta) + 2\gamma\sigma - 2(\gamma+\sigma)r. \end{aligned}$$

Then $r$ can be written as a function of $|z|$:

$$r = \gamma - |z|\frac{\gamma - \sigma}{\gamma + \sigma}\cos\beta \overset{(2.61)}{=} \gamma - \frac{2|z|\cos\beta}{\gamma + \sigma}.$$

Hence,

$$\begin{aligned} r^2 &= \left(\gamma - \frac{2|z|\cos\beta}{\gamma + \sigma}\right)^2 \\ &\overset{(2.63)}{=} |z|^2 + \gamma^2 - 2|z|\gamma\cos\beta. \end{aligned}$$

From this we obtain the following expression for $|z|$:

$$|z|^2\left(1 - \frac{4\cos^2\beta}{(\gamma+\sigma)^2}\right) = 2|z|\gamma\cos\beta\left(1 - \frac{2}{\gamma+\sigma}\right).$$

From this follows that $|z| = 0$ or

$$|z| = 2\gamma \cos\beta \left(1 - \frac{2}{\sigma+\gamma}\right)\left(1 - \frac{4\cos^2\beta}{(\sigma+\gamma)^2}\right)^{-1},\ \beta \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right). \tag{2.65}$$

Note that for $|\beta| = \frac{\pi}{2}$ the above expression gives $|z| = 0$.

Hence, inserting (2.65) into (2.59), we obtain:

$$\max_{t\in E_\rho} G(t) = \max_{\beta\in\left[-\frac{\pi}{2},\ \frac{\pi}{2}\right]} g(\beta),$$

where

$$\begin{aligned} g(\beta) = \exp\Bigg(-2\left(\|x\|\|y\|\gamma\left(1 - \frac{2}{\sigma+\gamma}\right)\right)^{1/2}|s| \\ \times\sqrt{\cos\beta}\left(1 - \frac{4\cos^2\beta}{(\sigma+\gamma)^2}\right)^{-1/2}\cos\left(\alpha + \frac{\beta}{2}\right)\Bigg). \end{aligned} \tag{2.66}$$

This expression has to be maximized in $\beta$. Let $\mu := \frac{2}{\sigma+\gamma}$ and

$$R(\beta) := -\sqrt{\cos\beta}\cos\left(\alpha + \frac{\beta}{2}\right)\left(1 - \mu^2\cos^2\beta\right)^{-\frac{1}{2}}.$$

We consider two cases corresponding to $\alpha \geq 0$. The bounds for $\alpha \leq 0$ can be found from similar considerations.

1. $\alpha \in \left[0, \frac{\pi}{4}\right]$. In this case, for all $\beta \in \left(-\frac{\pi}{2},\ \frac{\pi}{2}\right)$,

   $$-\cos\left(\alpha + \frac{\beta}{2}\right) \geq 0$$

   and hence

   $$R(\beta) \leq 0.$$

2. $\alpha \in \left(\frac{\pi}{4}, \frac{\pi}{2}\right]$. The maximum of $R(\beta)$ is achieved in some $\beta = \beta_* \in \left[-\frac{\pi}{2},\ \frac{\pi}{2}\right]$, s.t.

   $$\cos\left(\alpha + \frac{\beta_*}{2}\right) \leq 0,$$

   which shows that

   $$\beta_* \in \left[\pi - 2\alpha,\ \frac{\pi}{2}\right].$$

   Setting $p := \cos\beta$, with $\beta$ lying inside the above interval:

   $$\begin{aligned} R(\beta) &= -\sqrt{p}(1-\mu^2p^2)^{-\frac{1}{2}}\left(\cos\alpha\sqrt{\frac{1+p}{2}} - \sin\alpha\sqrt{\frac{1-p}{2}}\right) \\ &\leq \sqrt{p}(1-\mu^2p^2)^{-\frac{1}{2}}\left(\sin\alpha\sqrt{\frac{1-p}{2}} - \cos\alpha\sqrt{\frac{1+p}{2}}\right) \\ &\leq \sqrt{\frac{p(1-p)}{2}}\left(1-\mu^2p^2\right)^{-\frac{1}{2}}(\sin\alpha - \cos\alpha). \end{aligned}$$

Next we find the maximum of the function $f(p) = \sqrt{\frac{p(1-p)}{2(1-\mu^2 p^2)}}$ on $[0,\ 1]$. It is achieved at

$$p_* = \frac{1-\sqrt{1-\mu^2}}{\mu^2}$$

and equals

$$f(p_*) = \frac{\sqrt{1-\sqrt{1-\mu^2}}}{2\mu}.$$

Hence,

$$R(\beta) \leq \frac{\sqrt{1-\sqrt{1-\mu^2}}}{2\mu}(\sin\alpha - \cos\alpha).$$

The bound in the statement of the lemma can be deduced noting that

$$\rho^{-1} = \frac{1-\sqrt{1-\mu^2}}{\mu}, \tag{2.67}$$

$$\rho = \frac{1+\sqrt{1-\mu^2}}{\mu} \tag{2.68}$$

and $\gamma = \frac{1}{\mu}+1$. More precisely, the coefficient in the exponent near $|s|\|y\|(\sin\alpha - \cos\alpha)$ in (2.66) is

$$\begin{aligned}\lambda(\rho) &= 2\sqrt{\rho\gamma(1-\mu)}\frac{\sqrt{1-\sqrt{1-\mu^2}}}{2\mu} \\ &= \sqrt{\rho}\sqrt{\gamma(1-\mu)}\frac{\sqrt{\rho^{-1}}}{\sqrt{\mu}},\end{aligned}$$

where we applied (2.67). Inserting the explicit expression of $\gamma$ in terms of $\mu$, we obtain

$$\lambda(\rho) = \frac{\sqrt{1-\mu^2}}{\mu},$$

which, using (2.67) and (2.68), gives the explicit expression for $\lambda(\rho)$.

□

Theorem 2.2.13 and Lemma 2.2.14 allow us to formulate the following bound.

**Corollary 2.2.15.** *Let $\|x\| > \|y\| > 0$ and $1 < \xi < \rho = \frac{\|x\|}{\|y\|}$. Then for $s = |s|\mathrm{e}^{i\alpha}$ s.t. $\alpha \in [-\frac{\pi}{2},\ \frac{\pi}{2}]$ the following bound holds true:*

$$\begin{aligned}|h_n(is\|x\|)j_n(is\|y\|)| \leq{}& 2\xi^{-n-1}\sqrt{\xi^2+\xi^{-2}}\max\left(1, \mathrm{e}^{\|y\||s|(|\sin\alpha|-\cos\alpha)\lambda(\rho)}\right) \\ &\times\left(|s|\|y\|\sqrt{\rho}(1-\xi^{-2})\sqrt{\rho-\xi+\frac{1}{\rho}-\frac{1}{\xi}}\right)^{-1},\end{aligned} \tag{2.69}$$

*where $\lambda(\rho)$ is given by (2.58).*

*Proof.* To apply Theorem 2.2.13, we need to bound the perimeter of the ellipse (analytically, it is expressed via the complete elliptic integral of the second kind):

$$l(E_\xi) < 2\pi\sqrt{\frac{a^2+b^2}{2}},$$

where $a$, $b$ are correspondingly the lengths of the semi-major and semi-minor axes of the ellipse $E_\xi$. Hence

$$l(E_\xi) < 2\pi\sqrt{\xi^2 + \frac{1}{\xi^2}}.$$

□

For $s = |s|e^{i\alpha}$, $|\alpha| \leq \frac{\pi}{4}$, the length of the expansion (2.56) can be bounded by a constant that is independent of $|s|$, $\alpha$ for the range of $|s| > \sigma > 0$ for a fixed $\sigma > 0$. This is not the case for $\frac{\pi}{4} < |\alpha| \leq \frac{\pi}{2}$: Corollary 2.2.15 shows that

$$N = O\left(|s|\|y\|\left(|\sin\alpha| - \cos\alpha\right)\right),$$

which is tight for smaller values of $|s|$ and $\alpha$ close to $\frac{\pi}{2}$, however is pessimistic as $|s| \to +\infty$, as Figure 2.5 shows.

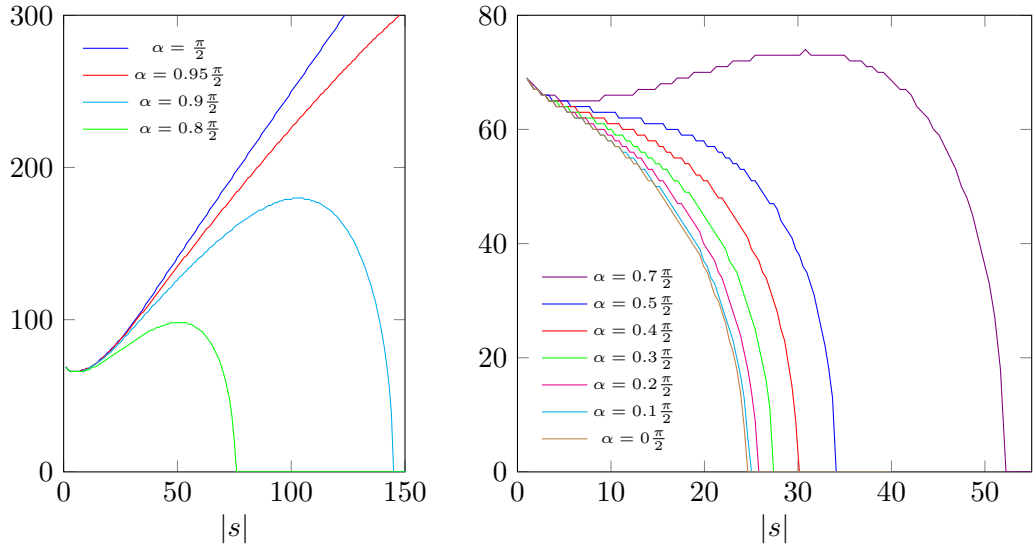


*Figure 2.5: In both plots the dependence of the length of the expansion (2.56) on $|s|$ ($s = |s|e^{i\alpha}$) for $\epsilon = 10^{-12}$ and various $\alpha$ is shown.*

**Remark 2.2.16.** *Our numerical experiments show that the length of the expansion for* $\operatorname{Re} s > 0$, *moderate values of $\rho$ ($\rho \geq 1.5$) and moderate values of $|s|$ satisfies:*

$$N \ll |s|\|x\|. \tag{2.70}$$

*The reason for this is that in the presence of decay, i.e. when* $\operatorname{Re} s > 0$, *the length of the expansion is not larger the length of the expansion in the no-decay case (keeping* $\operatorname{Im} s$ *fixed), see also Figure 2.6 and Remark 2.2.12. For the latter there exist tight formulas showing (2.70), see e.g. [52].*

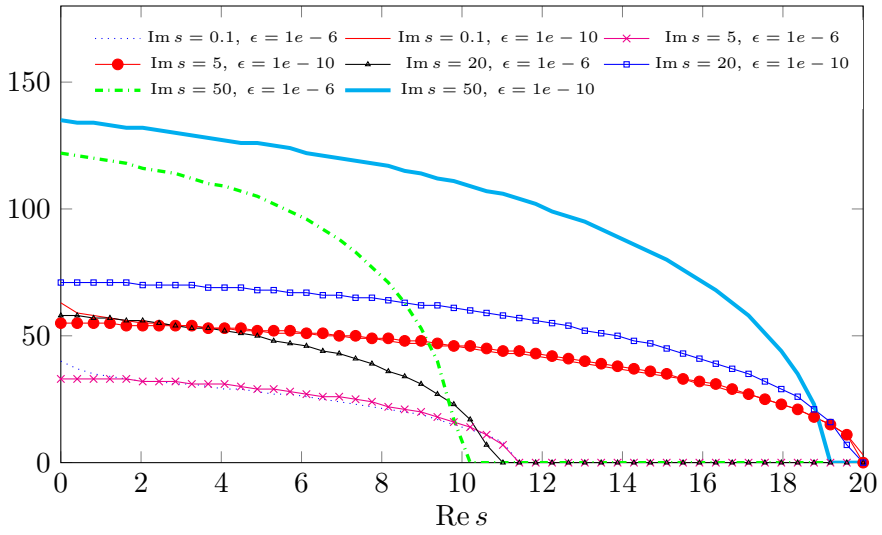


*Figure 2.6: The length of the expansion for $s = s_r + is_i$, as defined in (2.56), with varying $s_r$ and fixed $s_i$; $\|x\| = 3$, $\|y\| = 2$.*

Recall that the criteria based on (2.56) is typically used to choose the length of the multipole expansion, see [57, 70]. Numerical experiments, see Figure 2.5, show that the length of the expansion for larger values of $|s|\|y\|$ can be smaller than for smaller $|s|\|y\|$. In the fast multipole algorithm, $\|y\|$ stands for the diameter of a cluster, and $\|x\|$ for the distance between the centers of the admissible clusters, see also Section 2.2.2. Assuming that $\rho = \frac{\|x\|}{\|y\|}$ is fixed, we can conclude that for larger clusters one may need the expansion of the smaller length than for smaller ones. It is not obvious if such choice of the length of the expansion leads to the deterioration of the accuracy when doing multipole-to-multipole and local-to-local transforms. This motivates the need for the analysis of the error of the multilevel fast multipole method for a complex wavenumber case.

#### 2.2.3.3 Multilevel FMM Error Control

Give the block-cluster tree $\mathcal{T}_{I\times I}$, let $\tau_\alpha$ and $\tau_\beta$ be two admissible clusters (the block cluster $(\tau_\alpha, \tau_\beta)$ belongs to the set of admissible leaves of the block-cluster tree). In this section we consider the error of the approximation of $h_0(is\|x-y\|)$ by $\tilde{h}_0$ computed in the course of the fast multipole algorithm:

$$E = |h_0(is\|x-y\|) - \tilde{h}_0|, \ s \in \mathbb{C}, \tag{2.71}$$

where $x \in \tau_\alpha$, $y \in \tau_\beta$. .

There exist several empirical formulas [52, 61, 91, 102] that provide tight estimates for the length of expansions in the fast multipole algorithm. In works [57, 70] authors suggest that it can be chosen analyzing the convergence of the Gegenbauer's series (2.56). We adopt this approach.

In [130] the authors analyzed the full error of the fast multipole algorithm in the case $is \in \mathbb{R}$. Their error analysis uses superexponential decay of spherical Bessel functions $j_n(is\|y\|)$ in the regime $n \gg |s|\|y\|$ and the geometric convergence of the quadrature on the unit sphere for interpolating of multipole expansions in the course fast multipole method.

A straightforward application of this error analysis to the case of the complex wavenumber may result in pessimistic error bounds, since it wouldn't take into account the decay of spherical Hankel functions.

In this section we derive an explicit expression for the error of the multilevel fast multipole method for the case of general $s \in \mathbb{C}$ and comment on the choice of lengths of multipole expansions.

We study the following simple case.

**Assumption 2.2.17.** *Let the clusters $\tau_\alpha$ and $\tau_{\alpha'} \in \mathrm{sons}(\tau_\alpha)$, $\tau_\beta$ and $\tau_{\beta'} \in \mathrm{sons}(\tau_\beta)$ be given. The points $x \in \tau_{\beta'}$, $y \in \tau_{\alpha'}$. Additionally, $\tau_\alpha$, $\tau_\beta$ are admissible, in the sense of Definition 2.1.7. The points $x, y$ are chosen so that $\|x - y\| = \mathrm{dist}(\tau_\alpha, \tau_\beta)$ (typically, it is assumed that the error of the approximation provided by the fast multipole method is larger in close points of the admissible clusters). We assume that $\tau_{\alpha'}$ is a leaf, and so is $\tau_{\beta'}$. By $y_\alpha$, $x_\beta$, $y_{\alpha'}$, $x_{\beta'}$ we denote the centers of the bounding boxes of the clusters $\tau_\alpha$, $\tau_\beta$, $\tau_{\alpha'}$, $\tau_{\beta'}$.*

We assume that all spherical harmonic transforms are done exactly, see Section 2.2.2.2. Recall that with each level of the cluster tree we associate a set of quadrature points on the unit sphere given by (2.25), i.e.

$$(\hat{s}_k)_{k=1}^{2n_\theta^2}. \tag{2.72}$$

We set $n_\theta = N$ at the level where the children clusters are located and $n_\theta = M$ at the level of parent clusters. Let us also define

$$N_* = \min(N, M).$$

The fast multipole algorithm proceeds in the following stages.

1. Evaluation of the multipole expansion for the cluster $\tau_{\beta'}$. The function $f(\hat{s}) = \mathrm{e}^{-s(\hat{s}, x - x_{\beta'})}$ is sampled on the grid (2.72) of size $N \times 2N$, see Remark 2.2.5.

2. Evaluation of the multipole expansion for the cluster $\tau_\beta$. This is done in two stages. First, the multipole expansion for the cluster $\tau_{\beta'}$ is re-sampled on the grid of size $M \times 2M$ with the help of the spherical harmonic transform operator (and possibly the spherical harmonic expansion of $f(\hat{s})$ is truncated to $\min(N, M) = N_*$ summands, see Section 2.2.2.2 and Remark 2.2.9). The result of this operation is the vector of values of the function

$$b(\hat{s}) = \sum_{n=0}^{N_*-1} \sum_{m=-n}^{n} \beta_n^m Y_n^m(\hat{s}),$$
$$\beta_n^m = Q_N \left[ \mathrm{e}^{-s(\hat{q}, x - x_{\beta'})} Y_n^{m*}(\hat{q}) \right]$$

   in the points of the grid (2.72) of size $M \times 2M$. The expression for $\beta_n^m$ is obtained using Lemma 2.2.4. An alternative expression for $b(\hat{s})$ can be obtained using (2.19):

$$b(\hat{s}) = \sum_{n=0}^{N_*-1} \frac{2n+1}{4\pi} Q_N \left[ \mathrm{e}^{-s(\hat{q}, x - x_{\beta'})} P_n(\hat{s} \cdot \hat{q}) \right].$$

   Next,

$$B(\hat{s}) = \mathrm{e}^{-s(x_{\beta'} - x_\beta, \hat{s})} b(\hat{s})$$

   is evaluated at the points of the grid (2.72) of size $M \times 2M$.

3. Multipole-to-local translation. At each point of the grid $\hat{s}_k, k = 1, \ldots, 2M^2$, $B(\hat{s}_k)$ is multiplied by

$$M_{\alpha,\beta}(\hat{s}_k) = \frac{1}{4\pi} \sum_{\ell=0}^{L-1} (2\ell+1)(-i)^\ell h_\ell(is\|y_\alpha - x_\beta\|) P_\ell \left( \frac{y_\alpha - x_\beta}{\|y_\alpha - x_\beta\|} \cdot \hat{s}_k \right), \tag{2.73}$$

where $L \in \mathbb{N}$. The result of this operation is the vector of values of the function

$$F(\hat{s}) = M_{\alpha,\beta}(\hat{s}) B(\hat{s}) \tag{2.74}$$

in the points of the grid (2.72) of size $M \times 2M$.

4. Local-to-local translation. First, at each point of the grid $F(\hat{s})$ is multiplied by $\mathrm{e}^{-s(y_\alpha - y_{\alpha'}, \hat{s})}$ evaluated at this point. The result of this operation is the vector of values of

$$A(\hat{s}) = \mathrm{e}^{-s(y_\alpha - y_{\alpha'}, \hat{s})} F(\hat{s})$$

in the points of the grid (2.72) of size $M \times 2M$. Next, the (adjoint) spherical harmonic transform operator is applied to $A(\hat{s})$, possibly truncating its spherical harmonic expansion and re-sampling it on the grid (2.25) of size $N \times 2N$. The result of this operation is the vector of values of the function

$$\begin{aligned} a(\hat{s}) &= \sum_{n=0}^{N_*-1} \sum_{m=-n}^{n} \alpha_n^m Y_n^m(\hat{s}), \\ \alpha_n^m &= Q_M \left[ A(\hat{q}) Y_n^{m*}(\hat{q}) \right], \end{aligned}$$

in the points of the grid (2.72) of size $N \times 2N$. The explicit expression for the coefficients $\alpha_n^m$ is:

$$\alpha_n^m = Q_M \left[ \mathrm{e}^{-s(y_\alpha - y_{\alpha'}, \hat{q})} M_{\alpha,\beta}(\hat{q}) \mathrm{e}^{-s(x_{\beta'} - x_\beta, \hat{s})} b(\hat{q}) Y_n^{m*}(\hat{q}) \right].$$

Using the addition theorem for Legendre functions (2.19),

$$a(\hat{s}) = \sum_{n=0}^{N_*-1} \frac{2n+1}{4\pi} Q_M \left[ A(\hat{q}) P_n(\hat{q} \cdot \hat{s}) \right].$$

Finally, the result is evaluated at the point $y$, giving the approximation

$$\tilde{h}_0 = Q_N \left[ \mathrm{e}^{-s(y_{\alpha'} - y, \hat{s})} a(\hat{s}) \right].$$

Making use of the linearity of the quadrature rule, we end up with the following approximation to $h_0(is\|x - y\|)$:

$$\begin{aligned} \tilde{h}_0 = Q_M^{\hat{s}} \Bigg[ & \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ \mathrm{e}^{s(y - y_{\alpha'}, \hat{q})} P_k(\hat{q} \cdot \hat{s}) \right] \\ & \times \mathrm{e}^{-s(y_\alpha - y_{\alpha'}, \hat{s})} M_{\alpha,\beta}(\hat{s}) \mathrm{e}^{-s(x_{\beta'} - x_\beta, \hat{s})} \sum_{n=0}^{N_*-1} \frac{2n+1}{4\pi} Q_N^{\hat{r}} \left[ \mathrm{e}^{-s(x - x_{\beta'}, \hat{r})} P_n(\hat{r} \cdot \hat{s}) \right] \Bigg]. \end{aligned}$$

Our goal is to rewrite this approximation in a more convenient form. For the sake of brevity of presentation we made all computations in Appendix A. We will need the following auxiliary quantities:

$$\begin{aligned} U(\hat{s}) &= \mathrm{e}^{s(y-y_\alpha+x_\beta-x_{\beta'},\hat{s})} M_{\alpha,\beta}(\hat{s}), \\ V(\hat{s}) &= \mathrm{e}^{s(y_{\alpha'}-y_\alpha+x_\beta-x_{\beta'},\hat{s})} M_{\alpha,\beta}(\hat{s}), \\ r_K(x,\hat{s}) &= \sum_{n=K}^{+\infty} (2n+1) i^n j_n(is\|x\|) P_n(\hat{x}\cdot\hat{s}). \end{aligned}$$

As demonstrated in Appendix A, the error of the fast multipole method can be written as a sum of terms of the type:

$$\mathcal{E}_1 = Q_M^{\hat{s}} \left[ \mathrm{e}^{s(y-y_\alpha,\hat{s})} M_{\alpha,\beta}(\hat{s}) \mathrm{e}^{s(x_\beta-x,\hat{s})} \right] - h_0(is\|x-y\|), \tag{2.75}$$

and

$$\begin{aligned} \mathcal{E}_2 &= -Q_M^{\hat{s}} \left[ U(\hat{s}) r_{N_*}(x-x_{\beta'},\hat{s}) \right], \\ \mathcal{E}_3 &= Q_M^{\hat{s}} \left[ U(\hat{s}) \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(x-x_{\beta'},\hat{q}) P_k(\hat{q}\cdot\hat{s}) \right] \right], \\ \mathcal{E}_4 &= Q_M^{\hat{s}} \left[ V(\hat{s}) r_{N_*}(y_{\alpha'}-y,\hat{s}) r_{N_*}(x-x_{\beta'},\hat{s}) \right], \\ \mathcal{E}_5 &= -Q_M^{\hat{s}} \left[ V(\hat{s}) r_{N_*}(x-x_{\beta'},\hat{s}) \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q}) P_k(\hat{q}\cdot\hat{s}) \right] \right], \\ \mathcal{E}_6 &= Q_M^{\hat{s}} \left[ V(\hat{s}) \sum_{m=0}^{N_*-1} \frac{2m+1}{4\pi} Q_N^{\hat{r}} \left[ r_{2N-N_*+1}(x-x_{\beta'},\hat{r}) P_m(\hat{r}\cdot\hat{s}) \right] \right. \\ &\quad \left. \times \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q}) P_k(\hat{q}\cdot\hat{s}) \right] \right]. \end{aligned} \tag{2.76}$$

Our goal is to show how $L$ in (2.73), $N$ and $M$ have to be chosen to control each of the terms $\mathcal{E}_j$, $j = 1, \ldots, 6$. We will make use of the trivial bound, see also Lemma 2.2.4:

$$|Q_M[f(\hat{q})]| = \left| \sum_{k=1}^{2M^2} w_k f(\hat{q}_k) \right| \le \sup_{\hat{q}\in\mathbb{S}^2} |f(\hat{q})| \sum_{k=1}^{2M^2} w_k = 4\pi \sup_{\hat{q}\in\mathbb{S}^2} |f(\hat{q})|. \tag{2.77}$$

The sum of the quadrature weights $w_k = 4\pi$, because $\sum_{k=1}^{2M^2} w_k$ equals the quadrature rule of Lemma 2.2.4 applied to evaluate the integral $\int_{\mathbb{S}^2} d\hat{s} = 4\pi$, which in turn is computed by this quadrature rule exactly (as an integral over the product of two spherical harmonics $Y_0^0(\hat{s}) \equiv 1$), see Lemma 2.2.4.

Recall that

$$|P_n(t)| \le 1, \qquad t \in [-1,\ 1]\,. \tag{2.78}$$

We will make use of (2.22):

$$e^{iz\cos\theta} = \sum_{n=0}^{\infty}(2n+1)i^n j_n(z)P_n(\cos\theta). \tag{2.79}$$

For the sake of brevity, from now on

$$c_{\alpha\beta} := y_\alpha - x_\beta.$$

First, let us assume that $M \geq L$, where $L$ is as in (2.73). Due to the monotonicity result (2.55) and the bound (2.78),

$$\begin{aligned}
|M_{\alpha,\beta}(\hat{s})| &= \frac{1}{4\pi}\left|\sum_{\ell=0}^{L-1}(2\ell+1)(-i)^\ell h_\ell(is\|c_{\alpha\beta}\|)P_\ell\left(\hat{c}_{\alpha\beta}\cdot\hat{s}\right)\right| \\
&\leq \frac{1}{4\pi}\sum_{\ell=0}^{L-1}(2\ell+1)|h_\ell(is\|c_{\alpha\beta}\|)| \\
&\leq \frac{1}{4\pi}|h_{L-1}(is\|c_{\alpha\beta}\|)|\sum_{\ell=0}^{L-1}(2\ell+1) \\
&\leq \frac{L^2}{4\pi}|h_{L-1}(is\|c_{\alpha\beta}\|)|.
\end{aligned} \tag{2.80}$$

The following lemma shows that the error $\mathcal{E}_1$ is not related to multipole-to-multipole (local-to-local) translations and equals the error of the one-level fast multipole method. Similar statements (though formulated slightly differently) have been proved in [130, 152].

**Lemma 2.2.18.** *Let $\epsilon > 0$ be fixed. Let $L$ in (2.73) be s.t.*

$$\sum_{m=L}^{\infty}(2m+1)|j_m(is\|c_{\alpha\beta}+x-y\|)h_m(is\|c_{\alpha\beta}\|)| \leq \left(1+L^2\right)^{-1}\epsilon,$$

*and $M \geq L$. Then the following bound holds for $\mathcal{E}_1 = \mathcal{E}_1(M,L)$ defined by (2.75):*

$$|\mathcal{E}_1| < \epsilon.$$

Before proving this lemma we would like to remark that such $L$ exists, since the series $\sum_{m=0}^{\infty}(2m+1)|j_m(is\|c_{\alpha\beta}+x-y\|)h_m(is\|c_{\alpha\beta}\|)|$ converges geometrically, see Section 2.2.3.2.

*Proof of Lemma 2.2.18.* Let

$$H := Q_M^{\hat{s}}\left[\mathrm{e}^{s(y-y_\alpha,\hat{s})}M_{\alpha,\beta}(\hat{s})\mathrm{e}^{s(x_\beta-x,\hat{s})}\right].$$

Setting $v := y_\alpha - y - x_\beta + x$, we obtain:

$$\begin{aligned}
H &:= Q_M^{\hat{s}}\left[\mathrm{e}^{-s(v,\hat{s})}M_{\alpha,\beta}(\hat{s})\right] \\
&\overset{(2.79),(2.73)}{=} \frac{1}{4\pi}\sum_{k=0}^{\infty}(2k+1)i^k j_k(is\|v\|)\sum_{\ell=0}^{L-1}(2\ell+1)(-i)^\ell h_\ell(is\|c_{\alpha\beta}\|)Q_M^{\hat{s}}\left[P_\ell\left(\hat{c}_{\alpha\beta}\cdot\hat{s}\right)P_k(\hat{v}\cdot\hat{s})\right].
\end{aligned}$$

We split

$$H = \frac{1}{4\pi} \sum_{k=0}^{2M-L} (2k+1) i^k j_k(is\|v\|) \sum_{\ell=0}^{L-1} (2\ell+1)(-i)^\ell h_\ell(is\|c_{\alpha\beta}\|) Q_M^{\hat{s}} \left[ P_\ell \left( \hat{c}_{\alpha\beta} \cdot \hat{s} \right) P_k(\hat{v} \cdot \hat{s}) \right]$$
$$+ \frac{1}{4\pi} \sum_{k=2M-L+1}^{+\infty} (2k+1) i^k j_k(is\|v\|) \sum_{\ell=0}^{L-1} (2\ell+1)(-i)^\ell h_\ell(is\|c_{\alpha\beta}\|) Q_M^{\hat{s}} \left[ P_\ell \left( \hat{c}_{\alpha\beta} \cdot \hat{s} \right) P_k(\hat{v} \cdot \hat{s}) \right].$$

Applying Lemma 2.2.8 to the above and using $M \geq L$, we obtain:

$$\begin{aligned} H &= \sum_{k=0}^{L-1} (2k+1) j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|) P_k \left( \hat{c}_{\alpha\beta} \cdot \hat{v} \right) \\ &\quad + \sum_{k=2M-L+1}^{+\infty} (2k+1) i^k j_k(is\|v\|) Q_M^{\hat{s}} \left[ M_{\alpha,\beta} \left( \hat{s} \right) P_k \left( \hat{s} \cdot \hat{v} \right) \right] \\ &\overset{(2.24)}{=} h_0(is\|x-y\|) - \sum_{k=L}^{+\infty} (2k+1) j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|) P_k \left( \hat{c}_{\alpha\beta} \cdot \hat{v} \right) \\ &\quad + \sum_{k=2M-L+1}^{+\infty} (2k+1) i^k j_k(is\|v\|) Q_M^{\hat{s}} \left[ M_{\alpha,\beta} \left( \hat{s} \right) P_k \left( \hat{s} \cdot \hat{v} \right) \right]. \end{aligned}$$

Inequalities (2.77,2.78,2.80) let us bound

$$|\mathcal{E}_1| \leq \sum_{k=L}^{+\infty} (2k+1) \left| j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|) \right| + L^2 \sum_{k=L}^{\infty} (2k+1) \left| j_k(is\|v\|) h_{L-1}(is\|c_{\alpha\beta}\|) \right|.$$

From the monotonicity of spherical Hankel functions (2.55), it follows:

$$\begin{aligned} |\mathcal{E}_1| &\leq \sum_{k=L}^{+\infty} (2k+1) \left| j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|) \right| + L^2 \sum_{k=L}^{\infty} (2k+1) |j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|)| \\ &\leq \left(1 + L^2\right) \sum_{k=L}^{\infty} (2k+1) |j_k(is\|v\|) h_k(is\|c_{\alpha\beta}\|)| \leq \epsilon. \end{aligned}$$

□

The errors $\mathcal{E}_j,\ j = 2, \ldots, 6$, occur due to the multipole-to-multipole and local-to-local transforms. To show how these errors can be controlled, we will need the following auxiliary quantities. Let us set

$$R_K(d) = \sum_{m=K}^{+\infty} (2m+1) |j_m(d)|.$$

Clearly $|r_K(x - x_{\beta'}, \hat{s})| < R_K(\|x - x_{\beta'}\|)$ for arbitrary $\hat{s} \in \mathbb{S}_2$. Moreover, since the series (2.79) converges supergeometrically (Remark 2.2.2), $R_K(d)$ decays rapidly as $K \to +\infty$.

Given $d_\alpha, d_\beta, d_{\alpha'}, d_{\beta'}$ being the diameters of bounding boxes of clusters $\tau_\alpha, \tau_\beta, \tau_{\alpha'}, \tau_{\beta'}$, let us introduce auxiliary quantities:

$$\begin{aligned} r_p &= \frac{1}{2}\max(d_\alpha, d_\beta), \\ r_c &= \frac{1}{2}\max(d_{\alpha'}, d_{\beta'}), \\ r_d &= \max\left(\|y_\alpha - y_{\alpha'}\|,\ \|x_\beta - x_{\beta'}\|\right). \end{aligned}$$

The following simple lemma demonstrates how the errors $\mathcal{E}_j$, $j = 2, \dots, 6$, can be made small.

**Lemma 2.2.19.** *Given $\epsilon > 0$, $L$ as in (2.73), let $N$, $M$ be chosen so that $N_* = \min(N, M)$ satisfies:*

$$E^{(1)} = L^2 N_*^2 \mathrm{e}^{\operatorname{Re} s(r_p + r_d)} \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{N_*}(r_c) \le \epsilon, \tag{2.81}$$

$$E^{(2)} = L^2 N_*^4 \mathrm{e}^{2\operatorname{Re} s r_d} \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{N_*}^2(r_c) \le \epsilon. \tag{2.82}$$

*Then the following bound holds for $\mathcal{E}_j = \mathcal{E}_j(L, M, N)$, $j = 2, \dots, 6$, defined by (2.76):*

$$|\mathcal{E}_j| < \epsilon, \qquad j = 2, \dots, 6.$$

*Proof.* We bound each of the errors $\mathcal{E}_j$, $j = 2, \dots, 6$:

$$\begin{aligned} |\mathcal{E}_2| &= \left| Q_M^{\hat{s}} \left[ \mathrm{e}^{s(y - y_\alpha + x_\beta - x_{\beta'}, \hat{s})} M_{\alpha,\beta}(\hat{s}) r_{N_*}(x - x_{\beta'}, \hat{s}) \right] \right| \\ &\overset{(2.77,2.80)}{\le} \mathrm{e}^{\operatorname{Re} s(r_p + r_d)} L^2 \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{N_*}(r_c) \le \frac{\epsilon}{N_*^2}, \end{aligned}$$

where the last bound follows from (2.81).

$$\begin{aligned} |\mathcal{E}_3| &= \left| Q_M^{\hat{s}} \left[ \mathrm{e}^{s(y - y_\alpha + x_\beta - x_{\beta'}, \hat{s})} M_{\alpha,\beta}(\hat{s}) \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(x - x_{\beta'}, \hat{q}) P_k(\hat{q} \cdot \hat{s}) \right] \right] \right| \\ &\overset{(2.77,2.78,2.80)}{\le} L^2 N_*^2 \mathrm{e}^{\operatorname{Re} s(r_p + r_d)} \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{2N-N_*+1}(r_c) \\ &\le L^2 N_*^2 \mathrm{e}^{\operatorname{Re} s(r_p + r_d)} \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{N_*}(r_c), \end{aligned}$$

where we used $N \ge N_*$. The bound (2.81) gives $\mathcal{E}_3 < \epsilon$. Similarly, we bound

$$\begin{aligned} |\mathcal{E}_4| &= \left| Q_M^{\hat{s}} \left[ \mathrm{e}^{s(y_{\alpha'} - y_\alpha + x_\beta - x_{\beta'}, \hat{s})} r_{N_*}(y_{\alpha'} - y, \hat{s}) M_{\alpha,\beta}(\hat{s}) r_{N_*}(x - x_{\beta'}, \hat{s}) \right] \right| \\ &\overset{(2.77,2.80)}{\le} L^2 \mathrm{e}^{2\operatorname{Re} s r_d} R_{N_*}^2(r_c) |h_{L-1}(is\|c_{\alpha\beta}\|)| \le \frac{\epsilon}{N_*^4}, \end{aligned}$$

where the last bound was obtained using (2.82).

$$\begin{aligned} |\mathcal{E}_5| &= \left| Q_M^{\hat{s}} \left[ \mathrm{e}^{s(y_{\alpha'} - y_\alpha + x_\beta - x_{\beta'}, \hat{s})} r_{N_*}(x - x_{\beta'}, \hat{s}) M_{\alpha,\beta}(\hat{s}) \right.\right. \\ &\qquad \left.\left. \times \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(y_{\alpha'} - y, \hat{q}) P_k(\hat{q} \cdot \hat{s}) \right] \right] \right| \\ &\overset{(2.77,2.80)}{\le} \mathrm{e}^{2\operatorname{Re} s r_d} N_*^2 R_{N_*}(r_c) L^2 |h_{L-1}(is\|c_{\alpha\beta}\|)| R_{2N-N_*+1}(r_c) \\ &\le L^2 N_*^2 R_{N_*}^2(r_c) |h_{L-1}(is\|c_{\alpha\beta}\|)| \le \frac{\epsilon}{N_*^2}, \end{aligned}$$

where $N_* \le N$ and (2.82) were used. Finally, the error

$$\begin{aligned}|\mathcal{E}_6| = \Big| Q_M^{\hat{s}} \Big[ & \mathrm{e}^{s(y_{\alpha'} - y_\alpha + x_\beta - x_{\beta'}, \hat{s})} M_{\alpha,\beta}(\hat{s}) \\ & \times \sum_{m=0}^{N_*-1} \frac{2m+1}{4\pi} Q_N^{\hat{r}} \left[ r_{2N-N_*+1}(x - x_{\beta'}, \hat{r}) P_m(\hat{r} \cdot \hat{s}) \right] \\ & \times \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(y_{\alpha'} - y, \hat{q}) P_k(\hat{q} \cdot \hat{s}) \right] \Big] \Big| \\ & \overset{(2.77),(2.80)}{\le} \mathrm{e}^{2 \operatorname{Re} s r_d} N_*^4 L^2 |h_{L-1}(is\|c_{\alpha\beta}\|) R_{2N-N_*+1}^2(r_c)| \le \epsilon.\end{aligned}$$

□

**Remark 2.2.20.** *The bounds provided in Lemma 2.2.19 are not fully optimal, due to the use of (2.77, 2.80). We would like to show that values $E^{(1)}, E^{(2)}$ decay with $\operatorname{Re} s > 0$, independently of the choice $N$, $M$ (however, with $L$ satisfying Lemma 2.2.18).*

*Let the admissibility condition be fixed (assuming that the diameters $d_\alpha$, $d_\beta$ of the bounding boxes of clusters $\tau_\alpha$ and $\tau_\beta$ are equal):*

$$\|c_{\alpha\beta}\| \ge \frac{3}{4}(d_\alpha + d_\beta) = 3r_p. \tag{2.83}$$

*We will need two ingredients.*

1. *Exponentially fast decay of spherical Hankel functions as $\operatorname{Re} s \to +\infty$. As $|s| \operatorname{diam} \tau_\alpha \gg 1$, see Remark 2.2.16,*

$$L \ll |s| \|c_{\alpha\beta}\|.$$

*Hence, the spherical Hankel function $h_{L-1}(is\|c_{\alpha,\beta}\|)$ is in the asymptotic regime (2.52), namely*

$$|h_{L-1}(is\|y_\alpha - x_\beta\|)| \sim (1 + \gamma(|s|\|c_{\alpha\beta}\|)) \frac{\mathrm{e}^{-\operatorname{Re} s \|c_{\alpha\beta}\|}}{|s|\|c_{\alpha\beta}\|}, \tag{2.84}$$

