The Panel-Clustering Method for the Wave Equation in Two Spatial Dimensions.[☆]

Silvia Falletta[a], Stefan A. Sauter[b]

[a] *Dip. Scienze Matematiche "G.L.Lagrange", Politecnico di Torino, C.so Duca degli Abruzzi 24,10129 Torino, Italy, e-mail:* silvia.falletta@polito.it
[b] *Institut für Mathematik, Universität Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland, e-mail:* stas@math.uzh.ch

Abstract

We consider the numerical solution of the wave equation in a two-dimensional domain and start from a boundary integral formulation for its discretization. We employ the convolution quadrature (CQ) for the temporal and a Galerkin boundary element method (BEM) for the spatial discretization. Our main focus is the sparse approximation of the arising sequence of boundary integral operators by panel clustering. This requires the definition of an appropriate admissibility condition such that the arising kernel functions can be efficiently approximated on admissible blocks.

The resulting method has log-linear complexity $\mathcal{O}\left(N\left(N+M\right)q^{4+s}\right)$, $s \in \{0,1\}$, where N is the number of time points, M denotes the dimension of the boundary element space, and $q = \mathcal{O}\left(\log N + \log M\right)$ is the order of the panel-clustering expansion.

Numerical experiments will illustrate the efficiency and accuracy of the proposed CQ-BEM method with panel clustering.

Keywords: Wave equation; convolution quadrature; boundary element method; panel clustering; modified Bessel function

Mathematics Subject Classification (2000): 65N38, 35L05, 33C10, 65F50, 78M16

[☆]This work was partially supported by the Ministero dell'Istruzione, dell'Università e della Ricerca of Italy, under the research programs PRIN09: "Boundary element methods for time dependent problems"and PRIN 2012: "Metodologie innovative nella modellistica differenziale numerica".

1. Introduction

The efficient and reliable simulation of scattered waves in unbounded exterior domains is a numerical challenge and the development of *fast* numerical methods is far from being matured. We are here interested in a boundary integral formulation of the problem to avoid the use of an artificial boundary with approximate transmission conditions [25], [2], [10], [17], [7] but allows to recast the problem (under certain assumptions which will be detailed later) as an integral equation on the surface of the scatterer. As our model problem we consider the homogeneous wave equation

$$\begin{array}{ll} \partial_t^2 u = \Delta u & \text{in } \Omega \times (0,T)\,, \\ u(\cdot,0) = \partial_t u(\cdot,0) = 0 & \text{in } \Omega\,, \\ u = g & \text{on } \Gamma \times (0,T)\,, \end{array} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is either a bounded domain or the exterior of an unbounded domain and $\Gamma := \partial\Omega$. The methods for solving this problem can be split into a) *frequency domain* methods where an incident plane wave at prescribed frequency excites a scattered field and a time periodic ansatz reduces the problem to a purely spatial Helmholtz equation and b) *time-domain* methods where the excitation is allowed to have a broad temporal band width and, possibly, an a-periodic behavior with respect to time.

In our paper we will focus on time-domain methods for the wave equation which is particularly important to model electric or acoustic systems shortly after they are "switched on", i.e., before the system has reached a time-harmonic steady state.

The formulation of (1) as a space-time integral equation by the *retarded acoustic single layer potential* can be written in the form

$$\int_0^t \int_\Gamma k\left(\|x-y\|, t-\tau\right) \varphi\left(y,\tau\right) d\Gamma_y d\tau = g\left(x,t\right) \qquad \forall (x,t) \in \Gamma \times (0,T), \tag{2}$$

where k is the fundamental solution for the acoustic wave equation.

Among the most popular methods for discretizing this equation are: a) the *convolution quadrature* (CQ) method [29], [30], [22], [28], [5], [12] and b) the direct *space-time Galerkin discretization* of (2) (see, e.g., [4], [18], [19], [35], [34], [38]).

The goal of this paper is to present fast solution methods for solving the wave equation in two spatial dimensions via (2) and to base the discretization on the CQ-method. The kernel function is given by applying the inverse Laplace transform $\mathcal{L}^{-1}$ to the *transfer function* K :

$$k\left(r,\bullet\right) := \mathcal{L}^{-1}\left(K\left(r,\bullet\right)\right) = \frac{1}{2\pi\,\mathrm{i}} \int_{I_\sigma} \mathrm{e}^{z\bullet}\, K\left(r,z\right) dz \quad \text{with} \quad K\left(r,z\right) := \frac{1}{2\pi} K_0\left(rz\right)$$

along a vertical contour

$$I_\sigma = \sigma + \mathrm{i}\,\mathbb{R} \quad \text{for some } \sigma > 0,$$

and K_0 being the modified Bessel function (see, e.g., [1, 9.6]). For this problem, we will introduce the panel-clustering method for the sparse representation of the discrete CQ-BEM operators. For problems in *three* spatial dimensional domains $\Omega \subset \mathbb{R}^3$ and Γ being a two-dimensional Lipschitz manifold, a fast version of the convolution quadrature *with BDF2* for the temporal discretization has been developed in [23], [27], [6]. Although there is a reduction with respect to memory and CPU time compared to the conventional approach the arising method is not of optimal complexity $\mathcal{O}(NM)$ (modulo additional factors depending only logarithmically on N and M). In this paper, we consider the panel-clustering method for the CQ-BEM *with BDF1* in two spatial dimensions and prove the log-linear scaling with respect to the total number of unknowns for both, CPU time and memory requirement.

It is well known that the fundamental solution of a second order partial differential equation (PDE) in *even* (spatial) dimensions is more complicated than in *odd* dimensions and new techniques for its approximation have to be developed. The speedup and memory savings of the resulting method is substantial and more significant than for the methods described in [23], [27]: More precisely, if N denotes the number of time steps and M is the dimension of the boundary element space, the storage and computational complexity is $\mathcal{O}\left(N\left(N+M\right)q^{4+s}\right)$ with $q = \mathcal{O}\left(\log\left(NM\right)\right)$ and $s \in \{0,1\}$ instead of $\mathcal{O}\left(NM^2\right)$ for the classical CQ-BEM method. If we assume $M \sim N$, we obtain an optimal complexity (up to logarithmic terms) with respect to the total number of freedoms. We note in passing that boundary integral equations can be used to define transparent transmission conditions at artificial boundaries for wave propagation problems; the above mentioned CQ-BEM method has been proposed in [11] for an efficient discretization of such conditions. The new method we propose here also allows for a sparse realization of such exact non-local transmission conditions in log-linear complexity.

Our new panel-clustering method for the two-dimensional wave equation requires the generalization and combination of quite different discretization techniques such as convolution quadrature, boundary element method, and panel clustering for complicated kernel functions. We recall the definitions of the basic algorithms in order to keep the presentation self contained and to estimate the complexity of the different steps of the algorithm. The paper is organized as follows.

In Section 2, we formulate the convolution quadrature method for the two-dimensional wave equation and introduce the boundary element method for its spatial discretization.

In Section 3, the panel-clustering method based on an abstract *admissibility condition* is introduced while Section 4 is devoted to its implementation. This algorithmic formulation of the method will also play an essential role for the complexity estimates of the method.

The error analysis is carried out in Section 5. We employ functional-type estimates for certain derivatives of modified Bessel and exponential functions, recently presented by the authors in [14], to derive a non-standard admissibility

condition for the panel-clustering approximation of the arising kernel functions. The local approximation error will be estimated and used for the stability and consistency analysis.

In Section 6, we will prove that the storage and computational complexity of the resulting CQ-BEM method with panel clustering behaves like $\mathcal{O}\left(N\left(N+M\right)q^{4+s}\right)$, where $q=\mathcal{O}\left(\log\left(NM\right)\right)$ and $s\in\{0,1\}$.

We will present the results of numerical experiments in Section 7 which demonstrate that the theoretical complexity and error estimates are sharp for the considered model problems.

2. Convolution Quadrature

The starting point is to write (2) as a system of integro-differential equations

$$\int_{\Gamma}\left(\frac{1}{2\pi\,\mathrm{i}}\int_{I_\sigma}K\left(\left\|x-y\right\|,z\right)u\left(z,y,t\right)dz\right)d\Gamma_y=g\left(x,t\right) \tag{3a}$$

$$\partial_t u\left(z,x,t\right)-zu\left(z,x,t\right)-\varphi\left(x,t\right)=0 \tag{3b}$$

for all $x\in\Gamma$, $z\in I_\sigma$, and $t\in(0,T)$ with the initial condition $u\left(z,x,0\right)=0$ (see [29]).

The convolution quadrature is based on a time stepping scheme of (3b) on an equidistant time mesh $t_j=j\Delta t$ with $j=0,\ldots,N$ and $\Delta t=T/N$. The semi-discrete approximations φ_k to the unknown density $\varphi\left(\cdot,t_k\right)$ are given by setting $g_n\left(x\right):=g\left(x,t_n\right)$ and solving

$$\sum_{j=0}^{n}\int_{\Gamma}\omega_{n-j}\left(\left\|x-y\right\|\right)\varphi_j\left(y\right)d\Gamma_y=g_n\left(x\right),\qquad n=0,\ldots,N,\quad x\in\Gamma \tag{4}$$

with weight functions $\omega_j\left(r\right)$ defined implicitly by the formal Taylor series

$$K\left(r,\frac{\gamma\left(\zeta\right)}{\Delta t}\right)=\sum_{n=0}^{\infty}\omega_n\left(r\right)\zeta^n$$

and $\gamma\left(\zeta\right)$ denoting the differentiation symbol of the ODE solver (quotient of the generating polynomials). A-stable backward differentiation formulas of order 1 and 2 are given by

$$\gamma\left(\zeta\right):=\begin{cases}1-\zeta & \text{BDF1 method,}\\ \dfrac{\left(1-\zeta\right)\left(3-\zeta\right)}{2} & \text{BDF2 method.}\end{cases} \tag{5}$$