*where $\gamma(|s|\|c_{\alpha\beta}\|, L) = O\left(\frac{L^2}{|s|\|c_{\alpha\beta}\|} + \frac{L^4}{|s|^2\|c_{\alpha\beta}\|^2}\right)$.*

2. *Exponentially fast (but with a relatively small rate) growth of $|R_{N_*}(r_c)|$. Let us show that*

$$R_{N_*}(r_c) \le C_1 |s|^2 r_c^2 \mathrm{e}^{\operatorname{Re} s r_c}, \qquad C_1 > 0. \tag{2.85}$$

*The expression (2.45) allows to derive a trivial bound:*

$$|j_n(isr_c)| = \left| \frac{1}{2}(-i)^n \int_{-1}^{1} \mathrm{e}^{-sr_c} P_n(t) dt \right| \overset{(2.78)}{\le} \frac{1}{2} \mathrm{e}^{\operatorname{Re} s r_c}. \tag{2.86}$$

*Recall that spherical Bessel functions satisfy the bound (2.50):*

$$|j_n(z)| \le \mathrm{e}^{\operatorname{Im} z} \frac{\mathrm{e}}{2|z|\sqrt{\pi}} \left( \frac{|z|\mathrm{e}}{2(n+1)} \right)^{n+1}.$$

*Let $N' \in \mathbb{N}$, $N' \geq |z|\mathrm{e}$. Then for all $n \geq N'$, it holds that*

$$|j_n(z)| \leq \mathrm{e}^{\operatorname{Im} z} \frac{\mathrm{e}}{2|z|\sqrt{\pi}} \left( \frac{|z|\mathrm{e}}{2(n+1)} \right)^{n+1} \leq \mathrm{e}^{\operatorname{Im} z} \frac{\mathrm{e}}{2|z|\sqrt{\pi}} \left( \frac{1}{2} \right)^{n+1}. \tag{2.87}$$

*Then*

$$R_{N_*}(r_c) \leq R_0(r_c) = \sum_{n=0}^{\infty} (2n+1)|j_n(isr_c)| = \sum_{n=0}^{N'-1} (2n+1)|j_n(isr_c)| + \sum_{n=N'}^{+\infty} (2n+1)|j_n(isr_c)|.$$

*The first term can be bounded using (2.86) by*

$$\frac{1}{2} N'^2 \mathrm{e}^{\operatorname{Re} sr_c} < C|s|^2 r_c^2 \mathrm{e}^{\operatorname{Re} sr_c},$$

*for a constant $C$ that does not depend on $s$ and $r_c$. The second term can be bounded with the help of (2.87)*

$$\sum_{n=N'}^{+\infty} (2n+1)|j_n(isr_c)| \leq C' \mathrm{e}^{\operatorname{Re} sr_c},$$

*for some $C' > 0$. This lets obtaining (2.85).*

*Now let us consider the errors $E^{(1)}$ and $E^{(2)}$. We insert (2.84,2.85) into the expression for $E^{(1)}$ to obtain*

$$E^{(1)} \sim C_1 L^2 N_*^2 \left(1 + \gamma\left(|s|\|c_{\alpha\beta}\|, L\right)\right) |s|^2 r_c^2 \frac{\mathrm{e}^{\operatorname{Re} s(r_p + r_d + r_c - \|c_{\alpha\beta}\|)}}{|s|\|c_{\alpha\beta}\|}.$$

*Clearly, $r_d < r_p$, $r_c < r_p$, hence, using (2.83),*

$$r_p + r_d + r_c - \|c_{\alpha\beta}\| < 0,$$

*which shows the exponential decay of $E^{(1)}$ with $\operatorname{Re} s$. Similarly, we can substitute $R_{N_*}(r_c)$ and $h_{L-1}(is\|c_{\alpha\beta}\|)$ in (2.82) by their estimates (2.85, 2.84) to obtain:*

$$E^{(2)} = L^2 N_*^4 \mathrm{e}^{2\operatorname{Re} sr_d} \left|h_{L-1}(is\|c_{\alpha\beta}\|)\right| R_{N_*}^2(r_c) \sim L^2 N_*^4 C_1^2 |s|^4 r_c^4 \left(1 + \gamma(|s|\|c_{\alpha\beta}\|)\right) \frac{\mathrm{e}^{\operatorname{Re} s(2r_d + 2r_c - \|c_{\alpha\beta}\|)}}{|s|\|c_{\alpha\beta}\|}.$$

*We consider two cases, the first one when an octree based partitioning of the domain is used, and another one when the binary partitioning is employed (see [48]). We assume that bounding boxes of clusters $\tau_\alpha, \tau_\beta$ are cuboids with sides $a, b, c$. Then $r_p = \frac{1}{2}\sqrt{a^2 + b^2 + c^2}$.*

1. *if an octree partitioning is used, $r_c = \frac{1}{2} r_p = r_d$, hence, using the admissibility condition (2.83), we obtain:*

$$E^{(2)} \sim L^2 N_*^4 C_1^2 |s|^3 r_c^4 \left(1 + \gamma(|s|\|c_{\alpha\beta}\|)\right) \frac{\mathrm{e}^{-\operatorname{Re} sr_p}}{\|c_{\alpha\beta}\|}.$$

2. *if the binary tree partitioning is employed, the parent cluster is split into two children clusters. We assume w.l.o.g.* $r_d = \frac{a}{4}$ *and* $r_c = \frac{1}{2}\sqrt{\frac{a^2}{4} + b^2 + c^2}$. *Hence, using (2.83), we obtain:*

$$E^{(2)} \sim L^2 N_*^4 C_1^2 |s|^3 r_c^4 \left(1 + \gamma(|s|\|c_{\alpha\beta}\|)\right) \frac{\mathrm{e}^{\operatorname{Re} s(\frac{a}{2} + \sqrt{\frac{a^2}{4}+b^2+c^2} - \frac{3}{2}\sqrt{a^2+b^2+c^2})}}{\|c_{\alpha\beta}\|},$$

*which decays exponentially with* $\operatorname{Re} s \to +\infty$.

#### 2.2.3.4 Numerical Stability and Control of Roundoff Errors

There are two sources of round-off errors when the high-frequency fast multipole method is applied to Helmholtz problems with complex wavenumbers. The first one is connected to exponential growth of spherical Hankel functions $h_n(z)$ when $n \gg |z|$ and is also inherent to the HF FMM applied to the problems with purely real wavenumber, see e.g. [152]. The second one is intrinsic to the HF FMM applied to the Helmholtz equation with large decay and was studied in [188]. Importantly, these errors occur in different situations: the first one appears only when small clusters are considered, while the second one is likely to appear when applying the high-frequency fast multipole method to distant (and hence, due to the definition of the admissibility condition, large) clusters. In the following section we study the effect of these errors on the high-frequency FMM.

The low-frequency breakdown of the fast multipole method occurs when performing the multipole-to-local transform between small admissible clusters. One of the ways to control this error was suggested in [57]: there numerically determined bounds on size of clusters were used (e.g. to achieve an accuracy $10^{-3}$, the authors recommend to use HF FMM only for clusters whose size exceeds $\frac{1}{4}$ of a wavelength). Indeed, such strategy has to be adapted to different admissibility conditions, as well as to the presence of decay, which can (though not always) decrease the magnitude of rounding errors.

Our strategy of the roundoff error control is based on the following observation. In the simplest case of the one-level fast multipole method $h_0(is\|x-y\|)$ is approximated by the scalar product

$$\begin{aligned} h_0(is\|x-y\|) &\approx \sum_{\ell=1}^{2M^2} w_\ell \mathrm{e}^{-s(x-x_\beta,\hat{s}_\ell)} M_{\alpha,\beta}(\hat{s}_\ell) \mathrm{e}^{s(y-y_\alpha,\hat{s}_\ell)} = A^T B, \qquad (2.88)\\ A &= \left[w_1 \mathrm{e}^{-s(x-x_\beta,\hat{s}_1)}, \ldots, w_{2M^2} \mathrm{e}^{-s(x-x_\beta,\hat{s}_{2M^2})}\right]^T,\\ B &= \left[M_{\alpha,\beta}(\hat{s}_1)\mathrm{e}^{s(y-y_\alpha,\hat{s}_1)} \ldots, M_{\alpha,\beta}(\hat{s}_{2M^2})\mathrm{e}^{s(y-y_\alpha,\hat{s}_{2M^2})}\right]^T. \end{aligned}$$

The following lemma from [125, Section 3.1] bounds the error of the evaluation of the scalar product in the finite precision arithmetic.

**Lemma 2.2.21.** *Given* $x, y \in \mathbb{R}^n$, *let* $s_n = x^T y$ *and* $\hat{s}_n = fl(x^T y)$ *be the inner product* $x^T y$ *computed with no overflow or underflow in the finite precision arithmetic compliant with the standard model, i.e. for all floating point numbers* $a, b$

$$fl(a \circ b) = a \circ b(1+\delta),\ |\delta| < \epsilon_m; \circ = +, -, *, \backslash, \qquad (2.89)$$

*where $\epsilon_m$ is a machine accuracy. Then,*

$$|\hat{s}_n - s_n| \le \gamma_n \sum_{i=1}^{n} |x_i y_i|,$$
$$\gamma_n = \frac{n\epsilon_m}{1 - n\epsilon_m}. \tag{2.90}$$

Questions of the accuracy of the complex floating point arithmetic are considered in [125, Lemma 3.5]. In a nutshell, it is possible to implement the basic arithmetic operations so that

$$fl(a \circ b) = a \circ b(1+\delta),\ |\delta| < \epsilon_m, ; \circ = +, -,$$
$$fl(ab) = ab(1+\delta),\ |\delta| < \sqrt{2}\gamma_2,$$
$$fl\left(\frac{a}{b}\right) = \frac{a}{b}(1+\delta),\ |\delta| < \sqrt{2}\gamma_4,$$

where $\gamma_n$ is given by (2.90).

Hence, for complex $s$, the roundoff error of the evaluation (2.88) can be bounded by

$$\epsilon_{\text{roundoff}} \le \gamma_M \sum_{\ell=1}^{2M^2} |A_\ell||B_\ell|$$
$$\le 2CM^2\gamma_M \mathrm{e}^{\operatorname{Re} sd} \max_{\hat{s}\in\mathbb{S}^2} |M_{\alpha,\beta}(\hat{s})|, \tag{2.91}$$

where $C$ is a constant coming from the use of the complex arithmetic and $d = \max(\|x - x_\beta\|, \|y - y_\alpha\|)$. The low-frequency (occurring when $|s|d$ is smaller than a fixed value) roundoff error can be controlled by checking if

$$(L-1)\,|h_{L-1}(is\|c_\alpha - c_\beta\|)|\,\epsilon_m \tag{2.92}$$

is smaller than given $\epsilon' > 0$.

In the case of high decay, the cancellation errors can occur when performing multipole-to-multipole (local-to-local) transforms. First, recall that the cluster basis of the high-frequency fast multipole method is the matrix of the form

$$V_{kj} = \int_{\tau_\alpha} \mathrm{e}^{-\operatorname{Re} s_k(y - y_\alpha, \hat{s_k})} \phi_j(y) d\Gamma_y, \hat{s_k} \in \mathbb{S}^2.$$

This implies that the entries of the cluster basis for large $\operatorname{Re} s$ can vary greatly in magnitude. When performing multipole-to-multipole transform one needs to do the spherical harmonic transform that includes many additions and subtractions of these numbers. This can potentially lead to cancellation errors. Hence, for clusters of diameter $d$ we check if

$$\mathrm{e}^{\operatorname{Re} s\frac{d}{2}}\epsilon_m > \epsilon, \tag{2.93}$$

where $\epsilon$ is a desired precision. In practice, however, such high-decay roundoff errors are not likely to cause problems, at least for moderate accuracies, due to reasons explained next. Briefly, if $\operatorname{Re} s\frac{d}{2}$ is big enough for the term $\mathrm{e}^{\operatorname{Re} s\frac{d}{2}}\epsilon_m$ to cause difficulties, then all the admissible blocks at the level where the clusters with diameter $d$ are located can be

approximated by zero matrices (as well as the blocks corresponding to admissible block-clusters on the higher levels of the block-cluster tree), see (2.9). Therefore, they will not contribute to the final result, and there is no need for performing multipole-to-multipole (local-to-local) translations at these levels. Next we support this argument with more detail.

The round-off error during the multipole-to-multipole (or local-to-local) translation may exceed the desired precision $\epsilon$ only when

$$\operatorname{Re} s\frac{d}{2} > \log\frac{\epsilon}{\epsilon_m}. \tag{2.94}$$

The multipole-to-multipole (local-to-local) translation has to be performed only in the following cases.

1. When the cluster $\tau$ with the bounding box of the diameter $d$ has at least one admissible neighbor, i.e. there exists an admissible block-cluster $(\tau, \sigma)$;

2. When there exists a cluster $\tau'$ s.t. $\operatorname{level}(\tau') < \operatorname{level}(\tau)$, $\tau$ is one of the descendants of $\tau'$ and $\tau'$ has at least one admissible neighbor.

Otherwise there is no need to translate an expansion, see Remark 2.1.15.

Let us consider the first case, i.e. let the cluster $\tau$ have at least one admissible neighbor $\sigma$, and show that the corresponding block can be approximated by a zero matrix. The same arguments will apply in the second case as well. The diameter of the bounding box of the cluster $\sigma$ we denote by $d$ and assume that it equals the diameter of the bounding box of the cluster $\tau$.

From the definition of admissible clusters (2.1.7) it follows that the distance between $\tau$, $\sigma$ can be bounded from below in terms of the sizes of these clusters. Let us denote the centers of the bounding box of the clusters $\tau, \sigma$ by $c_\tau$, $c_\sigma$. Then:

$$\|c_\sigma - c_\tau\| \geq \eta d.$$

From the triangle inequality and the above equation it follows that

$$\begin{aligned}\operatorname{dist}(\tau,\sigma) &\geq \|c_\sigma - c_\tau\| - 2\frac{d}{2},\\ \operatorname{dist}(\tau,\sigma) &\geq (\eta-1)d.\end{aligned}$$

Using (2.94) we obtain a lower bound on the distance between the clusters $\tau$, $\sigma$:

$$\operatorname{dist}(\tau,\sigma) \geq \frac{2(\eta-1)}{\operatorname{Re} s}\log\frac{\epsilon}{\epsilon_m}. \tag{2.95}$$

Now let us show that the distance between clusters $\tau$, $\sigma$ is large enough to efficiently approximate the corresponding block by a zero matrix, see (2.9). This will imply that at this level all admissible blocks can be approximated by zero matrices, and therefore there is no need in performing multipole-to-multipole (local-to-local) translations. The zero-rank approximation applies, see (2.9), when

$$\left|\frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|}\right| < \epsilon,$$

for all $x \in \Omega_\tau$, $y \in \Omega_\sigma$. Equivalently,

$$\frac{\mathrm{e}^{-\mathrm{dist}(\tau,\sigma)\,\mathrm{Re}\,s}}{4\pi\mathrm{dist}(\tau,\sigma)} < \epsilon.$$

For this condition to hold true, it is sufficient that, see (2.95),

$$\frac{\exp\left(-2(\eta-1)\log\frac{\epsilon}{\epsilon_m}\right)}{4\pi\mathrm{dist}(\tau,\sigma)} \leq \epsilon,$$

or

$$\frac{1}{4\pi\mathrm{dist}(\tau,\sigma)}\left(\frac{\epsilon_m}{\epsilon}\right)^{2(\eta-1)} < \epsilon.$$

Since also $\mathrm{dist}(\tau,\sigma) \geq (\eta-1)d$, the block-cluster is to be approximated by a zero matrix when

$$d > \frac{\epsilon_m^{2\eta-2}}{\epsilon^{2\eta-1}4\pi(\eta-1)}.$$

For moderate accuracies, $\epsilon \leq 10^{-6}$, and $\eta = 1.5$,

$$d \geq 4 \cdot 10^{-5}.$$

This is a reasonable condition that allows to deal with problems having about $O(\frac{1}{d^2}) = 10^8 - 10^9$ spatial unknowns. For higher accuracies or smaller values of $d$, there exist two (in practice similar) strategies to avoid problems associated with the cancellation:

- increase $\eta$ (e.g. $\eta = 2$ ensures that no cancellation errors occur for $\epsilon \geq 10^{-8}$ and $d \geq 4 \cdot 10^{-9}$); this may result in the increase of the near-field;

- split each of the admissible block-clusters located at the level where the high-decay breakdown occurs into several smaller admissible block-clusters (that will be located at the next level of the block-cluster tree) and construct the multipole-to-local approximation for them separately. This allows to avoid performing the multipole-to-multipole and local-to-local translation operators.

We used the first strategy, due to the ease of the implementation.

#### 2.2.3.5 Choice of the Parameters of the Fast Multipole Method: Summary

The complexity of the high-frequency fast multipole method depends on the choice of the cluster tree and lengths of multipole and local expansions.

The cluster tree has to be constructed so that the diameter of the bounding boxes of leaf clusters is $O\left(\frac{1}{|s|}\right)$. Typically, an octree is used, see [57]. In this work we employ the binary cluster tree, similarly to [48, 89, 90]. In [48] it was suggested to make a cluster a leaf if the number of degrees of freedom inside this cluster does not exceed some fixed $n_0$. We use such strategy as well (with $n_0 = 20$), however, with an additional correction: leaf clusters can be located only at the levels $\ell \geq \ell_0$, where $\ell_0$ is given a priori and increases

logarithmically with $M$. The reason for the latter requirement is that in some cases it may happen that some of clusters with a few boundary elements occur at very coarse levels of the cluster tree. The size of bounding boxes of clusters located at one level of the cluster tree is the same (see Remark 2.2.11), and hence if such clusters were leaves, they would have large inadmissible neighbors. This strategy is in agreement with [57].

To choose the length of multipole and local expansions, we suggest the following scheme. Let $r_\ell$ be the half-diameter of the bounding box of a cluster at the level $\ell$.

First, let us consider the multipole-to-local operator. Let an admissible block-cluster $b = (\tau_\alpha, \tau_\beta)$ be located at the level $\ell > 0$ of the block-cluster tree. Let the bounding box of the cluster $\tau_\alpha$ be centered at $y_\alpha$ and the bounding box of the cluster $\tau_\beta$ be centered at $x_\beta$. Additionally, $c_{\alpha\beta} = y_\alpha - x_\beta$. The choice of the length $n_b$ of the truncated expansion for the corresponding multipole-to-local translation operator, see also (2.40), can be determined by checking (c.f. Lemma 2.2.18)

$$\left| h_0(is\|\|c_{\alpha\beta}\| - 2r_\ell\|) - \sum_{m=0}^{n_b-1} (2m+1) j_m(2isr_\ell) h_m(is\|c_{\alpha\beta}\|) \right| \leq \epsilon. \tag{2.96}$$

If the low-frequency breakdown happens, see (2.92), we set formally $n_b = 0$. Such cluster is to be approximated with the help of $\mathcal{H}$-matrix techniques.

The value $n_b$ depends on $r_\ell$ and $\|c_{\alpha\beta}\|$ only. Since there is a fixed number of different $\|c_{\alpha\beta}\|$ per level, the check (2.96) can be performed once for each different $\|c_\alpha\beta\|$. The complexity of this operation is obviously sublinear.

Now we have all ingredients to determine the length of multipole expansions. Denoting the set of admissible clusters located at the level $\ell$ by $\mathcal{L}_+^\ell$, let us introduce two auxiliary quantities:

$$n_\ell = \max_{b \in \mathcal{L}_+^\ell} n_b, \tag{2.97}$$

$$H_\ell = \sup_{b \in \mathcal{L}_+^\ell} \left| h_{n_b-1}(is\|c_{\alpha\beta}\|) \right|. \tag{2.98}$$

In principle, for most practical situations setting the length of the multipole expansion at the level $\ell$ to $2n_\ell^2$ is sufficient to achieve the desired accuracy. Such strategy was suggested in [57, 70] for the case of the real wavenumber. It is, however, not obvious (though feasible, due to the rapid decay of spherical Bessel functions) whether setting $N = n_\ell$ guarantees that conditions of Lemma 2.2.19 hold for general complex $s \in \mathbb{C}$.

The analysis in the previous section shows that to control the multipole-to-multipole (local-to-local) errors, the length of the expansion has to satisfy conditions of Lemmas 2.2.18 and 2.2.19. Through the multipole-to-multipole (local-to-local) translation the error propagates to the coarser (finer) cluster tree levels. Though we do not present the analysis of the error after several multipole-to-multipole (local-to-local) translations, we suggest performing the following checks.

Given a cluster $\tau_{\alpha,\ell}$ located at the level $\ell$, let $\tau_{\alpha,\ell-1}$, $\tau_{\alpha,\ell-2}$, $\ldots$, $\tau_{\alpha,k}$ be s.t.

$$\tau_{\alpha,j} \in \mathrm{sons}(\tau_{\alpha,j-1}), \qquad k < j \leq \ell. \tag{2.99}$$

Here $k$ is the smallest level at which there is at least one admissible block cluster. Let $r_d^{j,\ell,\alpha}$ be the distance between the centers of the bounding boxes of clusters $\tau_{\alpha,j}$ and $\tau_{\alpha,\ell}$. Given

levels $\ell,\ j$, the maximum of $r_d^{j,\ell,\alpha}$ over the pairs of clusters $\tau_{\alpha,\ell}, \tau_{\alpha,j}$ subject to (2.99) we denote by

$$\tilde{r}_d^{j,\ell} = \max_{(\tau_{\alpha,\ell},\tau_{\alpha,j})} r_d^{j,\ell,\alpha}.$$

This quantity can be computed in time not worse than linear (or even $O(\log^2 M)$), due to the uniform partition of the domain, and hence it does not affect the complexity of the algorithm. Alternatively, it also can be estimated by

$$\tilde{r}_d^{j,\ell} \leq r_j.$$

In practice, instead of checking (2.81, 2.82), we look for $N_\ell$, such that for all $k \leq j < \ell$ (i.e. for all highest levels with respect to the current level),

$$\begin{aligned} N_\ell n_j \left| j_{N_\ell}(isr_\ell) \right| H_j \mathrm{e}^{\mathrm{Re}\, s(r_j + \tilde{r}_d^{j,\ell})} &< \epsilon, \\ N_\ell n_j \left| j_{N_\ell}(isr_\ell) \right|^2 H_j \mathrm{e}^{2\,\mathrm{Re}\, s\tilde{r}_d^{j,\ell}} &< \epsilon. \end{aligned} \tag{2.100}$$

Lemma 2.2.19 shows that the length of the expansion of the level $\ell$ has to be chosen not smaller than $2N_j^2,\ j > \ell$. Let $\mathcal{N}_\ell = \max(\max_{q \geq \ell} N_q,\ n_\ell)$ (c.f. Lemma 2.2.19, Lemma 2.2.18). Then we choose the length of the expansion at the level $\ell$ to be equal to $2\mathcal{N}_\ell^2$.

Although the conditions of Lemma 2.2.19 may seem quite restrictive, in practice $\max_{q \geq \ell} N_q$ is rarely larger than $n_\ell$.

All these checks are of the complexity not larger than linear, and hence do not affect the asymptotic complexity of the fast multipole algorithm.

## 2.3 Numerical Comparison of $\mathcal{H}$-Matrix Techniques and the High-Frequency Fast Multipole Method for the Helmholtz Equation with Decay

In this section we compare the performance of $\mathcal{H}$-matrices and the fast multipole method for the Helmholtz equation with decay. We present results of numerical experiments for $\mathcal{H}$- and $\mathcal{H}^2$-matrices (constructed with the help of the expansions coming from the fast multipole method) approximating the Helmholtz boundary single-layer operator on the unit sphere, both for complex and real wavenumbers. In the end of this section we propose a heuristic that allows to make the choice whether an $\mathcal{H}-$ or an $\mathcal{H}^2-$matrix should be constructed in a specific situation.

All experiments in this section were done on the cluster of the Max Planck Institute for Mathematics in the Sciences, on a single processor Intel Xeon X5650 with 2.67 GHz. For the computation we used $\mathcal{H}$LIBpro library, see [132]. Spherical Bessel and Hankel functions were computed with the help of the Amos library [9].

### 2.3.1 Real Wavenumber

In this section we present results of numerical experiments for the Helmholtz equation without decay. Our goal is to validate correctness of the fast multipole implementation as well as to compare efficiency of the FMM with $\mathcal{H}$-matrices for this simple case. Similar experiments have been already performed in [48].

The experiments are done with two accuracies: $\epsilon = 10^{-6}$ and $\epsilon = 10^{-9}$. The relative accuracy for $\mathcal{H}$-matrices is set to $10^{-5}$ and $10^{-8}$ in both of the experiments (such choice is based on the observation that setting the ACA+ accuracy to some value often guarantees an accuracy several magnitudes higher).

The results are presented in Tables 2.1, 2.2 and 2.3. Matrix-vector multiplication errors are measured as

$$\epsilon_r = \max_v \frac{\|Mv - Dv\|_2}{\|Dv\|_2},$$

based on results of 100 matrix-vector multiplications with random complex vectors whose entries lie in $[-1,\ 1]$. Here $M$ is an $\mathcal{H}-$matrix or an $\mathcal{H}^2-$matrix constructed with the help of the high-frequency fast multipole expansions. The matrix $M$ is assembled with a given accuracy $\epsilon$. The matrix $D$ is a dense matrix (or a highly accurate $\mathcal{H}$-matrix) . The matrix construction time for $\mathcal{H}^2$-matrices includes time needed to construct the $\mathcal{H}$-matrix part, transfer matrices, multipole-to-local operators as well as the leaf cluster basis. Let us also remark that the time for the construction of the leaf basis is the time to evaluate the integrals (2.31) using the precomputed quadrature points, weights and cluster centers (see the end of Section 2.2.2.1), and the precomputation itself is included into the matrix construction time as well.

| $M$ | $s$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $\epsilon_r(\mathcal{H})$ | $\epsilon_r(\mathcal{H}^2)$ |
|---|---|---|---|---|---|---|---|
| 2048 | -8i | 22.7 | 15 (1.5) | 0.02 | 0.6 | 6e-6 | 1.5e-6 |
| 4232 | -11.3i | 62.4 | 35 (3.5) | 0.05 | 1.2 | 3.4e-6 | 1.5e-6 |
| 8192 | -16i | 152 | 63 (7.9) | 0.11 | 2.3 | 4.3e-6 | 1.3e-6 |
| 16200 | -22.6i | 334 | 122 (13.6) | 0.28 | 4.5 | 4.9e-6 | 1.4e-6 |
| 32768 | -32i | 877 | 257 (35.3) | 0.7 | 8.6 | 6e-6 | 1.7e-6 |
| 65448 | -45.3i | 7756 | 514 (62) | 1.8 | 19.5 | 1.7e-5 | 2.4e-5 |
| 129970 | -64.0i | - | 1023 (139) | - | 42.2 | - | - |

*Table 2.1: Construction times $T_c$, matrix-vector multiplication times $T_{mv}$ and computed relative errors for the accuracy setup $\epsilon = 10^{-6}$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. The deterioration of the accuracy for large matrices is due to insufficient accuracy of quadrature.*

The results in Table 2.2 demonstrate that the high-frequency fast multipole method is of almost linear complexity, while the complexity of $\mathcal{H}$-matrices scales somewhat worse, though better than predicted theoretically. This is connected to the fact that low-rank approximations constructed using $\mathcal{H}$-matrix techniques are close to optimal and take into account the geometry of the problem.

Results of numerical experiments in Table 2.3 show that matrix-vector multiplication times of $\mathcal{H}^2$-matrices constructed with a high accuracy can be slightly smaller compared to that of $\mathcal{H}^2$-matrices constructed with a lower accuracy. This happens because the increase of the accuracy requires more matrix blocks to be approximated with the help of $\mathcal{H}$-matrix techniques (due to the low-frequency breakdown), and, as it can be seen from the numerical results, the matrix-vector multiplication times of $\mathcal{H}$-matrices are in practice much smaller compared to that of $\mathcal{H}^2$-matrices.

Our numerical experiments support the results presented in [48]: if many matrix-vector multiplications are needed, $\mathcal{H}$-matrices are advantageous over $\mathcal{H}^2$-matrices. However, in terms of the matrix construction times $\mathcal{H}^2$-matrices perform practically always better.

| $N_n$ | $s_n$ | $\log_2 \frac{T_c^n}{T_c^{n-1}}(\mathcal{H})$ | $\log_2 \frac{T_{mv}^n}{T_{mv}^{n-1}}(\mathcal{H})$ | $\log_2 \frac{T_c^n}{T_c^{n-1}}(\mathcal{H}^2)$ | $\log_2 \frac{T_{mv}^n}{T_{mv}^{n-1}}(\mathcal{H}^2)$ |
|---|---|---|---|---|---|
| 4232 | -11.3i | 1.5 | 1.3 | 1.2 (1.2) | 1 |
| 8192 | -16i | 1.3 | 1.14 | 0.9 (1.2) | 0.9 |
| 16200 | -22.6i | 1.1 | 1.3 | 1 (0.8) | 1 |
| 32768 | -32i | 1.4 | 1.3 | 1.1 (1.4) | 0.9 |
| 65448 | -45.3i | 3.1 | 1.4 | 1 (0.8) | 1.2 |
| 129970 | -64.0i | - | - | 1 (1.2) | 1.1 |

*Table 2.2: The rate of the times for the matrix assembly and the times for the matrix-vector multiplication for the current discretization and the twice coarser one for different techniques with the accuracy setup $\epsilon = 10^{-6}$.*

| $M$ | $s$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $\epsilon_r(\mathcal{H})$ | $\epsilon_r(\mathcal{H}^2)$ |
|---|---|---|---|---|---|---|---|
| 2048 | -8i | 77.6 | 82.6 (1.6) | 0.017 | 0.4 | 1e-8 | 3.1e-9 |
| 4232 | -11.3i | 215.1 | 190.4 (2.6) | 0.05 | 1 | 9.2e-9 | 3.7e-9 |
| 8192 | -16i | 561.6 | 384.9 (5.9) | 0.16 | 1.8 | 7e-9 | 4e-9 |
| 16200 | -22.6i | 1488.4 | 693 (11.5) | 0.4 | 5 | 6e-9 | 3.3e-9 |
| 32768 | -32i | 3704.8 | 1323.3 (22.2) | 1 | 11 | 7e-9 | 4e-9 |
| 65448 | -45.3i | - | 2735.4 (55.4) | - | 25.4 | - | - |
| 129970 | -64.0i | - | 5138 (97.7) | - | 54.5 | - | - |

*Table 2.3: Construction times $T_c$, matrix-vector multiplication times $T_{mv}$ and computed relative errors for the accuracy setup $\epsilon = 10^{-9}$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. For the last two experiments we did not construct an $\mathcal{H}$-matrix approximation: for the given accuracy setting, it appears to be too expensive.*

### 2.3.2 Complex Wavenumber

In this section we present results of numerical experiments for the Helmholtz equation with decay. First, we fix $M = 16200$ and $|s| = 22.6$ and study how the complexity of $\mathcal{H}$- and $\mathcal{H}^2$-matrix approximations changes with the argument of $s = |s|e^{i\alpha}$, $\alpha \in \left[-\frac{\pi}{2}, 0\right]$. We set the desired accuracy of $\mathcal{H}^2$-matrix approximation to $\epsilon = 10^{-6}$ and the relative accuracy of the $\mathcal{H}$-matrix approximation to $10^{-5}$. The results of this experiment are presented in Table 2.4.

Let us remark that in the Runge-Kutta convolution quadrature algorithm we construct the leaf cluster basis on the fly: we need only a few matrix-vector multiplications, hence there is no need to store it. This allows to reduce storage costs significantly. Let $T_c$ be the matrix construction time (including the time for assembling the leaf cluster basis), $T_{cb}$ be the leaf cluster basis construction time and $T_{mv}$ be the time for the matrix-vector multiplication. Then in our algorithm the time to construct the $\mathcal{H}^2$-matrix equals $T_c - T_{cb}$ and the time to perform matrix-vector multiplication is $T_{mv} + T_{cb}$.

Results in Table 2.4 show the following effects of the presence of decay. First, the construction time of matrix approximations increases significantly. This is due to the fact that the evaluation of the Helmholtz kernel with decay $\frac{e^{-s\|x-y\|}}{4\pi\|x-y\|}$, $s \in \mathbb{C}$, is more computationally expensive compared to the evaluation $\frac{e^{i\kappa\|x-y\|}}{4\pi\|x-y\|}$, $\kappa \in \mathbb{R}$. This remains valid for the high-frequency fast multipole method, since the evaluation of the integral kernel is required for the near-field. Similar arguments apply to the leaf cluster basis.

| $\phi$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $N_{mv}$ |
|---|---|---|---|---|---|
| $-\frac{\pi}{2}$ | 334 | 122 (13.6) | 0.28 | 4.5 | 12 |
| $-\frac{6\pi}{14}$ | 427 | 183 (29.7) | 0.17 | 4.3 | 9 |
| $-\frac{5\pi}{14}$ | 324 | 175 (27.4) | 0.09 | 4.3 | 7 |
| $-\frac{4\pi}{14}$ | 260 | 166 (25.5) | 0.07 | 4.1 | 5 |
| $-\frac{3\pi}{14}$ | 231 | 149 (26.9) | 0.06 | 4.6 | 4 |
| $-\frac{2\pi}{14}$ | 213 | 158 (26.5) | 0.06 | 3.8 | 3 |
| $-\frac{\pi}{14}$ | 209 | 156 (26.2) | 0.05 | 3.6 | 3 |
| 0 | 170 | 128 (21) | 0.05 | 3.5 | 3 |

*Table 2.4: Construction times $T_c$, matrix-vector multiplication times $T_{mv}$ and computed relative errors for the accuracy setup $\epsilon = 10^{-6}$, $M = 16200$, $|s| = 22.6$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. $N_{mv}$ stands for the number of the matrix-vector multiplications needed for the $\mathcal{H}$-matrix approximation to outperform the $\mathcal{H}^2$-approximation (where the time for the matrix-vector multiplication includes the construction of the leaf cluster basis). In all the experiments the relative error of the $\mathcal{H}$- and $\mathcal{H}^2$-approximations did not exceed $1.6 \cdot 10^{-6}$.*

We have to remark that this is not always the case: the presence of sufficiently large decay (c.f. Tables 2.1 and 2.5) can also reduce the time of the $\mathcal{H}$-matrix construction (due to the drastic decrease of ranks of $\mathcal{H}$-matrix blocks).

The results in 2.4, 2.5 and 2.1 show that the matrix-vector multiplication times of the HF FMM in the presence of decay are smaller than in the case of no-decay. This can be also explained by the reduction of the length of the multipole expansions. Similarly, the matrix-vector multiplication costs for $\mathcal{H}$-matrices are reduced compared to the no-decay case.

Next, we can see that if more than 4 matrix-vector multiplications is needed, for a given accuracy $\epsilon = 10^{-6}$ $\mathcal{H}$-matrices outperform $\mathcal{H}^2$-matrices as soon as $|\alpha| \leq \frac{\pi}{4}$ in $s = |s|\mathrm{e}^{i\alpha}$. We check this result for larger and smaller matrices in Table 2.5.

| $M$ | $s$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $N_{mv}$ |
|---|---|---|---|---|---|---|
| 2048 | 8-8i | 24.4 | 17.5 (3.2) | 0.007 | 0.75 | 3 |
| 4232 | 11.3-11.3i | 60.1 | 41.7 (8.3) | 0.015 | 0.9 | 3 |
| 8192 | 16-16i | 106.6 | 84.5 (18.5) | 0.03 | 2.05 | 2 |
| 16200 | 22.6-22.6i | 204.3 | 159 (32) | 0.05 | 3.2 | 3 |
| 32768 | 32-32i | 427 | 374 (77) | 0.1 | 6.5 | 2 |
| 65448 | 45.3-45.3i | 878.2 | 758 (137) | 0.2 | 10.4 | 2 |
| 129970 | 64-64.0i | 1798.5 | 1548 (309) | 0.44 | 22.2 | 2 |

*Table 2.5: Construction times $T_c$, matrix-vector multiplication times $T_{mv}$ and computed relative errors for the accuracy setup $\epsilon = 10^{-6}$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. $N_{mv}$ stands for the number of the matrix-vector multiplications needed for the $\mathcal{H}$-matrix approximation to outperform the $\mathcal{H}^2$-approximation (where the time for the matrix-vector multiplication includes the construction of the leaf cluster basis). In all experiments the relative error of the matrix-vector product did not exceed $2.4 \cdot 10^{-6}$.*

Numerical results in Table 2.5 suggest that, similarly to the case of the Helmholtz equation with decay, the assembly time of $\mathcal{H}$-matrices is larger than that of $\mathcal{H}^2$-matrices. However, if in the case of purely real wavenumber for the matrices of size $10^4 - 10^5$ the

difference varies from 2 to 15 times, in the case of the prevailing decay (i.e. for $s = |s|e^{i\alpha}$, $\alpha \in \left[-\frac{\pi}{4}, 0\right]$) the difference is not that significant. In our experiments it never exceeded 2 times for matrices of size $10^4 - 10^5$. This shows that $\mathcal{H}$-matrix approximations in this case are more efficient, even if a small number of matrix-vector multiplications is needed.

This is not the case when a higher accuracy is required (see Table 2.6): the $\mathcal{H}$-matrices outperform $\mathcal{H}^2$-matrices for larger problems only when more than 8 matrix-vector multiplications is needed.

| $M$ | $s$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $N_{mv}$ |
|---|---|---|---|---|---|---|
| 2048 | 8-8i | 87 | 82.2 (3.8) | 0.01 | 0.8 | 2 |
| 4232 | 11.3-11.3i | 219 | 171.2 (7.2) | 0.03 | 1.8 | 7 |
| 8192 | 16-16i | 484 | 301 (19) | 0.05 | 3.2 | 10 |
| 16200 | 22.6-22.6i | 925 | 544.4 (34.4) | 0.1 | 6.8 | 11 |
| 32768 | 32-32i | 1709 | 1103 (83.4) | 0.2 | 12.2 | 8 |
| 65448 | 45.3-45.3i | 3396 | 2220 (157.3) | 0.37 | 22.4 | 8 |
| 129970 | 64-64.0i | 6924.3 | 4238.4 (353.4) | 0.7 | 46.2 | 8 |

*Table 2.6: Times of the matrix construction and matrix-vector multiplication for the accuracy setup $\epsilon = 10^{-9}$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. $N_{mv}$ stands for the number of the matrix-vector multiplications needed for the $\mathcal{H}$-matrix approximation to outperform the $\mathcal{H}^2$-approximation (where the time for the matrix-vector multiplication includes the construction of the leaf cluster basis). The $\mathcal{H}^2$-matrix relative error never exceeded $10^{-8}$. For $\mathcal{H}$-matrices it deteriorated to $2 \cdot 10^{-7}$ for the two largest problems, which is likely to be connected to the insufficiently high Galerkin quadrature order.*

An actual choice whether $\mathcal{H}$-matrices or $\mathcal{H}^2$-matrices have to be employed is rather difficult. For the case of moderate accuracies ($10^{-6} - 10^{-7}$) we suggest the following heuristic. $\mathcal{H}^2$-matrices should be constructed in all the cases except:

- in the case of prevailing decay: $s = |s|e^{i\alpha}$, $|\alpha| \leq \frac{\pi}{4}$ (and more than two matrix-vector multiplications are needed);
- when at the first few levels of the admissible block-cluster tree there are admissible block-clusters that cannot be approximated by FMM expansions because of the low-frequency breakdown.

This heuristic is not difficult to adapt to higher accuracies. As the results in Tables 2.7, 2.6 show, for $\epsilon = 10^{-9}$ $\mathcal{H}^2$-matrices outperform $\mathcal{H}$-matrices for $|\alpha| < \frac{\pi}{4}$ if more than 8-12 matrix-vector multiplications are needed. Hence, if high accuracies are needed, we suggest using $\mathcal{H}^2$-matrices in all the cases but when the frequencies are low enough, see also the above heuristic.

| $\phi$ | $T_c(\mathcal{H})$ | $T_c(\mathcal{H}^2)$ | $T_{mv}(\mathcal{H})$ | $T_{mv}(\mathcal{H}^2)$ | $N_{mv}$ |
|---|---|---|---|---|---|
| $-\frac{\pi}{2}$ | 1488 | 693 (11.5) | 0.4 | 5 | 51 |
| $-\frac{6\pi}{14}$ | 1868 | 558 (31) | 0.3 | 4.9 | 38 |
| $-\frac{5\pi}{14}$ | 1450 | 546 (28.8) | 0.2 | 5.2 | 28 |
| $-\frac{4\pi}{14}$ | 1141 | 493 (28.8) | 0.13 | 5.2 | 20 |
| $-\frac{3\pi}{14}$ | 966 | 483 (28.6) | 0.12 | 5 | 16 |
| $-\frac{2\pi}{14}$ | 874 | 477 (26.6) | 0.11 | 4.3 | 14 |
| $-\frac{\pi}{14}$ | 818 | 459 (30) | 0.1 | 4.3 | 12 |
| 0 | 656 | 373 (21.2) | 0.1 | 4.7 | 12 |

*Table 2.7: Construction times $T_c$, matrix-vector multiplication times $T_{mv}$ and computed relative errors for the accuracy setup $\epsilon = 10^{-9}$, $M = 16200$, $|s| = 22.6$. The times are given in seconds. In brackets the time to construct the leaf cluster basis is shown. $N_{mv}$ stands for the number of the matrix-vector multiplications needed for the $\mathcal{H}$-matrix approximation to outperform the $\mathcal{H}^2$-approximation (where the time for the matrix-vector multiplication includes the construction of the leaf cluster basis). In all the experiments the relative error of the $\mathcal{H}$-matrix approximations did not exceed $7.1e-9$ and of the $\mathcal{H}^2$-approximations $3.3e-9$ (and for the case with non-zero decay $1.1e-9$).*

# Chapter 3

# Fast Runge-Kutta Convolution Quadrature Algorithm

In this section we present the fast convolution quadrature algorithm based on two main ideas:

- the reuse of the near-field;
- the application of data-sparse techniques for the approximation of the far-field.

In the first part of this section we study Runge-Kutta convolution weights. Namely, we prove that convolution kernels $w_n^h(d)$ defined as coefficients of the expansion, see (1.52),

$$\frac{\mathrm{e}^{-\Delta(\xi)\frac{d}{h}}}{4\pi d} = \sum_{k=0}^{\infty} w_k^h(d)\xi^k,$$
$$\Delta(\xi) = \left(A + \frac{\xi}{1-\xi}A^{-1}\mathbb{1}b^T A^{-1}\right)^{-1},$$

decay exponentially fast outside of $d \approx nh$. We demonstrate as well the dependence of the speed of decay on the order of the underlying Runge-Kutta method. Based on these results, the article [25] has been submitted. Similar properties are known to hold for the BDF2 method and were used in works [115, 116, 131] to improve the complexity of the BDF2 convolution quadrature algorithm. We study numerically the applicability of the ideas of the aforementioned works to Runge-Kutta convolution quadrature and analyze the associated difficulties.

Our approach is conceptually different from the one used in [115,116,131]. We construct the new method based on the algorithm of linear complexity, rather than back substitution of quadratic complexity (see [116]). This approach allows us to avoid the actual evaluation of the convolution weights, thus enabling the use of fast techniques based on analytic expansions. Computational and storage costs of the improved algorithm scale linearly, up to logarithmic factors.

We dedicate the rest of the section to the description of fast Runge-Kutta convolution quadrature, see also the submitted paper [24].

## 3.1 Sparsity of Runge-Kutta Convolution Weights and its Use

It is well-known that the strong Huygens principle holds for the wave equation in odd dimensions. As a consequence, the full discretization of the related TDBIE (using MOT or Galerkin methods with, for example, hat basis functions, see Section 1.2.1) leads to sparse matrices, see also Section 1.2.2. It is however a priori not clear if similar properties are exhibited by convolution quadrature. The positive answer had been given in [131] for BDF2 convolution quadrature, and the negative one in [21] for CQ based on the trapezoidal rule. In the same work it was numerically demonstrated that an analogue of Huygens principle can hold for the Runge-Kutta CQ discretized wave equation (depending on the underlying Runge-Kutta method).

In this section we prove a property of Runge-Kutta convolution weights that can be viewed as a counterpart of the strong Huygens principle. We show that under some (mild) assumptions on the Runge-Kutta method, convolution weights $w_n^h(d)$ decay away from $d \approx nh$. Additionally, we analyze the dependence of the speed of decay on the order of the Runge-Kutta method and demonstrate that the obtained bounds are close to optimal.

We show how these properties can be used to evaluate convolution weights $w_n^h(d)$ for some range of $d$ with a machine accuracy. These results will be of use for the fast Runge-Kutta convolution quadrature algorithm.

### 3.1.1 Decay of Convolution Weights

Let us introduce scaled convolution weights $\mathrm{w}_n(d) := 4\pi d w_n^h(hd)$. Functions $\mathrm{w}_n(d)$ are coefficients of the following expansion (see (1.52)):

$$\exp\left(-\Delta(\xi)d\right) = \sum_{n=0}^{\infty} \mathrm{w}_n\left(d\right)\xi^n. \tag{3.1}$$

Our task in this section is to find the estimates for scaled convolution weights $\mathrm{w}_n(d)$ and next use these results to show that similar bounds hold also for $w_n^h(d)$.

All over this section we assume that the Runge-Kutta method satisfies Assumption 1.2.15.

A scaled convolution weight $\mathrm{w}_n(d)$ for $d > 0,\ n \geq 1$, can also be expressed as

$$\mathrm{w}_n(d) = \frac{1}{2\pi i} \oint\limits_{\gamma} R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}\mathrm{e}^{-zd}dz, \tag{3.2}$$

see [139]. Here, $\gamma$ represents a contour that encloses all the eigenvalues of $A^{-1}$ (which are also singularities of $R(z)$, see Section 1.2.8.1).

To prove the main estimates, we will need to choose the contour $\gamma$ carefully.

First, we consider an open set $\Upsilon_r$, $r > 0$:

$$\Upsilon_r = \left\{z \in \mathbb{C} \ :\ |R(z)| > r\right\}.$$

The contour $\gamma_r$ is defined as the boundary of this set, i.e., $\gamma_r := \partial\Upsilon_r$. Hence, $|R(z)| = r$ holds for all $z \in \gamma_r$. Next, we prove some properties of sets $\Upsilon_r$.

Let

$$A_+ = \left\{z \in \mathbb{C} \ :\ |R(z)| > |e^z|,\ \mathrm{Re}\, z > 0\right\}$$

denote the order star of $R(z)$ restricted to the right-half complex plane, see [122]. In fact $A_+$ denotes just $m$ bounded fingers containing $m$, counting multiplicities, singularities of $R$. Since $|R(iy)| < 1$ for $y \neq 0$, the origin is the only point of the intersection of the closure of the order star with the imaginary axis and hence

$$\overline{A}_+ \subset \Upsilon_1 \cup \{0\}.$$

Prior to continuing, let us recall that the stability function of a Runge-Kutta method is a rational function:

$$R(z) = \frac{P(z)}{Q(z)}, \tag{3.3}$$

where $P(z),\ Q(z)$ are polynomials with real coefficients. The degree of the polynomial $Q(z)$ defines the number of stages of the Runge-Kutta method. It is typically assumed that

$$P(0) = Q(0) = 1.$$

Additionally, we introduce an auxiliary polynomial, see [122, Chapter IV.3]:

$$E(y) = |Q(iy)|^2 - |P(iy)|^2 = e_0 y^{2s} + O(y^{2s+2}). \tag{3.4}$$

Proposition 3.4 in [122] shows that

$$E(y) = O(y^{p+1}).$$

**Remark 3.1.1.** *Note that for Runge-Kutta methods satisfying Assumption 1.2.15, the coefficient*

$$e_0 > 0. \tag{3.5}$$

*This follows from*

$$|R(iy)|^2 = \frac{|P(iy)|^2}{|Q(iy)|^2} = 1 - \frac{E(y)}{|Q(iy)|^2},$$

*which, after expansion into the series in $y$ (taking into account that $Q(0) = 1$), gives:*

$$|R(iy)|^2 = \frac{|P(iy)|^2}{|Q(iy)|^2} = 1 - e_0 y^{2s} + O(y^{2s+2}).$$

*Since, by Assumption 1.2.15, for all $y \in \mathbb{R} \setminus \{0\}$*

$$|R(iy)| < 1,$$

*the coefficient $e_0$ satisfies (3.5).*

We will need the following lemma. We believe this result to be known, however, we were not able to find the precise statement in the literature. The proof of this result can be found in Appendix B.

**Lemma 3.1.2.** *There exist $a, \nu > 0$, such that the domain*

$$\{(x, y) : \ |y| < \nu x^{\frac{1}{\ell}}, \ 0 < x < a\}$$

*belongs to $\Upsilon_1$ (and intersects all the order star fingers). Here*

$$\ell = \begin{cases} p+1, & \text{if } p \text{ is odd,} \\ 2s, & \text{if } p \text{ is even,} \end{cases}$$

*where $s$ is as in (3.4).*

**Lemma 3.1.3.** *Under Assumption 1.2.15, the set $\Upsilon_1$ is located in the open right-half plane and is bounded and connected (possibly multiply).*

*Proof.* The boundedness follows directly from the assumption of stiff accuracy $R(\infty) = 0$. A-stability and the bound $|R(iy)| < 1, \ y \in \mathbb{R} \setminus \{0\}$, imply that $\Upsilon_1$ is located in the open right-half plane.

Let $\widetilde{\Upsilon}_1$ be a connected (possibly multiply) component of $\Upsilon_1$. Then, by the maximum principle, $\widetilde{\Upsilon}_1$ must contain a singularity of $R(z)$ and the closure of the corresponding finger. According to Lemma 3.1.2, the intersection of $\widetilde{\Upsilon}_1$ with all the other fingers is nonempty. Since $\widetilde{\Upsilon}_1$ contains all the singularities of $\Upsilon_1$, by the maximum modulus principle applied to $R(z)$, it coincides with $\Upsilon_1$.

□

**Remark 3.1.4.** *The domain $\Upsilon_1$ is not necessarily simply connected: it can have a hole; for example, there can exist a bounded domain $\Upsilon'$, s.t. $R(z)$ vanishes in one of its interior points, $|R(z)| < 1$ inside $\Upsilon'$ and $\partial\Upsilon' \subset \partial\Upsilon_1$.*

**Remark 3.1.5.** *In a small enough vicinity of $r = 1$, $\Upsilon_r$ stays bounded and connected.*

*This follows from the fact that $z \in \partial\Upsilon_r$ is equivalent to (see (3.3))*

$$P(z) - Q(z)re^{i\phi} = 0, \qquad \phi \in [0, \ 2\pi) \, , \ Q(z) \neq 0.$$

*The roots of the polynomial depend continuously on its coefficients, and hence there exists $\delta_*$ s.t.*

$$\text{the domain } \Upsilon_r \text{ is bounded and connected for } |r-1| < \delta_*. \tag{3.6}$$

**Corollary 3.1.6.** *If the stability function of a Runge-Kutta method coincides with a Padé approximant for the exponential, the domain $\Upsilon_1$ is simply connected.*

*Proof.* For the proof we need two ingredients:

1. Ehle's Conjecture [183, Theorem 7]. Any Padé approximation $R(z) = \frac{P(z)}{Q(z)}$, $\deg P = k$, $\deg Q = m$ is $A$-stable iff $m - 2 \leq k \leq m$.

2. All zeros of such Padé approximants lie in the open left-half plane, see [79].

The existence of a bounded domain $\Upsilon'$, s.t. $|R(z)| < 1$ inside $\Upsilon'$ and $\partial\Upsilon' \subset \partial\Upsilon_1$ (i.e. a hole in $\Upsilon_1$), contradicts the maximum modulus principle applied to the analytic function $\frac{1}{R(z)}$, $z \in \mathbb{C}, \ \operatorname{Re} z > 0$. □

Curves $\gamma_r = \partial\Upsilon_r$, $r > 1$, can be drawn by plotting the eigenvalues of $\Delta(\xi)$ for all $\xi \in \mathbb{C} : \ |\xi| = 1/r$, see [21]. We repeat the respective result with minor modifications to cover the case $r \leq 1$. Let us first make a remark on the domain of definition of $\Delta(\xi)$. From formula (1.46), values of $\xi$ for which $\Delta(\xi) = (A + \frac{\xi}{1-\xi}\mathbb{1}b^T)^{-1} = A^{-1}(I + \frac{\xi}{1-\xi}A^{-1}\mathbb{1}b^T)^{-1}$ is not defined satisfy:

$$\frac{\xi - 1}{\xi} \text{ is an eigenvalue of } A^{-1}\mathbb{1}b^T.$$

The matrix $Q = A^{-1}\mathbb{1}b^T$ is a rank one matrix, therefore all its eigenvalues but one are equal to 0. It is not difficult to see that one of the left eigenvectors of $Q$ is $b$ (recall that $b^T A^{-1}\mathbb{1} = 1$ for stiffly accurate Runge-Kutta methods, see (1.40)) and the corresponding eigenvalue equals 1. Now note that $\frac{\xi-1}{\xi} \neq 1$ for any $\xi \in \mathbb{C}$. Hence the only value in which $\Delta(\xi)$ is not defined is $\xi = 1$. However $\xi = 1$ is a removable singularity, since, due to Sherman-Morrison-Woodbury formula:

$$\begin{aligned}
\Delta(\xi) = \lim_{\xi\to 1}\left(A + \frac{\xi}{1-\xi}\mathbb{1}b^T\right)^{-1} &= A^{-1} - \frac{\xi}{1-\xi}\frac{A^{-1}\mathbb{1}b^T A^{-1}}{1 + \frac{\xi}{1-\xi}b^T A^{-1}\mathbb{1}} \\
&= A^{-1} - \xi A^{-1}\mathbb{1}b^T A^{-1}, \qquad \text{for } \xi : \ 1 + \frac{\xi}{1-\xi}b^T A^{-1}\mathbb{1} \neq 0, \\
\lim_{\xi\to 1}\Delta(\xi) &= A^{-1} - A^{-1}\mathbb{1}b^T A^{-1}.
\end{aligned}$$

The last expression thus defines $\Delta(\xi)$ for all $\xi \in \mathbb{C}$.

**Lemma 3.1.7.** *For Runge-Kutta methods satisfying Assumption 1.2.15 any eigenvalue $\mu$ of $\Delta(\xi) = A^{-1} - \xi A^{-1}\mathbb{1}b^T A^{-1}$, $\xi \in \mathbb{C}$, is either an eigenvalue of $A^{-1}$ or $R(\mu) = \frac{1}{\xi}$.*

*Proof.* For $|\xi| < 1$ this result was proved in [21]; the proof can be trivially extended to the case $|\xi| \geq 1$ for Runge-Kutta methods satisfying Assumption 1.2.15. We repeat here this proof to cover the case of general $\xi$. Assume $v \in \mathbb{C}^m$, $\Delta(\xi)v = \mu v$, where $\mu \in \mathbb{C}$ is not an eigenvalue of $A^{-1}$. Additionally, let $b^T A^{-1}v \neq 0$. Then,

$$\begin{aligned}
\mu A v &= v - \xi\mathbb{1}b^T A^{-1}v, \\
(I - \mu A)v &= \xi\mathbb{1}b^T A^{-1}v, \\
\frac{1}{\xi}b^T A^{-1}v &= b^T A^{-1}(I - \mu A)^{-1}\mathbb{1}b^T A^{-1}v.
\end{aligned}$$

Dividing both sides by $b^T A^{-1}v$ and using the expression (1.41), we obtain

$$R(\mu) = \frac{1}{\xi}.$$

Now note that $b^T A^{-1}v = 0$ implies that $\mu$ is an eigenvalue of $A^{-1}$ and $v$ is the corresponding eigenvector:

$$\mu v = \Delta(\xi)v = A^{-1}v - \xi A^{-1}\mathbb{1}b^T A^{-1}v = A^{-1}v.$$

□

**Remark 3.1.8.** *Lemma 3.1.7 provides us with an efficient way to plot curves $\gamma_r$.*

*In [21] the multiplicity of the eigenvalues of $\Delta(\xi)$ for the 2- and 3-stage Radau IIA Runge-Kutta methods was discussed. In both cases $\Delta(\xi)$ has only simple eigenvalues for $|\xi| = 1$. For the 2-stage version, eigenvalues of multiplicity greater than 1 occur only for $|\xi| = -5 \pm 3\sqrt{3}$. For the 3-stage version the critical values are $|\xi| \approx 0.069366077$ and $|\xi| \approx 15.7581353$.*

*For a plot of the curves $\gamma_r$ for the 3-stage Radau IIA at critical values $r = \frac{1}{|\xi|}$ and $r = 1$ see Figure 3.1.*

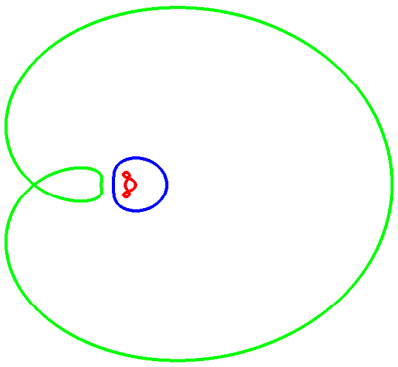

*Figure 3.1: Curves $\gamma_r$, for the 3-stage Radau IIA method, are plotted for $r = 1$ (middle curve in blue) and the critical values $r = r_1 \approx \frac{1}{0.069366077}$ (inner curve in red) and $r = r_2 \approx \frac{1}{15.7581353}$ (outer curve in green). For $r > r_1$ the curve splits into three disjoint curves, and for $r < r_2$ into two.*

Let us fix $r > 0$ satisfying (3.6) and choose a positively oriented contour $\gamma = \gamma_r$. We will assume that the domain $\Upsilon_1$ is simply connected (though all the arguments can be trivially extended to the multiple connectedness case). By Remark 3.1.5, $\Upsilon_r$, for sufficiently small $|r - 1|$, is also simply connected.

**Remark 3.1.9.** *Recall that $m$ is the number of stages of a Runge-Kutta method, as well as the degree of the polynomial $Q(z)$ in $R(z) = \frac{P(z)}{Q(z)}$. Then the length of the curve $\gamma_r$ is bounded, see [7, Lemma 3], by:*

$$|\gamma_r| \leq 4md(\gamma_r),$$

*where $d(\gamma_r)$ is the diameter of the curve $\gamma_r$.*

From (3.2) the following bound on the Euclidean norm of $\mathrm{w}_n(d)$ follows:

$$\begin{aligned}\|\mathrm{w}_n(d)\| &\leq \frac{1}{2\pi}\left\|\int_{\gamma_r} R(z)^{n-1}\mathrm{e}^{-zd}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}dz\right\| \\ &\leq \frac{1}{2\pi}|\gamma_r|r^{n-1}\max_{z\in\gamma_r}\|\mathrm{e}^{-zd}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}\|.\end{aligned}$$

Denoting by $Q_A(z) = (I - Az)^{-1}$, one can deduce the bound

$$\begin{aligned}\max_{z\in\gamma_r}\|(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}\| &\leq \max_{z\in\gamma_r}\|Q_A(z)\|^2\|\mathbb{1}b^T\| \\ &\leq \max_{z\in\gamma_r}\|Q_A(z)\|^2\|b\|\sqrt{m},\end{aligned}$$

which implies that

$$\|\mathrm{w}_n(d)\| \le \frac{1}{2\pi} r^{n-1} |\gamma_r| \|b\| \sqrt{m} \max_{z\in\gamma_r} |\mathrm{e}^{-zd}| \max_{z\in\gamma_r} \|Q_A(z)\|^2. \tag{3.7}$$

To understand the behavior of a scaled convolution weight $\mathrm{w}_n(d)$ we need to find a bound on $\max\limits_{z\in\gamma_r} |\mathrm{e}^{-zd}|$. To do so, we use the fact that the stability function $R(z)$ is an approximant to $\mathrm{e}^z$, see (1.39), and thus $\max\limits_{z\in\gamma_r} |\mathrm{e}^{-zd}|$ can be expressed via the value of $|R(z)|$ on $\gamma_r$.

For a Runge-Kutta method of order $p$ we can write

$$R(z) = \mathrm{e}^z + f(z),$$

where $f(z) = O(z^{p+1})$.

Let us consider $z \in \gamma_r$ and $d \in \mathbb{R}_{>0}$. Multiplying the last equation by $\mathrm{e}^{-z} R(z)^{-1}$, we obtain:

$$g(z) := \mathrm{e}^{-z} = R(z)^{-1} \left(1 + \mathrm{e}^{-z} f(z)\right).$$

Our goal is to find a (tight) bound on $|g(z)|^d = |\mathrm{e}^{-zd}|$ for $z \in \gamma_r$. The maximum of $|g(z)|$ for $z \in \gamma_r$ is achieved in $z_0 \in \gamma_r$, such that

$$\operatorname{Re} z' \ge \operatorname{Re} z_0, \qquad \text{for all} \quad z' \in \gamma_r. \tag{3.8}$$

Such point is not necessarily unique. Let $z_0 = x_0 + iy_0$. Then

$$\begin{aligned} \max_{z\in\gamma_r} |g(z)|^d = \mathrm{e}^{-x_0 d} &= |R(z_0)|^{-d} \left|1 + f(z_0)\mathrm{e}^{-z_0}\right|^d \\ &= r^{-d} \left|1 + f(z_0)\mathrm{e}^{-z_0}\right|^d. \end{aligned} \tag{3.9}$$

Hence, to bound $g(z)$, $z \in \gamma_r$, it is required to understand how $z_0$ behaves, at least for some range of $r$.

Although a point $z_0 = x_0 + iy_0$ as defined above is not necessarily unique, but, as we argue later, as $r \to 1$, all such points are located close to the origin, namely

$$|z_0| \le C|r - 1|^\alpha,$$

for some $C, \alpha > 0$. This question has been studied in detail in the work [120] examining the behavior of $R(z)$ in the order star [183]. This and the fact that $f(z) = O(z^{p+1})$ will allow us to obtain the required bounds on scaled convolution weights.

**Remark 3.1.10.** *The points $z_0$ lying on the contour $|R(z)| = r$ that have the smallest real part can be alternatively characterized by two properties:*

1. *For all $z:\ \operatorname{Re} z < \operatorname{Re} z_0$, it holds that*

$$|R(z_0)| > |R(z)|. \tag{3.10}$$

*This is due to the analyticity of $R(z)$, as well as the definition of the contour $\gamma_r$ and points $z_0$.*

2. *Let us fix* $x_0 = \operatorname{Re} z_0$. *Note that* $|R(x_0 + iy)|$, $y \in \mathbb{R}$, *is bounded. Then* $y_0$, *see (3.8), are the points in which* $|R(x_0 + iy)|$ *achieves its maximum. Let us show this. We assume the contrary, namely, that there exists* $y' \in \mathbb{R}$ *s.t.*

$$|R(x_0 + iy')| > |R(x_0 + iy_0)|.$$

*Since* $R(z)$ *is analytic in* $z' = x_0 + iy'$, *there exists an* $\epsilon$*-neighborhood of* $z'$ *s.t.*

$$|R(z)| > |R(x_0 + iy_0)|, \qquad |z - z'| < \epsilon.$$

*Taking* $z = z' - \epsilon = (x_0 - \epsilon) + iy'$, *we arrive at the contradiction to (3.10).*

Here we will employ some results from [120].

**Definition 3.1.11.** *( [120]) Given a rational function* $R(z)$ *we define the error growth function as the real-valued function* $\phi(x) := \sup_{\operatorname{Re} z < x} |R(z)|$.

**Theorem 3.1.12.** *(Theorem 7 in [120]) Let* $R(z) = \frac{P(z)}{Q(z)}$ *be an A-stable approximation to* $\mathrm{e}^z$ *of exact order* $p \geq 1$, *namely:*

$$R(z) = \frac{P(z)}{Q(z)} = \mathrm{e}^z + C_{p+1} z^{p+1} + O(z^{p+2}), \qquad \text{for} \quad z \to 0,\ C_{p+1} \neq 0. \tag{3.11}$$

*Furthermore, assume* $|R(iy)| < 1$ *for* $y \neq 0$, *and* $|R(\infty)| < 1$. *Then we have for* $x \to 0$*:*

- *if* $p$ *is odd,*
$$\phi(x) = \mathrm{e}^x + O(x^{p+1}),$$
- *if* $p$ *is even and* $(-1)^{p/2} C_{p+1} x < 0$,
$$\phi(x) = \mathrm{e}^x + O(x^{p+1}).$$
- *if* $p$ *is even and* $(-1)^{p/2} C_{p+1} x > 0$,
$$\phi(x) = \mathrm{e}^x + O(x^{1+p/(2s-p)}),$$
*where* $s$ *is defined by (3.4).*

**Remark 3.1.13.** *( [120]) For* $x < \operatorname{Re} \lambda_{min}$, *with* $\lambda_{min}$ *being an eigenvalue of* $A^{-1}$ *with the smallest real part,* $\phi(x)$ *is a strictly monotonically increasing continuous function.*

The following proposition shows that for $r \to 1$ $x_0 = \min_{z \in \gamma_r} \operatorname{Re} z$ is close to $r - 1$.

**Proposition 3.1.14.** *Let* $R(z)$ *be the stability function of the Runge-Kutta method satisfying Assumption 1.2.15, let (3.11) hold and let* $x_0 = \min_{z \in \gamma_r} \operatorname{Re} z$. *Then for* $r \to 1$*:*

- *if* $p$ *is odd,*
$$x_0 = r - 1 + O((r-1)^2).$$
- *if* $p$ *is even and* $(-1)^{p/2} C_{p+1} x_0 < 0$,
$$x_0 = r - 1 + O((r-1)^2).$$

- *if $p$ is even and $(-1)^{p/2}C_{p+1}x_0 > 0$,*

$$x_0 = r - 1 + o(|r-1|).$$

*Proof.* On the contour $\gamma_r$,

$$|R(z)| = r.$$

Since the error growth function $\phi(x)$ is a strictly monotonically increasing continuous function, see Remark 3.1.13, $\phi(x_0) = r$. The statement of the proposition follows from the application of the implicit function theorem to the 3 cases of Theorem 3.1.12 and the fact that $\phi(0) = 1$, $\frac{d\phi}{dx}(0) = 1$. □

The next proposition shows that when $r \approx 1$, points $z_0$ defined by (3.8) lie in a small circle centered at the origin.

**Proposition 3.1.15.** *Let $R(z)$ be the stability function of the Runge-Kutta method satisfying Assumption 1.2.15 and (3.11). Then there exist $\delta_0 > 0$ and $K > 0$, s.t. for all $r$ with $|r-1| < \delta_0$ the points $z_0 \in \gamma_r$ defined by (3.8) lie inside one of the circles specified below.*

1. *for $p$ odd:*

$$|z_0| \le K|r-1|.$$

2. *for $p$ even:*

   (a) *if $r > 1$ and $(-1)^{\frac{p}{2}}C_{p+1} < 0$ or $r < 1$ and $(-1)^{\frac{p}{2}}C_{p+1} > 0$,*

$$|z_0| \le K|r-1|.$$

   (b) *if $r > 1$ and $(-1)^{\frac{p}{2}}C_{p+1} > 0$ or $r < 1$ and $(-1)^{\frac{p}{2}}C_{p+1} < 0$,*

$$|z_0| \le K|r-1|^{\frac{1}{2s-p}},$$

   *where $s$ is defined by (3.4).*

*The constant $K$ depends only on the Runge-Kutta method.*

*Proof.* The proof of this statement closely follows the proof of Theorem 7 in [120]. Recall that $z_0$ is chosen so that

$$|R(z_0)| = r,$$

and $x_0 = \operatorname{Re} z_0 < \operatorname{Re} z,\ z \in \gamma_r$. Then

$$\phi(\operatorname{Re} z_0) = \sup_{\operatorname{Re} z < x_0} |R(z)| = r,$$

as shown in the proof of Proposition 3.1.14. Also,

$$\max_y |R(x_0 + iy)| = r,$$

see Remark 3.1.10. Hence, to bound $z_0$, we have to look for $y_0$ at which an extremum is achieved:

$$\max_{y\in\mathbb{R}} |R(x_0+iy)| = |R(x_0+iy_0)|.$$

As argued in the proof of Theorem 7 in [120], for $x \to 0$ the maximum $\max_{y\in\mathbb{R}} |R(x+iy)|$, has to lie inside the order star close to the origin. Indeed, $\max_{y\in\mathbb{R}} |R(iy)| = 1$ and is achieved in $y = 0$; as $z \to +\infty$ $|R(z)| < 1$. Then, for small $x$, due to the smoothness of $|R(x+iy)|$ in $x$, $y$, the maximum $\max_{y\in\mathbb{R}} |R(x+iy)|$ has to lie close to the origin.

Let us fix $r:\ |r-1| < \epsilon$, for some small $\epsilon > 0$ ($x_0 = O(\epsilon)$, as shown in the previous proposition), and consider the following cases as $\epsilon \to 0$.

1. $p$ is odd.

   As shown in the proof of Theorem 7 in [120], the local extrema of $|R(x_0+iy)|,\ y \in \mathbb{R}$, lie asymptotically on the lines $y_k = x_0 \tan((k-1)\pi/p),\ k = 1, 2, \ldots, p$. Since $y_0$ is equal to $y_k$ for some $k = 1, 2, \ldots, p$, we can bound

   $$|z_0| = \left( x_0^2 + x_0^2 \sup_{k=1,2,\ldots,p} \tan^2\left((k-1)\frac{\pi}{p}\right)\right)^{\frac{1}{2}} \le C|x_0|,$$

   where $C > 0$ depends only on the Runge-Kutta method.

   Proposition 3.1.14 gives an explicit expression for $x_0$:

   $$x_0 = r - 1 + O((r-1)^2).$$

   Hence,

   $$|z_0| \le K|r-1|,$$

   for some $K > 0$.

2. $p$ is even.

   (a) As proved in Theorem 7 in [120], for $(-1)^{p/2} C_{p+1} x_0 < 0$ and $x_0$ sufficiently small, $|z_0|$ is asymptotically bounded:

   $$|z_0| \le C|x_0|,\ C > 0.$$

   The statement of the proposition is obtained with the help of the same arguments as in the previous case and the fact that $\operatorname{sgn} x_0 = \operatorname{sgn}(r-1)$.

   (b) For the last case, namely $(-1)^{p/2} C_{p+1} x_0 > 0$, in the proof of Theorem 7 in [120] it was shown that the maximum of $|R(x_0+iy)|$ in $y \in \mathbb{R}$ is achieved near the imaginary axis in the points

   $$y^{2s-p} = Dx_0,$$

   where $D \in \mathbb{R}$ is a constant (that depends on the Runge-Kutta method) and $s$ is defined by (3.4).

Then

$$|z_0| = \left(x_0^2 + |D|^{\frac{2}{2s-p}}|x_0|^{\frac{2}{2s-p}}\right)^{\frac{1}{2}}$$
$$= |x_0|^{1/(2s-p)}\left(|D|^{\frac{2}{2s-p}} + |x_0|^{2-\frac{2}{2s-p}}\right)^{\frac{1}{2}}.$$

According to Proposition 3.4 in [122] $2s \geq p+1$, therefore, for even $p$, $2s \geq p+2$. This implies that $|x_0|^{2-2/(2s-p)} = o(1)$ and hence

$$|z_0| \leq K|r-1|^{\frac{1}{2s-p}},$$

for some $K > 0$.

□

Now we have all the estimates necessary to prove the next proposition on the decay of scaled convolution weights.

**Proposition 3.1.16.** *Let $\mathrm{w}_n(d)$, $n \geq 0$, be scaled convolution weights for an $m$-stage Runge-Kutta method of order $p$ that satisfies Assumption 1.2.15 and (3.11).*

*Let $s$ be defined by (3.4). Then there exist positive constants $G$, $G'$, $C$, $C'$ and $\bar{\delta} \in (0,\ 1)$, such that for $n \geq 1$ and $0 < \delta < \bar{\delta}$ the following estimates hold:*

1. *$p$ is odd*

$$\begin{aligned} \|\mathrm{w}_n(d)\| &\leq G(1-\delta)^{n-d}(1+C\delta^{p+1})^d \qquad \text{for} \quad d \leq n, \\ \|\mathrm{w}_n(d)\| &\leq G'(1+\delta)^{n-d}(1+C'\delta^{p+1})^d \qquad \text{for} \quad d > n; \end{aligned} \tag{3.12}$$

2. *$p$ is even*

   (a) *$C_{p+1}(-1)^{\frac{p}{2}} > 0$*

$$\begin{aligned} \|\mathrm{w}_n(d)\| &\leq G(1-\delta)^{n-d}(1+C\delta^{p+1})^d \qquad \text{for} \quad d \leq n, \\ \|\mathrm{w}_n(d)\| &\leq G'(1+\delta)^{n-d}(1+C'\delta^{\frac{p+1}{2s-p}})^d \qquad \text{for} \quad d > n; \end{aligned} \tag{3.13}$$

   (b) *$C_{p+1}(-1)^{\frac{p}{2}} < 0$*

$$\begin{aligned} \|\mathrm{w}_n(d)\| &\leq G(1-\delta)^{n-d}(1+C\delta^{\frac{p+1}{2s-p}})^d \qquad \text{for} \quad d \leq n, \\ \|\mathrm{w}_n(d)\| &\leq G'(1+\delta)^{n-d}(1+C'\delta^{p+1})^d \qquad \text{for} \quad d > n. \end{aligned} \tag{3.14}$$

*The scaled convolution weight $\mathrm{w}_0(d)$ satisfies:*

$$\|\mathrm{w}_0(d)\| \leq \exp(-\mu d), \tag{3.15}$$

*for some $\mu > 0$.*

*Constants $G$, $G'$, $C$, $C'$, $\bar{\delta}$, $\mu$ depend only on the Runge-Kutta method and do not depend on $n$ or $d$.*

*Proof.* Let us start with the case $\mathrm{w}_0(d)$. From the definition of scaled convolution weights

$$\exp\left(-\Delta(\xi)d\right) = \sum_{n=0}^{\infty} \mathrm{w}_n\left(d\right)\xi^n,$$
$$\Delta(\xi) = A^{-1} - \xi A^{-1}\mathbb{1}b^T A^{-1},$$

it follows that $\mathrm{w}_0(d) = \exp(-A^{-1}d)$. All the eigenvalues of $A$ lie on the right from the imaginary axis (due to A-stability of the Runge-Kutta method). Same holds for the eigenvalues of $A^{-1}$. The bound on $\mathrm{w}_0(d)$ can then be obtained from the definition of the matrix exponential.

For a general case $\mathrm{w}_n(d)$, $n \geq 1$, we use the bounds derived before, inserting (3.9) into (3.7):

$$\begin{aligned}\|\mathrm{w}_n(d)\| &\leq \frac{1}{2\pi} r^{n-1}\|b\|\sqrt{m}|\gamma_r| \max_{z\in\gamma_r}|\mathrm{e}^{-zd}| \max_{z\in\gamma_r}\|Q_A(z)\|^2 \\ &= \frac{1}{2\pi} r^{n-d-1}|\gamma_r| \max_{z\in\gamma_r}\|Q_A(z)\|^2\|b\|\sqrt{m}|1+f(z_0)\mathrm{e}^{-z_0}|^d, \end{aligned} \tag{3.16}$$

where $z_0$ is such that for all $z' \in \gamma_r$

$$\operatorname{Re} z' \geq \operatorname{Re} z_0$$

and $f(z) = R(z) - \mathrm{e}^z$. Here $r \in \mathbb{R}_{>0}$ is fixed. Properly speaking, $z_0$ depends on $r$.

Let us first derive the bound for $|1 + f(z_0)\mathrm{e}^{-z_0}|$. For $|z| < \frac{1}{\lambda_0}$, where $\lambda_0$ is the spectral radius of $A$, we can expand $R(z) = 1 + zb^T(I - Az)^{-1}\mathbb{1}$ with the help of Neumann series to obtain an explicit expression for $f(z)$:

$$f(z) = R(z) - \mathrm{e}^z = z\sum_{\ell=p}^{\infty} b^T A^\ell \mathbb{1} z^\ell - \sum_{\ell=p+1}^{\infty} \frac{z^\ell}{\ell!}.$$

For $|z| < \frac{1}{\|A\|}$, we can trivially bound

$$|1 + f(z)\mathrm{e}^{-z}|^d \leq \left(1 + C|z|^{p+1}\right)^d, \tag{3.17}$$

where $C$ depends on the Runge-Kutta method, but does not depend on $z$ or $d$.

Now let $d \leq n$. We choose $r < 1$, $r = 1 - \delta$, $0 < \delta < \delta_*$, where $\delta_*$ is a constant from (3.6), which allows to choose the contour $\gamma_r$, s.t. $|R(z)| = r$ for all $z \in \gamma_r$.

Then the bound (3.16), using (3.17), can be rewritten as:

$$\|\mathrm{w}_n(d)\| \leq \frac{1}{2\pi}(1-\delta)^{n-d-1}|\gamma_{1-\delta}| \max_{z\in\gamma_{1-\delta}}\|Q_A(z)\|^2\|b\|\sqrt{m}\left(1 + C|z_0|^{p+1}\right)^d,$$

where $z_0$ is such that $\operatorname{Re} z_0 < \operatorname{Re} z$ for all $z \in \gamma_{1-\delta}$.

The length of the curve $\gamma_{1-\delta}$ as well as $\max\limits_{z\in\gamma_{1-\delta}}\|Q_A(z)\|$, for $0 < \delta < \delta_*$, can be bounded by constants that depend on the Runge-Kutta method only, see also Lemma 3.1.3 and Remarks 3.1.5 and 3.1.9.