In general, the weights $\omega_n\left(r\right)$ can be computed by using the contour integral representation

$$\omega_n\left(r\right)=\frac{1}{n!}\left(\partial_\zeta^n K\left(r,\frac{\gamma\left(\zeta\right)}{\Delta t}\right)\right)\Big|_{\zeta=0}=\frac{1}{2\pi\,\mathrm{i}}\oint_{\mathcal{C}}\frac{K\left(r,\frac{\gamma(\zeta)}{\Delta t}\right)}{\zeta^{n+1}}d\zeta, \tag{6}$$

where the contour $\mathcal{C}$ can be chosen as a circle about the origin and radius smaller than one. However, for certain time stepping schemes it is possible to determine ω_n explicitly (see, e.g., [27]). For the BDF1 method, we get

$$\omega_n(r) = \frac{1}{2\pi}\tilde{\omega}_n\left(\frac{r}{\Delta t}\right) \quad \text{with} \quad \tilde{\omega}_n(x) := \frac{(-x)^n}{n!}K_0^{(n)}(x). \tag{7}$$

We introduce the integral operators

$$(\mathcal{K}_j\psi)(x) := \int_\Gamma \omega_j(\|x-y\|)\,\psi(y)\,d\Gamma_y \qquad \forall\, x \in \Gamma$$

so that (4) can be written in the compact form

$$\sum_{j=0}^{n} \mathcal{K}_{n-j}\varphi_j = g_n, \qquad n = 0,\ldots,N. \tag{8}$$

Our goal is to employ the Galerkin boundary element method for the spatial discretization and hence, we multiply with test functions ψ to obtain

$$\sum_{j=0}^{n} \langle \mathcal{K}_{n-j}\varphi_j, \psi\rangle = \langle g_n, \psi\rangle, \qquad n = 0,\ldots,N. \tag{9}$$

For $s \in [-1,1]$, let $H^s(\Gamma)$ denote the usual fractional order Sobolev spaces which are well defined on Lipschitz curves/surfaces Γ. In (9), $\langle\cdot,\cdot\rangle$ denotes the anti-duality pairing in $H^{1/2}(\Gamma)\times H^{-1/2}(\Gamma)$ considered as a continuous extension of the $L^2(\Gamma)$ scalar product.

For the Galerkin boundary element method we introduce a mesh $\mathcal{G}$ on Γ consisting of m panels τ_i. For simplicity, we assume that $\mathcal{G}$ is conforming, i.e., the intersection of any different two panels τ_i, τ_i is either empty, or a common point. The maximal mesh width is denoted by

$$\Delta x := \max\{\Delta_\tau : \tau \in \mathcal{G}\} \quad \text{with} \quad \Delta_\tau := \operatorname{diam}\tau.$$

For any $\tau \in \mathcal{G}$, we choose a bijective pullback $\chi_\tau : \hat{\tau} \to \tau$, where the reference element $\hat{\tau}$ is the unit interval. For $p \in \mathbb{N}_0$ and $k \in \{0,1\}$, let

$$S_{\mathcal{G}}^{p,k} := \left\{\psi \in L^2(\Gamma) \mid \forall \tau \in \mathcal{G} : \psi|_\tau \circ \chi_\tau \in \mathbb{P}_p\right\} \cap H^k(\Gamma), \tag{10}$$

where $\mathbb{P}_p$ is the space of univariate polynomials of maximal degree p. Hence, for $k = 0$ we obtain discontinuous boundary elements while for $k = 1$ the boundary element functions are continuous. If the indices p, k, and $\mathcal{G}$ are clear from the context, we write S short for $S_{\mathcal{G}}^{p,k}$. Let b_i, $1 \le i \le M$, denote the usual nodal basis of S. We assume that $\{b_i : 1 \le i \le M\}$ forms a partition of unity on Γ. For a detailed introduction to Galerkin boundary element methods we refer to [33].

The CQ-BEM discretization of (2) is given by: Find $\varphi_j^{\Delta x} \in S$, $0 \le j \le N$, such that

$$\sum_{j=0}^{n} \left\langle \mathcal{K}_{n-j} \varphi_j^{\Delta x}, \psi \right\rangle = \langle g_n, \psi \rangle , \qquad \forall \psi \in S \text{ and } n = 0, \ldots, N. \tag{11}$$

Let $P^{\Delta x} : L^2(\Gamma) \to S$ be the orthogonal projection, i.e.,

$$\left(P^{\Delta x} u, v\right)_{L^2(\Gamma)} = (u, v)_{L^2(\Gamma)} \qquad \forall v \in S. \tag{12}$$

Then, the Galerkin operator is defined by

$$\mathcal{K}_j^{\Delta x} := P^{\Delta x} \mathcal{K}_j \left(P^{\Delta x}\right)^* \qquad \text{with the adjoint } \left(P^{\Delta x}\right)^* \text{ of } P^{\Delta x}$$

and (11) is equivalent to

$$\sum_{j=0}^{n} \mathcal{K}_{n-j}^{\Delta x} \varphi_j^{\Delta x} = g_n^{\Delta x} \qquad \forall n = 0, \ldots, N \quad \text{with} \quad g_n^{\Delta x} := P^{\Delta x} g_n.$$

The matrix representation of (11) is given by introducing

$$\varphi_j^{\Delta x} = \sum_{k=1}^{M} \alpha_{j,k}^{\Delta x} b_k, \qquad \mathbf{r}_j^{\Delta x} := \left(\langle g_j, b_k \rangle\right)_{k=1}^{M} \qquad \forall 0 \le j \le N$$

and

$$\mathbf{K}_j^{\Delta x} = \left(a_{j,\ell,m}\right)_{\ell,m=1}^{M} \quad \text{with} \quad a_{j,\ell,m} := \left(\langle \mathcal{K}_j b_m, b_\ell \rangle\right)_{m,\ell=1}^{M}$$

and then to solve the algebraic system of equations for the coefficient vectors $\boldsymbol{\alpha}_j := \left(\alpha_{j,k}^{\Delta x}\right)_{k=1}^{M}$:

$$\sum_{j=0}^{n} \mathbf{K}_{n-j}^{\Delta x} \boldsymbol{\alpha}_j = \mathbf{r}_n^{\Delta x} \qquad \forall\, 0 \le n \le N. \tag{13}$$

Since $\mathbf{K}_0^{\Delta x}$ is nonsingular and $g(x, t_0) \equiv u(x, t_0) = 0$ (for the compatibility conditions on the data), we have $\varphi_0^{\Delta x} = 0$. Therefore, the linear system is solved for $1 \le n \le N$.

3. Fast Solution Method by Panel Clustering

The linear system (13) has dimension $(NM) \times (NM)$ and its efficient generation and solution are the major bottlenecks in the overall numerical solution process. In a first step one has to generate the block matrices $\mathbf{K}_j^{\Delta x} \in \mathbb{C}^{M \times M}$ for $0 \le j \le N$, and then the block triangular Toeplitz system has to be solved.

One fast approach employs the block Toeplitz structure of (13): the *generation* of the matrices $\mathbf{K}_j^{\Delta x}$ requires $\mathcal{O}\left(NM^2\right)$ operations while its *solution*

then can be performed with FFT-type techniques (cf. [26]) in $\mathcal{O}\left(NM^2\right)$ operations up to logarithmic terms – instead of $\mathcal{O}\left(N^2M^2\right)$ operations for a naive implementation.

In our paper we present an approach which aims for a complexity (up to logarithmic terms) of $\mathcal{O}\left(NM\right)$ operations for the generation of the linear system and $\mathcal{O}\left(N^2M\right)$ for its solution. The development of a *fast* solver with linear complexity is the topic of future research.

The approximation is based on *panel clustering* which was introduced in [27] for the BDF2 discretization of the *three*-dimensional wave equation. The panel-clustering method (see, e.g., [24], [16], [36], [33]) is a sparse representation of the operators $\mathcal{K}_j^{\Delta x}$ which allows for a fast matrix-vector multiplication

$$\boldsymbol{\beta} = \mathbf{K}_n^{\Delta x}\boldsymbol{\alpha}.$$

Before going into the details we state that the kernel functions $\omega_j\left(r\right)$ have a peak around $r \approx t_j$, decay exponentially for $r \gg t_j$, and have a "tail" towards $r = 0$ which is not tending to zero. The idea of the panel clustering is to subdivide the domain $\Gamma \times \Gamma$ into blocks $(c_1, c_2) \subset \Gamma \times \Gamma$ such that ω_n allows for a separable approximation

$$\omega_n\left(\|x-y\|\right) \approx \sum_{(\nu,\mu)\in\mathcal{M}_q} \kappa_n^{(\nu,\mu)}\left(c_1,c_2\right)\Phi_{c_1,\nu}^{(1)}\left(x\right)\Phi_{c_2,\mu}^{(2)}\left(y\right), \tag{14}$$

for certain expansion functions $\Phi_{c,\nu}^{(1)}$, $\Phi_{c,\nu}^{(2)}$, expansion coefficients $\kappa_n^{(\nu,\mu)}\left(c_1,c_2\right)$, and index set $\mathcal{M}_q$.

3.1. Cluster Tree and Block Partitioning

We begin with the basic definitions. Let $\mathcal{I} := \{1,\ldots,M\}$ denote the index set for the spatial basis functions b_i, $i \in \mathcal{I}$.

Definition 1. *A cluster is a subset of* $\mathcal{I}$. *The* area *of a cluster* c *is*

$$\operatorname{supp}\left(c\right) := \bigcup_{j\in c}\operatorname{supp}\left(b_j\right)$$

and its diameter

$$d_c := \operatorname{diam}\left(\operatorname{supp}\left(c\right)\right).$$

The cluster box Q_c *is the (closed) minimal axes parallel rectangle which contains* $\operatorname{supp}\left(c\right)$ *and its barycentre* m_c *is the* cluster center.

The minimal distance $\delta_{\min}$ *and the* maximal distance $\delta_{\max}$ *of a* pair *of clusters* (c_1, c_2) *are*

$$\delta_{\min}\left(c_1,c_2\right) := \min\left\{\|z\| : z \in Q_{c_1} - Q_{c_2}\right\}, \quad \delta_{\max}\left(c_1,c_2\right) := \max\left\{\|z\| : z \in Q_{c_1} - Q_{c_2}\right\},$$

where

$$Q_{c_1} - Q_{c_2} := \left\{x - y : x \in Q_{c_1}, y \in Q_{c_2}\right\}.$$

For the efficiency of the algorithm it is important not to consider all possible clusters but impose the following tree structure.

Definition 2. *The cluster tree* $\mathbb{T}$ *is a set of clusters which has the following tree structure*

1. $\mathcal{I} \in \mathbb{T}$
2. *For any* $c \in \mathbb{T}$ *it holds*
   (a) *either* $c = \{i\}$ *consists of a single index (then* c *is said to be a* leaf*),*
   (b) *or there are "sons"* $s_i \in \mathbb{T}$, $1 \leq i \leq n_c$, *which are disjoint and satisfy* $c = \bigcup_{i=1}^{n_c} s_i$. *The set* $\{s_i : 1 \leq i \leq n_c\}$ *is denoted by* $\operatorname{sons}(c)$.