Now let us choose $\delta$ sufficiently small, so that Proposition 3.1.15 can be applied to estimate $\left(1 + C|z_0|^{p+1}\right)^d$. This allows us to obtain the required expressions for the case $n > d$.

The bound for $d > n$ can be obtained similarly setting $r = 1+\delta$, with $\delta$ chosen sufficiently small. □

**Remark 3.1.17.** *Note that for even $p$ the above bounds imply that when $2s - p < p + 1$ scaled convolution weights decay exponentially. However, $2s \leq 2m$ ($m$ is the number of stages and the degree of the denominator in $R(z) = \frac{P(z)}{Q(z)}$), and thus for exponential decay it suffices that $p \geq m$.*

We have shown that scaled convolution weights $\mathrm{w}_n(d)$ exhibit exponential decay outside of a neighborhood of $n \approx d$, which is an expression of the strong Huygens principle. Additionally, the above estimates suggest that the size of the approximate support of a convolution weight $\mathrm{w}_n(d)$ increases with $d$, $n$. Let us examine this in more detail. We define the approximate support $\mathrm{w}_n(d)$, $n > 0$, as

$$\begin{aligned} \operatorname{supp}_\epsilon \mathrm{w}_n &= \left[d_1^{(n,\epsilon)},\ d_2^{(n,\epsilon)}\right], \\ d_1^{(n,\epsilon)} &= \sup\left\{d:\ \|\mathrm{w}_n(d')\| < \epsilon,\ \text{ for all } 0 \leq d' < d\right\}, \\ d_2^{(n,\epsilon)} &= \inf\left\{d:\ \|\mathrm{w}_n(d')\| < \epsilon,\ \text{ for all } d' > d\right\}. \end{aligned} \tag{3.18}$$

The set

$$\left\{d:\ \|\mathrm{w}_n(d')\| < \epsilon,\ d' < d\right\}$$

is non-empty for $n \geq 1$, since $\mathrm{w}_n(0) = 0$ (as we show in Lemma 3.1.21) and $\mathrm{w}_n(d)$ is smooth in $d \geq 0$. We assume that the set

$$\left\{d:\ \|\mathrm{w}_n(d')\| < \epsilon,\ d' > d\right\}$$

is non-empty for all $n \geq 0$. By Proposition 3.1.16, this holds for Runge-Kutta methods of odd order, while for Runge-Kutta methods of even order we require that $2s - p < p + 1$ (where $p$ is the order and $s$ is as in (3.4)), see also Remark 3.1.17.

Hence, the values $d_1^{(n,\epsilon)}$, $d_2^{(n,\epsilon)}$ are defined for all $n \geq 1$.

To find the estimates on $d_1^{(n,\epsilon)}$, $d_2^{(n,\epsilon)}$, $n > 0$, we make use of the bounds of Proposition 3.1.16 that can be written in a more general form:

$$\begin{aligned} \|\mathrm{w}_n(d)\| &\leq G(1-\delta)^{n-d}(1 + C\delta^\alpha)^d \qquad \text{for all } d \leq n, \\ \|\mathrm{w}_n(d)\| &\leq G'(1+\delta)^{n-d}(1 + C'\delta^{\alpha'})^d \qquad \text{for all } d > n, \end{aligned}$$

for all $\delta < \bar{\delta}$, for constants $C, C', G, G', \alpha, \alpha', \bar{\delta} > 0$ depending only on the Runge-Kutta method. The estimates on values $d_1^{(n,\epsilon)}$, $d_2^{(n,\epsilon)}$ can be found from

$$\begin{aligned} G(1-\delta)^{n-d}(1 + C\delta^\alpha)^d &\leq \epsilon, \\ G'(1+\delta)^{n-d}(1 + C'\delta^{\alpha'})^d &\leq \epsilon. \end{aligned}$$

The solution to the first inequality is given by

$$\begin{aligned} d &< n \log\frac{1}{1-\delta}\left(\log\frac{1}{1-\delta} + \log(1 + C\delta^\alpha)\right)^{-1} - \log\frac{G}{\epsilon}\left(\log\frac{1}{1-\delta} + \log(1 + C\delta^\alpha)\right)^{-1} \\ &= n - \left(\log\frac{1}{1-\delta} + \log(1 + C\delta^\alpha)\right)^{-1}\left(\log\frac{G}{\epsilon} + n\log(1 + C\delta^\alpha)\right) \end{aligned}$$

For bounded $\delta$, there exist constants $c_1, c_2, C' > 0$ s.t. (recall that $\alpha > 1$):

$$d < n - \frac{c_1 n \delta^{\alpha-1} + c_2 \delta^{-1} \log \frac{G}{\epsilon}}{1 - C'\delta}. \tag{3.19}$$

The choice $\delta = cn^{-\frac{1}{\alpha}} \log^{\frac{1}{\alpha}} \frac{G}{\epsilon}$, with $c$ being a small constant, ensures that

$$d_1^{n,\epsilon} \geq n - C_1 n^{\frac{1}{\alpha}} \log^{1-\frac{1}{\alpha}} \frac{G}{\epsilon}, \tag{3.20}$$

for some $C_1$ depending on the Runge-Kutta method. Similarly,

$$d_2^{n,\epsilon} \leq n + C_1' n^{\frac{1}{\alpha'}} \log^{1-\frac{1}{\alpha'}} \frac{G'}{\epsilon}, \tag{3.21}$$

for $C_1' > 0$ that depends on the Runge-Kutta method. To check this estimate, we plot the dependence of $\Delta_1^{n,\epsilon} = n - d_1^{n,\epsilon}$ and $\Delta_2^{n,\epsilon} = d_2^{n,\epsilon} - n$ on $n$ in Figure 3.2 for different Runge-Kutta methods, with $d_1^{n,\epsilon}$ and $d_2^{n,\epsilon}$ determined numerically.

Our estimates predict that for methods with $p$ odd, namely BDF1 ($p = 1$), 2-stage Radau IIA ($p = 3$) and 3-stage Radau IIA ($p = 5$), $\Delta_1^{n,\epsilon}$, $\Delta_2^{n,\epsilon}$ have to increase as $O\left(n^{\frac{1}{p+1}}\right)$. This is in quite close agreement with the results in Figure 3.2.

For Runge-Kutta methods of even orders obtained estimates predict that for larger $n$ and $d$ the width of a convolution weight gets larger in a non-symmetric manner: $\Delta_1^{n,\epsilon}$ can get larger with increasing $n$ faster than $\Delta_2^{n,\epsilon}$ or vice versa. For larger $n$, $d$ the nonsymmetricity will become more and more visible. This can be illustrated through an example of Lobatto IIIC method of 6th order. Numerical experiments indicate that with increasing $n$ the part of the approximate support of the convolution weight $\mathrm{w}_n(d)$ of Lobatto IIIC method corresponding to $d < n$ increases slower than the part of the approximate support related to $d > n$. This effect can be explained by estimates (3.13) as follows. The stability function of the 4-stage Lobatto IIIC method is the $(2, 4)$-Padé approximation to $\mathrm{e}^z$. For such approximants the sign of the error term $C_{p+1}$ is negative (see, for example, [122, Theorem 3.11]); then the sign of $C_{p+1}(-1)^{\frac{p}{2}}$ is positive. According to the estimates (3.13), $\Delta_1^{n,\epsilon} = O\left(n^{\frac{1}{p+1}}\right)$, with $p = 6$, while $\Delta_2^{n,\epsilon} = O\left(n^{\frac{2s-p}{p+1}}\right)$, with $s = 4$ for Lobatto IIIC (this value can be obtained examining $|R(iy)|^2 - 1 = O(y^{2s})$ for small $y \in \mathbb{R}$). Again, the results in Figure 3.2 are in quite close agreement with these estimates.

**Remark 3.1.18.** *From the proof it can be seen that the effect of the dispersion of convolution weights is due to the term* $|1 + f(z_0)\mathrm{e}^{-z_0}|$ *which was bounded by a constant greater than* 1. *We have not observed in the numerical experiments any case when this term is noticeably smaller than* 1, *which would force convolution weights* $\mathrm{w}_n(d)$ *to decay exponentially with increasing* $n \approx d$.

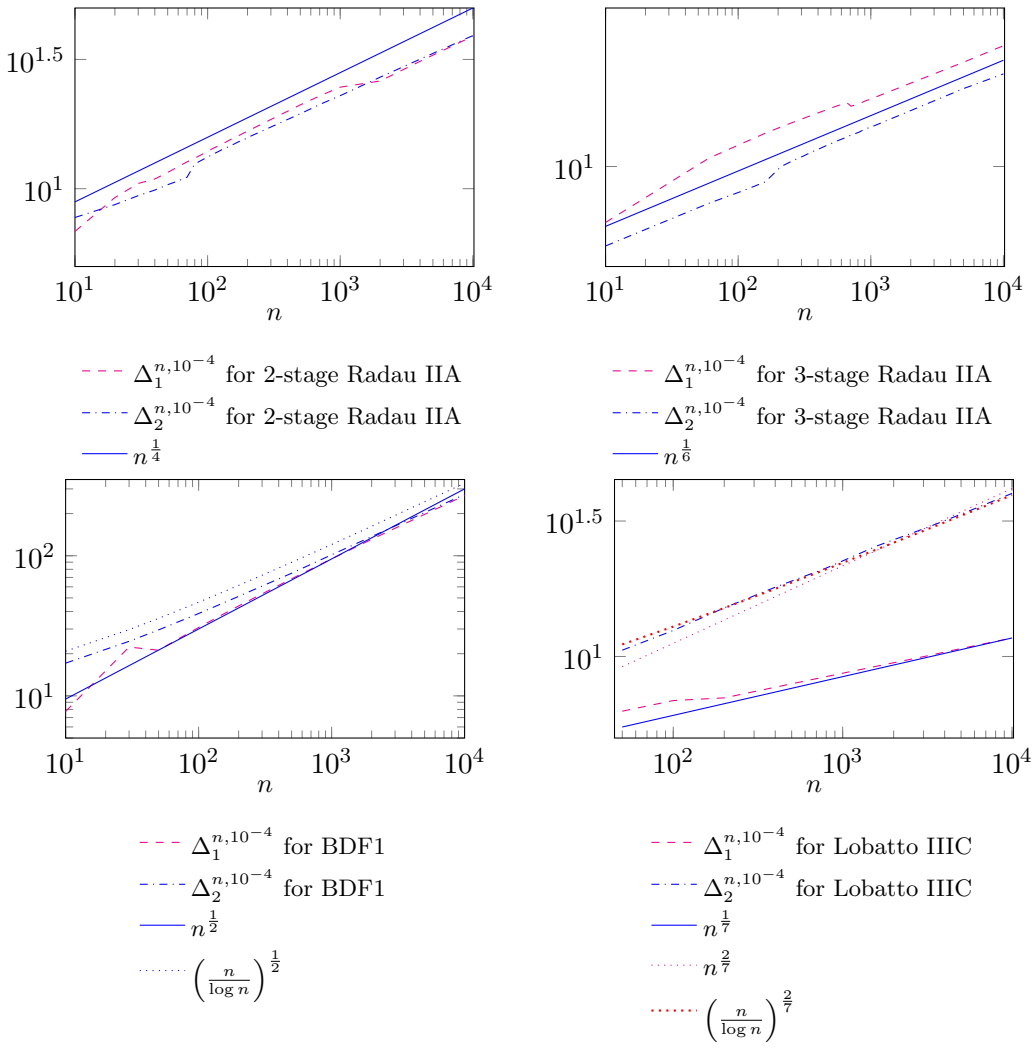


*Figure 3.2: Dependence of $\Delta_1^{n,10^{-4}}$, $\Delta_2^{n,10^{-4}}$ on $n$ for different Runge-Kutta methods.*

### 3.1.2 Efficient Evaluation of Convolution Weights

This section is dedicated to the description of procedures for the efficient evaluation of convolution weights $w_n^h(d)$. First we briefly recall the conventional algorithm to compute convolution weights, see also Section 1.2.9, with the accuracy $\sqrt{\epsilon_m}$, where $\epsilon_m$ is the machine precision.

Let

$$\mathcal{K}_d(\xi) = \frac{\exp\left(-\Delta(\xi)\frac{d}{h}\right)}{4\pi d}.$$

The expansion (1.48) shows that $w_n^h(d)$ is the $n$th Taylor coefficient of $\mathcal{K}_d(\xi)$. Therefore, Cauchy integral formula gives another representation of $w_n^h(d)$,

$$w_n^h(d) = \frac{1}{2\pi i}\oint_{\mathcal{C}} \mathcal{K}_d(\xi)\xi^{-n-1}d\xi.$$

Let us choose the contour $\mathcal{C}$ as the circle of radius $\rho < 1$ centered at the origin. Discretizing

this integral with the composite trapezoid rule gives the approximation

$$w_n^h(d) = \rho^{-n} \sum_{j=0}^{N} \mathcal{K}_d\left(\rho e^{ij\frac{2\pi}{N+1}}\right) e^{-ijn\frac{2\pi}{N+1}} + O(\rho^{N+1}), \qquad n = 0, 1, \ldots, N. \tag{3.22}$$

In practice, the parameter $\rho > 0$ cannot be chosen arbitrarily small in finite precision arithmetic. The best accuracy that can be achieved is $\sqrt{\epsilon}_m$ with the choice $\rho = \epsilon_m^{\frac{1}{2N}}$, see [124,137]. Using FFT, $w_n^h(d)$ can be computed in $O(N \log N)$ time for all $n = 0, 1, \ldots, N$. However, if $d$ is restricted, it is possible to avoid the scaling parameter $\rho$ as described in the next proposition. Recall that

$$w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d}. \tag{3.23}$$

**Proposition 3.1.19.** *Let $w_n^h$, $n \geq 0$, be Runge-Kutta convolution weights (1.48), and let the Runge-Kutta method satisfy Assumption 1.2.15 and (3.11). Additionally, if the order $p$ of the Runge-Kutta method is even and $C_{p+1}(-1)^{\frac{p}{2}} < 0$, let*

$$\frac{p+1}{2s-p} < 1,$$

*where $s$ is defined by (3.4).*

*Let $K, h > 0$ be fixed and $D = Kh$. Then there exist $\mu_1, \mu_2, \mu_3 > 0$, s.t. for all $\epsilon > 0$ and for all $L \in \mathbb{N}$ satisfying*

$$L \geq \mu_1 \log \frac{1}{\epsilon} + \mu_2 K + \mu_3,$$

*the following holds true:*

1. *There exists an $L$-term approximation to the convolution kernel $\mathcal{K}_d(\xi) = \frac{\exp\left(-\Delta(\xi)\frac{d}{h}\right)}{4\pi d}$:*

$$\left| \mathcal{K}_d(\xi) - \sum_{\ell=0}^{L-1} w_\ell^h(d) \xi^\ell \right| \leq \frac{\epsilon}{4\pi d}$$

   *for all $\xi \in \mathbb{C} : \ |\xi| \leq 1$ and $0 \leq d \leq D$.*

2. *Convolution weights can be approximated with the accuracy $\epsilon$ by an $L$-term discrete Fourier transform of the convolution kernel.*

$$\left| w_n^h(d) - \frac{1}{L} \sum_{\ell=0}^{L-1} \mathcal{K}_d(\mathrm{e}^{i\ell\frac{2\pi}{L}}) \mathrm{e}^{-i\ell n\frac{2\pi}{L}} \right| \leq \frac{\epsilon}{4\pi d}$$

   *for all $n < L$ and $0 \leq d \leq D$.*

*Proof.* Let us prove the first statement using the bounds on convolution weights derived in Proposition 3.1.16. The second statement then straightforwardly follows from the first statement by the application of the aliasing formula.

By definition, for all $\xi : |\xi| < 1$

$$K_d(\xi) = \sum_{\ell=0}^{\infty} w_\ell^h(d) \xi^\ell = \sum_{\ell=0}^{L-1} w_\ell^h(d) \xi^\ell + \sum_{\ell=L}^{\infty} w_\ell^h(d) \xi^\ell.$$

Let us show that given $\epsilon > 0$, there exists $L$ s.t. (see also (3.23)):

$$E_L(\xi) = \left\| \sum_{\ell=L}^{\infty} w_\ell^h \xi^\ell \right\| = \left\| \frac{1}{4\pi d} \sum_{\ell=L}^{\infty} \mathrm{w}_\ell \left( \frac{d}{h} \right) \xi^\ell \right\| < \frac{\epsilon}{4\pi d}. \tag{3.24}$$

First, let $L > K$. In a generalized form, the bounds on scaled convolution weights $\mathrm{w}_n\left(\frac{d}{h}\right)$ for $n \geq L$ and $d \leq Kh < nh$ can be stated as

$$\left\| \mathrm{w}_n \left( \frac{d}{h} \right) \right\| \leq G\,(1-\delta)^{n-\frac{d}{h}}\,(1+A\delta^\alpha)^{\frac{d}{h}},$$

for some $0 < \delta < \bar{\delta}$ and $A, G, \alpha, \bar{\delta} > 0$ being constants. Then, after inserting this bound into the expression (3.24) for $E_L(\xi)$,

$$\begin{aligned} E_L(\xi) &= \left\| \frac{1}{4\pi d} \sum_{\ell=L}^{\infty} \mathrm{w}_\ell \left( \frac{d}{h} \right) \xi^\ell \right\| \leq \frac{1}{4\pi d} \sum_{\ell=L}^{\infty} \left\| \mathrm{w}_\ell \left( \frac{d}{h} \right) \right\| \\ &\leq \frac{G}{4\pi d} \left( \frac{1+A\delta^\alpha}{1-\delta} \right)^{\frac{d}{h}} \sum_{\ell=L}^{\infty} (1-\delta)^\ell \\ &\leq \frac{G}{4\pi d} \left( \frac{1+A\delta^\alpha}{1-\delta} \right)^{\frac{d}{h}} (1-\delta)^L \delta^{-1} \\ &\leq \frac{G}{4\pi d\delta} \left( \frac{1+A\delta^\alpha}{1-\delta} \right)^{K} (1-\delta)^L, \end{aligned} \tag{3.25}$$

where we used that $d < D = Kh$. From the above it can be seen that $L$ has to be chosen larger than $K$ and so that

$$\frac{G}{4\pi d\delta} \left( \frac{1+A\delta^\alpha}{1-\delta} \right)^{K} (1-\delta)^L < \frac{\epsilon}{4\pi d}.$$

Namely,

$$\begin{aligned} L \geq K &\left( \log(1+A\delta^\alpha) \log^{-1} \frac{1}{1-\delta} + 1 \right) \\ &+ \log \frac{1}{\epsilon} \log^{-1} \frac{1}{1-\delta} + \log \frac{G}{\delta} \log^{-1} \frac{1}{1-\delta}. \end{aligned}$$

This proves the statement of the proposition for $|\xi| < 1$. For $|\xi| = 1$ the correctness of the statement can be seen from the bound (3.25) which is valid for $|\xi| = 1$. □

### 3.1.3 Bounds for Non-Scaled Convolution Weights

For consistency, in this section we show how to deal with convolution weights $w_n^h(d)$ for small $d > 0$.

The next proposition is a corollary of Proposition 3.1.16 and shows that (non-scaled) convolution weights $w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d}$ also experience exponential decay away from $\frac{d}{h} \approx n$.

**Proposition 3.1.20.** *Let $w_n^h$, $n \geq 0$, be convolution weights for an $m$-stage Runge-Kutta method of order $p$ that satisfies Assumption 1.2.15 and (3.11).*

*Let $s$ be defined by (3.4). Then there exist positive constants $G$, $G'$, $C$, $C'$ and $\bar{\delta} \in (0,\, 1)$, such that for $n \geq 1$ and $0 < \delta < \bar{\delta}$ the following estimates hold:*

1. *p is odd*

$$\begin{aligned} \|w_n^h(d)\| &\leq \frac{G}{h}(1-\delta)^{n-\frac{d}{h}}(1+C\delta^{p+1})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} \leq n, \\ \|w_n^h(d)\| &\leq \frac{G'}{d}(1+\delta)^{n-\frac{d}{h}}(1+C'\delta^{p+1})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} > n; \end{aligned} \tag{3.26}$$

2. *p is even*

   (a) $C_{p+1}(-1)^{\frac{p}{2}} > 0$

$$\begin{aligned} \|w_n^h(d)\| &\leq \frac{G}{h}(1-\delta)^{n-\frac{d}{h}}(1+C\delta^{p+1})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} \leq n, \\ \|w_n^h(d)\| &\leq \frac{G'}{d}(1+\delta)^{n-\frac{d}{h}}(1+C'\delta^{\frac{p+1}{2s-p}})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} > n; \end{aligned} \tag{3.27}$$

   (b) $C_{p+1}(-1)^{\frac{p}{2}} < 0$

$$\begin{aligned} \|w_n^h(d)\| &\leq \frac{G}{h}(1-\delta)^{n-\frac{d}{h}}(1+C\delta^{\frac{p+1}{2s-p}})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} \leq n, \\ \|w_n^h(d)\| &\leq \frac{G'}{d}(1+\delta)^{n-\frac{d}{h}}(1+C'\delta^{p+1})^{\frac{d}{h}} \qquad \text{for} \quad \frac{d}{h} > n. \end{aligned} \tag{3.28}$$

*The convolution weight $w_0^h(d)$ satisfies:*

$$\|w_0^h(d)\| \leq \frac{\exp(-\mu\frac{d}{h})}{4\pi d},$$

*for some $\mu > 0$.*

*Constants $G$, $G'$, $C$, $C'$, $\bar{\delta}$, $\mu$ depend only on the Runge-Kutta method and do not depend on $n$, $d$ and $h$.*

To prove this result, we will need the following technical lemma.

**Lemma 3.1.21.** *Given an $m$-stage Runge-Kutta method satisfying Assumption 1.2.15, the following statements hold true.*

1. *For all $n \geq 1$ a scaled convolution weight $\mathrm{w}_n(d)$ has a zero of the multiplicity $n$ at $d = 0$.*
2. *For all $n \geq 2$ a convolution weight $w_n^h(d)$ has a zero of the multiplicity $n-1$ at $d = 0$.*
3. *Given the stability function of the Runge-Kutta method $R(z)$ and a contour $\gamma$ enclosing all the singularities of $R(z)$, for all $n \geq 1$,*

$$\oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}dz = 0. \tag{3.29}$$

*Proof.* Let us prove the statements successively.

1. The generating function of scaled convolution weights $\mathrm{e}^{-\Delta(\xi)d}$, see (3.1), can be expanded into Taylor series in $\xi$ and then into series in $d$:

$$\exp\left(-\Delta(\xi)d\right) = \sum_{n=0}^{\infty} \mathrm{w}_n(d)\xi^n,$$
$$\exp\left(-\Delta(\xi)d\right) = \sum_{n=0}^{\infty} \frac{(-\Delta(\xi))^n}{n!} d^n.$$

   For Runge-Kutta methods of interest

$$\Delta(\xi) = A^{-1} - \xi A^{-1} b^T \mathbb{1} A^{-1}.$$

   Matching the powers of $\xi$ in both expansions we obtain the following expression for $\mathrm{w}_n(d)$, $n \geq 0$:

$$\mathrm{w}_n(d) = \sum_{m=n}^{\infty} d^m f_m^n\left(A, b\right),$$

   where $f_m^n\left(A, b\right),\ m \geq n,\ n \geq 0$ are matrix-valued functions of $A$ and $b$. From this the first statement of the lemma follows.

2. The second statement immediately follows from the first one, using $w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d}$.

3. The third statement is the corollary of the first one as well. The scaled convolution weights can be written as the following integral (3.2):

$$\mathrm{w}_n(d) = \frac{1}{2\pi i} \oint_{\gamma} R(z)^{n-1}(I - Az)^{-1} \mathbb{1} b^T (I - Az)^{-1} \mathrm{e}^{-zd} dz, \tag{3.30}$$

   where $\gamma$ is a contour that encloses all the eigenvalues of $A^{-1}$ (singularities of $R(z)$) and $n \geq 1$. Then

$$\mathrm{w}_n(0) = \frac{1}{2\pi i} \oint_{\gamma} R(z)^{n-1}(I - Az)^{-1} \mathbb{1} b^T (I - Az)^{-1} dz = 0$$

   for all $n \geq 1$.

□

Now we have the required ingredients to prove Proposition 3.1.20.

*Proof of Proposition 3.1.20.* Let us start with the case $w_0^h(d)$. From the definition of scaled convolution weights it follows that $w_0^h(d) = \frac{\mathrm{w}_0\left(\frac{d}{h}\right)}{4\pi d}$, and the required bound can be obtained from (3.15). Note, however, that the convolution weight $w_0^h(d)$ has a singularity at $d = 0$.

Bounds for the case $\frac{d}{h} > n$ can be obtained straightforwardly from expressions (3.12, 3.13, 3.14) applied to $w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d}$.

The case $\frac{d}{h} \leq n$ has to be treated separately: we cannot directly apply Proposition 3.1.16 for bounding $w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d}$, since for small $d$ this bound would be far from optimal.

Lemma 3.1.21 shows that convolution weights $w_n^h(d)$, $n \geq 1$, have a zero at $d = 0$ of order at least $n - 1$.

We will proceed as follows. First, we will derive a modified representation for convolution weights using Lemma 3.1.21. Next, ideas from the proof of Proposition 3.1.16 will be used to demonstrate that away from $n$ convolution weights $w_n^h(d)$ decay exponentially; more precisely, bounds, similar to that derived for scaled convolution weights, hold also for $w_n^h(d)$.

Let $d \neq 0$. We express $\mathrm{e}^{-z\frac{d}{h}}$ as an integral of a parameter $0 \leq \rho \leq 1$:

$$\mathrm{e}^{-z\frac{d}{h}} = 1 - \frac{zd}{h} \int_0^1 \mathrm{e}^{-z\frac{d}{h}\rho} d\rho.$$

Then the definition (3.2) can be rewritten:

$$\begin{aligned} w_n^h(d) = \frac{\mathrm{w}_n\left(\frac{d}{h}\right)}{4\pi d} &= \frac{1}{8\pi^2 i d} \oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}\mathrm{e}^{-z\frac{d}{h}} dz \\ &= \frac{1}{8\pi^2 i d} \oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1} dz - \\ &- \frac{1}{8\pi^2 i h} \int_0^1 \oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1} z\mathrm{e}^{-z\frac{d}{h}\rho} dz d\rho. \end{aligned}$$

The first term in the above sum equals 0, due to (3.29). The absolute value of the second term, namely,

$$\frac{1}{8\pi^2 i h} \int_0^1 \oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1} z\mathrm{e}^{-z\frac{d}{h}\rho} dz d\rho,$$

can be estimated using the mean value theorem. We first bound the value of the integral

$$I(\rho, d) = \frac{1}{8\pi^2 i h} \oint_\gamma R(z)^{n-1}(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1} z\mathrm{e}^{-z\frac{d}{h}\rho} dz$$

repeating the arguments of the proof of Proposition 3.1.16. Note that two changes have to be made. First, $d$ has to be substituted with $\frac{d}{h}$. And second, instead of bounding $\|(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}\|$ we now bound $\|(I - Az)^{-1}\mathbb{1}b^T(I - Az)^{-1}z\|$, for $z$ lying on a contour $\gamma = \gamma_r$, by a constant that does not depend neither on $d$, nor on $h$ or $n$, but only on the Runge-Kutta method.

It is not difficult to see that there exist positive constants $G$, $C$, $\bar{\delta} \in (0,\ 1)$, $q > 0$ such that for $n \geq 1$ and $0 < \delta < \bar{\delta}$ the following estimate holds:

$$|I(\rho, d)| \leq \frac{1}{h} G(1 - \delta)^{n - \frac{d}{h}\rho}(1 + C\delta^q)^{\frac{d}{h}\rho},$$

for $\frac{d}{h}\rho \leq n$. In the above expression $q$ is either $p + 1$ or $\frac{p+1}{2s-p}$, as in Proposition 3.1.16. Clearly this estimate is valid for all $d : \frac{d}{h} \leq n$ and $\rho \in [0,\ 1]$.

Next, we bound $\int_0^1 I(\rho, d)d\rho$ as:

$$\begin{aligned}\left|\int_0^1 I(\rho,d)d\rho\right| &\le \max_{\rho\in[0,\,1]}|I(\rho,d)|\\ &\le \frac{1}{h}G(1-\delta)^n \max_{\rho\in[0,\,1]}\left(\frac{1+C\delta^q}{1-\delta}\right)^{\frac{d}{h}\rho}\\ &\le \frac{1}{h}G(1-\delta)^{n-\frac{d}{h}}\,(1+C\delta^q)^{\frac{d}{h}}\,.\end{aligned}$$

This finishes the proof of the statement. □

To find the approximate support of convolution weights $w_n^h$ we can argue similarly as in the case of scaled convolution weights $\mathrm{w}_n$. Let

$$\begin{aligned}\operatorname{supp}_\epsilon w_n^h &= \left[d_1^{(n,\epsilon,h)},\ d_2^{(n,\epsilon,h)}\right], \qquad (3.31)\\ d_1^{(n,\epsilon,h)} &= \sup\Big\{d:\ \|w_n^h(d')\|<\epsilon,\ \text{ for all } 0\le d'<d\Big\},\\ d_2^{(n,\epsilon,h)} &= \inf\Big\{d:\ \|w_n^h(d')\|<\epsilon,\ \text{ for all } d'>d\Big\}.\end{aligned}$$

The value $d_1^{(n,\epsilon,h)}$ is defined for $n>1$ and $d_2^{(n,\epsilon,h)}$ for all $n\ge 0$ under the same assumptions as in (3.18). From Proposition 3.1.20 it follows that

$$\|w_n^h(d)\| \le \frac{G}{h}(1-\delta)^{n-\frac{d}{h}}(1+C\delta^\alpha)^{\frac{d}{h}} \qquad \text{for } d\le nh, \qquad (3.32)$$

$$\|w_n^h(d)\| \le \frac{G'}{d}(1+\delta)^{n-\frac{d}{h}}(1+C'\delta^{\alpha'})^{\frac{d}{h}} \qquad \text{for } d>nh, \qquad (3.33)$$

for all $0<\delta<\bar\delta$, for constants $C, C', G, G', \alpha, \alpha', \bar\delta>0$ depending only on the Runge-Kutta method. As before, we assume that $\alpha, \alpha'>1$. The estimate on $d_1^{(n,\epsilon,h)}$ can be found from the first inequality (3.32). Namely, given $\epsilon>0$, $\|w_n^h(d)\|<\epsilon$ for $d$ satisfying, see (3.19),

$$d < nh - h\frac{c_1 n\delta^{\alpha-1} + c_2\delta^{-1}\log\frac{G}{h\epsilon}}{1-C'\delta},$$

for some small $\delta$ and constants $C', c_1, c_2>0$. The choice $\delta = c\left(\frac{\log\frac{G}{\epsilon h}}{n}\right)^{\frac{1}{\alpha}}$, with a small constant $c$, ensures that $\|w_n^h(d)\|<\epsilon$ for all

$$d < nh - C_2 h n^{\frac{1}{\alpha}}\log^{\frac{\alpha-1}{\alpha}}\frac{G}{\epsilon h},$$

for some constant $C_2>0$. The right-hand side of the above expression serves as an estimate for $d_1^{(n,\epsilon,h)}$. Similarly an estimate on $d_2^{(n,\epsilon,h)}$ can be found. The inequality (3.33) holds whenever

$$\|w_n^h(d)\| \le \frac{G'}{nh}(1+\delta)^{n-\frac{d}{h}}(1+C'\delta^{\alpha'})^{\frac{d}{h}} \qquad \text{for all } d>nh.$$

Let us assume that

$$\frac{G}{\epsilon nh} > 1. \tag{3.34}$$

Then, repeating the same steps as before, we obtain the following estimate on $d$ for which (3.33) holds:

$$d > nh + C_2' n^{\frac{1}{\alpha'}} h \log^{\frac{\alpha'-1}{\alpha'}} \frac{G}{\epsilon nh},$$

with some $C_2' > 0$. From this it follows that the approximate support of a convolution weight $w_n^h(d)$, $n > 1$, lies within the interval

$$\operatorname{supp}_\epsilon w_n^h \subseteq \left[ \max\left(0, nh - C_2 h n^{\frac{1}{\alpha}} \log^{\frac{\alpha-1}{\alpha}} \frac{G}{\epsilon h}\right),\ nh + C_2' n^{\frac{1}{\alpha'}} h \log^{\frac{\alpha'-1}{\alpha'}} \frac{G'}{\epsilon nh} \right], \tag{3.35}$$

where $\alpha, \alpha' > 1$.

An efficient evaluation of the convolution weights $w_n^h(d)$ for a restricted range of $d$ with the accuracy close to the machine accuracy can be done as described in the following proposition.

**Proposition 3.1.22.** *Let $w_n^h$, $n \geq 0$, be Runge-Kutta convolution weights (1.48), and let the Runge-Kutta method satisfy Assumption 1.2.15 and (3.11). Additionally, if the order $p$ of the Runge-Kutta method is even and $C_{p+1}(-1)^{\frac{p}{2}} < 0$, let*

$$\frac{p+1}{2s-p} < 1,$$

*where $s$ is defined by (3.4).*

*Let $K, h > 0$ be fixed and $D = Kh$. There exist $\mu_1, \mu_2, \mu_3 > 0$, s.t. for all $\epsilon > 0$ and for all $L \in \mathbb{N}$ satisfying*

$$L \geq \mu_1 \log \frac{1}{\epsilon h} + \mu_2 K + \mu_3,$$

*the following holds true:*

1. *There exists an $L$-term approximation to the convolution kernel $\mathcal{K}_d(\xi) = \frac{\exp\left(-\Delta(\xi)\frac{d}{h}\right)}{4\pi d}$:*

$$\left| \mathcal{K}_d(\xi) - \sum_{\ell=0}^{L-1} w_\ell^h(d) \xi^\ell \right| \leq \epsilon$$

   *for all $\xi \in \mathbb{C} :\ |\xi| \leq 1$ and $0 < d \leq D$.*

2. *Convolution weights can be approximated with the accuracy $\epsilon$ by an $L$-term discrete Fourier transform of the convolution kernel.*

$$\left| w_n^h(d) - \frac{1}{L} \sum_{\ell=0}^{L-1} \mathcal{K}_d(\mathrm{e}^{i\ell\frac{2\pi}{L}}) \mathrm{e}^{-i\ell n\frac{2\pi}{L}} \right| \leq \epsilon$$

   *for all $n < L$ and $0 < d \leq D$.*

*Proof.* The proof mimics the proof of Proposition 3.1.19, but with the use of bounds of Proposition 3.1.20 rather than of Proposition 3.1.16. □

**Remark 3.1.23.** *Note that in the above proposition the dependence of* $L = L(\log \frac{1}{\epsilon h}, \frac{D}{h})$ *on* $\log \frac{1}{h}$ *cannot be removed. To illustrate this fact in Figure 3.3 we plot* $N = \inf\{n \in \mathbb{N} : \left\|w_l^h(d)\right\| < \epsilon,\ \text{for all } l \geq n,\ d \leq D\}$ *for different values of* $h$ *and fixed* $\frac{D}{h} = 10$ *for 3-stage Radau IIA method of the fifth order.*

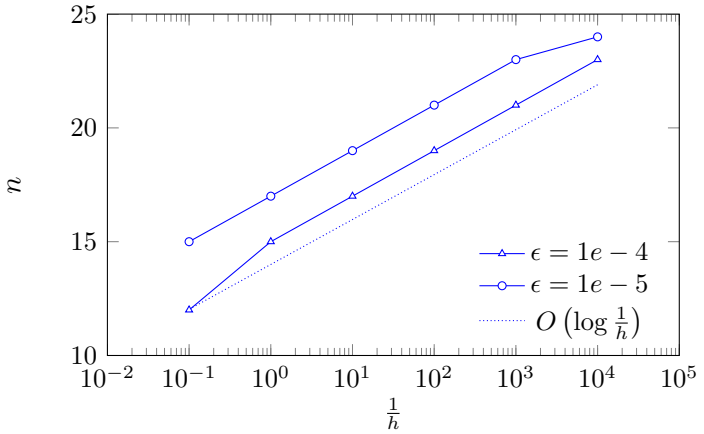


*Figure 3.3: Dependence of* $N = N\left(\log \frac{1}{\epsilon h}, \frac{D}{h}\right)$ *on* $\frac{1}{h}$, $\frac{D}{h} = 10$.

## 3.2 Applicability of Sparse Convolution Quadrature to Runge-Kutta CQ

In [115,116,131] the authors developed an efficient algorithm that exploits fast decay of the convolution weights for the multistep BDF2 method (see also Section 1.2.9). However, a straightforward extension of these ideas to Runge-Kutta convolution quadrature presents difficulties related to the assembly of Galerkin discretizations of operators

$$\left(W_n^h \phi\right)(x) = \int_\Gamma w_n^h(\|x-y\|)\phi(y)d\Gamma_y, \qquad n \geq 1,$$

as well as to the use of back substitution. Let us list some of those:

1. For a Runge-Kutta method with the stage number $m > 1$ no closed form expression is known for a convolution weight $w_n^h(d)$, $n > 0$;

2. Convolution weights $w_n^h(d)$, $n \geq 0$, are real matrices of size $m \times m$ (the real-valuedness can be seen noting that $\Delta(\xi) = \overline{\Delta(\bar{\xi})}$, $\xi \in \mathbb{C}$), hence, given $n > 0$, the construction of the Galerkin discretization of $W_n^h$ requires the assembly of $m^2$ matrices (see Section 3.1.2); the evaluation of $W_n^h \boldsymbol{\lambda}_k$ requires $m^2$ Galerkin matrix-vector multiplications;

3. The solution of the lower-triangular Toeplitz system (1.54), namely,

$$\begin{pmatrix} W_0^h & 0 & \cdots & 0 \\ W_1^h & W_0^h & \cdots & 0 \\ \cdots & & & \\ W_N^h & W_{N-1}^h & \cdots & W_0^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{g}_0 \\ \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_N \end{pmatrix} \tag{3.36}$$

with the help of back substitution is of the complexity $O(N^2)$.

In Sections 1.2.9, 3.1.2 it is shown that the actual computation of convolution weights $w_n^h(\|x-y\|)$ can be done using the fast Fourier transform, however, such procedure may be rather difficult to implement efficiently. From the proof of Lemma 3.1.21 it can be seen that $w_n^h(d),\ n \geq 1$, are analytic functions of $d > 0$, hence can be efficiently interpolated in $d$ by polynomials.

To check this, we perform the following numerical experiment.

Given $\epsilon > 0$, we compute degrees of interpolation polynomials for convolution weights $\left(w_n^h(d)\right)_{11}$ on the smallest interval $\left[d_1^{(n,h)},\ d_2^{(n,h)}\right]$ s.t. $|\left(w_n^h(d)\right)_{11}| < \epsilon$, for all $d < d_1^{(n,h)}$ and $d > d_2^{(n,h)}$. Polynomials are Chebyshev interpolants constructed with the help of Chebfun [180], with the tolerance set to $\epsilon$. Remarkably, if $n, h$ change so that $nh = \text{const}$, the degree of the corresponding interpolation polynomial remains almost constant, see Tables 3.1, 3.2.

| $n$ | $h$ | BDF1 | BDF2 | Radau IIA 2-stage | Radau IIA 3-stage |
|---|---|---|---|---|---|
| 10 | 0.05 | 26 | 36 | 38 | 48 |
| 20 | 0.025 | 27 | 37 | 38 | 48 |
| 50 | 0.01 | 25 | 38 | 36 | 46 |
| 100 | 0.005 | 24 | 42 | 36 | 47 |
| 200 | 0.0025 | 25 | 44 | 36 | 47 |
| 500 | 0.001 | 24 | 50 | 34 | 45 |

*Table 3.1: The degree of the interpolation polynomial for the convolution weight $\left(w_n^h(d)\right)_{11}$ on the interval $\left[d_1^{(n,h)},\ d_2^{(n,h)}\right]$ where $|\left(w_n^h(d)\right)_{11}| < \epsilon$, for all $d < d_1^{(n,h)}$ and $d > d_2^{(n,h)}$, $\epsilon = 10^{-6}$. The polynomial was constructed with a help of Chebfun [180], with the tolerance set to $\epsilon = 10^{-6}$. The order of the polynomial is almost constant for $nh = \text{const}$.*

The use of interpolation polynomials of as high degrees as shown in Tables 3.1 and 3.2 can appear inefficient. Hence intervals $\left[d_1^{(n,h)},\ d_2^{(n,h)}\right]$ have to be split into smaller subintervals, and the actual evaluation of convolution weights $w_n^h(d)$ has to be done based on the interpolation on these subintervals.

Next, we would like to show how the ideas of sparse BDF2 multistep convolution quadrature can be extended to construct the Galerkin discretizations of operators $W_n^h$. We consider the a priori cutoff strategy based on fast decay of the kernels of these operators (convolution weights) used in [115, 116, 131].

| $n$ | $h$ | BDF1 | BDF2 | Radau IIA 2-stage | Radau IIA 3-stage |
|---|---|---|---|---|---|
| 10 | 0.05 | 9 | 14 | 14 | 18 |
| 20 | 0.025 | 11 | 17 | 14 | 16 |
| 50 | 0.01 | 10 | 16 | 13 | 16 |
| 100 | 0.005 | 9 | 16 | 11 | 14 |
| 200 | 0.0025 | 9 | 17 | 10 | 14 |
| 500 | 0.001 | 7 | 17 | 12 | 14 |

*Table 3.2: The degree of the interpolation polynomial for the convolution weight $\left(w_n^h(d)\right)_{11}$ on the interval $\left[d_1^{(n,h)},\ d_2^{(n,h)}\right]$ where $|\left(w_n^h(d)\right)_{11}| < \epsilon$, for all $d < d_1^{(n,h)}$ and $d > d_2^{(n,h)}$, $\epsilon = 10^{-3}$. The polynomial was constructed with a help of Chebfun [180], with the tolerance set to $\epsilon = 10^{-3}$. The order of the polynomial remains almost constant for $nh = \text{const}$.*

Let $(\phi_i)_{i=1}^M$ be Galerkin test and trial basis functions; as previously, for simplicity, we use piecewise-constant functions. Let

$$\left(\mathbf{W}_n^h\right)_{ij} = \iint\limits_{\Gamma\times\Gamma} w_n^h(\|x-y\|)\phi_i(x)\phi_j(y)d\Gamma_y d\Gamma_x,$$
$$\left(\mathbf{W}_n^h\right)_{ij}^{kl} = \iint\limits_{\Gamma\times\Gamma} \left(w_n^h(\|x-y\|)\right)_{kl}\phi_i(x)\phi_j(y)d\Gamma_y d\Gamma_x,$$
$$i,j = 1,\dots,M, \qquad k,l = 1,\dots,m.$$

By $\Delta x$ we denote the meshwidth. The diameters $d_j$ of the supports of Galerkin basis functions $\phi_j,\ j = 1,\dots,M$, satisfy

$$b\Delta x \le d_j \le \Delta x,$$

for some $b > 0$ independent of $M,\ \Delta x$. Let the number of elements $M = O\left(\frac{1}{(\Delta x)^2}\right)$. We assume

$$\Delta x \approx Ch^\nu \tag{3.37}$$

for some $C > 0$, $\nu \ge 1$. This implies that with respect to the number of time steps $N$,

$$M = O(N^{2\nu}).$$

The case $\nu = 1$ is of particular interest for us, see the discussion in Section 1.2.11.3.

The estimates in Section 3.1.1 show that there exist $C_2, C_2', G, G' > 0$ and $0 < \alpha, \beta < 1$, s.t. for all $\epsilon, h > 0$

$$\left\|w_n^h(d)\right\| < \frac{\epsilon}{4\pi d}$$

as soon as, see (3.18, 3.20, 3.21),

$$d \notin \mathcal{I}_n^h = \left[\max\left(0, nh - C_2 n^\alpha h\log^{1-\alpha}\frac{G}{\epsilon}\right),\ nh + C_2' n^\beta h\log^{1-\beta}\frac{G'}{\epsilon}\right]. \tag{3.38}$$

There exists $n_0 = C \log \frac{G}{\epsilon}$, where $C = \text{const} > 0$ that depends on the Runge-Kutta method, such that for all $n > n_0$ it holds

$$\max\left(0, nh - C_2 n^\alpha h \log^{1-\alpha} \frac{G}{\epsilon}\right) = nh - C_2 n^\alpha h \log^{1-\alpha} \frac{G}{\epsilon}. \tag{3.39}$$

The main idea behind the a priori cutoff strategy is to avoid the evaluation of some of the entries of sparse matrices $\left(\mathbf{W}_n^h\right)^{kl},\ k,l = 1,\dots,m,\ n = 0,\dots,N$, that are close to zero because of (3.38).

Let us estimate the number of non-zero elements in $\left(\mathbf{W}_n^h\right)^{kl},\ k,l = 1,\dots,m, 0 \le n \le N$. We follow the strategy suggested in [131, Section 5.3.3] (there the bound on the number of panel clustering influence matrices is given; setting $b = 0$ in the equation (5.14) in [131] lets obtaining the desired estimate on the number of non-zero elements in a sparse matrix).

Let

$$I_n^h = \left\{(x,y) \in \Gamma \times \Gamma :\ \|x - y\| \in \mathcal{I}_n^h\right\}, \tag{3.40}$$

see (3.38).

In [116] it is shown (for the BDF2 method) that the stability and convergence of the method with the cut-off strategy is ensured if $\epsilon$ is chosen s.t.

$$\log \frac{1}{\epsilon} = O(\log M). \tag{3.41}$$

We make use of this estimate as well. Due to rapid decay of convolution weights, taking a slightly larger interval $\tilde{\mathcal{I}}_n^h \supset \mathcal{I}_n^h$ instead of $\mathcal{I}_n^h$ in (3.40) drastically reduces the cutoff error.

As $h, \Delta x \to 0$ preserving (3.37), the diameter of the approximate support (3.38) of a convolution weight $\mathrm{w}_n\left(\frac{d}{h}\right),\ n \ge 1$, exceeds $\Delta x$. The number of non-zero elements in each of the matrices $\left(\mathbf{W}_n^h\right)^{kl},\ k,l = 1,\dots,m,\ n = 1,\dots,N$, can be estimated by the formula:

$$\mathcal{C}_n = O\left(\frac{\mu\left(I_n^h\right)}{\mu_0}\right), \tag{3.42}$$

where, given a set $A$, $\mu(A)$ is its Lebesgue measure, and $\mu_0$ is the measure of the Cartesian product of two supports of basis functions on $\Gamma$. In our case,

$$\mu_0 = \mu(\pi_i \times \pi_j) = O\left((\Delta x)^4\right), \qquad i,j = 1,\dots,M.$$

Let us consider separately the following cases: $n = 0$, $0 < n \le n_0$ and $n_0 < n \le N$.

The convolution weight

$$\|w_0^h(d)\| < \frac{\epsilon}{4\pi d} \qquad \text{for } d \in \left[0,\ C_0 h \log \frac{1}{\epsilon}\right], \qquad C_0 > 0,$$

see (3.15).

To estimate the number of non-zero elements we make use of the formula (3.42). Similarly to [131], the assumption on the geometric shape of the surface is the following (one could think of the simplest case of a flat surface):

$$\mu(I_0^h) = O\left(h^2 \log^2 \frac{1}{\epsilon}\right).$$

Inserting this into (3.42) gives

$$\mathcal{C}_0 = O\left(\frac{h^2}{(\Delta x)^2} M \log^2 \frac{1}{\epsilon}\right) = O\left(M^{2-\frac{1}{\nu}} \log^2 \frac{1}{\epsilon}\right).$$

For $\nu = 1$, with the use of (3.41),

$$\mathcal{C}_0 = O\left(M \log^2 M\right).$$

Next, let us consider the case $0 < n \leq n_0 = C \log N$, see (3.39). The approximate support of $w_n^h$ lies within the interval, c.f. (3.38),

$$\left(0,\ nh + C_2' n^\beta h \log^{1-\beta} \frac{G'}{\epsilon}\right].$$

Adopting the assumption of [131] on the geometric shape of the surface:

$$\mu\left(I_n^h\right) = O\left(\left(nh + C_2' n^\beta h \log^{1-\beta} \frac{G'}{\epsilon}\right)^2\right).$$

For large values of $n$ (close to $C \log N$), this, combined with (3.41), gives

$$\mu\left(I_n^h\right) = O\left(h^2 \log^2 M\right).$$

Inserting this into (3.42) gives an upper bound on the number of elements in each of the matrices $\left(\mathbf{W}_n^h\right)^{kl}$, $k, l = 1, \ldots, m$, $1 \leq n \leq n_0$:

$$\mathcal{C}_n = O\left(M^{2-\frac{1}{\nu}} \log^2 M\right).$$

For the case $\nu = 1$, the storage costs scale not worse than

$$\mathcal{C}_n = O\left(M \log^2 M\right).$$

Finally, let us consider the case $n > n_0 = C \log N$. Similarly to [131], the geometric shape of the surface satisfies (one could again think of the simplest case of a flat surface):

$$\mu\left(I_n^h\right) = O\left(h^2 \left(n^{1+\alpha} \log^{1-\alpha} \frac{G}{\epsilon} + n^{1+\beta} \log^{1-\beta} \frac{G'}{\epsilon}\right)\right). \tag{3.43}$$

Then the total number of non-zero elements in a matrix $\left(\mathbf{W}_n^h\right)^{kl}$, $k, l = 1, \ldots, m$, $n = n_0 + 1, \ldots, N$, scales as

$$\mathcal{C}_n = O\left(M^{2-\frac{1}{\nu}} \left(n^{1+\alpha} \log^{1-\alpha} M + n^{1+\beta} \log^{1-\beta} M\right)\right) \tag{3.44}$$

$$= O\left(M^{2-\frac{1}{\nu}} n^{1+\max(\alpha,\beta)} \log^{1-\max(\alpha,\beta)} M\right). \tag{3.45}$$

For large $n = O(N)$ this gives

$$\mathcal{C}_n = O\left(M^{2-\frac{1}{\nu}} N^{1+\max(\alpha,\beta)} \log^{1-\max(\alpha,\beta)} M\right).$$

For the 3-stage Radau IIA method $\alpha = \beta = \frac{1}{6}$, see the discussion in Section 3.1. For $\nu = 1$ we need the storage of $O\left(n^{1+\frac{1}{6}} M \log^{\frac{5}{6}} M\right)$ for every of the matrices $\left(\mathbf{W}_n^h\right)^{kl}$, $k, l = 1, \ldots, m$, with $n \geq n_0$. For the rest of the matrices the storage costs scale as $O(M \log^2 M)$. The total storage costs then are

$$O\left(N^{2+\frac{1}{6}} M \log^{\frac{5}{6}} M\right) = O\left(M^{2+\frac{1}{12}} \log^{\frac{5}{6}} M\right).$$

Hence, a straightforward application of sparsity does not allow to construct the algorithm of a fully linear complexity. Nonetheless, the memory requirements are very close to those of many MOT and Galerkin methods ($O(M^2)$).

As a remedy, in [131] it was suggested to combine the a priori cutoff strategy with the panel clustering. Although the extension of these ideas to Runge-Kutta CQ may lead to the method with improved memory requirements, the total complexity of such algorithm does not scale better than $O(MN^2)$, due to the use of back substitution. This, combined with difficulties associated with the actual evaluation of convolution weights (see the beginning of Section 3.2), is not likely to allow the construction of the efficient large scale Runge-Kutta convolution quadrature algorithm based on the ideas from [115, 116, 131].

## 3.3 Fast Convolution Quadrature Algorithm

Let us come back to the recursive algorithm described in detail in Section 1.2.11. Recall that for this algorithm $O(N)$ Galerkin discretizations of boundary single-layer operators for the Helmholtz equation with decay need to be constructed. A straightforward application of the data-sparse techniques we considered in Section 2 (i.e. FMM and $\mathcal{H}$-matrices) would on its own lead to the algorithm of almost linear complexity. However, a significant drawback of this approach is large constants involved in complexity estimates. Our goal is to design a method that would reduce them.

The data-sparse techniques in question have two main bottlenecks:

- costly evaluation of singular and nearly singular integrals in the near-field;
- high matrix-vector multiplication costs of the high-frequency FMM.

We overcome the first problem by the use of fast decay of convolution weights $w_n^h(d)$ away from the neighborhood of $d \approx nh$. We show that within the whole recursive algorithm only a few matrices (namely $O(\log N)$) with the near-field need to be constructed, while for the rest we can assemble the far-field only. To motivate this strategy, we briefly survey the related works in Section 3.3.1.

In the end of this section we demonstrate that provided that for the approximation of different matrices a choice between $\mathcal{H}$-matrix techniques and the HF FMM is made properly, the problem of high matrix-vector multiplication costs of the HF FMM ceases to exist.

### 3.3.1 Motivation

The evaluation of the near-field integrals is commonly done with the help of coordinate transformation techniques, see [87, 118, 167–169]. Given the kernel $k(x, y)$ of a boundary

single layer operator (that maps from $H^{-\frac{1}{2}}(\Gamma)$ to $H^{\frac{1}{2}}(\Gamma)$), the evaluation of

$$\iint\limits_{\pi_i \times \pi_j} k(x,y)\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y, \qquad \operatorname{supp}\phi_i = \pi_i,\ \operatorname{supp}\phi_j = \pi_j,$$

with the accuracy sufficient to preserve the stability and convergence of the Galerkin method, requires that the quadrature order scales as $O(\log^3 M)$ if $\operatorname{dist}(\pi_i, \pi_j) = 0$, $O(\log^4 M)$ if $\operatorname{dist}(\pi_i, \pi_j) = O(\Delta x)$ (nearly singular integrals) and $O(1)$ if $\operatorname{dist}(\pi_i, \pi_j) = 1$. Thus the computation of the near-field (singular and nearly singular integrals) of one matrix is of $O(M \log^4 M)$ complexity. Within the recursive convolution quadrature algorithm $O(N)$ such matrices need to be assembled, hence resulting in the total complexity $O\left(NM \log^4 M\right)$.

The question of the efficient evaluation of singular and nearly singular integrals was addressed in recent works [16, 146, 147]. Particularly, in [16] such integrals were represented as functions of multiple parameters and efficiently computed using interpolation and tensor decomposition techniques. In [15] the effect of the application of such techniques on the full $\mathcal{H}$-matrix assembly time was numerically studied. For the Laplace boundary single layer operator on various geometries it was demonstrated that the 50%-70% reduction of the time required for the evaluation of the nearly singular and weakly singular integrals results in the 10%-20% reduction of the total $\mathcal{H}$-matrix assembly time. Given the bound on the ranks of $\mathcal{H}$-matrix $r$, the rest of the time is spent for the evaluation of $O(rM \log M)$ far-field integrals within the ACA+ procedure of the $\mathcal{H}$-matrix construction. If the evaluation of the far-field is done in a more efficient manner, the gain can be significantly larger. And this is the case for the fast multipole methods.

The precomputation time (i.e. time needed for the construction of the translation operators) for the HF FMM scales as $O(M \log M)$ (assuming $M = O(|\kappa|^2)$ for the wavenumber $\kappa = is$) and the constants involved are significantly smaller than that for the $\mathcal{H}$-matrix assembly. This can be seen in the experiments of [48], where the HF FMM precomputation times were reported to be in practice 9-20 times smaller than that for the $\mathcal{H}$-matrix construction. This can be also observed in the numerical experiments in Section 2.3. In [89, Tables 3.2-3.3] the time to compute the near-field for the HF FMM accelerated Burton-Miller formulation is compared to the time needed to construct the corresponding HF FMM translation matrices. The results show that for BEM discretizations with $10^3 - 10^5$ triangular boundary elements the computation of the near-field is typically order of magnitude slower than the assembly of translation matrices.

However the actual constants depend much on the implementation and the desired accuracy. Nevertheless, for large problems we should be able to see the improvement if we skip constructing the near-field. Asymptotic complexity estimates are improved as well. Indeed, while the application of ACA/ACA+ based $\mathcal{H}$-matrix techniques requires the evaluation of 4-dimensional integrals, for the use of the HF FMM in the far-field only the evaluation of two-dimensional integrals (for the cluster basis) is needed. We perform this step not during the precomputation stage, but when compute matrix-vector products (this allows to avoid storing the cluster basis for all matrices and thus improves memory costs). Therefore the relative improvement in the precomputation time if the near-field is not constructed is even more drastic.

Since in the course of the recursive algorithm described in Section 1.2.11 the matrix-vector multiplication with the same matrix block is performed multiple times, it makes sense to precompute the corresponding discretizations of boundary integral operators and keep them in memory, rather than recompute them every time the matrix-vector multiplication is

needed. For the matrices that are approximated with the help of the fast multipole method the near-field and translation operators can be stored. If only a small part of matrices has the near-field, the storage costs needed for HF FMM approximated matrices can be affected as well. Given the HF FMM approximation of $\mathbf{V}(s)$, the storage for its far-field part (translation matrices of the FMM) scales as

$$S_{ff}(s) = O(|s|^2 \log M) = O(M \log M),$$

where we assumed $M = O\left(|s|^2\right)$, while for the near-field

$$S_{nf}(s) = O(M).$$

Hence, as $M \to +\infty$, $S_{nf}$ is smaller than $S_{ff}$ (though only by a logarithmic term). The improvement in the storage costs can be achieved only in the case when the constants in $S_{ff}$ are so small that even for rather large $M$, $S_{nf} > S_{ff}$. As our numerical experiments in Section 4 show, in practice this is often the case.

The presence of decay, i.e. in the case when $s = s_1 + is_2$, $s_1 > 0$, facilitates the reduction of storage costs. If $s_1$ is large enough, for such discretizations $\mathbf{V}(s)$ the far-field part

$$S_{ff} \approx 0,$$

see also Figure 3.4 and Section 2.1.4.1.

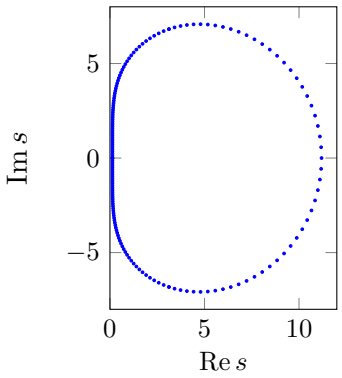


*Figure 3.4: Frequencies $s$ for which we need to construct discretizations of boundary single-layer operators $\boldsymbol{V}(\frac{s}{h})$; they are computed as eigenvalues of $\Delta(\xi)$, $|\xi| = 10^{-\frac{6}{105}}$. Here $h = 1$. While many frequencies are located close to the imaginary axis, a significant part of frequencies has $\operatorname{Re} s \gg 1$ (high-decay case). A large part of the far-field of the corresponding matrices $\boldsymbol{V}(\frac{s}{h})$ is negligibly small and can be a priori ignored when constructing these matrices, as described in Section 2.1.4.1.*

### 3.3.2 Near-Field Reuse

#### 3.3.2.1 Auxiliary Relations on Leaves of a Block-Cluster Tree

Before describing our strategy for dealing with the near-field, we introduce two auxiliary relations defined on leaves of a block-cluster tree, namely the near-field $d$-admissibility and the far-field $d$-admissibility. Recall that given a cluster $\tau$, the center of its bounding box we denote by $c_\tau$ and the diameter of the bounding box by $d_\tau$, see also Section 2.1.2.

**Definition 3.3.1.** *Given $d > 0$, we will call a leaf $(\tau, \sigma)$ near-field $d$-admissible if $\|c_\tau - c_\sigma\| < d - \frac{d_\tau}{2} - \frac{d_\sigma}{2}$.*

**Definition 3.3.2.** *Given $D > 0$, a leaf $(\tau, \sigma)$ is far-field $D$-admissible if $\|c_\tau - c_\sigma\| < D + \frac{d_\tau}{2} + \frac{d_\sigma}{2}$.*

**Remark 3.3.3.** *The following properties hold:*

1. *If $(\tau, \sigma)$ is near-field $d$-admissible then $(\forall x \in \Omega_\tau)(\forall y \in \Omega_\sigma),\ \|x - y\| < d$.*
2. *If $(\tau, \sigma)$ is not far-field $D$-admissible then $(\forall x \in \Omega_\tau)(\forall y \in \Omega_\sigma),\ \|x - y\| > D$.*

We will denote the set of near-field $d$-admissible leaves of a block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ by $\mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ and the set of far-field $D$-admissible leaves by $\mathcal{L}_D^+(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$.

**Remark 3.3.4.** *The following observation is crucial for our algorithm. Recall that $\mathcal{L}_-(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ is defined as the set of all non-admissible block-clusters of the block-cluster tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$. Then it is possible to choose $d$ s.t.*

$$\mathcal{L}_-(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}) \subset \mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}). \tag{3.46}$$

*This follows from the definition of the admissibility condition (Definition 2.1.7). Namely, given $\eta > 1$, non-admissible leaves $(\tau, \sigma)$ satisfy*

$$\|c_\tau - c_\sigma\| < \frac{\eta}{2}(d_\tau + d_\sigma),$$

*where $c_\tau$, $c_\sigma$ are the centers of bounding boxes of $\tau$, $\sigma$ and $d_\tau$, $d_\sigma$ are their diameters. The choice*

$$d = \gamma \sup_{(\tau,\sigma)\in\mathcal{L}_-} (d_\tau + d_\sigma), \qquad \gamma \geq \frac{\eta + 1}{2} \tag{3.47}$$

*ensures that (3.46) holds true.*

Now we have all the ingredients needed to describe fast Runge-Kutta convolution quadrature.

#### 3.3.2.2 Main Ideas and Algorithmic Realization

Consider the matrix-vector product (1.58), namely

$$\begin{pmatrix} \boldsymbol{h}_0 \\ \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_{n-\ell} \end{pmatrix} = \begin{pmatrix} W_\ell^h & W_{\ell-1}^h & \cdots & W_1^h \\ W_{\ell+1}^h & W_\ell^h & \cdots & W_2^h \\ \vdots & & & \\ W_n^h & W_{n-1}^h & \cdots & W_{n-\ell+1}^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_0 \\ \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{\ell-1} \end{pmatrix}. \tag{3.48}$$

After the discretization in space with the help of the Galerkin method (with trial and test basis functions $(\phi_j(x))_{j=1}^M$), the above system of equations can be rewritten as:

$$\begin{pmatrix} \boldsymbol{h}_0^j \\ \boldsymbol{h}_1^j \\ \vdots \\ \boldsymbol{h}_{n-\ell}^j \end{pmatrix} = \iint_{\Gamma\times\Gamma} \sum_{k=1}^{M} T^{\ell,n}(\|x - y\|) \begin{pmatrix} \boldsymbol{\lambda}_0^k \\ \boldsymbol{\lambda}_1^k \\ \vdots \\ \boldsymbol{\lambda}_{\ell-1}^k \end{pmatrix} \phi_k(y)\phi_j(x) d\Gamma_x d\Gamma_y, \qquad j = 1, \ldots, M, \tag{3.49}$$

where

$$\boldsymbol{h}_k^j = \int_\Gamma \boldsymbol{h}_k(x)\phi_j(x)d\Gamma_x, \qquad k = 0,\dots,n-\ell,\ j = 1,\dots,M,$$
$$\boldsymbol{\lambda}_k^j = \int_\Gamma \boldsymbol{\lambda}_k(x)\phi_j(x)d\Gamma_x, \qquad k = 0,\dots,\ell-1,\ j = 1,\dots,M,$$

and $T^{\ell,n}$ is the kernel function

$$T^{\ell,n}(\|x-y\|) = \begin{pmatrix} w_\ell^h(\|x-y\|) & \cdots & w_1^h(\|x-y\|) \\ w_{\ell+1}^h(\|x-y\|) & \cdots & w_2^h(\|x-y\|) \\ \vdots & & \\ w_n^h(\|x-y\|) & \cdots & w_{n-\ell+1}^h(\|x-y\|) \end{pmatrix}. \tag{3.50}$$

Let $d$ be chosen as in (3.47). The double integral in (3.49) can be split into a sum of two double integrals: one over the leaves of the block-cluster tree belonging to the set $\mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$ and the other being the remainder. Namely,

$$\iint_{\Gamma\times\Gamma} T^{\ell,n}(\|x-y\|)\phi_j(x)\phi_k(y)d\Gamma_x d\Gamma_y = \boldsymbol{N}_{jk} + \boldsymbol{F}_{jk}, \tag{3.51}$$

$$\boldsymbol{N}_{jk} = \iint_{\Omega_\sigma\times\Omega_\tau,\ (\sigma,\tau)\in\mathcal{L}_d} T^{\ell,n}(\|x-y\|)\phi_j(x)\phi_k(y)d\Gamma_x d\Gamma_y, \tag{3.52}$$

$$\boldsymbol{F}_{jk} = \iint_{\Omega_\sigma\times\Omega_\tau,\ (\sigma,\tau)\in\mathcal{L}_F} T^{\ell,n}(\|x-y\|)\phi_j(x)\phi_k(y)d\Gamma_x d\Gamma_y, \qquad j,k = 1,\dots,M,$$

where $\mathcal{L}_d = \mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$, $\mathcal{L}_F = \mathcal{L}(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}) \setminus \mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$. In this case $\boldsymbol{\mathcal{F}} = (\boldsymbol{F}_{jk})_{k,j=1}^M$ does not contain the near-field, since all non-admissible block-clusters belong to $\boldsymbol{\mathcal{N}} = (\boldsymbol{N}_{jk})_{k,j=1}^M$. Since $w_j^h$ are matrix-valued functions, $\boldsymbol{N}_{jk}$ and $\boldsymbol{F}_{jk}$ are tensors.

First, we demonstrate why such splitting may improve storage and computational costs. The bounds in Proposition 3.1.16 show that, for any given $\epsilon > 0$, there exists $L$,

$$\left\| w_j^h(\tilde{d}) \right\| < \frac{\epsilon}{4\pi\tilde{d}}, \qquad \text{for all} \quad j \geq L \quad \text{and} \quad \tilde{d} < d. \tag{3.53}$$

Let

$$\boldsymbol{\Omega}_d = \bigcup_{(\sigma,\tau)\in\mathcal{L}_d} \Omega_\sigma \times \Omega_\tau.$$

We assume w.l.o.g. that

$$L < \min(\ell, n-\ell+1). \tag{3.54}$$

Then some of the elements of the tensor $\boldsymbol{\mathcal{N}}$ are approximately equal to zero. Let us show

this. For $k \geq L$,

$$\left| \iint\limits_{\mathbf{\Omega}_d} w_k^h(\|x-y\|)\phi_j(x)\phi_i(y)d\Gamma_x d\Gamma_y \right| < \epsilon \left| \iint\limits_{\mathbf{\Omega}_d} \frac{|\phi_j(x)\phi_i(y)|}{4\pi\|x-y\|} d\Gamma_x d\Gamma_y \right|$$
$$\leq \epsilon \iint\limits_{\Gamma\times\Gamma} \frac{|\phi_j(x)\phi_i(y)|}{4\pi\|x-y\|} d\Gamma_x d\Gamma_y, \ i,j = 1,\dots M.$$

Recall that the boundary single-layer operator for the Laplacian is continuous from $L_2(\Gamma) \to L_2(\Gamma)$, see e.g. [29, 116, 169]. Hence, for some $C > 0$ that depends only on $\Gamma$ it holds:

$$\left| \iint\limits_{\mathbf{\Omega}_d} w_k^h(\|x-y\|)\phi_j(x)\phi_i(y)d\Gamma_x d\Gamma_y \right| \leq C\epsilon \|\phi_i\|_{L_2(\Gamma)} \|\phi_j\|_{L_2(\Gamma)}$$
$$= C\epsilon\mu_i\mu_j, \qquad i,j = 1,\dots,M, \tag{3.55}$$

where

$$\mu_i = \text{meas}(\text{supp}(\phi_i)), \qquad i,j = 1,\dots M.$$

Then $\epsilon$ can always be chosen so that up to a desired precision $\boldsymbol{\mathcal{N}}$ can be rewritten as

$$\boldsymbol{N}_{jk} \approx \iint\limits_{\mathbf{\Omega}_d} T_L^{\ell,n}(\|x-y\|)\phi_k(y)\phi_j(x)d\Gamma_x d\Gamma_y, \qquad k,j = 1,\dots,M, \tag{3.56}$$

where

$$T_L^{\ell,n}(\|x-y\|) = \begin{pmatrix} 0 & \cdots & w_{L-1}^h(\|x-y\|) & \cdots & w_2^h(\|x-y\|) & w_1^h(\|x-y\|) \\ 0 & \cdots & 0 & \cdots & w_3^h(\|x-y\|) & w_2^h(\|x-y\|) \\ \vdots & & & & & \\ 0 & \cdots & 0 & \cdots & w_{L-1}^h(\|x-y\|) & w_{L-2}^h(\|x-y\|) \\ 0 & \cdots & 0 & \cdots & 0 & w_{L-1}^h(\|x-y\|) \\ \vdots & & & & & \\ 0 & \cdots & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{3.57}$$

Hence, to approximate completely the near-field part of the matrix of the system (3.48), only $O(L)$ Galerkin matrices

$$\left(\widetilde{\boldsymbol{W}_\nu^h}\right)_{kj} = \iint\limits_{\mathbf{\Omega}_d} w_\nu^h(\|x-y\|)\phi_k(y)\phi_j(x)d\Gamma_x d\Gamma_y, \qquad k,j = 1,\dots,M, \ \nu = 1,\dots,L-1,$$

need to be constructed. In practice we do not assemble these matrices, but rather evaluate the matrix-vector product with $\boldsymbol{\mathcal{N}}$ with the help of either of two procedures we present below. Before describing these procedures, we would like to show that

$$L = O\left(\log\frac{1}{\epsilon}\right)$$

and does not depend on the size of the system (3.48). Recall that the diameter of non-admissible clusters

$$d_\tau = O\left(\Delta x\right),$$

where $\Delta x$ is the meshwidth (this is by construction of the admissible block-cluster tree, see also Lemma 2.1.10). Hence, by (3.47), for some $\gamma' > 0$,

$$d = \gamma' \Delta x.$$

Since $\Delta x \approx Ch$, for some $C > 0$, see Section 1.2.11.3,

$$d = \tilde{\gamma} h,\ \tilde{\gamma} > 0.$$

Importantly, $\tilde{\gamma}$ is constant and does not depend on $h$ and $\Delta x$. The estimate on $L$ can be obtained from Proposition 3.1.16, choosing a priori $L \geq \frac{d}{h} = \tilde{\gamma}$. Namely, there exist constants $\delta, G, A, \beta > 0$, s.t.

$$\left\| w_k^h(d') \right\| \leq \frac{G}{d'} (1-\delta)^{k - \frac{d'}{h}} (1 + A\delta^\beta)^{\frac{d'}{h}}, \quad \text{for all } d' < d, \text{ and } k \geq \frac{d}{h}.$$

Then $L$ can be estimated from:

$$\begin{aligned} \frac{G}{d}(1-\delta)^{L-\frac{d}{h}}(1+A\delta^\beta)^{\frac{d'}{h}} &< \frac{\epsilon}{4\pi d}, \\ G(1-\delta)^{L-\tilde{\gamma}}(1+A\delta^\beta)^{\tilde{\gamma}} &< \frac{\epsilon}{4\pi}. \end{aligned}$$

From this it follows that for a fixed accuracy $\epsilon$

$$L = O\left(\log \frac{1}{\epsilon}\right),$$

where the hidden constant depends on $\tilde{\gamma}$.

Therefore, to approximate the full near-field of the system (3.48) only $O(\log \frac{1}{\epsilon})$ matrices need to be constructed, **independently** of the size of this system.

**Remark 3.3.5.** *Increasing the value of $d$ allows to reuse a part of the far-field as well.*

**Remark 3.3.6.** *We do not address here the question how $\epsilon > 0$ has to be chosen to preserve the stability and convergence of the method. A full analysis would require the combination of the estimates of [29] and [116]. In particular, in [116] it is shown that the convergence of the sparse BDF2 convolution quadrature is preserved if the convolution weights are cut off with the accuracy $\epsilon$ satisfying $\log \frac{1}{\epsilon} = O\left(\log M\right) = O\left(\log N\right)$. We expect similar estimates to hold for our case as well, since all the errors are linear, and bounds for the errors and operator norms depend on $h, \Delta x$ polynomially or as powers (positive or negative) of $h, \Delta x$, c.f. [29] and Section 1.2.8.3.*

Next the question of the efficient evaluation of a matrix vector product with the system (3.56) is addressed. We suggest the use of either of two methods.

**Near-Field Matrix-Vector Multiplication with Diagonalization** The main idea of this approach is that the matrix (3.57) in (3.56) can be represented in the form of Toeplitz matrix, and hence easily diagonalized. Using (3.55), the matrix (3.56) can be rewritten in a form

$$\boldsymbol{N}_{jk} \approx \iint\limits_{\boldsymbol{\Omega}_d} \begin{pmatrix} 0 & \cdots & w^h_{L-1}(\|x-y\|) & \cdots & w^h_1(\|x-y\|) \\ 0 & \cdots & w^h_L(\|x-y\|) & \cdots & w^h_2(\|x-y\|) \\ \vdots & & & & \\ 0 & \cdots & w^h_{2L-3}(\|x-y\|) & \cdots & w^h_{L-1}(\|x-y\|) \\ \vdots & & & & \\ 0 & \cdots & 0 & \cdots & 0 \end{pmatrix} \phi_k(y)\phi_j(x)d\Gamma_x d\Gamma_y, \quad (3.58)$$

for all $k, j = 1, \ldots, M$. This matrix has the same structure as (3.48). The algorithm for the efficient evaluation of matrix-vector products involving such matrices is based on the embedding the matrix into a circulant matrix, scaling it with a properly chosen parameter $\rho$ and diagonalizing the resulting matrix with the help of the discrete Fourier transform, see Section 1.2.11.1.

In this particular case, however, $\|x-y\| < d$, for all $(x,y) \in \boldsymbol{\Omega}_d$, see Remark 3.3.3. Let us show that there is no need in the use of the scaling parameter $\rho$, see Section 1.2.11.1. From Proposition 3.1.19 it follows that the matrix

$$\begin{pmatrix} w^h_{L-1}(\|x-y\|) & w^h_{L-2}(\|x-y\|) & \cdots & w^h_1(\|x-y\|) \\ w^h_L(\|x-y\|) & w^h_{L-1}(\|x-y\|) & \cdots & w^h_2(\|x-y\|) \\ \vdots & & & \\ w^h_{2L-3}(\|x-y\|) & w^h_{2L-4}(\|x-y\|) & \cdots & w^h_{L-1}(\|x-y\|) \end{pmatrix}$$

after the extension to a circulant matrix can be approximated for $\|x-y\| < d$ by

$$\begin{pmatrix} w^h_0(\|x-y\|) & w^h_{2L-3}(\|x-y\|) & \cdots & w^h_1(\|x-y\|) \\ \vdots & & & \\ w^h_{L-1}(\|x-y\|) & w^h_{L-2}(\|x-y\|) & \cdots & w^h_L(\|x-y\|) \\ \vdots & & & \\ w^h_{2L-3}(\|x-y\|) & w^h_{2L-4}(\|x-y\|) & \cdots & w^h_0(\|x-y\|) \end{pmatrix} \approx \mathcal{F}^{-1}_{2L-2}\mathcal{P}_{2L-2}\mathcal{F}_{2L-2}, \quad (3.59)$$

where

$$\begin{aligned} \mathcal{P}_{2L-2} = \frac{1}{4\pi\|x-y\|}\mathrm{diag}\Bigg[\exp\left(-\Delta(1)\frac{\|x-y\|}{h}\right) \exp\left(-\Delta\left(\mathrm{e}^{-\frac{2\pi i}{2L-2}}\right)\frac{\|x-y\|}{h}\right) \cdots \\ \exp\left(-\Delta\left(\mathrm{e}^{-\frac{2\pi i}{2L-2}(2L-3)}\right)\frac{\|x-y\|}{h}\right)\Bigg]. \end{aligned} \quad (3.60)$$

**Remark 3.3.7.** *Such approximation can be done with arbitrary accuracy, by adjusting the value of $\epsilon > 0$ in (3.53).*

*For $\|x-y\| < d$, scaled convolution weights $\mathrm{w}_n\left(\frac{\|x-y\|}{h}\right)$ decay exponentially with $n \geq L$, see Proposition 3.1.16. Therefore, choosing $\epsilon$ being equal to the desired accuracy allows to approximate*

$$\frac{\exp\left(-\frac{\Delta(\xi)}{h}\|x-y\|\right)}{4\pi\|x-y\|} = \sum_{k=0}^{\infty} w^h_k(\|x-y\|)\xi^k, \qquad |\xi| \leq 1,$$

*by $L$ terms of the above series with an accuracy close to $\frac{\epsilon}{4\pi\|x-y\|}$. This shows some redundancy of the representation (3.58).*

Comparing (3.56) to (3.59, 3.60), one can see that the matrix vector multiplication with $\boldsymbol{\mathcal{N}}$ can be performed in $O(L\log L)$ steps. It requires assembling the Galerkin matrices $\boldsymbol{\mathcal{M}}^k$ defined as

$$\boldsymbol{\mathcal{M}}^k_{ij} = \iint\limits_{\boldsymbol{\Omega}_d} \frac{\exp\left(-\Delta\left(\mathrm{e}^{-i\frac{2\pi}{2L-2}k}\right)\frac{\|x-y\|}{h}\right)}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y, \tag{3.61}$$
$$i,j = 1\dots M,\ k = 0,\dots,L-1,$$

where we also used the fact that half of these matrices are complex conjugates of the rest, see Remark 1.2.25. Importantly, these matrices need to be constructed once and can then be reused for all the matrix-vector multiplications of type (3.48).

**Direct Near-Field Matrix-Vector Multiplication** Remark 3.3.7 shows that the representation (3.58), though allows to evaluate the matrix-vector product with $\boldsymbol{\mathcal{N}}$ efficiently, may be redundant, in the sense that it requires constructing more matrices than needed. Alternatively, one could perform a direct matrix-vector multiplication with the matrix (3.56) of size $L$ in quadratic time. Since $L = O(\log N)$ (see Remark 3.3.6), computing this matrix-vector product with a complexity of $O(L^2)$ may increase the time of the solution of the system of equations, but, as it is shown in this section, allows to decrease storage costs as well as the time needed to construct the matrices.