If $c = \{i\}$ *we set* $\operatorname{sons}(c) = \emptyset$.

Definition 3. *For* $c \in \mathbb{T}$, *the* mollified characteristic function *is*

$$b_c := \sum_{i \in c} b_i$$

and the restriction *of a function* $w : \Gamma \times \Gamma \to \mathbb{C}$ *to a pair of clusters* (c_1, c_2) *is*

$$w^{(c_1,c_2)}(x,y) := b_{c_1}(x)\, w(x,y)\, b_{c_2}(y). \tag{15}$$

Next we will derive a minimal decomposition of $\mathcal{I} \times \mathcal{I}$ into admissible *pairs* of clusters. Since the kernel functions ω_n depend on the time step n, the decomposition also depends on n and a given accuracy $\varepsilon > 0$. The admissibility condition will be a consequence of the error analysis and ensures that the kernel function $\omega_n(\|x-y\|)$ can be approximated by a separable expansion (cf. (14)) with accuracy ε.

3.1.1. The Admissible Condition for Panel-Clustering for the Wave Equation

In this section, we will formulate the admissibility condition for panel-clustering method for the wave equation which depends on the time step n. The definition will be justified by the local error analysis in Section 5.1. For the control parameters $\eta, c, \tilde{C}$, $\delta_0 = \mathcal{O}(1)$ and $0 < \varepsilon \ll 1$, the theory does not give sharp enough insights on their optimal choice but only proves that these constants are independent of the discretization parameters. We have performed numerical experiments (see Section 7) and it turned out that $\eta = \tilde{C} = 1$, $\delta_0 = \frac{3}{2}$, $c = 1/2$, and $\varepsilon = 10^{-8}$ are good choices for these parameters.

Definition 4. *The* admissibility function adm *depends on control parameters* $0 < \varepsilon \ll 1$, $1 < \delta_0 < 2$, *and* $\eta, c, \tilde{C} = \mathcal{O}(1)$. *For* $0 \leq n \leq N$, $(c_1, c_2) \in \mathbb{T} \times \mathbb{T}$ *the function* $\operatorname{adm}((c_1,c_2), n)$ *has the value:*

1. *"*$\operatorname{nonadm}, \mathrm{FFT}$*", if both,* c_1 *and* c_2, *are leaves and*

$$\max\{d_{c_1}, d_{c_2}\} > \eta \delta_{\min}(c_1, c_2); \tag{16}$$

2. *"*adm,far*", if the condition in Case 1 is violated and*

$$\delta_{\min}(c_1, c_2) \geq \delta_0 \max\left\{\Delta t, t_n\left(1 + \frac{1}{\sqrt{n}}\right)\right\} \tag{17}$$

as well as

$$n \geq \frac{\log^2 \frac{3}{\varepsilon}}{(\delta_0 - 1)^2} \tag{18}$$

hold;

3. *"*adm,near*", if the conditions in Cases 1,2 are violated and*

$$\begin{array}{rcl} (\delta_{\max}(c_1, c_2) \leq ct_n & \wedge & \max\{d_{c_1}, d_{c_2}\} \leq \eta\delta_{\min}(c_1, c_2)) \\ \vee & & \\ (\delta_{\min}(c_1, c_2) \geq (2+\sqrt{2})\, t_n & \wedge & \max\{d_{c_1}, d_{c_2}\} \leq \eta\delta_{\min}(c_1, c_2)) \end{array} \tag{19}$$

as well as

$$2 + 4\log(n+1) \leq 2\left\lceil \frac{\log \frac{\tilde{C}}{\varepsilon}}{\log 2} \right\rceil \leq \sqrt{n} \tag{20}$$

hold;

4. *"*adm,peak*", if the conditions in Cases 1-3 are violated and*

$$ct_n \leq \delta_{\max}(c_1, c_2) \wedge \delta_{\min}(c_1, c_2) \leq \left(2+\sqrt{2}\right) t_n \wedge \max\{d_{c_1}, d_{c_2}\} \leq \eta \frac{\delta_{\min}(c_1, c_2)}{\sqrt{n+1}} \tag{21}$$

holds;

5. *"*nonadm, direct*", if the conditions in Cases 1-4 are violated.*

Next, we will formulate the procedure for the decomposition of $\mathcal{I} \times \mathcal{I}$ into parts which correspond to the cases described in Definition 4. We start initializing both, the sets $\mathcal{C}_n^{\text{adm,near}}, \mathcal{C}_n^{\text{adm,far}}, \mathcal{C}_n^{\text{adm,peak}}$ of admissible pairs of clusters as well as the sets $\mathcal{C}_n^{\text{nonadm,FFT}}, \mathcal{C}_n^{\text{nonadm,direct}}$ of non-admissible pairs of leaves as the empty sets. For brevity we introduce

$$\overrightarrow{\mathcal{C}_n} := \left\{\mathcal{C}_n^{\text{adm,near}}, \mathcal{C}_n^{\text{adm,far}}, \mathcal{C}_n^{\text{adm,peak}}, \mathcal{C}_n^{\text{nonadm,FFT}}, \mathcal{C}_n^{\text{nonadm,direct}}\right\}$$

and call the procedure **divide** with

$$\textbf{divide}\left(\overrightarrow{\mathcal{C}_n}, (\mathcal{I}, \mathcal{I}), n\right);$$

and define the procedure as follows:

procedure divide$\left(\overrightarrow{\mathcal{C}_n}, (c_1, c_2), n\right)$;
begin
  if adm$((c_1, c_2), n) =$"nonadm, FFT"
    then $\mathcal{C}_n^{\text{nonadm,FFT}} := \mathcal{C}_n^{\text{nonadm,FFT}} \cup \{(c_1, c_2)\}$
  elseif adm$((c_1, c_2), n) =$"adm, far" **then** $\mathcal{C}_n^{\text{adm,far}} := \mathcal{C}_n^{\text{adm,far}} \cup \{(c_1, c_2)\}$

elseif $\operatorname{adm}((c_1,c_2),n) =$"adm, near" **then** $\mathcal{C}_n^{\mathrm{adm,near}} := \mathcal{C}_n^{\mathrm{adm,near}} \cup \{(c_1,c_2)\}$
elseif $\operatorname{adm}((c_1,c_2),n) =$"adm, peak" **then** $\mathcal{C}_n^{\mathrm{adm,peak}} := \mathcal{C}_n^{\mathrm{adm,peak}} \cup \{(c_1,c_2)\}$
else
  if[1] $c_1 = \{i\}$ **and** $c_2 = \{j\}$
    then $\mathcal{C}_n^{\mathrm{nonadm,direct}} := \mathcal{C}_n^{\mathrm{nonadm,direct}} \cup \{(c_1,c_2)\}$
  else for $(s_1,s_2) \in \operatorname{sons}(c_1,c_2)$ **divide**$\left(\overrightarrow{\mathcal{C}_n},(s_1,s_2),n\right)$;
end;

Remark 5. *Let* $\mathcal{C}_n^{\mathrm{adm}} := \mathcal{C}_n^{\mathrm{adm,near}} \cup \mathcal{C}_n^{\mathrm{adm,far}} \cup \mathcal{C}_n^{\mathrm{adm,peak}}$ *and* $\mathcal{C}^{\mathrm{nonadm}} := \mathcal{C}_n^{\mathrm{nonadm,FFT}} \cup \mathcal{C}_n^{\mathrm{nonadm,direct}}$. *The union of* $\mathcal{C}_n := \mathcal{C}_n^{\mathrm{adm}} \cup \mathcal{C}_n^{\mathrm{nonadm}}$ *is a disjoint partitioning of* $\mathcal{I} \times \mathcal{I}$. *Note that the set* $\mathcal{C}_n^{\mathrm{nonadm,FFT}}$ *does not depend on* n *and we write short* $\mathcal{C}^{\mathrm{nonadm,FFT}}$.

Since the basis functions b_i form a partition of unity, any function $w : \Gamma \times \Gamma \to \mathbb{C}$ satisfies

$$w = \sum_{(c_1,c_2)\in\mathcal{C}_n} w^{(c_1,c_2)}.$$

By using this decomposition the sesquilinear form $\langle \mathcal{K}_n u, v\rangle$ can be written in the form

$$\begin{aligned}\langle \mathcal{K}_n u, v\rangle = &\sum_{(\{i\},\{j\})\in\mathcal{C}_n^{\mathrm{nonadm}}} \int_{\mathrm{supp}(b_i)\times\mathrm{supp}(b_j)} \overline{v}(x)\, \omega_n^{(\{i\},\{j\})}(\|x-y\|)\, u(y)\, d\Gamma_y d\Gamma_x \\ &+ \sum_{(c_1,c_2)\in\mathcal{C}_n^{\mathrm{adm}}} \int_{\mathrm{supp}(c_1)\times\mathrm{supp}(c_2)} \overline{v}(x)\, \omega_n^{(c_1,c_2)}(\|x-y\|)\, u(y)\, d\Gamma_y d\Gamma_x.\end{aligned}$$

Now we assume that the localized kernel function $\omega_n^{(c_1,c_2)}$ can be approximated on admissible clusters $(c_1,c_2) \in \mathcal{C}_n^{\mathrm{adm}}$ by a separable expansion

$$\omega_n^{(c_1,c_2)}(\|x-y\|) \approx \sum_{(\nu,\mu)\in\mathcal{M}_q} \kappa_n^{(\nu,\mu)}(c_1,c_2)\, \Phi_{c_1,\nu}^{(1)}(x)\, \Phi_{c_2,\mu}^{(2)}(y), \tag{22}$$

where $\mathcal{M}_q \subset \mathbb{N}_0^4$ is an index set with q, typically, depending on the block (c_1,c_2) and a given tolerance ε; $\Phi_{c_1,\nu}^{(1)}$ and $\Phi_{c_2,\mu}^{(2)}$ are suitable expansion functions and $\kappa_n^{(\nu,\mu)}(c_1,c_2)$ are the cluster-cluster interaction coefficients. In Section 3.2 we explain how such a separable expansion can be derived from interpolation.