The matrices $\boldsymbol{\mathcal{M}}^k,\ k = 0,\dots,L-1$, in (3.61) contain only the near-field and (possibly) a part of the far-field. Therefore, if they are approximated with the help of $\mathcal{H}$-matrix techniques, the time for the computation of corresponding matrix-vector products is linear in their size and in practice is often insignificant.

Let

$$\begin{pmatrix} \boldsymbol{v}^i_0 \\ \boldsymbol{v}^i_1 \\ \vdots \\ \boldsymbol{v}^i_{n-\ell} \end{pmatrix} = \sum_{j=1}^{M} \boldsymbol{N}_{ij} \begin{pmatrix} \boldsymbol{\lambda}^j_0 \\ \boldsymbol{\lambda}^j_1 \\ \vdots \\ \boldsymbol{\lambda}^j_{\ell-1} \end{pmatrix}, \qquad i = 1,\dots,M.$$

This matrix-vector multiplication can be alternatively written as, see (3.56),

$$\boldsymbol{v}_j = \sum_{k=1}^{L-1-j} \iint\limits_{\boldsymbol{\Omega_d}} w^h_{k+j}(\|x-y\|)\boldsymbol{\lambda}_{\ell-k}d\Gamma_x d\Gamma_y,\ j = 0,\dots,L-2,$$
$$\boldsymbol{v}_k = 0, \qquad k = L-1,\dots,n-\ell.$$

Using Proposition 3.1.19,

$$\boldsymbol{v}_j \approx \frac{1}{L}\sum_{\ell_1=0}^{L-1} \iint\limits_{\boldsymbol{\Omega_d}} \frac{\exp\left(-\Delta\left(\mathrm{e}^{-i\ell_1\frac{2\pi}{L}}\right)\frac{\|x-y\|}{h}\right)}{4\pi\|x-y\|} \sum_{k=0}^{L-1-j} \mathrm{e}^{i\ell_1(k+j)}u_{\ell-k}, \tag{3.62}$$

for all $j = 0, \dots, L-2$. The error of such approximation of convolution weights $w_\nu^h(\|x-y\|)$, $\nu = 1, \dots, L-1$, is close to $\frac{\epsilon}{4\pi\|x-y\|}$, where $\epsilon$ is as in (3.53), see Remark 3.3.7. As before, the $L_2$-continuity of the single-layer boundary integral operator can be used, similarly to (3.55), to show how the respective errors can be controlled by a proper choice of $\epsilon > 0$, see also Remark 3.3.6.

From the above expression it follows that to perform the matrix-vector multiplication with $\boldsymbol{\mathcal{N}}$ it is sufficient to construct the near-field matrices $\widetilde{\boldsymbol{\mathcal{M}}}^k$:

$$\widetilde{\boldsymbol{\mathcal{M}}}_{ij}^k = \iint\limits_{\boldsymbol{\Omega}_d} \frac{\exp\left(-\Delta(e^{-i\frac{2\pi}{L}k})\frac{\|x-y\|}{h}\right)}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y,\ k = 0, \dots, \left\lfloor \frac{L}{2} \right\rfloor.$$

Note that the number of matrices $\widetilde{\boldsymbol{\mathcal{M}}}^k$, $k = 0, \dots, \lfloor \frac{L}{2} \rfloor$, is twice smaller than the number of matrices $\boldsymbol{\mathcal{M}}^k$, $k = 0, \dots, L-1$, see (3.61).

For most of the experiments we used the approach described in the current section, since the time overhead due to additional matrix-vector multiplications (using the method of the current section) was smaller than the time needed to construct additional matrices with the near-field (using the method of Section 3.3.2). We explicitly remark when we use the approach from the previous section. A heuristic to choose between both should be based on the number of the matrices reused and the number of time steps. The larger the number of matrices with the near-field is and the larger the number of time steps is, the likelier it is that the algorithm with the diagonalization will perform better. The precise limiting size of the time interval, as well as the critical number of matrices with the near-field that would require the use of the algorithm with the diagonalization has to be determined based on the extensive numerical experiments.

**Matrix-Vector Multiplication with the Far-Field Matrices** The matrix-vector multiplication, see (3.51),

$$\sum_{k=1}^{M} \boldsymbol{F}_{jk} \begin{pmatrix} \boldsymbol{\lambda}_0^k \\ \boldsymbol{\lambda}_1^k \\ \dots \\ \boldsymbol{\lambda}_{\ell-1}^k \end{pmatrix}, \qquad j = 1, \dots, M, \tag{3.63}$$

can be performed as described in Section 1.2.11.1, with the help of scaling and FFT. We have to assemble the Galerkin matrices

$$\boldsymbol{L}_{ij}^k = \iint\limits_{\substack{\Omega_\sigma \times \Omega_\tau, \\ (\sigma,\tau) \in \mathcal{L}(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}) \setminus \mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})}} \frac{\exp\left(-\Delta\left(\rho e^{-i\frac{2\pi}{n+1}k}\right)\frac{\|x-y\|}{h}\right)}{4\pi\|x-y\|}\phi_i(x)\phi_j(y)d\Gamma_x d\Gamma_y,$$

$$i, j = 1, \dots, M,\ k = 0, \dots, \lfloor \frac{n+1}{2} \rfloor,$$

where, given $\epsilon_0 > 0$, $\rho = \epsilon_0^{\frac{1}{2n+1}}$. Importantly, the near-field does not appear in this computation.

Some additional improvement in storage costs and computational complexity for these matrices can be achieved if one notices that the matrix-vector multiplication (3.63) involves

only convolution weights with indices up to $n$. The bounds stated in Proposition 3.1.16 imply that for all $\epsilon > 0$ there exists $D_n > 0$ such that for all $m \leq n$ and for all $D > D_n$

$$\left\| w_m^h(D) \right\| \leq \frac{\epsilon}{4\pi D}.$$

That is why one can construct the matrices $\boldsymbol{L}^k$, $k = 0, \ldots, \lfloor \frac{n+1}{2} \rfloor$ only on the far-field $D_n$-admissible block-clusters $(\sigma, \tau) \in \mathcal{L}^+_{D_n,F} = \mathcal{L}^+_{D_n}(\mathcal{T}_{\mathcal{I}\times\mathcal{I}}) \setminus \mathcal{L}_d(\mathcal{T}_{\mathcal{I}\times\mathcal{I}})$, see also Remark 3.3.3:

$$\boldsymbol{L}^k_{ij} = \iint\limits_{\substack{\Omega_\sigma \times \Omega_\tau, \\ (\sigma,\tau) \in \mathcal{L}^+_{D_n,F}}} \frac{\exp\left(-\Delta(\rho e^{i\frac{2\pi}{n+1}k})\frac{\|x-y\|}{h}\right)}{4\pi\|x-y\|} \phi_i(x)\phi_j(y) d\Gamma_x d\Gamma_y, \qquad i,j = 1, \ldots, M.$$

### 3.3.3 Remarks on the Application of Data-Sparse Techniques and Parallellization

In this section we would like to address several questions on the application of data-sparse techniques, $\mathcal{H}$-matrices and the high-frequency fast multipole method, for approximating Galerkin discretizations of boundary integral operators in the course of the recursive algorithm. Recall that the setup time (i.e. the matrix assembly time) of $\mathcal{H}^2$-matrices that use expansions coming from the HF FMM is much smaller than that of $\mathcal{H}$-matrices. However, the corresponding HF FMM accelerated matrix-vector multiplications are significantly slower than the matrix-vector multiplications with $\mathcal{H}$-matrices, even for discretizations with about $10^5$ boundary elements (see Section 2.3).

The structure of the system of equations we need to solve is shown in Figure 3.5. The

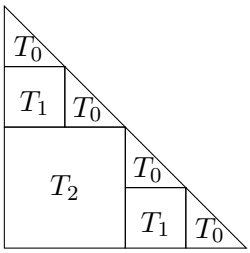


*Figure 3.5: The structure of the matrix of the convolution quadrature system of equations.*

solution of the small triangular system of size $J$ (where the matrix $T_0$ is involved) has to be performed $O\left(\frac{N}{J}\right) = O(N)$ times. Since this operation requires the construction of only a few matrices and performing many matrix-vector multiplications with them, it makes scene to approximate these matrices by $\mathcal{H}$-matrix techniques, see Section 1.2.11.2. Additionally, matrix-vector multiplications with matrix blocks at the lower levels of the recursive algorithm (in the figure these blocks are marked by $T_1$) need to be performed more often than that with the matrix blocks located at the higher levels of the recursive algorithm ($T_2$). Hence, for large problems it is reasonable to employ pure $\mathcal{H}$-matrix approximations in this case. For the rest of the Toeplitz blocks the choice whether $\mathcal{H}$- or $\mathcal{H} + \mathcal{H}^2$-based approximation is to be used is done as described in the end of Section 2.3.

The advantage of the recursive algorithm is its easy parallelizability. The precomputation of Galerkin discretizations of boundary integral operators can be done independently in

parallel. The same holds true for Galerkin matrix-vector multiplications needed to compute block Toeplitz matrix-vector products. However, if an optimally load balanced parallelization is needed, we suggest to parallelize two most time consuming operations, namely the $\mathcal{H}$-matrix assembly and the $\mathcal{H}^2$-matrix-vector multiplication.

### 3.3.4 Fast CQ Algorithm and Its Complexity

In this section the fast Runge-Kutta convolution quadrature algorithm is described. Compared to the conventional recursive algorithm, see Section 1.2.11, the multiplication with Toeplitz matrix blocks is replaced by the improved procedure of Section 3.3.2.2.

We substitute the procedure `Multiply` for the multiplication of the following matrix-vector product

$$\begin{pmatrix} \boldsymbol{h}_\ell \\ \boldsymbol{h}_{\ell+1} \\ \vdots \\ \boldsymbol{h}_{\ell+n-m} \end{pmatrix} = \begin{pmatrix} W_m^h & W_{m-1}^h & \cdots & W_1^h \\ W_{m+1}^h & W_m^h & \cdots & W_2^h \\ \vdots & & & \\ W_n^h & W_{n-1}^h & \cdots & W_{n-\ell+1}^h \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}_p \\ \boldsymbol{\lambda}_{p+1} \\ \vdots \\ \boldsymbol{\lambda}_{p+m-1} \end{pmatrix},$$

see Section 1.2.11, by the two procedures.

`MultiplyNF` ($m$, $n$, $p$, $\ell$, $\boldsymbol{\lambda}$, $\tilde{\boldsymbol{h}}$) - performs the matrix-vector multiplication with the near-field:

$$\begin{pmatrix} \tilde{\boldsymbol{h}}_\ell^j \\ \tilde{\boldsymbol{h}}_{\ell+1}^j \\ \cdots \\ \tilde{\boldsymbol{h}}_{\ell+n-m}^j \end{pmatrix} = \sum_{k=1}^{M} \boldsymbol{N}_{jk} \begin{pmatrix} \boldsymbol{\lambda}_p^k \\ \boldsymbol{\lambda}_{p+1}^k \\ \cdots \\ \boldsymbol{\lambda}_{p+m-1}^k \end{pmatrix}, \qquad j = 1, \ldots, M.$$

`MultiplyFF` ($m$, $n$, $p$, $\ell$, $\lambda$, $\bar{\boldsymbol{h}}$) - performs the matrix-vector multiplication with the far-field:

$$\begin{pmatrix} \bar{\boldsymbol{h}}_\ell^j \\ \bar{\boldsymbol{h}}_{\ell+1}^j \\ \cdots \\ \bar{\boldsymbol{h}}_{\ell+n-m}^j \end{pmatrix} = \sum_{k=1}^{M} \iint\limits_{\substack{\Omega_\sigma \times \Omega_\tau, \\ (\sigma,\tau) \in \mathcal{L}_{D_n,F}^+}} \begin{pmatrix} w_m^h(\|x-y\|) & \ldots & w_1^h(\|x-y\|) \\ w_{m+1}^h(\|x-y\|) & \ldots & w_2^h(\|x-y\|) \\ \cdots & & \\ w_n^h(\|x-y\|) & \ldots & w_{n-m+1}^h(\|x-y\|) \end{pmatrix} \times$$

$$\times \begin{pmatrix} \boldsymbol{\lambda}_p^k \\ \boldsymbol{\lambda}_{p+1}^k \\ \cdots \\ \boldsymbol{\lambda}_{p+m-1}^k \end{pmatrix} \phi_j(x)\phi_k(y) d\Gamma_x d\Gamma_y, \qquad j = 1, \ldots, M.$$

Let the parameter $J$ be fixed: every system of size smaller than $J$ is to be solved directly.

**function** Solve ($n_0$, $n_1$, $\boldsymbol{\lambda}$, $\boldsymbol{g}$)
**if** ($n_1 - n_0 \le J$) **then**
  SolveBasic($n_0$, $n_1$, $\boldsymbol{\lambda}$, $\boldsymbol{g}$);
**else**
  $n_{\frac{1}{2}} = \left\lceil \frac{n_0 + n_{\frac{1}{2}}}{2} \right\rceil$;
  Solve($n_0$, $n_{\frac{1}{2}}$, $\boldsymbol{\lambda}$, $\boldsymbol{g}$);
  MultiplyNF ($n_{\frac{1}{2}} - n_0 + 1$, $n_1 - n_0$, $n_0$, $n_{\frac{1}{2}} + 1$, $\boldsymbol{\lambda}$, $\boldsymbol{h}_1$);
  MultiplyFF ($n_{\frac{1}{2}} - n_0 + 1$, $n_1 - n_0$, $n_0$, $n_{\frac{1}{2}} + 1$, $\boldsymbol{\lambda}$, $\boldsymbol{h}_2$);
  $\boldsymbol{g}|_{n_{\frac{1}{2}}+1,\dots,n_1} = \boldsymbol{g}|_{n_{\frac{1}{2}}+1,\dots,n_1} - \boldsymbol{h}_1|_{n_{\frac{1}{2}}+1,\dots,n_1} - \boldsymbol{h}_2|_{n_{\frac{1}{2}}+1,\dots,n_1}$;
  Solve($n_{\frac{1}{2}} + 1$, $n_1$, $\boldsymbol{\lambda}$, $\boldsymbol{g}$);
**end if**
**endFunction**

Let us discuss the complexity of this algorithm based on the preliminary estimates in Section 1.2.11.3. Compared to the conventional recursive algorithm, see Section 1.2.11, the new algorithm performs an extra block matrix-vector multiplication with the near-field matrices. The computational complexity of each of matrix-vector multiplication with the near-field matrices is either $O(L \log LM) = O(\log N \log\log NM)$ (if the near-field matrix-vector multiplication with the diagonalization is used) or $O(L^2 M) = O(\log^2 NM)$, see Remark 3.3.6. Totally, there are $O(N)$ matrix blocks (3.49), hence the total complexity of the near-field related matrix-vector multiplications is

$$O(N \log^2 NM).$$

The number of matrix-vector multiplications with the far-field matrices is $O(N \log N)$, while each of this matrix-vector multiplications requires about $O(M \log M)$ operations (here the hidden constant depends on the accuracy of the approximation, see Remark 3.3.8). Hence, the total complexity of the algorithm is

$$O(N \log NM \log M + N \log^2 NM) = O(NM \log^2 M).$$

The memory costs for the near-field matrices scale linearly, $O(M \log N)$, while for the rest of the matrices as $O(NM \log M)$. As before, for the matrices with the far-field in this complexity estimate there is a hidden constant that depends on the accuracy of the approximation.

The construction times for $\mathcal{H}$-matrices scale as $O(NT_q M \log M)$, where $\mathcal{T}_q$ is the complexity of the evaluation of the integrals in BEM, see also the discussion in Section 3.3.1. Since we use the technique described in detail in [169], $T_q$ scales not worse than $O(\log^\alpha M)$, for $\alpha \ge 0$ (in our case $T_q = O(\log^4 M)$). The construction times for $\mathcal{H}^2$-matrices scale as $O(NM \log M)$. The hidden constants in these complexity estimates depend on the accuracy of the matrix approximations.

Combined with the use of data-sparse techniques and the complexity estimates 1.2.11.3, the computational complexity of the `Solve` procedure is not worse than $O(NM \log^2 M)$, the time to construct the matrices $O(NM \log M \log^k M)$, for $k > 1$, and the storage costs are $O(NM \log M)$.

**Remark 3.3.8.** *In [29] a rather restrictive condition on the accuracy $\epsilon$ of the separable expansions and $\mathcal{H}$-matrix approximation was imposed, suggesting that it has to be proportional to $h^\alpha$, $\alpha \geq 1$. However, as noted in the same work, this is not too prohibitive when applied to the HF FMM for the Helmholtz kernel and leads to logarithmic ($\log^2 \frac{1}{h} = \log^2 M$) increase of the complexity. Same holds true for $\mathcal{H}$-matrices. In our algorithm they are applied to approximate the discretizations of $\mathcal{V}(s)$ with $s$ either being small or $\left|\frac{\operatorname{Im} s}{\operatorname{Re} s}\right| < C$, for some $C > 0$. In both cases the $\mathcal{H}$-matrix complexity depends on the desired accuracy $\epsilon$ as $\log^k \frac{1}{\epsilon}$, for $k \geq 1$.*

# Chapter 4

# Numerical Experiments

In this section we present the results of the numerical experiments for the solution of the problem of wave scattering by a sound-soft obstacle. In particular, we solve the boundary integral equation (1.3), namely

$$g(t,x) = -u^{inc}(t,x) = \int_0^t \int_\Gamma \frac{\delta(t-\tau-\|x-y\|)}{4\pi\|x-y\|}\lambda(\tau,y)d\Gamma_y, \qquad x \in \Gamma, \tag{4.1}$$

on the interval $[0,\ T]$, $T > 0$. Knowing $\lambda(t,y)$, we compute the scattered field (see (1.1)) outside of the domain $\Omega$ using the indirect boundary integral formulation:

$$u(t,x) = \int_0^t \int_\Gamma \frac{\delta(t-\tau-\|x-y\|)}{4\pi\|x-y\|}\lambda(\tau,y)d\Gamma_y, \qquad x \in \Omega^c,\ t \in [0,\ T].$$

First we consider several different domains, including a unit sphere, a thin domain similar to the NASA almond, see [186], and a trapping domain. We demonstrate almost linear complexity of fast Runge-Kutta convolution quadrature, as well as show that it indeed outperforms conventional Runge-Kutta CQ, especially for large problems. At the end of this section we present the numerical evidence that the suggested approach allows to achieve higher accuracies, as well as that the matrix approximations can be done with the accuracy sufficient not to affect the convergence of the algorithm, see [29] for the related discussion.

In all the computations of this section the Helmholtz boundary single layer operators are discretized by the Galerkin method with piecewise constant test and trial basis functions. The matrices are approximated with accuracy $\epsilon_0 = 1e-6$, unless stated otherwise. For all the experiments the 3-stage Radau IIA method of the 5th order is used.

To cut off the convolution weights, we fix $L > 0$ and choose the parameter $d$, see (3.53), as

$$d = \sup\left\{\tilde{d} : \left\|\mathrm{w}_j\left(\frac{d'}{h}\right)\right\| < 5e-4,\ \text{for all } j \geq L,\ d' \in [0,\tilde{d}]\right\}. \tag{4.2}$$

Here $\mathrm{w}_j$ are scaled convolution weights, as defined in Section 3.1.1. We use this accuracy setup for all the experiments.

For long-time computations we employ the procedure described in [21] that allows to reduce the amount of matrices to be assembled. Let us briefly describe the main idea of

this method. Let the diameter of the domain be equal to $D$. Given $\epsilon > 0$, there exists $N_D$ s.t. for all $n > N_D$ and for all $\tilde{d} \leq D$

$$\left\| \mathrm{w}_n \left( \frac{\tilde{d}}{h} \right) \right\| < \epsilon.$$

Then Toeplitz matrix blocks of size $N_T > N_D$, see (1.58), can be substituted by Toeplitz blocks of size $N_D$. This allows to significantly reduce the number of matrices that need to be constructed. For our accuracy setting the choice $\epsilon = 5e-4$ was always sufficient.

All the experiments of this section were performed with the help of $\mathcal{H}$LIBpro [132] on three clusters of the Max Planck Institute, each having 8x Dual Core AMD Opteron 8220 CPUs with 2.8 GHz and 256 GB RAM. The computation time we show is the total CPU time (excluding the time needed for the communications between CPUs), i.e. CPU time needed to solve the scattering problem on one CPU. It includes the time of construction of all the matrices for the recursive CQ algorithm and the time for the actual solution of the lower triangular Toeplitz system.

As discussed before, we assemble the matrices once and store them on a disk. For the discretizations that are approximated with the help of the fast multipole method we keep all translation operators. We report storage costs, i.e. the disk space needed to keep the precomputed matrices, as the output of the shell command `du` (rather than `du --apparent-size` (though the difference in both never reached more than $\pm 2\%$)).

Additionally, we introduce the following notation:

- $\mathcal{H}$ denotes the approach that uses $\mathcal{H}$-matrices only and requires the construction of the near-field for all the matrices (the conventional RK CQ algorithm);
- $\mathcal{H}^{sp}$ is the approach based on $\mathcal{H}$-matrices with the near-field reuse;
- $\mathcal{H}^2$ is the algorithm that uses the fast multipole method but does not reuse the near-field;
- $\mathcal{H}^{2,sp}$ is fast Runge-Kutta convolution quadrature based on the near-field reuse and the HF FMM.

## 4.1 Experiments with a Sphere

In this section we consider sound-soft scattering by the unit sphere. In the first part we demonstrate that the approach with the near-field reuse allows to obtain the solution with the accuracy not worse than the accuracy of the conventional Runge-Kutta convolution quadrature method. In the second part we consider the scattering of wide-band incident waves and demonstrate the efficiency of the new algorithm.

### 4.1.1 Correctness of the Approach

As the first example, we consider the scattering problem for the unit sphere on the time interval $[0,\ 25]$ for which the explicit solution is known. We choose the incident wave as

$$u^{inc}(t,x) = u^{inc}(t) = -\mathrm{e}^{-\frac{(t-3)^2}{0.4^2}} \cos 3t, \qquad t \geq 0. \tag{4.3}$$

Importantly, $|u^{inc}(0)| < 10^{-24}$. The solution to (4.1) with such incident wave does not depend on spatial variables, see [165]:

$$\lambda(t) = -2 \sum_{k=0}^{\lfloor \frac{T}{2} \rfloor} \frac{d}{d\tau} u^{inc}(\tau) \Big|_{\tau=t-2k}.$$

We fix the time step $h = 0.125$ and choose the spatial discretization with $M = 16200$ triangles. The results of the computation with the conventional Runge-Kutta CQ based on $\mathcal{H}$-matrices ($\mathcal{H}$) and with fast Runge-Kutta CQ ($\mathcal{H}^{2,sp}$) are shown in Figure 4.1.

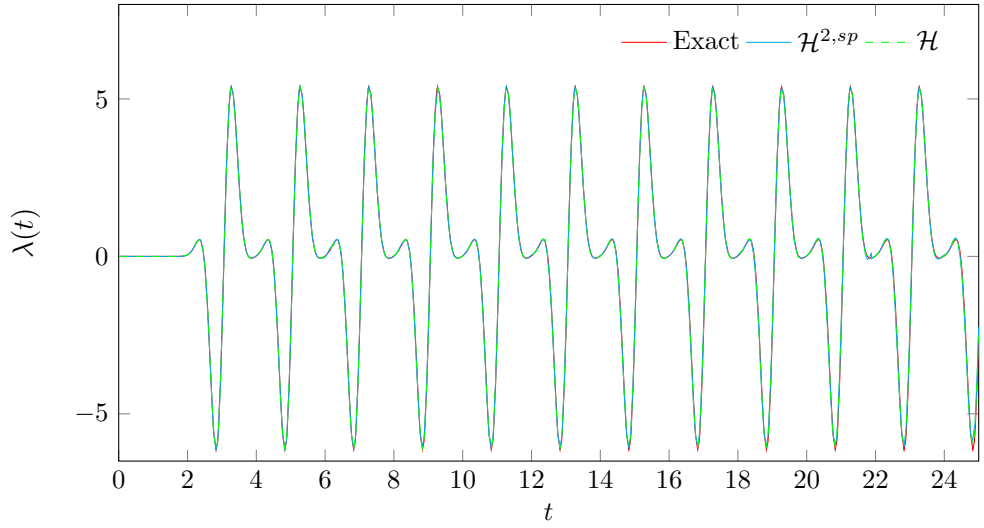


*Figure 4.1: The solution to the problem (4.3) at one of the points on the sphere. We plot the solution obtained at internal stages of Runge-Kutta convolution quadrature (see Section 1.2.8.1) as well.*

Let $\tilde{\lambda}_k(x)$, $k = 0, \dots, N$, be the boundary density at the time step $t = kh$ obtained numerically. We measure the relative error of the solution:

$$\epsilon_{\text{rel}} = \frac{\left( h \sum_{k=0}^{N} \|\tilde{\lambda}_k(x) - \lambda(kh, x)\|^2_{H^{-\frac{1}{2}}(\Gamma)} \right)^{\frac{1}{2}}}{\left( h \sum_{k=0}^{N} \|\lambda(kh, x)\|^2_{H^{-\frac{1}{2}}(\Gamma)} \right)^{\frac{1}{2}}}. \tag{4.4}$$

To compute $\|.\|_{H^{-\frac{1}{2}}(\Gamma)}$, we make use of the results of Proposition 1.1.2. For a given $s \in \mathbb{C}$ and $\phi \in H^{-\frac{1}{2}}(\Gamma)$, the equivalent norm in $H^{-\frac{1}{2}}(\Gamma)$ is given by

$$\|\phi\|^2 := \langle \mathcal{V}(s)\phi, \phi \rangle. \tag{4.5}$$

For the current experiment the estimate on the norm (4.5) was found using an $\mathcal{H}$-matrix approximation of $\boldsymbol{V}(s)$ for $s = 20$.

The relative error of the solution obtained with the help of fast Runge-Kutta CQ does not exceed $5.31 \cdot 10^{-4}$, and for the solution obtained with the help of $\mathcal{H}$-matrices $\epsilon_{\text{rel}} \approx 5.24 \cdot 10^{-4}$. This shows that the error that stems from the near-field reuse is negligible compared to the error coming from matrix approximations and the discretization.

### 4.1.2 Scattering of a Wide-Band Signal

Let us consider scattering of the following incident wave

$$u^{inc}(t,x) = -0.33 \sum_{i=1}^{3} \mathrm{e}^{-\frac{(t-\alpha_i \cdot x - 6\sigma - 1)^2}{\sigma^2}}, \tag{4.6}$$

with parameters $\alpha_1 = (-1,\ 0,\ 0)\,,\ \alpha_2 = (0,\ -1,\ 0)\,,\ \alpha_3 = (0,\ 0,\ -1)$. The Dirichlet data is given by $g(t,x) = -u^{inc}(t,x)$ and is almost zero in $t = 0$:

$$|g(0,x)| < 10^{-15}, \qquad x \in \Gamma.$$

In order to resolve the solution for higher frequencies, time and spatial discretizations have to be refined, preserving relations $h\omega \approx const$, $\frac{\Delta x}{h} \approx const$.

At each step of the experiment $k = 1, \ldots, 8$, $\sigma = \sigma_k$ is reduced by a factor $\sqrt{2}$, and the number of time steps $N_k$ on the interval $[0,\ 12.5]$ is increased by the same factor; for the spatial discretization $M_k \approx 2M_{k-1}$. To check the validity of the result obtained for a certain value of $\sigma$, we perform the experiment on a finer mesh and compare the scattered field outside of the domain computed on the coarse and fine meshes. The largest $\sigma_{max} = 0.8$, the smallest $\sigma_{min} = 0.07$.

For larger problems ($M \geq 65448$), it appears that the accuracy produced by the Galerkin integration with the chosen quadrature order is not sufficient to construct some of the matrices $\boldsymbol{V}(s)$ with chosen accuracy settings. Hence we have to increase the quadrature order. The precise theoretical reasoning for this can be found in [169]. In a nutshell, the numerical evaluation of

$$\int_{\tau} \int_{\sigma} \frac{\mathrm{e}^{-s\|x-y\|}}{4\pi\|x-y\|} d\Gamma_x d\Gamma_y,$$

with $\tau$, $\sigma$ being two panels of size $O(\Delta x)$, requires $O\left(\log^4 \frac{1}{\Delta x}\right)$ quadrature points if the distance between the panels is $O(\Delta x)$. Hence, if $\Delta x \to 0$ and $L$, see (4.2), is fixed (as in our case), the quadrature order indeed has to increase. In all the experiments in this section $L$ does not exceed 16.

For the first four experiments (discretizations with $N \leq 70$ and $M \leq 8192$) the use of the fast Runge-Kutta convolution quadrature algorithm does not allow to obtain a significant gain compared to the conventional Runge-Kutta algorithm. For the four largest problems, as Tables 4.1 and 4.2 show, the new algorithm is up to 2.4 times faster than conventional $\mathcal{H}$- or $\mathcal{H}^2$-matrix based approaches. The storage costs are reduced more than 3 times compared to the purely $\mathcal{H}$-matrix based CQ algorithm.

The new algorithm requires more time to solve the system of equations after all the matrices were constructed, which can be attributed to the use of the high-frequency fast multipole method, see the related discussion in Section 3.3.3. However, it reduces the matrix assembly time drastically compared to the $\mathcal{H}$-matrix based approach.

Figure 4.2 demonstrates almost linear complexity of the fast Runge-Kutta convolution quadrature algorithm. The time of the matrix construction and memory costs increase sublinearly for the above range of problems. The reason for this is that the assembly (and storage) costs of the full near-field of all the matrices are in this case significantly larger than that required for the far-field. Hence, if only a small part of the near-field is constructed, the total complexity is significantly improved.

| $\sigma$ | 0.2 | 0.14 | 0.1 | 0.07 |
|---|---|---|---|---|
| $h$ | 0.125 | 0.09 | 0.0625 | 0.045 |
| $M$ | 16200 | 32768 | 65448 | 129970 |
| $N$ | 100 | 139 | 200 | 278 |
| $L$ | 14 | 15 | 16 | 16 |
| $\mathcal{H}$, Gb | 33.2 | 118.6 | - | - |
| $\mathcal{H}^{sp}$, Gb | 22 | 78.9 | - | - |
| $\mathcal{H}^2$, Gb | 19.7 | 57.6 | 117.3 | - |
| $\mathcal{H}^{2,sp}$, Gb | 12.5 | 34.7 | 56.8 | 145.1 |

*Table 4.1: Storage costs for different discretizations and techniques stated in Gb for the problem with the right-hand side defined by (4.6), time interval* $[0, 12.5]$*. Here* $L$ *is the parameter from (4.2).*

| $\sigma$ | 0.2 | 0.14 | 0.1 | 0.07 |
|---|---|---|---|---|
| $h$ | 0.125 | 0.09 | 0.0625 | 0.045 |
| $M$ | 16200 | 32768 | 65448 | 129970 |
| $N$ | 100 | 139 | 200 | 278 |
| $L$ | 14 | 15 | 16 | 16 |
| $\mathcal{H}$, hr | 6.2 (0.6) | 25.9 (1.6) | - | - |
| $\mathcal{H}^{sp}$, hr | 3.9 (0.9) | 19.4 (4.2) | - | - |
| $\mathcal{H}^2$, hr | 6.4 (2) | 20.5 (5.9) | 97.8 (30.7) | - |
| $\mathcal{H}^{2,sp}$, hr | 3.9 (1.5) | 12.5 (4.2) | 40.1 (16) | 116.9 (48.6) |

*Table 4.2: CPU time (in hours) for the solution of the problem on different discretizations and with different techniques, time interval* $[0, 12.5]$*. In parentheses we show the CPU time needed to solve the system of equations after all the matrices were constructed.*

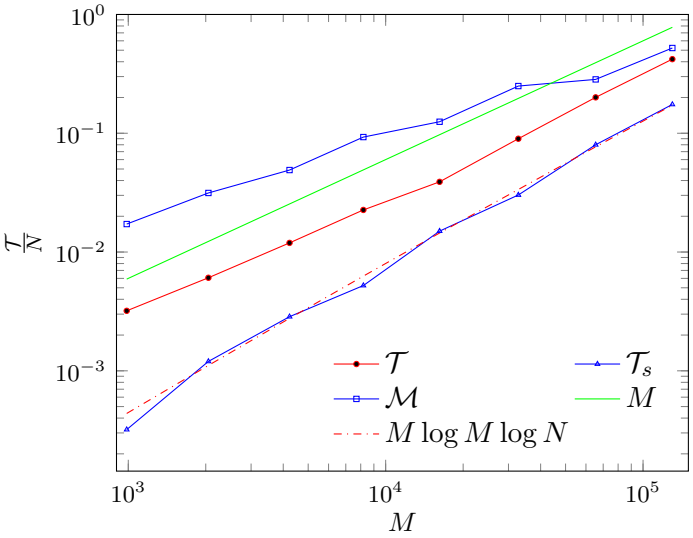


*Figure 4.2: The dependence of the total CPU time per time step* $\mathcal{T}$*, the CPU time without the time needed for the matrix assembly per time step* $\mathcal{T}_s$ *and the memory per time step* $\mathcal{M}$ *on the spatial discretization size* $M$*.*

The solutions to the problem for different $\sigma$ computed outside of the domain, at the point (2.5, 0, 0), are depicted in Figures 4.3 and 4.4. Here we depict as well the solutions obtained at internal stages of the Runge-Kutta method. Like in the previous section, the near-field reuse allows to obtain the solution with the same accuracy as the conventional $\mathcal{H}$-matrix based Runge-Kutta CQ algorithm.

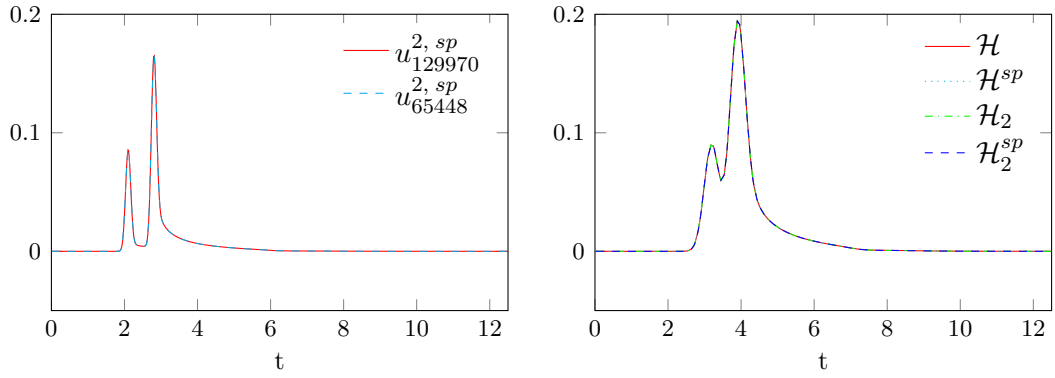


*Figure 4.3: In the left picture we depict solutions obtained on the discretizations of the domain with 65448 and 129970 triangles, $\sigma = 0.1$: on this scale the solutions are practically indistinguishable. In the right picture the solution computed for $\sigma = 0.28$ with different techniques is shown.*

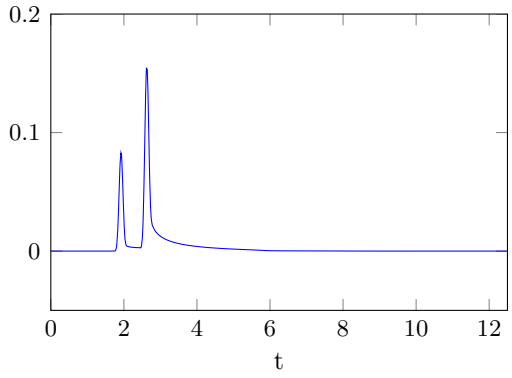


*Figure 4.4: The solution for $\sigma = 0.07$ at the point (2.5, 0, 0). We plot here the solutions obtained at internal stages of the Runge-Kutta method as well.*

The benefit of the suggested technique applied to scattering by a unit sphere is not as significant for smaller discretizations as for larger ones. Nevertheless, we can see a significant benefit from the use of the algorithm already for problems with 4.5 million unknowns. In further sections we show how the efficiency of the improved recursive algorithm depends on the domain and the problem size.

## 4.2 Experiments with an Elongated Domain

To demonstrate the efficiency of the algorithm, we perform a set of tests for the domain depicted in Figure 4.5. The domain and the mesh for it were generated with the help of Gmsh [94]. The length of this domain is 2.5, width 1 and height 0.32.

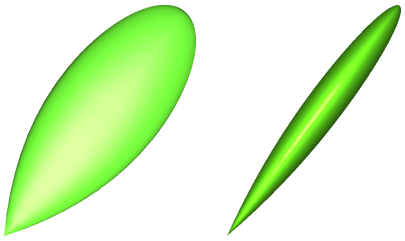

*Figure 4.5: The domain that we use in experiments. The domain is oriented parallel to x-axis; the incoming wave first hits the tip of the domain.*

The incident wave used in the experiments is the plane-wave modulated by a Gaussian:

$$u^{inc}(t,x) = -\cos(\omega(t-\alpha\cdot x-6\sigma-A))\mathrm{e}^{-\frac{(x-\alpha\cdot x-6\sigma-A)^2}{\sigma^2}}, \tag{4.7}$$

with parameters $\alpha = (-1,\ 0,\ 0)$, $A = 1.45$, $\sigma = \frac{6}{\omega}$. The Dirichlet data is given by $g(t,x) = -u^{inc}(t,x)$ and satisfies, for all $\sigma$ we considered,

$$|u^{inc}(0,x)| < 10^{-15}, \qquad x \in \Gamma.$$

As previously, in order to resolve the right-hand side for higher frequencies, time and spatial discretizations have to be refined, preserving relations $h\omega \approx const$, $\frac{\Delta x}{h} \approx const$.

At each step of the experiment $k = 1, ..., 4$, we double $\omega_k$, i.e. $\omega_k = 2\omega_{k-1}$, as well as increase the number of time steps $N_k$ on the interval $[0,\ 6.4]$ twice. For the spatial discretization $M_k \approx 4M_{k-1}$. To check the validity of the results, we perform the experiment for every frequency $\omega_k$, $k = 1, ..., 4$, on the finer mesh and compare the obtained solutions. The accuracy of the solution for the largest frequency, namely $\omega = 48$, is compared to the solution obtained on the time-space mesh with 92 million unknowns (or more than 276 million unknowns if fractional time steps (internal stages of the Runge-Kutta method) are taken into account).

We increase the number of matrices to be reused for larger problems in order to improve the performance of the algorithm: it makes sense to reuse also a part of the far-field as the problem size increases, see Remark 3.3.5. The parameter $L$, see (4.2), varies here from 21 to 37. For the two largest problems we employ the approach for the near-field reuse with the diagonalization, while for the smaller problems the direct approach is used (see Section 3.3.2.2). Additionally we increase the Galerkin quadrature order for the two largest problems.

Storage costs required for the solution of the problem with different approaches are shown in the Table 4.3, while computation times are given in Table 4.4. Numerical experiments show that the algorithm based on $\mathcal{H}^2$-matrices with the near-field reuse is more than 3 times faster than the conventional $\mathcal{H}$-matrix based method and allows to reduce storage costs 2-5 times. In the conventional $\mathcal{H}$-matrix based approach the matrix assembly time is significantly larger than the actual system solution time, and the use of the fast Runge-Kutta CQ algorithm allows to reduce this time significantly.

In Figures 4.6, 4.7 we plot the solutions outside of the domain, at the distance 1 from the tip of the domain (at the point $x_0 = (2.5,\ 0,\ 0)$). We show as well the error

$$e_n = e(nh) = |\tilde{u}_n^N(x_0) - \tilde{u}_n^{2N}(x_0)|, \qquad n = 1, \ldots, N. \tag{4.8}$$

| $\omega$ | 6 | 12 | 24 | 48 | 48 |
|---|---|---|---|---|---|
| $h$ | 0.12 | 0.06 | 0.03 | 0.015 | 0.01 |
| $M$ | 1134 | 4096 | 16072 | 64230 | 144092 |
| $N$ | 54 | 107 | 214 | 427 | 640 |
| $L$ | 21 | 24 | 24 | 26 | 37 |
| $\mathcal{H}$, Gb | 0.95 | 11.5 | 159.7 | - | - |
| $\mathcal{H}^{sp}$, Gb | 0.42 | 4.7 | 113.4 | - | - |
| $\mathcal{H}^2$, Gb | 1.15 | 9.8 | 71.7 | - | - |
| $\mathcal{H}^{2,sp}$, Gb | 0.42 | 4.2 | 30.9 | 169 | 414.3 |

*Table 4.3: Storage costs for different discretizations and techniques stated in Gb for the problem with the right-hand side defined by (4.7), time interval* $[0,\ 6.4]$.

| $\omega$ | 6 | 12 | 24 | 48 | 48 |
|---|---|---|---|---|---|
| $h$ | 0.12 | 0.06 | 0.03 | 0.015 | 0.01 |
| $M$ | 1134 | 4096 | 16072 | 64230 | 144092 |
| $N$ | 54 | 107 | 214 | 427 | 640 |
| $L$ | 21 | 24 | 24 | 26 | 37 |
| $\mathcal{H}$, hr | 0.38 (0.01) | 3 (0.08) | 43.9 (1) | - | - |
| $\mathcal{H}^{sp}$, hr | 0.13(0.02) | 1.1 (0.3) | 32.2 (2.4) | - | - |
| $\mathcal{H}^2$, hr | 0.28 (0.03) | 2.5 (0.5) | 23.6 (6.4) | - | - |
| $\mathcal{H}^{2,sp}$, hr | 0.12 (0.03) | 1.2(0.4) | 12.4 (5) | 135.8 (47.2) | 371.2 (157.8) |

*Table 4.4: Total CPU times for different discretizations and techniques stated in hours for the problem with the right-hand side defined by (4.7), time interval* $[0,\ 6.4]$. *In parentheses we show the CPU time needed to solve the system of equations after all the matrices were constructed.*

Here $\tilde{u}_n^N(x_0)$ is the scattered field at the point $x_0$ obtained on the discretization with $N$ time steps and $M$ spatial degrees of freedom. The quantity $\tilde{u}_n^{2N}(x_0)$ is the scattered field at the point $x_0$ computed with the help of fast Runge-Kutta CQ on the finer spatial discretization (with approximately $4M$ degrees of freedom) and $2N$ time steps

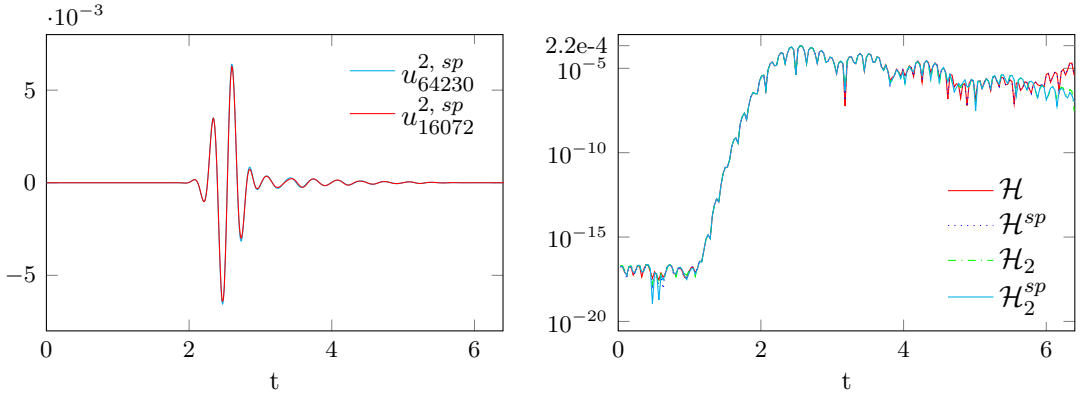


*Figure 4.6: In the left plot we depict the solution computed for* $\omega = 24$ *(also at internal stages), while in the right plot we show the errors (4.8) for the same solution obtained with the help of different techniques. The errors that stem from the near-field reuse are negligible compared to the matrix approximation and convolution quadrature errors.*

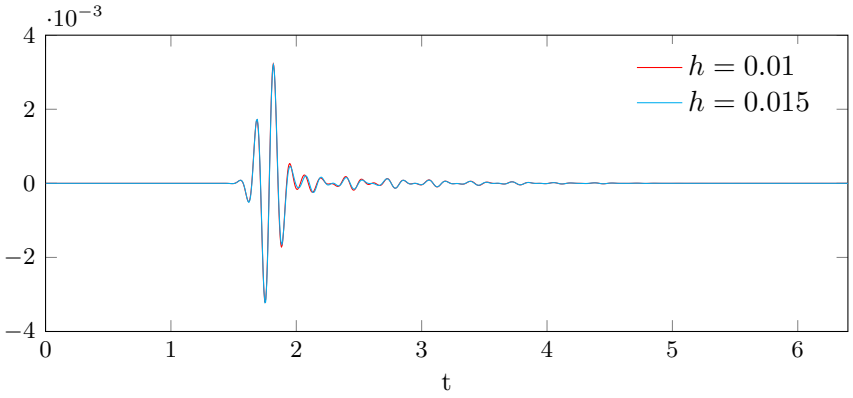


*Figure 4.7: The solution for $\omega = 48.0$ at the point $(2.5,\ 0,\ 0)$ obtained on two different discretizations, with $h = 0.015$ and $h = 0.01$ (we plot as well the solution computed on internal stages). Similarly to the case $\omega = 24.0$, both solutions are in a good agreement.*

The above results show that already for problems with 60000 unknowns the use of the FMM-based approach with the near-field reuse allows to obtain noticeable performance gain without deterioration of accuracy compared to the conventional $\mathcal{H}$-matrix- and FMM-based CQ algorithms.

## 4.3 Experiments with a Trapping Domain

In [41] it is shown that for a class of 2-dimensional domains (that contain an elliptic cavity) the condition number of the combined field integral formulation for the exterior Helmholtz problem grows exponentially with the frequency. Hence, for larger frequencies, the scattering problem seems to be better suited for the solution in the time domain. We consider the 3-dimensional domain of the diameter 2.0 formed by rotating the 2D trapping domain in question; this domain is depicted in Figure 4.8. The domain and mesh for it were generated with the help of Gmsh [94].

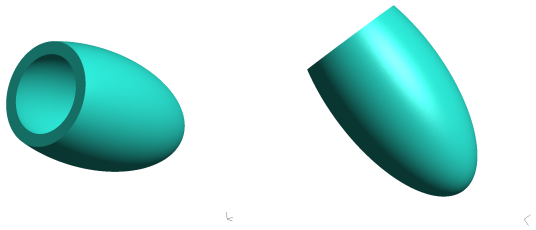

*Figure 4.8: The trapping domain that we use in experiments. The domain is oriented parallel to $x$-axis; the incoming wave first hits the cavity.*

We solve the scattering problem with the incident wave

$$u^{inc}(t,x) = -\exp\left(-\omega^2(t - x\cdot\alpha - 6\sigma - A)^2\right), \tag{4.9}$$

where $\omega$ varies from $\frac{11}{3} \approx 3.667$ to 28 and $A$ is chosen so that in $t = 0$, for all $x \in \Gamma$ and all $\omega$ in the above range, $\left|u^{inc}(0,x)\right|$ does not exceed $2\cdot 10^{-10}$. For the three smallest experiments $A = 0.5$ and for the two largest experiments $A = 0.588$. As before, to control the error,

we compare the solution in a point outside of the domain to the solution in the same point computed on a finer discretization, see (4.8).

The vector $\alpha = (1,\ 0,\ 0)$ is oriented along the axis of the rotation of the domain. The results of the numerical experiments are shown in Tables 4.5 and 4.6. As before, we increase the Galerkin quadrature order for the largest problem. For all the experiments we employ the algorithm of the near-field reuse with the diagonalization, see Section 3.3.2.2.

The parameter $L$ from (4.2) is chosen in the range from 26 (for the smallest problem) to 43 (the largest one).

| $\omega$ | $\frac{11}{3}$ | $\frac{29}{3}$ | $\frac{41}{3}$ | 20 | 28 |
|---|---|---|---|---|---|
| $h$ | 0.075 | 0.028 | 0.02 | 0.014 | 0.01 |
| $M$ | 1344 | 9588 | 21900 | 39612 | 89202 |
| $N$ | 70 | 188 | 263 | 375 | 525 |
| $L$ | 26 | 36 | 41 | 41 | 43 |
| $\mathcal{H}$, Gb | 1.6 | 78 | - | - | - |
| $\mathcal{H}^{sp}$, Gb | 1.2 | 56.6 | 230.2 | - | - |
| $\mathcal{H}^2$, Gb | 2.9 | 40.6 | 99.2 | 277 | 608 |
| $\mathcal{H}^{2,sp}$, Gb | 1.7 | 28.1 | 66.2 | 161.2 | 352 |

*Table 4.5: Storage costs for different discretizations and techniques stated in Gb for the problem with the right-hand side defined by (4.9), time interval* $[0,\ 5.25]$.

Results in Table 4.5 show that storage costs for $\mathcal{H}$-matrix based techniques grow prohibitively large even for quite small problems, hence we do not construct $\mathcal{H}$-matrix based approximations for problems with $M \geq 39612$. For the trapping domain the $\mathcal{H}$-matrix based algorithm with the near-field reuse is twice faster than the conventional $\mathcal{H}$-matrix based approach already when dealing with problems having more than 95000 unknowns. However, both methods have prohibitively high memory requirements. For smaller problems, the FMM-based algorithm with the near-field reuse is slower than the algorithm with the near-field reuse that uses $\mathcal{H}$-matrices only, but is less memory-consuming. The use of the fast multipole method with the near-field reuse for larger discretizations allows to reduce storage costs about 3.5 times compared to $\mathcal{H}$-matrix based approaches.

The FMM-based algorithm with the near-field reuse is twice faster than the conventional FMM-based CQ algorithm for the problem with 46 million unknowns, while being only 1.5 times more efficient for smaller problems. Moreover, in this case the near-field reuse allows to reduce the system solution time after the matrices have been constructed: for the largest problem it takes 315 hours for the FMM based approach without the near-field reuse vs 156 hours for the FMM based approach with the near-field reuse. This can be explained as follows. With the choice of the parameter $L$ as in this section also a part of the far-field is reused, and hence fewer multipole-to-local translations have to be done when computing the FMM accelerated matrix-vector product. This, combined with the fact that the computational complexity of the method for the near-field reuse with the diagonalization is quite low, see Section 3.3.4, results in improved computational times for the solution of the Toeplitz system of equations.

In Figures 4.9 and 4.10 the scattered field computed in the point $(0,\ 0,\ 0)$ located inside the cavity is shown. These plots demonstrate that the wave is trapped inside the cavity.

| $\omega$ | $\frac{11}{3}$ | $\frac{29}{3}$ | $\frac{41}{3}$ | 20 | 28 |
|---|---|---|---|---|---|
| $h$ | 0.075 | 0.028 | 0.02 | 0.014 | 0.01 |
| $M$ | 1344 | 9588 | 21900 | 39612 | 89202 |
| $N$ | 70 | 188 | 263 | 375 | 525 |
| $L$ | 26 | 36 | 41 | 41 | 43 |
| $\mathcal{H}$, hr | 0.8 (0.02) | 19.7 (0.4) | - | - | - |
| $\mathcal{H}^{sp}$, hr | 0.37 (0.03) | 8.7 (0.7) | 33.6(4.2) | - | - |
| $\mathcal{H}^2$, hr | 1.1 (0.26) | 21.8 (3.7) | 59.7 (19) | 161.8 (57.5) | 745 (315) |
| $\mathcal{H}^{2,sp}$, hr | 0.7 (0.1) | 15.1 (2.3) | 41 (10.7) | 105 (37) | 373 (156) |

*Table 4.6: CPU times for different discretizations and techniques stated in hours for the problem with the right-hand side defined by (4.9), time interval* $[0,\ 5.25]$*. In parentheses we show the CPU time needed to solve the system of equations after all the matrices were constructed.*

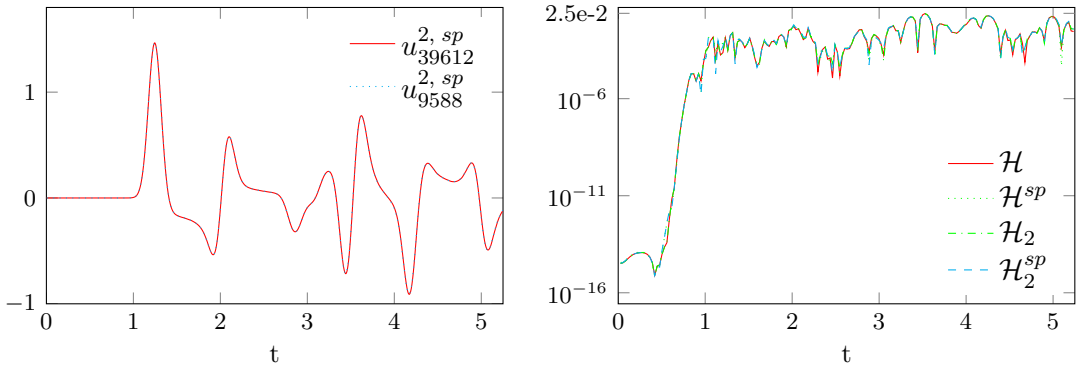


*Figure 4.9: In the left plot we depict the scattered field computed for the incident wave with* $\omega = \frac{29}{3}$ *inside the cavity in the point* $(0, 0, 0)$ *(also computed at internal stages of the Runge-Kutta method), while in the right plot we show the errors of the solution obtained with different techniques measured at the same point (see also formula (4.8)). We can see that the errors coming from the near-field reuse are negligible compared to the discretization and matrix approximation errors.*

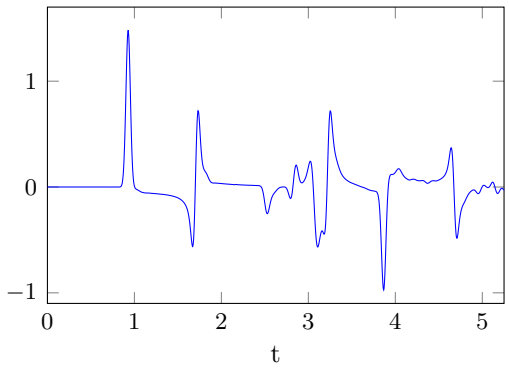


*Figure 4.10: The scattered field for* $\omega = 28$ *at the point* $(0,\ 0,\ 0)$ *(computed also at internal stages of the Runge-Kutta method).*

## 4.4 Solution Obtained with a Higher Accuracy

The goal of this section is to demonstrate that the fast convolution quadrature algorithm is capable of producing a more accurate solution. We consider the problem (4.3) for the unit sphere on the interval $[0,\ 25]$, on a larger discretization, with $M = 32768$ and the time step $h = 0.09$. The comparison of both techniques is shown in Table 4.7 and Figure 4.11. Let us remark that we increased the Galerkin quadrature order compared to the experiment in Section 4.1.1. The matrices are constructed with the accuracy $\epsilon_0 = 10^{-8}$.

| Technique | Storage, Gb | Problem solution time, hr | $\epsilon_{\text{rel}}$ |
|---|---|---|---|
| $\mathcal{H}$-matrices | 139.6 | 65 (4.4) | 9.15e-5 |
| $\mathcal{H}$-matrices with near-field reuse | 83.4 | 37 (7.3) | 9.14e-5 |
| $\mathcal{H}^2$-matrices | 78.6 | 60.8 (17.3) | 9.14e-5 |
| $\mathcal{H}^2$-matrices with near-field reuse | 41.4 | 35.3 (14) | 9.14e-5 |

*Table 4.7: Storage costs and CPU times for the problem (4.3). In parentheses we show the CPU time needed to solve the system of equations after all the matrices were constructed. The errors $\epsilon_{\text{rel}}$ are measured as described in Section 4.1.1, namely, using formulas (4.4, 4.5) with $s = 20$.*

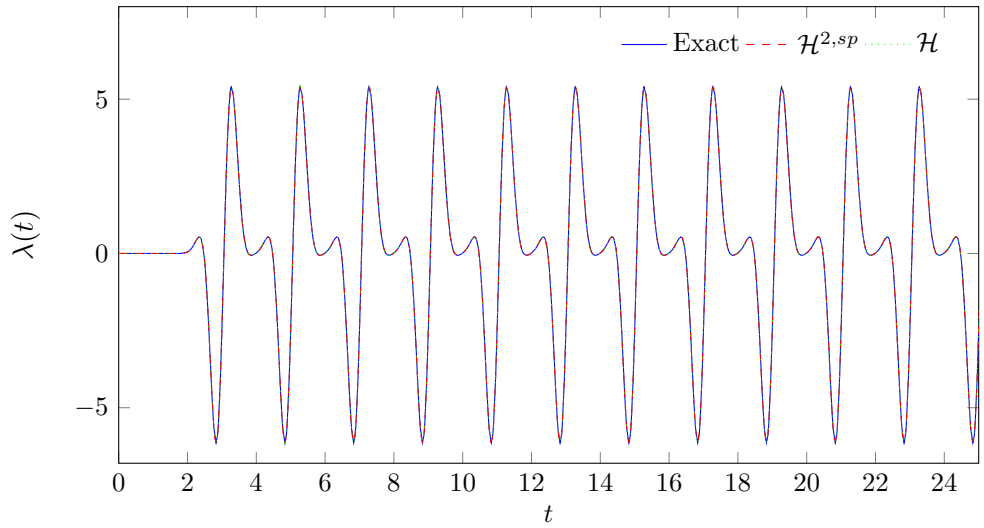


*Figure 4.11: The solution to the problem (4.3) at one of the points on the sphere. We plot the solution obtained at internal stages of Runge-Kutta convolution quadrature as well.*

## 4.5 Convergence

### 4.5.1 Convergence in Time

Section 1.2.8.4 provides tight theoretical estimates on the rate of convergence of Runge-Kutta convolution quadrature in time. Namely, given a Runge-Kutta method with the stage order $q$, the rate of the convergence for the boundary density does not exceed $h^q$. Hence, for the 3-stage Radau IIA based CQ we expect the convergence rate $h^3$.

We compute the solution of the scattering problem with the incident wave

$$u_i = -\cos((t - \alpha \cdot x - 3))\mathrm{e}^{-\frac{(x-\alpha\cdot x-3)^2}{0.1}}, \qquad \alpha = (1,\ 0,\ 0), \tag{4.10}$$

for the domain shown in Figure 4.5 discretized with 8780 triangles, on the time interval $[0, 10]$, for different time steps $h_N = \frac{10}{N}$, $N = 20,\ 40,\ 80,\ 160,\ 320$. We compute the absolute error

$$e_N = \left( h_N \sum_{j=0}^{N} \|\lambda_j^N - \lambda_j^{N_{max}}\|^2_{H^{-\frac{1}{2}}(\Gamma)} \right)^{\frac{1}{2}},$$

where $\lambda_j^N$ is the boundary density at the time step $t_j = jh_N$ computed on the discretization with the time step $h_N$ and $\lambda_j^{N_{max}}$ is the boundary density at the time step $t_j$ obtained on the finest discretization, $N_{max} = 320$.

As before, to compute $\|.\|_{H^{-\frac{1}{2}}(\Gamma)}$, we make use of the results of Proposition 1.1.2, see formula (4.5). For the present experiment the estimate on the norm (4.5) was found using an $\mathcal{H}$-matrix approximation of $\boldsymbol{V}(s)$ for $s \approx 56.5$.

All the computations are performed using the fast Runge-Kutta CQ algorithm, with highly accurate matrix approximations, $\epsilon_0 = 10^{-10}$. The results presented in Figure 4.12 show that the matrix approximations can be done with the accuracy sufficient not to affect the convergence of the method.

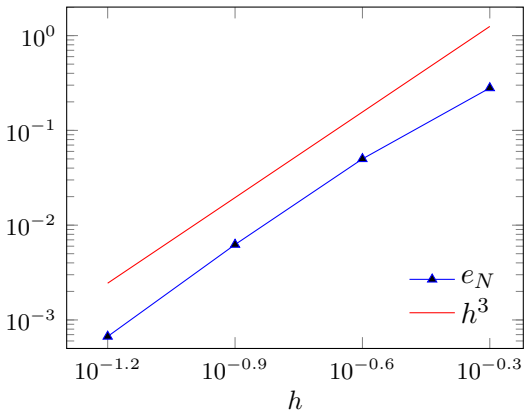


*Figure 4.12: Convergence of the boundary density in time*

### 4.5.2 Convergence in Space

To check the convergence rate in space, we solve the scattering problem for the unit sphere with the spatial mesh having $M = 2048,\ 4232,\ 8192,\ 16200,\ 32768$ triangles, on the interval $[0,\ 4]$ divided into $N = 80$ time steps. We use constant in space Dirichlet data:

$$g(t) = \mathrm{e}^{-2t}t^5 H(t).$$

The solution in this case is given by [165]:

$$\lambda(t) = 2g'(t) + 2g'(t-2) = 10\mathrm{e}^{-2t}t^4 - 4\mathrm{e}^{-2t}t^5 + \left(10\mathrm{e}^{-2(t-2)}(t-2)^4 - 4\mathrm{e}^{-2(t-2)}(t-2)^5\right) H(t-2).$$

The error is computed using the formula

$$e = \left( h \sum_{k=0}^{N} \|\tilde{\lambda}_k(x) - \lambda(kh, x)\|^2_{H^{-\frac{1}{2}}(\Gamma)} \right)^{\frac{1}{2}},$$

where $\tilde{\lambda}_j(x),\ j = 0, \ldots, N$, is the numerical solution and $\lambda(t, x)$ is the exact solution.

The convergence in this case is tested for rather accurate matrix approximations, with $\epsilon = 10^{-10}$. The results are shown in Table 4.8. The norm (4.5) is estimated using the $\mathcal{H}$-matrix approximation of $\boldsymbol{V}(s)$, with $s \approx 2$.

The theoretical convergence rate in $H^{-\frac{1}{2}}$-norm is $(\Delta x)^{\frac{3}{2}}$. As before, we assume $\Delta x \approx M^{-\frac{1}{2}}$.

| $n$, id of the experiment | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $M$ | 2048 | 4232 | 8192 | 16200 | 32768 |
| $e^{(n)}$ | 0.009 | 0.0044 | 0.0023 | 0.00117 | 0.00058 |
| $\log_2 \frac{e^{(n)}}{e^{(n+1)}}$ | 1.03 | 0.94 | 0.98 | 1.01 | - |
| Theoretical | 0.79 | 0.715 | 0.74 | 0.76 | - |

*Table 4.8: The convergence rate in space* $\log_2 \frac{e^{(n)}}{e^{(n+1)}}$ *compared to the theoretical* $\frac{3}{2}\log_2 \frac{M^{(n+1)}}{M^{(n)}}$. *We can see that it is slightly better than theoretical, and behaves as* $(\Delta x)^2$ *rather than* $(\Delta x)^{\frac{3}{2}}$.

We can see that Runge-Kutta convolution quadrature with the near-field reuse is convergent in space, and the rate of convergence for this particular right-hand side is even better than predicted. Other numerical experiments (see e.g. the experiment with the domain with the elliptic cavity) demonstrate that often very coarse spatial discretizations are sufficient to produce an accurate solution to the scattering problem with the help of convolution quadrature.

# Conclusions and Future Work

In this work we developed a fast recursive Runge-Kutta convolution quadrature algorithm for the solution of the wave scattering problem in three dimensions. This method requires the construction of Galerkin discretizations of boundary integral operators for the Helmholtz equation with decay.

Fast Runge-Kutta convolution quadrature is based on two ingredients: the use of fast data-sparse techniques, namely the high-frequency fast multipole method and $\mathcal{H}$-matrices, and decay properties of Runge-Kutta convolution weights (that are the consequence of the Huygens principle). The use of the data-sparse techniques allows to solve the scattering problem in almost linear time. Exponentially fast decay of convolution weights $w_n^h(d)$ away from the neighborhood of $d \approx nh$ allows to skip constructing the diagonal and near-diagonal matrix blocks for most of the boundary integral operator discretizations, thus avoiding the evaluation of many singular and near-singular BEM integrals.

In the first part of this work the applicability of the high-frequency fast multipole method to the Helmholtz equation with a complex wavenumber was addressed. We did not encounter major problems associated to the presence of decay, although we expect the cancellation errors to be larger than in the case of a purely real wavenumber. The presence of decay allows to reduce the fast multipole expansions length, thus improving the efficiency of the FMM approximation, which is confirmed by numerical experiments. Nevertheless, as our numerical experiments show, for moderate accuracies in the case of prevailing decay $\mathcal{H}$-matrices are more efficient than the high-frequency FMM even when only a few matrix-vector multiplications are needed.

The second part of the thesis is dedicated to the study of decay properties of convolution weights. We prove that $w_n^h(d) \approx nh$ away from $d \approx nh$. Using the approximating properties of the Runge-Kutta stability function, we show that the size of the (approximate) support of a convolution weight $w_n^h$ increases with $n$ for a fixed $h$ as $O\left(n^\alpha\right)$, where $\alpha$ depends on the order of the Runge-Kutta method. In a nutshell, the higher the order is, the smaller is the width of the convolution weight, though some Runge-Kutta methods of high order can be characterized by higher dispersion of convolution weights. The obtained values of $\alpha$ are close to optimal for Runge-Kutta methods we considered. We demonstrate how this property of convolution weights can be used to construct only $O(\log N)$ matrices with diagonal and near-diagonal blocks (near-field) and reuse them in all the stages of the recursive convolution quadrature algorithm.

Compared to the $\mathcal{H}$-matrix based convolution quadrature, the $\mathcal{H}$-matrix based algorithm with the near-field reuse allows to solve small scattering problems 1.5-2 times faster. For larger problems the high-frequency fast multipole based approach with the near-field reuse performs better, being 2-3 times faster and requiring 2-5 times less disk space. The performance of the algorithm was checked on problems from 25000 to 92 million unknowns.

In general, the gain from the use of the suggested approach depends on the problem size and on the geometry of a domain. We demonstrate that the near-field reuse does not influence the convergence properties of convolution quadrature and also allows to obtain highly accurate solutions.

The near-field reuse approach relies only on the decay properties of convolution weights and hence can be extended to solve problems other than acoustic scattering, e.g. Maxwell equations, see [30], though the theoretical justification for these cases may be required. More work is needed for optimizing the construction of matrix approximations. Since the assembly of Galerkin matrices for various frequencies can be treated as a multiparametric problem, tensor decomposition methods can be used to improve it, see, e.g. [15]. The difficulty here is the non-analyticity of high-frequency fast multipole operators in the frequency, which possibly can be overcome by the use of other fast multipole methods, e.g. [81]. The design of faster techniques for the matrix-vector multiplications involving Helmholtz potentials would significantly improve the presented approach.

More work can be done for creating the convolution quadrature based method that would take into account an a priori information about the solution and geometric properties of the domain, similarly to some of the fast MOT methods we considered in the this work.

# Appendices

# Appendix A

# The Error of the Fast Multipole Algorithm

We consider the approximation to $h_0(is\|x-y\|)$ in the course of the fast multipole algorithm as defined in Section 2.2.3.3, namely

$$\begin{aligned}\tilde{h}_0 = Q_M^{\hat{s}} \Bigg[ \sum_{n=0}^{N_*-1} \frac{2n+1}{4\pi} Q_N^{\hat{r}} \left[ \mathrm{e}^{-s(x-x_{\beta'},\hat{r})} P_n(\hat{r}\cdot\hat{s}) \right] \\ \times \mathrm{e}^{-s(x_{\beta'}-x_\beta,\hat{s})} M_{\alpha,\beta}(\hat{s}) \mathrm{e}^{-s(y_\alpha-y_{\alpha'},\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ \mathrm{e}^{s(y-y_{\alpha'},\hat{q})} P_k(\hat{q}\cdot\hat{s}) \right] \Bigg] .\end{aligned} \tag{A.1}$$

Here $N_* = \min(N, M)$. Let

$$\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s}) := \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ \mathrm{e}^{s(y-y_{\alpha'},\hat{q})} P_k(\hat{q}\cdot\hat{s}) \right] .$$

Then (A.1) can be rewritten as:

$$\tilde{h}_0 = Q_M^{\hat{s}} \left[ \mathcal{P}_N(-x_{\beta'}, -x_\beta, -x, \hat{s}) M_{\alpha,\beta}(\hat{s}) \mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s}) \right] . \tag{A.2}$$

From the expression (A.1) one can see that $\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s})$ approximates $\mathrm{e}^{s(y-y_\alpha)}$. Let us show this. First, let $v = y_{\alpha'} - y$. According to (2.22),

$$\begin{aligned}\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s}) &:= \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ \mathrm{e}^{-s(v,\hat{q})} P_k(\hat{q}\cdot\hat{s}) \right] \\ &= \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} \sum_{m=0}^{+\infty} (2m+1) i^m j_m(is\|v\|) Q_N^{\hat{q}} \left[ P_m(\hat{q}\cdot\hat{v}) P_k(\hat{q}\cdot\hat{s}) \right] .\end{aligned}$$

Making use of Lemma 2.2.8:

$$\begin{aligned}
\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s}) &= \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} \sum_{m=0}^{2N-N_*} (2m+1) i^m j_m(is\|v\|) \\
&\quad \times Q_N^{\hat{q}} \left[ P_m(\hat{q}\cdot\hat{v}) P_k(\hat{q}\cdot\hat{s}) \right] \\
&\quad + \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} \sum_{m=2N-N_*+1}^{+\infty} (2m+1) i^m j_m(is\|v\|) \\
&\quad \times Q_N^{\hat{q}} \left[ P_m(\hat{q}\cdot\hat{v}) P_k(\hat{q}\cdot\hat{s}) \right] \\
&= \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{m=0}^{\min(2N-N_*,N_*-1)} (2m+1) i^m j_m(is\|v\|) P_m(\hat{s}\cdot\hat{v}) \\
&\quad + \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} \sum_{m=2N-N_*+1}^{+\infty} (2m+1) i^m j_m(is\|v\|) \\
&\quad \times Q_N^{\hat{q}} \left[ P_m(\hat{q}\cdot\hat{v}) P_k(\hat{q}\cdot\hat{s}) \right].
\end{aligned}$$

Let us introduce

$$r_K(x,\hat{r}) = \sum_{n=K}^{+\infty} (2n+1) i^n j_n(is\|x\|) P_n(\hat{x}\cdot\hat{r}). \tag{A.3}$$

Then, using (2.22) and $2N - N_* > N_* - 1$, $\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s})$ can be represented as a sum of three terms:

$$\begin{aligned}
\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s}) &= \mathrm{e}^{s(y-y_\alpha,\hat{s})} - \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} r_{N_*}(y_{\alpha'}-y,\hat{s}) \\
&\quad + \mathrm{e}^{s(y_{\alpha'}-y_\alpha,\hat{s})} \sum_{k=0}^{N_*-1} \frac{2k+1}{4\pi} Q_N^{\hat{q}} \left[ r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q}) P_k(\hat{q}\cdot\hat{s}) \right].
\end{aligned}$$

Hence, $\mathcal{P}_N(y_{\alpha'}, y_\alpha, y, \hat{s})$ approximates $\mathrm{e}^{s(y-y_\alpha,\hat{s})}$ with the error that is a sum of two errors, one coming from the truncation of series (2.22) to a finite number of terms (see (A.3)), and another induced by the imprecise quadrature.

The obtained explicit expressions for $\mathcal{P}_N(\cdot,\cdot,\cdot,\cdot)$ have to be inserted into (A.2). After computations, it is possible to see that

$$\tilde{h}_0 = h_0(is\|x-y\|) + \sum_{n=1}^{9} \mathcal{A}_n,$$

where

$$\mathcal{A}_1 = Q_M^{\hat{s}}\left[\mathrm{e}^{s(y-y_\alpha,\hat{s})}M_{\alpha,\beta}(\hat{s})\mathrm{e}^{s(x_\beta-x,\hat{s})}\right] - h_0(is\|x-y\|),$$

$$\mathcal{A}_2 = -Q_M^{\hat{s}}\left[\mathrm{e}^{s(y-y_\alpha+x_\beta-x_{\beta'},\hat{s})}M_{\alpha,\beta}(\hat{s})r_{N_*}(x-x_{\beta'},\hat{s})\right],$$

$$\mathcal{A}_3 = -Q_M^{\hat{s}}\left[\mathrm{e}^{s(x_\beta-x-y_\alpha+y_{\alpha'},\hat{s})}M_{\alpha,\beta}(\hat{s})r_{N_*}(y_{\alpha'}-y,\hat{s})\right],$$

$$\mathcal{A}_4 = Q_M^{\hat{s}}\left[\mathrm{e}^{s(y-y_\alpha+x_\beta-x_{\beta'},\hat{s})}M_{\alpha,\beta}(\hat{s})\sum_{k=0}^{N_*-1}\frac{2k+1}{4\pi}Q_N^{\hat{r}}\left[r_{2N-N_*+1}(x-x_{\beta'},\hat{r})P_k(\hat{r}\cdot\hat{s})\right]\right],$$

$$\mathcal{A}_5 = Q_M^{\hat{s}}\left[\mathrm{e}^{s(x_\beta-x-y_\alpha+y_{\alpha'},\hat{s})}M_{\alpha,\beta}(\hat{s})\sum_{k=0}^{N_*-1}\frac{2k+1}{4\pi}Q_N^{\hat{q}}\left[r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q})P_k(\hat{q}\cdot\hat{s})\right]\right],$$

$$\mathcal{A}_6 = Q_M^{\hat{s}}\left[\mathrm{e}^{s(y_{\alpha'}-y_\alpha+x_\beta-x_{\beta'},\hat{s})}r_{N_*}(y_{\alpha'}-y,\hat{s})M_{\alpha,\beta}(\hat{s})r_{N_*}(x-x_{\beta'},\hat{s})\right],$$

$$\mathcal{A}_7 = -Q_M^{\hat{s}}\left[\mathrm{e}^{s(y_{\alpha'}-y_\alpha+x_\beta-x_{\beta'},\hat{s})}r_{N_*}(x-x_{\beta'},\hat{s})M_{\alpha,\beta}(\hat{s})\right.$$
$$\left.\times\sum_{k=0}^{N_*-1}\frac{2k+1}{4\pi}Q_N^{\hat{q}}\left[r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q})P_k(\hat{q}\cdot\hat{s})\right]\right],$$

$$\mathcal{A}_8 = -Q_M^{\hat{s}}\left[\mathrm{e}^{s(y_{\alpha'}-y_\alpha+x_\beta-x_{\beta'},\hat{s})}r_{N_*}(y_{\alpha'}-y,\hat{s})M_{\alpha,\beta}(\hat{s})\right.$$
$$\left.\times\sum_{k=0}^{N_*-1}\frac{2k+1}{4\pi}Q_N^{\hat{r}}\left[r_{2N-N_*+1}(x-x_{\beta'},\hat{r})P_k(\hat{r}\cdot\hat{s})\right]\right],$$

$$\mathcal{A}_9 = Q_M^{\hat{s}}\left[\mathrm{e}^{s(y_{\alpha'}-y_\alpha+x_\beta-x_{\beta'},\hat{s})}M_{\alpha,\beta}(\hat{s})\right.$$
$$\times\sum_{m=0}^{N_*-1}\frac{2m+1}{4\pi}Q_N^{\hat{r}}\left[r_{2N-N_*+1}(x-x_{\beta'},\hat{r})P_m(\hat{r}\cdot\hat{s})\right]$$
$$\left.\times\sum_{k=0}^{N_*-1}\frac{2k+1}{4\pi}Q_N^{\hat{q}}\left[r_{2N-N_*+1}(y_{\alpha'}-y,\hat{q})P_k(\hat{q}\cdot\hat{s})\right]\right].$$

# Appendix B

# Proof of Lemma 3.1.2

Recall the definition of $E$-polynome (3.4,3.5):

$$E(y) = |Q(iy)|^2 - |P(iy)|^2 = e_0 y^{2s} + O(y^{2s+2}),\ e_0 > 0. \tag{B.1}$$

**Lemma B.1.** *There exist $a, \nu > 0$, such that the domain*

$$\{(x, y) :\ |y| < \nu x^{\frac{1}{\ell}},\ 0 < x < a\}$$

*belongs to $\Upsilon_1$ (and intersects all the order star fingers). Here*

$$\ell = \begin{cases} p+1, & \text{if } p \text{ is odd,} \\ 2s, & \text{if } p \text{ is even,} \end{cases}$$

*where $s$ is as in (B.1).*

*Proof.* Let us first remark that this proof is similar to the proof of Theorem 7 in [120].

We rewrite the stability function as

$$R(z) = \mathrm{e}^z + C_{p+1} z^{p+1} + r(z), \qquad C_{p+1} \neq 0, \tag{B.2}$$

where $r(z) = O(z^{p+2})$.

Let us consider $(x, y)$ satisfying

$$|R(x + iy)| = 1.$$

Clearly, $R(0) = 1$ and $\left.\frac{\partial |R(x+iy)|}{\partial x}\right|_{(0,0)} = 1$. By the implicit function theorem, there exists $\epsilon > 0$ and the unique continuously differentiable function $f : B_\epsilon(0) \to \mathbb{R}$, s.t.

$$f(0) = 0, \ \text{ and } \ |R(f(y), y)| = 1.$$

Then, for $y \to 0$,

$$x = f(0) + f'(0)y + O(y^2) = O(y).$$

To prove the statement of the lemma, we explicitly study the behavior of the function $f(y)$ in the vicinity of 0.

Let us consider the following cases.

1. *$p$ is odd.*

$$
\begin{aligned}
|R(x+iy)|^2 = \mathrm{e}^{2x} + 2C_{p+1}\,\mathrm{Re}\left(\mathrm{e}^{x+iy}(x-iy)^{p+1}\right) + \\
2\,\mathrm{Re}\left(\mathrm{e}^{x+iy}r(x-iy)\right) + C_{p+1}^2(x^2+y^2)^{p+1} + |r(x+iy)|^2 \\
+ 2C_{p+1}\,\mathrm{Re}\left(r(x+iy)(x-iy)^{p+1}\right).
\end{aligned} \tag{B.3}
$$

Next we expand this expression into the Taylor series in $x$ and $y$, singling out higher order terms (retaining that $x = O(y)$):

$$
\begin{aligned}
\mathrm{e}^{2x} &= 1 + x + O(x^2), \\
\mathrm{Re}\left(\mathrm{e}^{x+iy}(x-iy)^{p+1}\right) &= \mathrm{Re}\sum_{k=0}^{p+1}\binom{p+1}{k}x^k i^{p+1-k}y^{p+1-k}(1+r_1^{(2)}(x))(1+r_2^{(2)}(y)),
\end{aligned} \tag{B.4}
$$

where $r_1^{(2)}(x) = \mathrm{e}^x - 1 = O(x)$, $r_2^{(2)}(y) = O(y)$.