[1] The set of sons for a pair of clusters is defined by

$$\operatorname{sons}(c_1,c_2) := \begin{cases} \{(s_1,s_2) : s_1 \in \operatorname{sons}(c_1), s_2 \in \operatorname{sons}(c_2)\} & \text{if } \operatorname{sons}(c_1)\neq\emptyset \wedge \operatorname{sons}(c_2)\neq\emptyset, \\ \{(s_1,c_2) : s_1 \in \operatorname{sons}(c_1)\} & \text{if } \operatorname{sons}(c_1)\neq\emptyset \wedge \operatorname{sons}(c_2)=\emptyset, \\ \{(c_1,s_2) : s_2 \in \operatorname{sons}(c_2)\} & \text{if } \operatorname{sons}(c_1)=\emptyset \wedge \operatorname{sons}(c_2)\neq\emptyset, \\ \emptyset & \text{if } \operatorname{sons}(c_1)=\emptyset \wedge \operatorname{sons}(c_2)=\emptyset. \end{cases}$$

Replacing the kernel function $\omega_n^{(c_1,c_2)}$ for the admissible integrals we obtain the following approximate representation of the bilinear form

$$\langle \mathcal{K}_n u, v\rangle \approx \sum_{(\{i\},\{j\})\in\mathcal{C}_n^{\mathrm{nonadm}}} \int_{\mathrm{supp}(b_i)\times\mathrm{supp}(b_j)} \overline{v}(x)\,\omega_n^{(\{i\},\{j\})}(\|x-y\|)\,u(y)\,d\Gamma_y d\Gamma_x \tag{23}$$

$$+ \sum_{(c_1,c_2)\in\mathcal{C}_n^{\mathrm{adm}}} \sum_{(\nu,\mu)\in\mathcal{M}_q} \kappa_n^{(\nu,\mu)}(c_1,c_2)\,J_{c_1,\nu}^{(1)}(\overline{v})\,J_{c_2,\mu}^{(2)}(u)$$

where the *farfield coefficients* are given by

$$J_{c,\nu}^{(1)}(v) := \int_{\mathrm{supp}(c)} \Phi_{c,\nu}^{(1)}(x)\,v(x)\,d\Gamma_x \quad \text{and} \quad J_{c,\nu}^{(2)}(u) := \int_{\mathrm{supp}(c)} \Phi_{c,\nu}^{(2)}(x)\,u(x)\,d\Gamma_x.$$

Let us assume (cf. (27)) that the function systems $\Phi_{c,\nu}^{(1)}$ and $\Phi_{c,\nu}^{(2)}$ satisfy the following recursion: For all $s\in\mathrm{sons}(c)$ it holds

$$\Phi_{c,\nu}^{(1)} = \sum_{s\in\mathrm{sons}(c)} \sum_{\mu\in M_q} t_{(c,s),(\nu,\mu)}^{(1)} \Phi_{s,\mu}^{(1)} \quad \text{and} \quad \Phi_{c,\nu}^{(2)} = \sum_{s\in\mathrm{sons}(c)} \sum_{\mu\in M_q} t_{(c,s),(\nu,\mu)}^{(2)} \Phi_{s,\mu}^{(2)}$$

for some *transfer* coefficients $t_{(c,s),(\nu,\mu)}^{(1)}$, and $t_{(c,s),(\nu,\mu)}^{(2)}$. Then the farfield coefficients can be computed by a recursion over the cluster tree: for all admissible clusters $c\in\mathcal{C}_n^{\mathrm{adm}}$ and sons $s\in\mathrm{sons}(c)$ we have

$$J_{c,\nu}^{(1)}(v) = \sum_{s\in\mathrm{sons}(c)} \sum_{\mu\in M_q} t_{(c,s),(\nu,\mu)}^{(1)} J_{s,\mu}^{(1)}(v) \text{ and } J_{c,\nu}^{(2)}(u) = \sum_{s\in\mathrm{sons}(c)} \sum_{\mu\in M_q} t_{(c,s),(\nu,\mu)}^{(2)} J_{s,\mu}^{(2)}(u). \tag{24}$$

3.2. Construction of a Separable Expansion by Interpolation

Let $\widehat{\Theta} := \left\{\widehat{\xi}_i : 0\le i\le q\right\}$ denote the Chebyshev interpolation points on the interval $[-1,1]$ and let $\widehat{L}_i\in\mathbb{P}_q$, $0\le i\le q$, be the corresponding Lagrange basis. For an interval $\tau := [a,b]$ we define the affine coordinate transform $\chi_\tau(\hat{x}) := \frac{1-\hat{x}}{2}a + \frac{1+\hat{x}}{2}b$ and the transformed Lagrange functions $L_{\tau,i} := \widehat{L}_i\circ\chi_\tau^{-1}$. The transformed interpolation points are $\xi_{\tau,i} := \chi_\tau\left(\widehat{\xi}_i\right)$.

Let $M_q := \{0\le i\le q\}^2$. For a rectangle $\tau_1\times\tau_2$ we define the two-dimensional Lagrange functions for $\nu=(\nu_1,\nu_2)\in M_q$ by

$$L_{Q,\nu} := L_{\tau_1,\nu_1}\otimes L_{\tau_2,\nu_2}$$

and the two-dimensional interpolation points by $\xi_{Q,\nu} := (\xi_{\tau_1,\nu_1},\xi_{\tau_2,\nu_2})^{\mathsf{T}}$. Let $(c_1,c_2)\in\mathcal{C}_n^{\mathrm{adm}}$. We approximate $\omega_n^{(c_1,c_2)}(\|x-y\|)$ for $(x,y)\in\mathrm{supp}(c_1)\times\mathrm{supp}(c_2)$ by interpolation

$$\omega_n^{(c_1,c_2)}(\|x-y\|) \approx \omega_{n,q}^{(c_1,c_2)}(x,y) \tag{25}$$

$$:= \sum_{(\nu,\mu)\in M_q\times M_q} \omega_n\left(\|\xi_{Q(c_1),\nu}-\xi_{Q(c_2),\mu}\|\right) b_{c_1}(x)\,L_{Q(c_1),\nu}(x)\,b_{c_2}(y)\,L_{Q(c_2),\mu}(y).$$

By setting

$$\mathcal{M}_q := M_q \times M_q, \quad \kappa_n^{(\nu,\mu)}(c_1,c_2) := \omega_n\left(\|\xi_{Q(c_1),\nu} - \xi_{Q(c_2),\mu}\|\right), \quad \Phi_{c,\nu}^{(1)} = \Phi_{c,\nu}^{(2)} = b_c L_{Q(c),\nu}, \tag{26}$$

we have derived an expansion of the form (22) which obviously satisfy the refinement relation

$$\Phi_{c,\nu}^{(1)} = \sum_{s\in\mathrm{sons}(c)} \sum_{\mu\in M_q} t_{(c,s),(\nu,\mu)}^{(1)} \Phi_{s,\mu}^{(1)} \quad \text{with} \quad t_{(c,s),(\nu,\mu)}^{(1)} := \Phi_{c,\nu}^{(1)}\left(\xi_{Q(s),\mu}\right). \tag{27}$$

3.3. Definition of the Panel-Clustering Method

Let

$$\omega_n^{\mathrm{pc}}(x,y) := \sum_{(c_1,c_2)\in\mathcal{C}_n^{\mathrm{nonadm}}} \omega_n^{(c_1,c_2)}(x,y) + \sum_{(c_1,c_2)\in\mathcal{C}_n^{\mathrm{adm}}} \omega_{n,q}^{(c_1,c_2)}(x,y), \tag{28}$$

with $\omega_{n,q}^{(c_1,c_2)}$ as in (25). The panel-clustering approximation to the sesquilinear form $\langle \mathcal{K}_n\cdot,\cdot\rangle : S\times S\to\mathbb{C}$ is then given by

$$\langle \mathcal{K}_n\phi,\psi\rangle \approx \langle \mathcal{K}_n^{\mathrm{pc}}\phi,\psi\rangle = \int_\Gamma \overline{\psi}(x)\left(\int_\Gamma \omega_n^{\mathrm{pc}}(x,y)\,\phi(y)\,d\Gamma_y\right) d\Gamma_x \qquad \forall \phi,\psi\in S. \tag{29}$$

We have now all ingredients to formulate the CQ-BEM with panel clustering for the discretization of the retarded potential integral equation (2). Let $\mathcal{K}_n$ be defined as in (29). Then, we are seeking $\varphi_j^{\mathrm{pc},\Delta x}\in S$, $0\le j\le N$, such that

$$\sum_{k=0}^{N} \left\langle \mathcal{K}_{n-k}^{\mathrm{pc}}\varphi_k^{\mathrm{pc},\Delta x},\psi\right\rangle = \int_\Gamma g_n\overline{\psi}, \qquad \forall \psi\in S \text{ and } n=0,\ldots,N. \tag{30}$$

4. Algorithmic Realization

The efficient solution of the CQ-BEM with panel clustering (cf. (30)) employs a two-fold hierarchy: the geometric hierarchy via Definition 2 and the hierarchy for the expansion functions by (24). Note that the "algebraization" of this two-fold hierarchy is the key idea behind the $\mathcal{H}^2$-matrices which have been introduced in [21] and further developed in [8].

4.1. Evaluation of the Kernel Functions ω_n

Before we will formulate the algorithms for the different sub-problems in the panel-clustering procedure, we will explain how the kernel functions ω_n can be evaluated efficiently. This a non-trivial task since the first definition in (6) involves the evaluation of high order derivatives of the modified Bessel function K_0 and this is numerically unstable, while the contour integral representation in (6) can be efficiently treated by FFT only if a value $\omega_n(r)$ for a fixed r is needed for all $0\le n\le N$ as explained below.

We distinguish between the nearfield $\mathcal{C}^{\mathrm{nonadm,FFT}}$ which is independent of the time step n and the remaining parts of the partition which depend on n. We set

$$\delta_{\min} := \min\left\{\delta_{\min}(c_1,c_2) : (c_1,c_2) \in \left(\bigcup_{n=0}^{N} \mathcal{C}_n\right) \setminus \mathcal{C}^{\mathrm{nonadm,FFT}}\right\}.$$

Note that condition (16) implies that $\delta_{\min} = \mathcal{O}\left(M^{-1}\right)$. Let $D := \operatorname{diam}\Gamma$ and our goal is to approximate $\omega_n : [\delta_{\min}, D] \to \mathbb{C}$ uniformly for all $0 \le n \le N$.

Recall the well-known error estimate for one-dimensional Chebyshev interpolation $I_{\tau,q}(f) \in \mathbb{P}_q$ for a sufficiently smooth function $f : \tau \to \mathbb{C}$ on an interval $\tau = [a,b]$ of length L

$$|f(r) - I_{\tau,q}(f)(r)| \le \frac{L^{q+1}}{2^{2q+1}(q+1)!}\left\|f^{(q+1)}\right\|_{L^\infty(\tau)}.$$

We employ the general estimate from [14, Theorem 1] (see (38)) to obtain

$$\left|\omega_n^{(q)}(r)\right| \le Cq!\left(\frac{\beta\sqrt{n+1}}{r}\right)^q$$

and, in turn,

$$|\omega_n(r) - I_{\tau,q}(\omega_n)(r)| \le \tilde{C}\left(\frac{\beta L\sqrt{n+1}}{4a}\right)^{q+1}.$$

Hence, if we choose $L \le \frac{4\eta a}{\beta\sqrt{n+1}}$ for some $0 < \eta < 1$, we obtain exponential convergence. This condition is satisfied if we partition $[\delta_{\min}, D]$ into intervals $\tau_i = [b_{i+1}, b_i] \subset [0, D]$ with $b_{i+1} := b_i - L_i$ which satisfy

$$L_i \le \frac{4\eta}{\beta\sqrt{N+1}} b_{i+1}.$$

Such a partitioning can be easily constructed via the recursion $b_0 := D$ and

$$\begin{aligned} &L_i := \tfrac{4\eta}{\beta\sqrt{N+1}}\left(1 + \tfrac{4\eta}{\beta\sqrt{N+1}}\right)^{-1} b_i, \quad (\text{i.e., } L_i = \tfrac{4\eta}{\beta\sqrt{N+1}}(b_i - L_i),)\\ &b_{i+1} := b_i - L_i. \end{aligned}$$

Let $\varepsilon := \frac{4\eta}{\beta\sqrt{N+1}}\left(1 + \frac{4\eta}{\beta\sqrt{N+1}}\right)^{-1}$ and assume $0 < \varepsilon < 1$ which is satisfied for practical cases. Then, we get

$$b_i = (1-\varepsilon)^i D. \tag{31}$$

We run this procedure for $i = 0, \ldots, i_*$, where i_* is the smallest integer such that $b_{i_*} < \delta_{\min}$, i.e.,

$$i_* = \left\lceil \frac{\log(D/\delta_{\min})}{\log\left(\frac{1}{1-\varepsilon}\right)} \right\rceil = \mathcal{O}\left(\frac{\log \delta_{\min}^{-1}}{\varepsilon}\right) = \mathcal{O}\left(\sqrt{N+1}\log M\right). \tag{32}$$

Hence, if we choose the sequence $(b_i)_{i=0}^{i_*}$ as in (31) we have partitioned $[\delta_{\min}, D]$ into i_* intervals τ_i, where the Chebyshev interpolation satisfies

$$|\omega_n(r) - I_{\tau_i,q}(\omega_n)(r)| \leq \tilde{C}\eta^{q+1} \qquad \forall t \in \tau_i$$

uniformly for all $0 \leq n \leq N$.

Let $\xi_{\tau_i,j}$, $0 \leq j \leq q$, denote the Chebyshev points in τ_i and let $\Theta_q := \{\xi_{\tau_i,j} : 0 \leq j \leq q, \quad 0 \leq i \leq i_*\}$ be the collection of all these Chebyshev points. Then, it is sufficient to compute the values $(\omega_n(\xi))_{\xi\in\Theta_q}$ in order to obtain a highly accurate Chebyshev interpolation of ω_n in $[\delta_{\min}, D]$. These values can be computed by using the contour integral representation (6) along its approximation by a trapezoidal rule with $2N+1$ points as recommended in [29, 31]:

$$\omega_n(r) = \frac{1}{4\pi^2\,\mathrm{i}} \oint_{\mathcal{C}} \frac{K_0\left(r\frac{\gamma(\zeta)}{\Delta t}\right)}{\zeta^{n+1}} d\zeta \approx \frac{\lambda^{-n}}{2N+1} \sum_{\ell=0}^{2N} K_0(rs_\ell)\zeta_{2N+1}^{\ell n}, \qquad (33)$$

where $\zeta_{2N+1} = \exp\left(\dfrac{2\pi\,\mathrm{i}}{2N+1}\right), \quad s_\ell = \dfrac{\gamma(\lambda\zeta_{2N+1}^{-\ell})}{\Delta t}.$

Remark 6. *As explained, e.g., in [6, Rem. 5.11], λ in (33) should be chosen in the range $\sqrt{\mathrm{eps}} < \lambda^N < 1$, where* eps *is the machine accuracy. In IEEE double precision this is approximately 10^{-16}; therefore the accuracy of the method is limited by the choice $\lambda > 10^{-8/N}$.*

By employing FFT techniques the values $(\omega_n(r))_{n=0}^N$ can be computed in $\mathcal{O}(N \log N)$ operations. Hence, the computation of all $\omega_n(\xi)$, $\xi \in \Theta_q$ and $0 \leq n \leq N$ requires $\mathcal{O}\left(N^{3/2} q (\log N)(\log M)\right)$ operations and $\mathcal{O}\left(N^{1/2} q \log M\right)$ quantities must be stored (cf. (32)). The *evaluation* of $\omega_n(r)$ for some point $r \in [\delta_{\min}, D]$ requires only $\mathcal{O}(q)$ operations.

We remark here that the choice of the degree q of the Chebyshev interpolation of ω_n is fixed to 6 in the forthcoming numerical tests, and it turns out that the corresponding approximation error is negligible compared to the other errors.

4.2. Preprocessing

4.2.1. Generation of the Cluster Tree

Let $Q_0 = (a_1, a_1 + L_1) \times (a_2, a_2 + L_2)$ be an axes-parallel rectangle which contains Γ. We set $\mathcal{Q}_0 := \{Q_0\}$ and for $\ell \geq 1$,

$$\mathcal{Q}_\ell := \left\{Q_{\ell,k} : 1 \leq k \leq 2^{2\ell}\right\}$$

is the set of scaled, axis parallel rectangles $Q_{\ell,k} := A_{\ell,k} + \left(0, 2^{-\ell}L_1\right) \times \left(0, 2^{-\ell}L_2\right)$ which cover Q_0. We emphasize that this tree has not be generated physically. The following procedure generates the cluster tree – in fact it generates the tree levels $\mathbb{T}_\ell$, $0 \leq \ell \leq \ell_{\max}$, and then $\mathbb{T} = \bigcup_{\ell=0}^{\ell_{\max}} \mathbb{T}_\ell$.

procedure generate_cluster_tree$(Q, \mathcal{I}, \mathbb{T})$
begin
  $\ell := 0; s := \sharp\mathcal{I}$;
  while $s > 1$ **do begin**
    $\ell_{\max} := \ell$; $s := 1$;
    for $i \in \mathcal{I}$ **do begin**
      determine k such that[2] the cluster center satisfies $m_{\{i\}} \in Q_{\ell,k}$;
      $c_{\ell,k} := c_{\ell,k} \cup \{i\}$;
      $\mathbb{T}_\ell := \mathbb{T}_\ell \cup \{c_{\ell,k}\}$
      if $\sharp c_{\ell,k} > s$ **then** $s := \sharp c_{\ell,k}$;
      if $\ell \neq 0$ **then begin**
        determine j such that $Q_{\ell,k} \subset Q_{\ell-1,j}$
        father $(c_{\ell,k}) := c_{\ell-1,j}$;
        sons $(c_{\ell-1,j}) :=$ sons $(c_{\ell-1,j}) \cup \{c_{\ell,k}\}$;
        for $\nu, \mu \in M_q$ **do for** $r \in \{1,2\}$ **do** $t^{(r)}_{(c_{\ell-1,j}, c_{\ell,k}),(\nu,\mu)} := \Phi^{(r)}_{c_{\ell-1,j},\nu}\left(\xi_{Q(c_{\ell,k}),\mu}\right)$;
      end;
    end;
    $\ell := \ell + 1$;
  end;
end;

4.2.2. Generation of the Matrix for the Non-Admissible Part

Remark 7. *For the non-admissible pairs of leaves, the entries of the system matrix have to approximated as usual by numerical quadrature. For any pair of panels (τ_i, τ_j), which lies in the support of such a non-admissible pair of leaves, we have implemented the following quadrature rules in our computer program: For the singular cases, where $\tau_i = \tau_j$, we are using the k-smoothing change of variables with $k = 3$ in combination with an 8-point Gauss-Legendre rule as explained, e.g., in [13]. For the remaining cases, we directly apply 8-point Gauss-Legendre quadrature. It turns out that the effect of this numerical quadrature to the overall discretization error is negligible compared to the other approximation errors for the problems considered in our numerical experiments.*

a) $(\{i\}, \{j\}) \in \mathcal{C}^{\text{nonadm,direct}}$.

The (sparse) matrices $\mathbf{K}_n^{(i,j)} := \left(a^{(i,j)}_{n,\ell,k}\right)_{(\{i\},\{j\}) \in \mathcal{C}_n^{\text{nonadm,direct}}}$ for the non-admissible part are defined for all pairs $(\{i\}, \{j\}) \in \mathcal{C}_n^{\text{nonadm,direct}}$ by

$$a^{(i,j)}_{n,\ell,k} = \begin{cases} \int_{\text{supp}(b_k)} b_\ell(x) \int_{\text{supp}(b_\ell)} \omega_n^{(i,j)}(\|x-y\|)\, b_k(y)\, d\Gamma_y d\Gamma_x & (k,\ell) \in \mathcal{I} \times \mathcal{I}, \\ 0 & \text{otherwise.} \end{cases}$$

[2] If the cluster center $m_{\{i\}} \in \partial Q_{\ell,k}$, a convention has to be employed such that $m_{\{i\}}$ belongs only to one and only one rectangle in $\{Q_{\ell,k}, 1 \leq k \leq 2^{2\ell}\}$.