Since $p+1$ is even,

$$
\mathrm{Re}\left(\mathrm{e}^{x+iy}(x-iy)^{p+1}\right) = (-1)^{\frac{p+1}{2}}y^{p+1} + O(y^{p+2}).
$$

Finally,

$$
\mathrm{Re}\left(\mathrm{e}^{x+iy}r(x-iy)\right) = O(y^{p+2}),
$$

which follows from the definition of $r(z) = O(z^{p+2})$, and

$$
\begin{aligned}
(x^2+y^2)^{p+1} &= O(y^{2p+2}), \\
|r(x+iy)|^2 &= O(y^{2p+2}), \\
\mathrm{Re}\left(r(x+iy)(x-iy)^{p+1}\right) &= O(y^{2p+2}).
\end{aligned}
$$

Summarizing the above,

$$
|R(x+iy)|^2 = 1 + 2x + 2C_{p+1}(-1)^{\frac{p+1}{2}}y^{p+1} + O(y^{p+2} + x^2). \tag{B.5}
$$

Hence,

$$
|R(x+iy)|^2 = 1 + y^{p+1}\left(2C_{p+1}(-1)^{\frac{p+1}{2}} + 2xy^{-p-1} + O(y) + O(x^2y^{-p-1})\right). \tag{B.6}
$$

We look for $x, y$ lying in an $\epsilon$-neighborhood of 0 and satisfying

$$
|R(x+iy)|^2 = 1.
$$

From the expression (B.6) we deduce that $x = f(y)$ with

$$
x = -C_{p+1}(-1)^{\frac{p+1}{2}}y^{p+1} + O(y^{p+2}).
$$

Note as well that $|R(iy)|^2 < 1$, hence from (B.2)

$$
C_{p+1}(-1)^{\frac{p+1}{2}} < 0.
$$

Hence there exists $\tilde{a}, \nu > 0$ s.t. $|R(x+iy)|^2 > 1$ for all

$$
\left\{(x,\ y) \mid 0 < x < \tilde{a},\ |y| < \nu x^{\frac{1}{p+1}}\right\}.
$$

2. *$p$ is even.*

In this case we will make use of properties of the $E$-polynomial (B.1), similarly to how it was done in the proof of Theorem 7 in [120].

Let us define

$$\psi_y(x) = |R(x+iy)|^2.$$

For a fixed $y$ we can expand the above expression into Taylor series in $x$:

$$\psi_y(x) = |R(iy)|^2 + x\frac{d\psi_y}{dx}(0) + O(x^2). \tag{B.7}$$

Using (B.1), we can rewrite the first term:

$$|R(iy)|^2 = \tfrac{|P(iy)|^2}{|Q(iy)|^2} = 1 - \tfrac{E(y)}{|Q(iy)|^2},$$

which, after expansion into Taylor series (and using the convention $Q(0) = 1$), gives

$$|R(iy)|^2 = 1 - e_0 y^{2s} + O(y^{2s+2}).$$

Next, we need an expression for $\frac{d}{dx}\psi_y(0)$. From (B.3,B.4), for non-even $p$ (using the same arguments as previously),

$$|R(x+iy)|^2 = 1 + 2x + 2C_{p+1}(p+1)xy^p(-1)^{\frac{p}{2}} + O(xy^{p+1} + y^{p+2} + x^2)$$

Hence

$$\frac{d\psi_y}{dx}(0) = 2 + 2C_{p+1}(p+1)y^p(-1)^{\frac{p}{2}} + O(y^{p+1}).$$

Substituting the above into (B.7), we obtain

$$\begin{aligned}\psi_y(x) &= 1 - e_0 y^{2s} + x(2 + 2C_{p+1}(p+1)y^p(-1)^{\frac{p}{2}} + O(y^{p+1})) + O(y^{2s+2}) + O(x^2)\\ &= 1 + y^{2s}\Big(-e_0 + 2xy^{-2s} + 2C_{p+1}(p+1)xy^{p-2s}(-1)^{\frac{p}{2}}\\ &+O\left(xy^{p+1-2s} + x^2y^{-2s}\right) + O(y^2)\Big).\end{aligned}$$

Recall that $e_0 > 0$. From the above expression we can see that $|R(x+iy)|^2 = 1$ in the vicinity of zero for $(x, y)$:

$$x = e_0 y^{2s} + O(y^{2s+1}).$$

Hence there exists $\tilde{q}, \nu > 0$ s.t. $|R(x+iy)|^2 > 1$ for all

$$\left\{(x,\ y) \mid 0 < x < \tilde{q},\ |y| < \nu x^{\frac{1}{2s}}\right\}.$$

Since the bounds derived are asymptotically optimal, this domain indeed intersects all the order star fingers. □

# Appendix C

# Singular Value Decomposition

The following well-known lemma can be found in e.g. [111, Lemma C.2.3, p. 374].

**Lemma C.1.** *Let $M \in \mathbb{R}^{m\times n}$ and let*

$$M = U\Sigma V^T$$

*be its singular value decomposition. The matrix*

$$R_k = U\Sigma_k V^T,$$

*where*

$$(\Sigma_k)_{ij} = \begin{cases} \sigma_i, \ i = j \leq \min(k, m, n), \\ 0, \quad else, \end{cases}$$

*is a rank-k matrix that approximates the matrix $M$ with the error*

$$\|M - R_k\|_2 = \sigma_{k+1},$$
$$\|M - R_k\|_F = \sqrt{\sum_{i=k+1}^{\min(m,n)} \sigma_i^2}.$$

**Remark C.2.** *[111, Corollary C.2.4, p.375] The matrix $R_k$ as defined in Lemma C.1 solves the following two minimization problems:*

$$\min_{\text{rank}(R)\leq k} \|M - R\|_2 \qquad \textit{and} \qquad \min_{\text{rank}(R)\leq k} \|M - R\|_F.$$

*In the case when $\sigma_k > \sigma_{k+1}$, such matrix is unique.*

# References


[1] *NIST digital library of mathematical functions. http://dlmf.nist.gov/, release 1.0.5 of 2012-10-01.*

[2] T. Abboud, M. Pallud, and C. Teissedre, *SONATE: a parallel code for acoustics*, tech. report, Internal Report IMACS - Hewlett Packard. http://imacs.polytechnique.fr/Reports/sonate-parallel.pdf.

[3] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55 of National Bureau of Standards Applied Mathematics Series, For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1972.

[4] A. Aimi, M. Diligenti, A. Frangi, and C. Guardasoni, *Neumann exterior wave propagation problems: computational aspects of 3D energetic Galerkin BEM*, Computational Mechanics, 51 (2013), pp. 475–493.

[5] A. Aimi, M. Diligenti, C. Guardasoni, I. Mazzieri, and S. Panizzi, *An energy approach to space-time Galerkin BEM for wave propagation problems*, Internat. J. Numer. Methods Engrg., 80 (2009), pp. 1196–1240.

[6] A. Aimi, M. Diligenti, and S. Panizzi, *Energetic Galerkin BEM for wave propagation Neumann exterior problems*, CMES Comput. Model. Eng. Sci., 58 (2010), pp. 185–219.

[7] A. Ambroladze and H. Wallin, *Rational interpolants with preassigned poles, theoretical aspects*, Studia Math., 132 (1999), pp. 1–14.

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

[9] D. E. Amos, *Algorithm 644: a portable package for Bessel functions of a complex argument and nonnegative order*, ACM Trans. Math. Software, 12 (1986), pp. 265–273.

[10] R. J. Astley, *Transient wave envelope elements for wave problems*, Journal of Sound and Vibration, 192(1) (1996), pp. 245–261.

[11] R. J. Astley, *Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods*, Springer Berlin Heidelberg, 2008, ch. Infinite Elements.

[12] K. Aygun, M. Lu, N. Liu, A.E. Yilmaz, and E. Michielssen, *A parallel PWTD accelerated time marching scheme for analysis of EMC/EMI problems*, 2 (2003), pp. 863 – 866 Vol.2.

[13] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[14] H. Bagci, A.E. Yilmaz, J.-M. Jin, and E. Michielssen, *Time Domain Adaptive Integral Method for Surface Integral Equations*, vol. 59 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2008.

[15] J. Ballani, *Fast evaluation of near-field boundary integrals using tensor approximations*, dissertation, Universitt Leipzig, 2012.

[16] ——, *Fast evaluation of singular BEM integrals based on tensor approximations*, Numerische Mathematik, 121 (2012), pp. 433–460.

[17] J. Ballani, L. Banjai, S. Sauter, and A. Veit, *Numerical solution of exterior Maxwell problems by Galerkin BEM and Runge-Kutta convolution quadrature*, Numer. Math., 123 (2013), pp. 643–670.

[18] A. Bamberger and T. Ha-Duong, *Formulation variationnelle espace-temps pour le calcul par potentiel retardé de la diffraction d'une onde acoustique (I)*, Mathematical Methods in the Applied Sciences, 8 (1986), pp. 405–435.

[19] ——, *Formulation variationnelle pour le calcul de la diffraction d'une onde acoustique par une surface rigide*, Mathematical Methods in the Applied Sciences, 8 (1986), pp. 598–608.

[20] L. Banjai, *A boundary element method for the solution of Helmholtz problems for a large range of complex wavenumbers.* Presentation at the 23rd Annual GAMM Seminar Leipzig 'Integral Equation Methods for High-Frequency Scattering Problems'.

[21] L. Banjai, *Multistep and multistage convolution quadrature for the wave equation: Algorithms and experiments*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2964–2994.

[22] L. Banjai, *Time-domain Dirichlet-to-Neumann map and its discretization*, (2012). Preprint 5/2012, Max Planck Institute for Mathematics in the Sciences, Leipzig.

[23] L. Banjai and W. Hackbusch, *Hierarchical matrix techniques for low- and high-frequency Helmholtz problems*, IMA Journal of Numerical Analysis, 28 (2008), pp. 46–79.

[24] L. Banjai and M. Kachanovska, *Fast convolution quadrature for the wave equation in three dimensions.* Submitted.

[25] ——, *Sparsity of Runge-Kutta convolution weights for three-dimensional wave equation.* Submitted, 2012.

[26] L. Banjai, A. Laliena, and F.-J. Sayas, *Fully discrete Kirchhoff formulas with CQ-BEM*, Preprint, http://arxiv.org/abs/1301.0267, (2013).

[27] L. Banjai and Ch. Lubich, *An error analysis of Runge-Kutta convolution quadrature*, BIT Numerical Mathematics, 51 (2011), pp. 483–496.

[28] L. Banjai, Ch. Lubich, and J.M. Melenk, *Runge-Kutta convolution quadrature for operators arising in wave propagation*, Numer. Math., 119 (2011), pp. 1–20.

[29] L. Banjai and S. Sauter, *Rapid solution of the wave equation in unbounded domains*, SIAM J. Numerical Analysis, 47 (2008), pp. 227–249.

[30] L. Banjai and M. Schanz, *Wave propagation problems treated with convolution quadrature and BEM*, Lecture Notes in Applied and Computational Mechanics, 63 (2012), pp. 145–184.

[31] A. Bayliss and E. Turkel, *Radiation boundary conditions for wave-like equations*, Comm. Pure Appl. Math., 33 (1980), pp. 707–725.

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

[33] ——, *Hierarchical LU decomposition-based preconditioners for BEM*, Computing, 74 (2005), pp. 225–247.

[34] M. Bebendorf, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, vol. 63 of Lecture Notes in Computational Science and Engineering (LNCSE), Springer-Verlag, 2008. ISBN 978-3-540-77146-3.

[35] M. Bebendorf and R. Grzhibovskis, *Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation*, Math. Methods Appl. Sci., 29 (2006), pp. 1721–1747.

[36] M. Bebendorf and W. Hackbusch, *Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with $L^\infty$-coefficients*, Numer. Math., 95 (2003), pp. 1–28.

[37] E. Bécache, S. Fauqueux, and P. Joly, *Stability of perfectly matched layers, group velocities and anisotropic waves*, J. Comput. Phys., 188 (2003), pp. 399–433.

[38] E. Bécache and P. Joly, *On the analysis of Bérenger's perfectly matched layers for Maxwell's equations*, M2AN Math. Model. Numer. Anal., 36 (2002), pp. 87–119.

[39] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200.

[40] ——, *Three-dimensional perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 127 (1996), pp. 363–379.

[41] T. Betcke, S. N. Chandler-Wilde, I. G. Graham, S. Langdon, and M. Lindner, *Condition number estimates for combined potential integral operators in acoustics and their boundary element discretisation*, Numer. Methods Partial Differential Equations, 27 (2011), pp. 31–69.

[42] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, *AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems*, Radio Science, 31 (1996), pp. 1225–1251.

[43] A. Boag, V. Lomakin, and E. Michielssen, *Nonuniform grid time domain (NGTD) algorithm for fast evaluation of transient wave fields*, IEEE Trans. Antennas and Propagation, 54 (2006), pp. 1943–1951.

[44] S. Börm, *Efficient numerical methods for non-local operators*, vol. 14 of EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2010.

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

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

[47] J. Breuer, *Schnelle Randelementmethoden zur Simulation von elektrischen Wirbelstromfeldern sowie ihrer Wärmeproduktion und Kühlung*, PhD thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2005.

[48] D. Brunner, M. Junge, P. Rapp, M. Bebendorf, and L. Gaul, *Comparison of the fast multipole method with hierarchical matrices for the Helmholtz-BEM*, CMES: Computer Modeling in Engineering & Sciences, 58 (2010), pp. 131–160.

[49] J. C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons Ltd., Chichester, second ed., 2008.

[50] E. Candès, L. Demanet, and L. Ying, *A fast butterfly algorithm for the computation of Fourier integral operators*, Multiscale Model. Simul., 7 (2009), pp. 1727–1750.

[51] Q. Carayol and F. Collino, *Error estimates in the fast multipole method for scattering problems. I. Truncation of the Jacobi-Anger series*, M2AN Math. Model. Numer. Anal., 38 (2004), pp. 371–394.

[52] ——, *Error estimates in the fast multipole method for scattering problems. II. Truncation of the Gegenbauer series*, M2AN Math. Model. Numer. Anal., 39 (2005), pp. 183–221.

[53] C. Cecka and E. Darve, *Fourier-Based Fast Multipole Method for the Helmholtz Equation*, SIAM J. Sci. Comput., 35 (2013), pp. A79–A103.

[54] D. J. Chappell, *A convolution quadrature Galerkin boundary element method for the exterior Neumann problem of the wave equation*, Math. Methods Appl. Sci., 32 (2009), pp. 1585–1608.

[55] ——, *Convolution quadrature Galerkin boundary element method for the wave equation with reduced quadrature weight computation*, IMA J. Numer. Anal., 31 (2011), pp. 640–666.

[56] Q. Chen, P. Monk, X. Wang, and D. Weile, *Analysis of convolution quadrature applied to the time-domain electric field integral equation*, Commun. Comput. Phys., 11 (2012), pp. 383–399.

[57] H. Cheng, W. Y. Crutchfield, Z. Gimbutas, L. F. Greengard, J. F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao, *A wideband fast multipole method for the Helmholtz equation in three dimensions*, J. Comput. Phys., 216 (2006), pp. 300–325.

[58] W. Ch. Chew, J.-M. Jin, E. Michielssen, and J. Song, eds., *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House, 2001.

[59] W. Ch. Chew and W. H. Weedon, *A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microwave and Optical Technology Letters, 7 (1994), pp. 599–604.

[60] C. W. Clenshaw, *A note on the summation of Chebyshev series*, Math. Tables Aids Comput., 9 (1955), pp. 118–120.

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

[62] F. Collino, *High order absorbing boundary conditions for wave propagation models: straight line boundary and corner cases*, in Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993), SIAM, Philadelphia, PA, 1993, pp. 161–171.

[63] F. Collino and P. B. Monk, *Optimizing the perfectly matched layer*, Comput. Methods Appl. Mech. Engrg., 164 (1998), pp. 157–171. Exterior problems of wave propagation (Boulder, CO, 1997; San F.co, CA, 1997).

[64] M. Costabel, *Boundary integral operators on Lipschitz domains: elementary results*, SIAM J. Math. Anal., 19 (1988), pp. 613–626.

[65] ——, *Time-dependent problems with the boundary integral equation method*, in Encyclopedia of Computational Mechanics, Erwin Stein, Rene de Borst, and Thomas J.R. Hughes., eds., John Wiley & Sons, Ltd., 2004.

[66] T. Cruse, *A direct formulation and numerical solution of the general transient elastodynamic problem ii.*, J. Math. Anal. Appl., 22 (1968), pp. 341–355.

[67] T. A. Cruse and F. J. Rizzo, *A direct formulation and numerical solution of the general transient elastodynamic problem i.*, J. Math. Anal. Appl., 22 (1968), pp. 244–259.

[68] E. Darrigrand, *Coupling of fast multipole method and microlocal discretization for the 3-D Helmholtz equation*, J. Comput. Phys., 181 (2002), pp. 126–154.

[69] E. Darve, *The fast multipole method I: error analysis and asymptotic complexity*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 98–128.

[70] E. Darve, *The fast multipole method: numerical implementation*, Journal of Computational Physics, 160 (2000), pp. 195 – 240.

[71] 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 (2004), pp. 603–628.

[72] P. J. Davies, *Numerical stability and convergence of approximations of retarded potential integral equations*, SIAM J. Numer. Anal., 31 (1994), pp. 856–875.

[73] P. J. Davies and D. B. Duncan, *Stability and convergence of collocation schemes for retarded potential integral equations*, SIAM J. Numer. Anal., 42 (2004), pp. 1167–1188 (electronic).

[74] ——, *Convolution-in-time approximations of time domain boundary integral equations*, SIAM J. Sci. Comput., 35 (2013), pp. B43–B61.

[75] P. J. Davis, *Interpolation and approximation*, Dover Publications Inc., New York, 1975. Republication, with minor corrections, of the 1963 original, with a new preface and bibliography.

[76] J. Diaz and P. Joly, *A time domain analysis of PML models in acoustics*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 3820–3853.

[77] V. Dominguez and F.J. Sayas, *Some properties of layer potentials and boundary integral operators for the wave equation*, to appear in Journal of Integral Equations and Applications, (2012).

[78] B.L. Ehle, *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*, ProQuest LLC, Ann Arbor, MI, 1969. Thesis (Ph.D.)–University of Waterloo (Canada).

[79] B. L. Ehle, *A-stable methods and Padé approximations to the exponential*, SIAM J. Math. Anal., 4 (1973), pp. 671–680.

[80] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.

[81] B. Engquist and L. Ying, *Fast directional multilevel algorithms for oscillatory kernels*, SIAM J. Sci. Comput., 29 (2007), pp. 1710–1737 (electronic).

[82] ——, *A fast directional algorithm for high frequency acoustic scattering in two dimensions*, Commun. Math. Sci., 7 (2009), pp. 327–345.

[83] ——, *Fast directional algorithms for the Helmholtz kernel*, J. Comput. Appl. Math., 234 (2010), pp. 1851–1859.

[84] M. A. Epton and B. Dembart, *Multipole translation theory for the three-dimensional Laplace and Helmholtz equations*, SIAM J. Sci. Comput., 16 (1995), pp. 865–897.

[85] A.A. Ergin, B. Shanker, and E. Michielssen, *The plane-wave time-domain algorithm for the fast analysis of transient wave phenomena*, Antennas and Propagation Magazine, IEEE, 41, pp. 39–52.

[86] A. A. Ergin, B.Shanker, and E. Michielssen, *Fast evaluation of three-dimensional transient wave fields using diagonal translation operators*, Journal of Computational Physics, 146 (1998), pp. 157 – 180.

[87] S. Erichsen and S. A. Sauter, *Efficient automatic quadrature in 3-d Galerkin BEM*, Comput. Methods Appl. Mech. Engrg., 157 (1998), pp. 215–224. Seventh Conference on Numerical Methods and Computational Mechanics in Science and Engineering (NMCM 96) (Miskolc).

[88] W. N. Everitt and D. S. Jones, *On an integral inequality*, Proc. Roy. Soc. London Ser. A, 357 (1977), pp. 271–288.

[89] M. Fischer, *The Fast Multipole Boundary Element Method and its Application to Structure-Acoustic Field Interaction*, PhD thesis, University of Stuttgart, 2004.

[90] Matthias Fischer, Holger Perfahl, and Lothar Gaul, *Approximate inverse preconditioning for the fast multipole BEM in acoustics*, Comput. Vis. Sci., 8 (2005), pp. 169–177.

[91] A. Frangi and M. Bonnet, *On the application of the fast multipole method to Helmholtz-like problems with complex wavenumber*, CMES. Computer Modeling in Engineering & Sciences, 58 (2010), pp. 271–295.

[92] S.D. Gedney, *An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices*, Antennas and Propagation, IEEE Transactions on, 44 (1996), pp. 1630–1639.

[93] N. Geng, A. Sullivan, and L. Carin, *Fast multipole method for scattering from an arbitrary PEC target above or buried in a lossy half space*, IEEE Trans. Antennas and Propagation, 49 (2001), pp. 740–748.

[94] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, International Journal for Numerical Methods in Engineering, 79 (2009), pp. 1309–1331.

[95] D. Givoli, *High-order local non-reflecting boundary conditions: a review*, Wave Motion, 39 (2004), pp. 319–326. New computational methods for wave propagation.

[96] D. Givoli, *Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods*, Springer Berlin Heidelberg, 2008, ch. Computational Absorbing Boundaries.

[97] S. Goreinov, *Mosaic-skeleton approximations of matrices, generated by asymptotically smooth and oscillatory kernels*, in Matrix Methods and Computations, E. Tyrtyshnikov, ed., INM RAS, Moscow, 1999, pp. 42–76. (in Russian).

[98] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.

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

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

[101] L. Grasedyck, R. Kriemann, and S. Le Borne, *Domain decomposition based $\mathcal{H}$-LU preconditioning*, Numer. Math., 112 (2009), pp. 565–600.

[102] E. Grasso, S. Chaillat, M. Bonnet, and J.-F. Semblat, *Application of the multi-level time-harmonic fast multipole BEM to 3-D visco-elastodynamics*, Eng. Anal. Bound. Elem., 36 (2012), pp. 744–758.

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

[104] L. F. Greengard and J. Huang, *A new version of the fast multipole method for screened Coulomb interactions in three dimensions*, J. Comput. Phys., 180 (2002), pp. 642–658.

[105] H. Groemer, *Geometric Applications of Fourier Series and Spherical Harmonics*, Cambridge University Press, 1996.

[106] M. J. Grote and J. B. Keller, *Exact nonreflecting boundary conditions for the time dependent wave equation*, SIAM J. Appl. Math., 55 (1995), pp. 280–297. Perturbation methods in physical mathematics (Troy, NY, 1993).

[107] N. A. Gumerov and R. Duraiswami, *Recursions for the computation of multipole translation and rotation coefficients for the 3-D Helmholtz equation*, SIAM J. Sci. Comput., 25 (2003/04), pp. 1344–1381.

[108] T. Ha-Duong, *On retarded potential boundary integral equations and their discretization*, in Topics in Computational Wave Propagation. Direct and Inverse Problems, P. Davies, D. Duncan, P. Martin, and B. Rynne, eds., Springer: Berlin, 2003, pp. 301–336.

[109] T. Ha-Duong, B. Ludwig, and I. Terrasse, *A Galerkin BEM for transient acoustic scattering by an absorbing obstacle*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 1845–1882.

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

[111] ——, *Hierarchische Matrizen: Algorithmen und Analysis*, Springer-Verlag Berlin Heidelberg, 2009.

[112] W. Hackbusch and S. Börm, *$\mathcal{H}^2$-matrix approximation of integral operators by interpolation*, Appl. Numer. Math., 43 (2002), pp. 129–143. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[113] W. Hackbusch, B. Khoromskij, and S. A. Sauter, *On $\mathcal{H}^2$-matrices*, Springer, Berlin, 2000, pp. 9–29.

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

[115] W. Hackbusch, W. Kress, and S. A. Sauter, *Sparse convolution quadrature for time domain boundary integral formulations of the wave equation by cutoff and panel-clustering*, 29 (2007), pp. 113–134.

[116] ——, *Sparse convolution quadrature for time domain boundary integral formulations of the wave equation*, IMA J. Numer. Anal., 29 (2009), pp. 158–179.

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

[118] W. Hackbusch and S. A. Sauter, *On numerical cubatures of nearly singular surface integrals arising in BEM collocation*, Computing, 52 (1994), pp. 139–159.

[119] Th. Hagstrom, *Radiation boundary conditions for the numerical simulation of waves*, in Acta numerica, 1999, vol. 8 of Acta Numer., Cambridge Univ. Press, Cambridge, 1999, pp. 47–106.

[120] E. Hairer, G. Bader, and Ch. Lubich, *On the stability of semi-implicit methods for ordinary differential equations*, BIT Numerical Mathematics, 22 (1982), pp. 211–232.

[121] E. Hairer, Ch. Lubich, and M. Schlichte, *Fast numerical solution of nonlinear Volterra convolution equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 532–541.

[122] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 2010.

[123] E. Hairer, G. Wanner, and S.P. Nørsett, *Solving Ordinary Differential Equations I*, vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 1993.

[124] P. Henrici, *Fast Fourier methods in computational complex analysis*, SIAM Review, 21 (1979), pp. 481–527.

[125] N. J. Higham, *Accuracy and stability of numerical algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[126] R. Jakob-Chien and B. K. Alpert, *A fast spherical filter with uniform resolution*, Journal of Computational Physics, 136 (1997), pp. 580–584.

[127] B. Kaltenbacher, M. Kaltenbacher, and I. Sim, *A modified and stable version of a perfectly matched layer technique for the 3-D second order wave equation in time domain with an application to aeroacoustics*, J. Comput. Phys., 235 (2013), pp. 407–422.

[128] B. Khoromskij, S. Sauter, and A. Veit, *Fast quadrature techniques for retarded potentials based on TT/QTT tensor approximation*, Computational Methods in Applied Mathematics, 11 (2011), pp. 342–362.

[129] L. Kielhorn and M. Schanz, *Convolution quadrature method-based symmetric Galerkin boundary element method for 3-D elastodynamics*, Internat. J. Numer. Methods Engrg., 76 (2008), pp. 1724–1746.

[130] S. Koc, J. Song, and W. Ch. Chew, *Error analysis for the numerical evaluation of the diagonal forms of the scalar spherical addition theorem*, SIAM J. Numer. Anal., 36 (1999), pp. 906–921.

[131] W. Kress and S. Sauter, *Numerical treatment of retarded boundary integral equations by sparse panel clustering*, IMA J. Numer. Anal., 28 (2008), pp. 162–185.

[132] R. Kriemann, *HLIBpro user manual. Technical Report 9/2008*, MPI for Mathematics in the Sciences, Leipzig, 2008.

[133] A. R. Laliena and F.-J. Sayas, *Theoretical aspects of the application of convolution quadrature to scattering of acoustic waves*, Numer. Math., 112 (2009), pp. 637–678.

[134] G. Lancioni, *Numerical comparison of high-order absorbing boundary conditions and perfectly matched layers for a dispersive one-dimensional medium*, Comput. Methods Appl. Mech. Engrg., 209/212 (2012), pp. 74–86.

[135] M. Lu, J. Sarvas, and E. Michielssen, *A simplified 3D plane wave time domain (PWTD) algorithm*, vol. 1, 2001, pp. 188 –191 vol.1.

[136] Ch. Lubich, *Convolution quadrature and discretized operational calculus I*, Numerische Mathematik, 52 (1988), pp. 129–145.

[137] ——, *Convolution quadrature and discretized operational calculus II*, Numerische Mathematik, 52 (1988), pp. 413–425.

[138] ——, *On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations*, Numerische Mathematik, 67 (1994), pp. 365–389.

[139] Ch. Lubich and A. Ostermann, *Runge-Kutta methods for parabolic equations and convolution quadrature*, Mathematics of Computation, 60 (1993), pp. 105–131.

[140] G. Manara, A. Monorchio, and R. Reggiannini, *A space-time discretization criterion for a stable time-marching solution of the electric field integral equation*, Antennas and Propagation, IEEE Transactions on, 45 (Mar), pp. 527–532.

[141] W. J. Mansur, *A time-stepping technique to solve wave propagation problems using the boundary element method*, PhD thesis, University of Southampton, 1983.

[142] W. McLean, *Strongly elliptic systems and boundary integral equations*, (2000), pp. xiv+357.

[143] J. Meng, A. Boag, V. Lomakin, and E. Michielssen, *A multilevel Cartesian non-uniform grid time domain algorithm*, J. Comput. Phys., 229 (2010), pp. 8430–8444.

[144] M. Messner, *Fast Boundary Element Methods in Acoustics*, Verlag der Technischen Universitaet Graz, 2012.

[145] M. Messner, M. Schanz, and E. Darve, *Fast directional multilevel summation for oscillatory kernels based on Chebyshev interpolation*, J. Comput. Phys., 231 (2012), pp. 1175–1196.

[146] P. Meszmer, *Hierarchical quadrature for multidimensional singular integrals*, J. Numer. Math., 18 (2010), pp. 91–117.

[147] P. Meszmer and J. Ballani, *Tensor structured evaluation of singular volume integrals*, Preprint MPI Leipzig, (2012).

[148] E. Michielssen and A. Boag, *Multilevel evaluation of electromagnetic fields for the rapid solution of scattering problems*, Microwave and Optical Technology Letters, 7 (1994), pp. 790–795.

[149] F. W. J. Olver, *Asymptotics and special functions*, AKP Classics, A K Peters Ltd., Wellesley, MA, 1997. Reprint of the 1974 original [Academic Press, New York].

[150] M. O'Neil, F. Woolfe, and V. Rokhlin, *An algorithm for the rapid evaluation of special function transforms*, Appl. Comput. Harmon. Anal., 28 (2010), pp. 203–226.

[151] D. Rabinovich, D. Givoli, and E. Bécache, *Comparison of high-order absorbing boundary conditions and perfectly matched layers in the frequency domain*, Int. J. Numer. Methods Biomed. Eng., 26 (2010), pp. 1351–1369.

[152] J. Rahola, *Diagonal forms of the translation operators in the fast multipole algorithm for scattering problems*, BIT, 36 (1996), pp. 333–358.

[153] S.M. Rao and D.R. Wilton, *Transient scattering by conducting surfaces of arbitrary shape*, Antennas and Propagation, IEEE Transactions on, 39 (Jan), pp. 56–61.

[154] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.

[155] ——, *Diagonal forms of translation operators for the Helmholtz equation in three dimensions*, Applied and Computational Harmonic Analysis, 1 (1993), pp. 82 – 93.

[156] B.P. Rynne and P.D. Smith, *Stability of time marching algorithms for the electric field integral equation*, Journal of Electromagnetic Waves and Applications, 4 (1990), pp. 1181–1205.

[157] L. Grasedyck S. Börm and W. Hackbusch, *Hierarchical Matrices*, Lecture notes for a winter school on the hierarchical matrices, 2003. http://www.mis.mpg.de/publications/other-series/ln/lecturenote-2103.html.

[158] A. Sadigh and E. Arvas, *Treating the instabilities in marching-on-in-time method from a different perspective [electromagnetic scattering]*, Antennas and Propagation, IEEE Transactions on, 41 (Dec), pp. 1695–1702.

[159] Takahiro Saitoh and Sohichi Hirose, *Parallelized fast multipole BEM based on the convolution quadrature method for 3-D wave propagation problems in time-domain*, IOP Conference Series: Materials Science and Engineering, 10 (2010), p. 012242.

[160] T. Saitoh, S. Hirose, T. Fukui, and T. Ishida, *Development of a time-domain fast multipole BEM based on the operational quadrature method in a wave propagation problem*, in Advances in Boundary Element Techniques VIII, V. Minutolo and M.H. Aliabadi, eds., EC, Ltd. UK, 2007.

[161] T. Saitoh, Ch. Zhang, and S. Hirose, *Large-scale multiple scattering analysis using fast multipole BEM in time-domain*, AIP Conference Proceedings, 1233 (2010), pp. 1196–1201.

[162] T. Sakuma, S. Schneider, and Y. Yasuda, *Fast solution methods*, in Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods, S. Marburg and B. Nolte, eds., Springer Berlin Heidelberg, 2008, pp. 333–366.

[163] H. E. Salzer, *Lagrangian interpolation at the Chebyshev points $X_{n,\nu} \equiv \cos(\nu\pi/n)$, $\nu = 0(1)n$; some unnoted advantages*, Comput. J., 15 (1972), pp. 156–159.

[164] J. Sarvas, *Performing interpolation and anterpolation entirely by fast Fourier transform in the 3-D multilevel fast multipole algorithm*, SIAM J. Numer. Anal., 41 (2003), pp. 2180–2196.

[165] S. Sauter and A. Veit, *A Galerkin method for retarded boundary integral equations with smooth and compactly supported temporal basis functions. Part II: Implementation and reference solutions*, Preprint (Universität Zürich, 03/2011).

[166] ——, *A Galerkin method for retarded boundary integral equations with smooth and compactly supported temporal basis functions*, Numer. Math., 123 (2013), pp. 145–176.

[167] S. A. Sauter, *Cubature techniques for 3-D Galerkin BEM*, in Boundary elements: implementation and analysis of advanced algorithms (Kiel, 1996), vol. 54 of Notes Numer. Fluid Mech., Vieweg, Braunschweig, 1996, pp. 29–44.

[168] S. A. Sauter and A. Krapp, *On the effect of numerical integration in the Galerkin boundary element method*, Numer. Math., 74 (1996), pp. 337–359.

[169] S. A. Sauter and Ch. Schwab, *Boundary element methods*, vol. 39 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2011. Translated and expanded from the 2004 German original.

[170] F.-J. Sayas, *Retarded potentials and time domain boundary integral equations: a road-map (march 19, 2013)*. The lecture notes for the workshop on the Theoretical and numerical aspects of inverse problems and scattering theory, La Coruna, Spain, July 4-8, 2011.

[171] A. Schädle, M. López-Fernández, and Ch. Lubich, *Fast and oblivious convolution quadrature*, SIAM J. Sci. Comput., 28 (2006), pp. 421–438 (electronic).

[172] Martin Schanz, *A boundary element formulation in time domain for viscoelastic solids*, Comm. Numer. Methods Engrg., 15 (1999), pp. 799–809.

[173] M. Schanz and H. Antes, *A new visco- and elastodynamic time domain: boundary element formulation*, Comput. Mech., 20 (1997), pp. 452–459.

[174] Y. Shi, M.-Y. Xia, R.-Sh. Chen, E. Michielssen, and M. Lu, *Stable electric field TDIE solvers via quasi-exact evaluation of MOT matrix elements*, IEEE Trans. Antennas and Propagation, 59 (2011), pp. 574–585.

[175] J.M. Song, C.-C. Lu, W.C. Chew, and S. W. Lee, *Fast Illinois solver code (FISC)*, Antennas and Propagation Magazine, IEEE, 40 (1998), pp. 27–34.

[176] J. Song, C.-C. Lu, and W. Ch. Chew, *Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects*, Antennas and Propagation, IEEE Transactions on, 45 (1997), pp. 1488–1493.

[177] O. Steinbach, *Numerical approximation methods for elliptic boundary value problems*, Springer, New York, 2008. Finite and boundary elements, Translated from the 2003 German original.

[178] E. P. Stephan, M. Maischak, and E. Ostermann, *Transient boundary element method and numerical evaluation of retarded potentials*, in Computational Science, ICCS 2008, M. Bubak, G. D. Albada, J. Dongarra, and P. M.A. Sloot, eds., vol. 5102 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 321–330.

[179] M. S. Tong and W. Ch. Chew, *Multilevel fast multipole acceleration in the Nyström discretization of surface electromagnetic integral equations for composite objects*, IEEE Trans. Antennas and Propagation, 58 (2010), pp. 3411–3416.

[180] L. N. Trefethen, *Computing numerically with functions instead of numbers*, Math. Comput. Sci., 1 (2007), pp. 9–19.

[181] S. V. Tsynkov, *Numerical solution of problems on unbounded domains. A review*, Appl. Numer. Math., 27 (1998), pp. 465–532. Absorbing boundary conditions.

[182] H. Wang and S. Xiang, *On the convergence rates of Legendre approximation*, Math. Comp., 81 (2012), pp. 861–877.

[183] G. Wanner, E. Hairer, and S. P. Nørsett, *Order stars and stability theorems*, BIT Numerical Mathematics, 18 (1978), pp. 475–489.

[184] G.N. Watson, *A Treatise on the theory of Bessel functions*, Cambridge University Press, Cambridge, England, 1944.

[185] R.A. Wildman, G. Pisharody, Daniel S. Weile, S. Balasubramaniam, and E. Michielssen, *An accurate scheme for the solution of the time-domain integral equations of electromagnetics using higher order vector bases and bandlimited extrapolation*, Antennas and Propagation, IEEE Transactions on, 52 (Nov.), pp. 2973–2984.

[186] A.C. Woo, H.T.G. Wang, M.J. Schuh, and M.L. Sanders, *Em programmer's notebook-benchmark radar targets for the validation of computational electromagnetics programs*, Antennas and Propagation Magazine, IEEE, 35 (1993), pp. 84–89.

[187] N. Yarvin and V. Rokhlin, *A generalized one-dimensional fast multipole method with application to filtering of spherical harmonics*, J. Comput. Phys., 147 (1998), pp. 594–609.

[188] Y. Yasuda and T. Sakuma, *Analysis of sound fields in porous materials using the fast multipole BEM*, in 37th International Congress and Exposition on Noise Control (Inter-noise 2008), 2008.

[189] A.E. Yilmaz, D.S. Weile, B. Shanker, Jian-Ming Jin, and E. Michielssen, *Fast analysis of transient scattering in lossy media*, Antennas and Wireless Propagation Letters, IEEE, 1 (2002), pp. 14 –17.

[190] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196 (2004), pp. 591–626.

[191] K. Yoshida, *Applications of Fast Multipole Method to Boundary Integral Equation Method*, PhD thesis, Kyoto University, Japan, 2001.

**Selbstständigkeitserklärung**

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den 9. Oktober 2013

.....
(Maryna Kachanovska)

**Daten zum Autor**

---

| | |
|---|---|
| **Name:** | Maryna Kachanovska |
| **Geburtsdatum:** | 21. Februar 1987 in Borowa, Oblast Kiew, Ukraine |
| **09/2003 - 07/2007** | Bachelorstudium der Angenwandte Mathematik<br>Nationale Technische Universität der Ukraine<br>'Kiewer Polytechnisches Institut' |
| **09/2007 - 06/2009** | Masterstudium der Angenwandte Mathematik,<br>Fachrichtung Informatik<br>Nationale Technische Universität der Ukraine<br>'Kiewer Polytechnisches Institut' |
| **seit 10/2009** | Doktorand am Max-Planck-Institut für<br>Mathematik in den Naturwissenschaften |