Note that only those indices $(k,\ell) \in \mathcal{I} \times \mathcal{I}$ lead to non-zero entries which belong to

$$\mathcal{I}^2(i,j) := \left\{ (\ell,k) \in \mathcal{I} \times \mathcal{I} : \operatorname{meas}\Big((\operatorname{supp}(b_\ell) \times \operatorname{supp}(b_k)) \cap (\operatorname{supp}(b_i) \times \operatorname{supp}(b_j)) \Big) > 0 \right\}.$$

The kernel function $\omega_n^{(i,j)}(\|x-y\|)$ can be evaluated by using the pre-computed approximations as explained in Section 4.1.

b) $(\{i\},\{j\}) \in \mathcal{C}^{\text{nonadm,FFT}}$.

Since the part $\mathcal{C}^{\text{nonadm,FFT}}$ of the non-admissible nearfield does not depend on the time step n, we can evaluate, for any pair $(\{i\},\{j\}) \in \mathcal{C}^{\text{nonadm,FFT}}$ and any quadrature point $(\xi_{i,\ell},\xi_{j,k}) \in \operatorname{supp}(\{i\}) \times \operatorname{supp}(\{j\})$, the kernel functions $\omega_n(\|\xi_{i,\ell}-\xi_{j,k}\|)$ for $0 \le n \le N$ by using the FFT techniques already described in Section 4.1.

4.2.3. Generation of the Cluster-Cluster Interaction Coefficients

For all $(c_1,c_2) \in \mathcal{C}_n^{\text{adm}}$ compute and store the cluster-cluster interaction coefficients $\kappa_n^{(\nu,\mu)}(c_1,c_2)$ for all $(\nu,\mu) \in M_q \times M_q$. If interpolation is used for approximation, these coefficients are given by (26) and the pre-computed approximations of the kernel functions ω_n as explained in Section 4.1 are employed.

4.2.4. Generation of the Basis Farfield Coefficients

For all $i,k \in \mathcal{I}$, for all $\nu \in M_q$, compute and store

$$J_{\{i\},k,\nu}^{(1),\text{basis}} := \int_{\operatorname{supp}(b_i)} \Phi_{\{i\},\nu}^{(1)}(x)\, b_k(x)\, d\Gamma_x \quad \text{and} \quad J_{\{i\},k,\nu}^{(2),\text{basis}} := \int_{\operatorname{supp}(b_i)} \Phi_{\{i\},\nu}^{(2)}(x)\, b_k(x)\, d\Gamma_x.$$

Note that, for $i \in \mathcal{I}$, only those indices $k \in \mathcal{I}$ lead to non-zero entries which belong to

$$\mathcal{I}_{\text{loc}}(i) := \{ k \in \mathcal{I} : \operatorname{meas}(\operatorname{supp}(b_i) \cap \operatorname{supp}(b_k)) > 0 \}.$$

4.3. Matrix-Vector Multiplication

Upwards Recursion:

The computation of the coefficients $J_{c,\nu}^{(2)} := J_{c,\nu}^{(2)}(u)$ for all $c \in \mathbb{T}$ is done by calling the procedure **upward_pass** for all $c \in \mathbb{T}$. The procedure is defined by

procedure upward_pass;
begin
  for $\ell = \ell_{\max}$ **downto** 0 **do begin**
    for $c_{\ell,k} \in \mathbb{T}_\ell$ **do begin**
      if $c_{\ell,k} = \{i\}$ is a leaf **then** $\left(J_{\{i\},\nu}^{(2)}\right)_{\nu \in M_q} := \left(\sum_{k \in \mathcal{I}_{\text{loc}}(i)} u_k J_{\{i\},k,\nu}^{(2),\text{basis}}\right)_{\nu \in M_q}$
      else for all $\nu \in M_q$ **do** $J_{c,\nu}^{(2)} := \sum_{s \in \text{sons}(c_{\ell,k})} \sum_{\mu \in M_q} t_{\nu,\mu,s}^{(2)} J_{s,\mu}^{(2)}$;
    end;
  end;
end;

Evaluation of the Cluster-Cluster Coupling:

For the cluster-cluster coupling we compute for all clusters $c_1 \in \mathbb{T}$ and $\nu \in M_q$ the values

$$B_n^{(\nu)}(c_1) := \sum_{\substack{c_2 \in \mathbb{T} \\ (c_1,c_2) \in \mathcal{C}_n^{\mathrm{adm}}}} \sum_{\mu \in M_q} \kappa_n^{(\nu,\mu)}(c_1,c_2)\, J_{c_2,\mu}^{(2)} \tag{34}$$

Downwards Recursion:

The evaluation of the matrix-vector multiplication is based on (23) and performed by calling the procedure **downward_pass** which is defined as follows.

procedure downward_pass;
begin
  for $\ell = 0$ **to** $\ell_{\max}$ **do begin**
    for $c \in \mathbb{T}_\ell$ **do begin**
      if $c = \mathcal{I}$ **then** $\left(y_n^{(\mu)}(c)\right)_{\mu \in M_q} := \left(B_n^{(\mu)}(c)\right)_{\mu \in M_q}$
      else begin
        determine the father of c, i.e., $c_{\ell-1,k}$ such that $c \subset c_{\ell-1,k}$;
        for all $\nu \in M_q$ **do** $y_n^{(\nu)}(c) := B_n^{(\nu)}(c) + \sum_{\mu \in M_q} t_{\mu,\nu,c}^{(1)} y_n^{(\mu)}(c_{\ell-1,k})$;
      end;
    end;
  end;
end;

Approximation of the Matrix-Vector Multiplication

The approximate evaluation of $\mathbf{v} := \mathbf{K}_n \mathbf{u}$ is computed by the following procedure. We assume that $\mathbf{v}$ is initialized by $\mathbf{0}$.

procedure mat_vec_mult;
begin
  upward_pass;
  for $c \in \mathbb{T}$ **for** $\nu \in M_q$ **compute** $B_n^{\nu}(c)$ (*according to (34)*);
  downward_pass;
  for $(\{i\},\{j\}) \in \mathcal{C}_n^{\mathrm{nonadm}}$ **for** $(\ell,k) \in \mathcal{I}^2(i,j)$ **do**

$$v_\ell := v_\ell + a_{n,\ell,k}^{(i,j)} u_k;$$

  for $i \in \mathcal{I}$ **for** $\ell \in \mathcal{I}_{\mathrm{loc}}(i)$ **do**

$$v_\ell := v_\ell + \sum_{\mu \in M_q} J_{\{i\},\ell,\mu}^{(1),\mathrm{basis}} y_n^{(\mu)}(\{i\}); \tag{35}$$

end;

5. Error Analysis

5.1. Local Error Analysis on Pairs of Clusters

Our separable approximation is based on polynomial Chebyshev interpolation and we will derive estimates for its accuracy depending on the order q. For this, let (c_1, c_2) be an admissible block with corresponding cluster boxes Q_{c_1} and Q_{c_2} (cf. Definition 1). Let $\omega_n^{(c_1,c_2)}$ and $\omega_{n,q}^{(c_1,c_2)}$ be defined by (15) and (25). We apply interpolation estimates for tensorized Chebyshev interpolation ([20, Lemma A.1] to obtain

$$\left|\omega_n^{(c_1,c_2)}\left(\|x-y\|\right) - \omega_{n,q}^{(c_1,c_2)}\left(x,y\right)\right| \leq C\frac{L^{q+1}\left(1+\log^3 q\right)}{2^{2q+1}\left(q+1\right)!} \max_{\substack{z\in Q_{c_1}-Q_{c_2}\\ i\in\{1,2\}}} \left|\partial_{z_i}^{q+1}\omega_n\left(\|z\|\right)\right|, \tag{36}$$

for all $x \in Q_{c_1}$ and $y \in Q_{c_2}$, where $C > 0$ is some constant independent of all parameters, L denotes the maximal side length of the boxes Q_{c_1} and Q_{c_2}. Hence, the accuracy depends on the growth of derivatives of the weight function ω_j on admissible pairs of clusters.

We use Lemma 6.7 in [27] to estimate $\partial_{z_i}^{q+1}\omega_n\left(\|z\|\right)$ in terms of the derivatives of the univariate function $\omega_n\left(r\right)$.

Lemma 8. *For a $q-$times differentiable function $f\left(r\right)$ it holds for $q \geq 1$*

$$\left|\partial_{z_i}^q f\left(\|z\|\right)\right| \leq \hat{C}^q q! \max_{1\leq\nu\leq q} \frac{1}{\nu!}\frac{\left|f^{(\nu)}\left(\|z\|\right)\right|}{\|z\|^{q-\nu}}. \tag{37}$$

The behavior of $\omega_n^{(m)}$ is analyzed in detail in [14] and Theorem 9 are direct consequences of [14, Theorem 1].

Theorem 9. *Let the time discretization be based on convolution quadrature with the BDF1 scheme and the transfer function be given by $K\left(r,z\right) = \dfrac{1}{2\pi}K_0\left(rz\right)$.*

1. *General estimate. For all $n \in \mathbb{N}_0$, $m \geq 1$, and $x > 0$, the estimate*

$$\left|\omega_n^{(m)}\left(r\right)\right| \leq \frac{\beta}{2\pi}\frac{m!}{\sqrt{n+1}}\left(\frac{\beta\sqrt{n+1}}{r}\right)^m \tag{38}$$

   holds for some $\beta \geq 1$ (independent of m, n, Δt, and r).
2. *Refined estimates for small and large arguments.*
   (a) *Small argument. There exists some constant $\beta > 1$ independent of $m, n, \Delta t, r$ such that for all $n \geq 0$ and $1 \leq m \leq \frac{1}{2}\sqrt{n}$ with the further restriction on m:*

$$\left\lfloor \frac{m\log\left(n+1\right)}{4\log 2} \right\rfloor \leq \frac{n+3}{4} \tag{39}$$

and all

$$0 < r \leq \min\left\{ \frac{t_n}{2C_0}, \frac{t_{n+1}}{4\beta} \right\} \tag{40}$$

it holds

$$\left| \omega_n^{(m)}(r) \right| \leq \frac{\beta}{2\pi} \frac{m!}{\sqrt{n+1}} \left(\frac{\beta}{r} \right)^m. \tag{41}$$

(b) *Large argument. For all* $n \geq 0$ *and* $m \geq 1 + 2\log(n+1)$ *it holds*

$$\left| \omega_n^{(m)}(r) \right| \leq m! \left(\frac{\beta}{r} \right)^m \qquad \forall r > \begin{cases} 0 & n = 0, 1, \\ t_n + t_m(\sqrt{n}+2) & n \geq 2, \end{cases} \tag{42}$$

for some constant $\beta \geq 1$ *(independent of* m*,* n*,* Δt*, and* r*).*

3. *Exponential decay. For* $m = 0$ *and* $r \geq \max\left\{ \Delta t, t_n\left(1 + \frac{1}{\sqrt{n}}\right) \right\}$*, the function* ω_n *is decaying exponentially*

$$|\omega_n(r)| \leq 3 \frac{\exp\left(\sqrt{n}\left(1 - \frac{r}{t_n\left(1+\frac{1}{\sqrt{n}}\right)} \right) \right)}{\sqrt{n+1}}. \tag{43}$$

The combination of (36) with Lemma 8 and Theorem 9 allows to determine the expansion order q (depending on $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm}}$) such that the error of the Chebyshev interpolation is below some given threshold $\varepsilon > 0$.

Farfield Blocks. Let $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm,far}}$ satisfy (17) for some $1 < \delta_0 = O(1)$. Then, from (43) we conclude that

$$\left| \omega_n^{(c_1,c_2)}(\|x-y\|) \right| \leq 3 \frac{\exp\left(-\sqrt{n}(\delta_0 - 1)\right)}{\sqrt{n+1}} \qquad \forall (x,y) \in Q_{c_1} \times Q_{c_2}$$

Hence, for given tolerance $\varepsilon > 0$, the condition (18) on the time step implies that the approximation of $\omega_n(\|x-y\|)$ on (c_1, c_2) by zero leads to an error $\leq \varepsilon$.

Remark 10. *For all blocks* $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm,far}}$*, the corresponding matrix blocks can be replaced by zero, and we formally express this by setting* $q := -1$ *and* $\omega_{n,q}^{(c_1,c_2)}(x,y) := 0$ *for all* $(x,y) \in \mathrm{supp}(c_1) \times \mathrm{supp}(c_2)$*.*

Admissible Blocks outside the "Peak-Zone". Let $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm,near}}$ satisfy (19) for some c depending on C_0 and β in (40), (42) and some $0 < \eta = O(1)$ which will be fixed later. These conditions along the choice of q as in (44) imply the conditions of Theorem 9.(2), i.e.,

$$\left| \partial_{z_i}^q \omega_n(\|z\|) \right| \leq C q! \left(\frac{\tilde{\beta}}{\|z\|} \right)^q,$$

where C and $\tilde{\beta}$ only depend on β in (41) and $\hat{C}$ in (37). The combination with (36) yields with $L := \max\{d_{c_1}, d_{c_2}\}$ for all $z = x - y \in Q_{c_1} - Q_{c_2}$:

$$\left|\omega_n^{(c_1,c_2)}(\|x-y\|) - \omega_{n,q}^{(c_1,c_2)}(x,y)\right| \leq \tilde{C}\left(1+\log^3 q\right)\left(\frac{L\tilde{\beta}}{4\|z\|}\right)^{q+1} \leq \check{C}\left(\frac{\tilde{\beta}\eta}{4}\right)^{q+1}$$

for an adjusted value of $\tilde{\beta}$. Hence the choice $\eta \leq 1/\tilde{\beta}$ results in

$$\sup_{z=x-y\in Q_{c_1}-Q_{c_2}} \left|\omega_n^{(c_1,c_2)}(\|x-y\|) - \omega_{n,q}^{(c_1,c_2)}(x,y)\right| \leq \frac{\tilde{C}}{4^q}$$

and an accuracy of ε is achieved by choosing

$$q := \left\lceil \frac{\log \frac{\tilde{C}}{\varepsilon}}{\log 4} \right\rceil. \tag{44}$$

In view of the conditions in Theorem 9(2a) on the order of the derivative, we have to assume that the time step n satisfies (20), while it is easy to verify that then (39) also holds. The condition in (42) can be written in the form

$$\delta_{\min}(c_1,c_2) \geq t_n\left(1+\frac{q}{\sqrt{n}}\left(1+\sqrt{2}\right)\right)$$

and is implied by the condition

$$\delta_{\min}(c_1,c_2) \geq \left(2+\sqrt{2}\right)t_n.$$

Remark 11. *For all blocks $(c_1,c_2) \in \mathcal{C}_n^{\mathrm{adm,near}}$, the order of the Chebyshev expansion is given by*

$$q = \left\lceil \frac{\log \frac{\tilde{C}}{\varepsilon}}{\log 4} \right\rceil.$$

Admissible Blocks at the "Peak". Let $(c_1,c_2) \in \mathcal{C}^{\mathrm{adm,peak}}$ satisfy (21), where, again, $0 < \eta = O(1)$ will be fixed later. Then, we employ the general estimate (38) (for $m \geq 1$) to obtain

$$\left|\partial_{z_i}^q \omega_n(\|z\|)\right| \leq \frac{\beta}{2\pi\sqrt{n+1}}\hat{C}^q q!\left(\frac{\beta\sqrt{n+1}}{\|z\|}\right)^q.$$

The combination with (36) results in the estimate for the Chebyshev interpolation (with $L := \max\{d_{c_1}, d_{c_2}\}$)

$$\sup_{z=x-y\in Q_{c_1}-Q_{c_2}} \left|\omega_n^{(c_1,c_2)}(\|x-y\|) - \omega_{n,q}^{(c_1,c_2)}(x,y)\right| \leq \check{C}\left(\frac{\hat{C}L\beta\sqrt{n+1}}{4\|z\|}\right)^{q+1} \leq \check{C}\left(\frac{\hat{C}\beta\eta}{4}\right)^{q+1}$$

with a constant $\check{C}$ depending only on $\hat{C}$ and β. The choice

$$\eta := \left(\hat{C}\beta\right)^{-1} \quad \text{and} \quad q := \left\lceil \frac{\log \frac{\tilde{C}}{\varepsilon}}{\log 4} \right\rceil$$

then leads to

$$\sup_{z=x-y\in Q_{c_1}-Q_{c_2}} \left|\omega_n^{(c_1,c_2)}\left(\|x-y\|\right) - \omega_{n,q}^{(c_1,c_2)}\left(x,y\right)\right| \leq \tilde{C}4^{-q} \leq \varepsilon.$$

Remark 12. *For all blocks $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm,peak}}$ the order of the Chebyshev expansion is*

$$q = \left\lceil \frac{\log \frac{\tilde{C}}{\varepsilon}}{\log 4} \right\rceil.$$

Remark 13. *Definition 4 along the choices of the expansion order q for the Chebyshev interpolation ensure that the panel-clustering approximation of the quadrature weight ω_n satisfies*

$$\left|\omega_n\left(\|x-y\|\right) - \omega_n^{\mathrm{PC}}\left(x,y\right)\right| \leq \varepsilon \qquad \forall x, y \in \Gamma. \tag{45}$$

5.2. Stability and Consistency Analysis

In Section 5.1, an admissibility condition has been derived for pairs of clusters so that the local approximation error is bounded by some given tolerance $\varepsilon > 0$. In this section, we will study the influence of these local errors to the overall solvability and accuracy of the full discretization. This will be a simple consequence of the perturbation theory which has been developed for the CQ-BDF2 discretization of the three-dimensional wave equation in [23], [31] but is applicable verbatim to our case. The theory requires the following estimate of $\|V^{-1}(s)\|_{H^{-1/2}(\Gamma)\leftarrow H^{1/2}(\Gamma)}$: Let $\sigma > 0$. Then, there exists $M(\sigma)$, such that

$$\|V^{-1}(s)\|_{H^{-1/2}(\Gamma)\leftarrow H^{1/2}(\Gamma)} \leq M(\sigma)|s|^2 \quad \forall\, \mathrm{Re}(s) > \sigma\,. \tag{46}$$

For a proof, we refer, e.g., to [37, Prop. 2.6.1]. In [23], it is also assumed that an inverse inequality for the boundary element space holds: There exists a constant $C_{\mathrm{inv}} > 0$ such that

$$\|\psi\|_{L^2(\Gamma)} \leq C_{\mathrm{inv}}\left(\Delta x\right)^{-1/2} \|\psi\|_{H^{-1/2}(\Gamma)} \qquad \forall \psi \in S \tag{47}$$

holds. For a proof for quasi-uniform meshes we refer to [9].

The following theorem is a direct consequence of [23, Theorem 4.3].

Theorem 14. *Let the discretization (30) be based on the panel-clustering CQ-BEM with the BDF1 scheme and polynomials of degree p in (10). We assume that the exact solution $\phi(\cdot, t)$ is in $H^{p+1}(\Gamma)$ for any $t \in [0, T]$. Then for any*

tolerance $\varepsilon > 0$ *in (45) with* $0 < \varepsilon < \frac{1-\mathrm{e}^{-\sigma\Delta t}}{2c_\Delta c_\sigma} (\Delta t)^2 \Delta x$*, the solutions* $\phi_n^{\mathrm{pc},\Delta x}$*,* $0 \le n \le N$*, in (30) exist and satisfy the error estimate*

$$\left\| \phi_{\Delta t,n}^{\mathrm{pc},\Delta x} - \phi(\cdot, t_n) \right\|_{H^{-1/2}(\Gamma)} \le C_g(t_n) \left(\frac{\varepsilon}{(\Delta t)^5 \Delta x} + \Delta t + (\Delta x)^{p+3/2} \right)$$

holds, where C_g *depends on the right-hand side* g *and on* σ*.*

Corollary 15. *Let the assumptions as in Theorem 14 be satisfied. Let*

$$\Delta t \sim (\Delta x)^{p+\frac{3}{2}}$$

and choose

$$\varepsilon \sim (\Delta t)^6 \Delta x.$$

Then the solution $\tilde{\phi}_{\Delta t,h}^n$ *exists and converges with optimal rate*

$$\left\| \tilde{\phi}_{\Delta t,h}^n - \phi(\cdot, t_n) \right\|_{H^{-1/2}(\Gamma)} \le C_g(t_n) \Delta t \sim C_g(t_n) (\Delta x)^{p+\frac{3}{2}}.$$

6. Complexity

In this section we will investigate the computational and storage complexity for the CQ-BEM with panel clustering. We impose some simplifying assumptions in order to reduce technicalities.

- The spatial mesh is quasi-uniform, i.e.,

$$\max_{\tau \in \mathcal{G}} \frac{\Delta x}{\Delta_\tau} \le C_{\mathrm{qu}}.$$

- There exist constants $C_1, C_2 > 0$ such that

$$C_1^{-1} 2^{-\ell} \le d_c \le C_1 2^{-\ell} \qquad \forall c \in \mathbb{T}_\ell \tag{48}$$

  and for all $(c_1, c_2) \in \mathcal{C}_n^{\mathrm{adm}}$ it holds

$$C_2^{-1} d_{c_2} \le d_{c_1} \le C_2 d_{c_2}. \tag{49}$$

  For a proof of assumptions (48) and (49) under moderate assumptions on Γ, we refer to the extended version of [32].

Next we will estimate the cardinalities of the various parts of the block partitioning of $\Gamma \times \Gamma$.

6.1. Farfield Blocks

For the farfield blocks in $\mathcal{C}_n^{\mathrm{adm,far}}$ the matrices are replaced by zero and hence no computational and storage complexity arises.

6.2. Admissible Blocks outside the Peak Zone

Generously, we estimate the area covered by the admissible blocks in $\mathcal{C}_n^{\text{adm,near}}$ by $|\Gamma \times \Gamma|$. The pairs of admissible blocks are graded by the same condition which is applied for the panel clustering method for the Laplace equation and it is well known (cf., e.g., [21], [33], [15])

$$\sharp \mathcal{C}_n^{\text{adm,near}} \leq CM.$$

6.3. Admissible Blocks at the Peak

The first two conditions in (21) imply that the area covered by the blocks in $\mathcal{C}_n^{\text{adm,peak}}$ can be estimated from above, generously, by $\mathcal{O}(t_n)$. Let $(c_1, c_2) \in \mathcal{C}_n^{\text{adm,peak}}$. The third conditions in (21) imply that (c_1, c_2) is not the root $\mathcal{I} \times \mathcal{I}$ but there exists a father $(\tilde{c}_1, \tilde{c}_2)$ with $(c_1, c_2) \in \text{sons}((\tilde{c}_1, \tilde{c}_2))$ which does not belong to $\mathcal{C}_n^{\text{adm,peak}}$. Since $\delta_{\max}(\tilde{c}_1, \tilde{c}_2) \geq \delta_{\max}(c_1, c_2)$ and $\delta_{\min}(\tilde{c}_1, \tilde{c}_2) \leq \delta_{\min}(c_1, c_2)$, the first two conditions in (21) are valid and, hence, the last one must be violated:

$$\max\{d_{\tilde{c}_1}, d_{\tilde{c}_2}\} > \eta \frac{\delta_{\min}(\tilde{c}_1, \tilde{c}_2)}{\sqrt{n+1}}.$$

Furthermore, we have

$$ct_n \leq \delta_{\max}(\tilde{c}_1, \tilde{c}_2) \leq \delta_{\min}(\tilde{c}_1, \tilde{c}_2) + d_{\tilde{c}_1} + d_{\tilde{c}_2} \leq (1+2\eta)\,\delta_{\min}(\tilde{c}_1, \tilde{c}_2) \tag{50}$$

and we obtain

$$\max\{d_{\tilde{c}_1}, d_{\tilde{c}_2}\} > \frac{c\eta}{(1+2\eta)} \frac{t_n}{\sqrt{n+1}}.$$

Assumptions (48) and (49) imply

$$\min\{d_{c_1}, d_{c_2}\} > \frac{c\eta}{(1+2\eta)} \frac{t_n}{\sqrt{n+1}}.$$

Hence,

$$\sharp \mathcal{C}_n^{\text{adm,peak}} \leq C \frac{t_n}{\left(\frac{c\eta}{(1+2\eta)} \frac{t_n}{\sqrt{n+1}}\right)^2} \leq C \frac{n+1}{t_n} \leq CN.$$

A summation over all $0 \leq n \leq N$ yields

$$\sum_{n=0}^{N} \sharp \mathcal{C}_n^{\text{adm,peak}} \leq CN^2.$$

6.4. Non-Admissible Pairs of Panels

The nearfield consists of the non-admissible pairs of panels $\mathcal{C}_n^{\text{nonadm}} := \mathcal{C}^{\text{nonadm,FFT}} \cup \mathcal{C}_n^{\text{nonadm,direct}}$. The condition (16) for $\mathcal{C}^{\text{nonadm,FFT}}$ is the same as the one for the standard panel-clustering method for Laplace's equation so that $\sharp \mathcal{C}^{\text{nonadm,FFT}} = O(M)$.

To estimate the cardinality of the remaining non-admissible pairs of leaves, it suffices to consider under what circumstances the last condition in (21) is

violated. Since $\delta_{\min}(c_1, c_2) \geq ct_n$ (cf. (50)) this case produces non-admissible pairs of panels only if

$$\Delta x > \tilde{c}\eta\sqrt{n}\Delta t \tag{51}$$

because, otherwise, the "procedure divide" will divide the blocks until the last condition (21) is satisfied and the pair becomes admissible. However, from (51) we conclude that the number of blocks which violate the third condition in (21) is bounded by $O(M)$ and, in most cases, is zero.

According to Remark 7 the amount of work for the approximation of each conventional matrix elements is $\mathcal{O}(1)$.

6.5. Complexity of the Panel-Clustering Algorithm

We have seen that the number of admissible and non-admissible blocks are bounded by

$$\sum_{n=0}^{N} \sharp\mathcal{C}_n^{\text{adm}} \leq CN(N+M) \quad \text{and} \quad \sharp\mathcal{C}_n^{\text{nonadm}} \leq CM.$$

The recursive structure of the "procedure divide" implies that the generation of the block partitions in the algorithm requires in total $O(N(N+M))$ operations for all time steps. Consequently the storage complexity for the generation of the cluster-cluster interaction coefficients and the basis farfield coefficients is bounded by $O\left(N(M+N)q^4\right)$, where $q \sim \log\frac{1}{\varepsilon}$ and $\varepsilon \sim (\Delta t)^6 \Delta x$. The total computational cost is $O\left(N(M+N)q^5\right)$ and this quintic scaling with respect to q is due to the fact that the evaluation of each pre-computed kernel function (cf. Section 4.1) requires $\mathcal{O}(q)$ operations.

The computation of the matrix entries related to pairs of leaves in $\mathcal{C}^{\text{nonadm,FFT}}$ requires the evaluation of the kernel functions ω_n, $0 \leq n \leq N$, at $\mathcal{O}(M)$ quadrature points (cf. Remark 7). As explained in Section 4.2.2 we employ FFT techniques; so this step requires $\mathcal{O}(MN\log N)$ operations for all $0 \leq n \leq N$.

The CPU time for the evaluation of a matrix-vector multiplication in the panel-clustering format is $O\left(N(M+N)q^4\right)$ and dominated by the evaluation of the cluster-cluster coupling. The additional storage amount for the matrix-vector multiplication is negligible.

7. Numerical Results

The first part of the numerical experiments concerns the number of blocks in the different parts of the partitions $\mathcal{C}_n^{\text{nonadm}}$, $\mathcal{C}_n^{\text{adm}}$ of $\Gamma \times \Gamma$.

Example 1.. Let us consider equation (1), where Ω is the disk of radius 1. We approximate the boundary Γ with the polygonal boundary whose nodes are obtained by a uniform partition of Γ into M intervals. For the space discretization, we consider piecewise constant functions associated to the uniform spatial mesh. For the time discretization, we choose a uniform partition of the interval $[0, T]$ into N subintervals. We construct the cluster tree according to the procedure

generate_cluster_tree and apply the procedure **divide** at each time step t_n by choosing $\varepsilon = 1E-08$ and $\eta = 1$. In Figures 1 we depict the sparse structure of $\mathbf{K}_n^{\Delta x}$ for different time steps t_n which illustrates the movement of the discrete light cone through the spatial domain $\Gamma \times \Gamma$ with increasing time step t_n.

We recall that the storage requirement for the panel clustering approximation is of order $\mathcal{O}(M+N)$ for each $n = 0, \cdots, N$, and the total storage is $\mathcal{O}(N(M+N))$. In Figure 2, we compare the computed storage requirements with the theoretical ones as well as with the memory storage required by the full matrix representation, which is $\mathcal{O}(M^2)$ for each $n = 0, \cdots, N$, and $\mathcal{O}(NM^2)$ globally. The bottom right picture in Figure 2 nicely illustrates the linear growth $\mathcal{O}(NM)$ of the storage requirements for the partitions of the panel-clustering method.

7.1. The construction of an exact solution

In order to test the panel-clustering algorithm and to show the efficiency of the sparse representation of the block matrices, it is important to have a reference solution at hand. Since no exact solution of the retarded potential equation is known in two dimensions, we are interested here in deriving an explicit representation of the solution of the semi-discrete problem obtained by considering a time discretization of the equation (2) in the case Γ is the unit circle.

To this aim, we start by determining the eigenfunctions ϕ_m and eigenvalues $\lambda_{n,m}$

$$\mathcal{K}_n \phi_m = \lambda_{n,m} \phi_m \tag{52}$$

of the integral operator

$$\mathcal{K}_n \phi(x) := \int_\Gamma \omega_n(\|x-y\|)\, \phi(y)\, d\Gamma_y$$

defined on the unit circle $\Gamma := \{x \in \mathbb{R}^2 \mid \|x\| = 1\}$.

It turns out that the eigenvalues $\lambda_{n,m}$ can be expressed in terms of Bessel functions and we recall the relevant definitions. From [1, 9.6.3 and 9.6.4], we have, for integers $n \in \mathbb{N}_0$, the relations

$$I_n(z) = (-\mathrm{i})^n J_n(\mathrm{i}\, z) \quad \text{and} \quad K_n(z) = \frac{\pi}{2}\, \mathrm{i}^{n+1} H_n^{(1)}(\mathrm{i}\, z)$$

with J_n and I_n being the Bessel and modified Bessel functions of first kind, respectively, K_n the modified Bessel functions of second kind and $H_n^{(1)}$ the Hankel functions. From [3, Theorem 8] it follows that, for the function $\phi_m(\alpha) := \exp(\pm \mathrm{i}\, m\alpha)$, $\alpha \in [-\pi, \pi[$ and any $m \in \mathbb{N}_0$, it holds

$$L_k \phi_m = \lambda_m(k)\, \phi_m \quad \text{with} \quad \lambda_m(k) := \frac{\pi\, \mathrm{i}}{2} J_m(k)\, H_m^{(1)}(k)$$

and

$$L_k \phi(x) := \int_\Gamma \frac{\mathrm{i}}{4} H_0^{(1)}(k\,\|x-y\|)\, \phi(y)\, d\Gamma_y$$

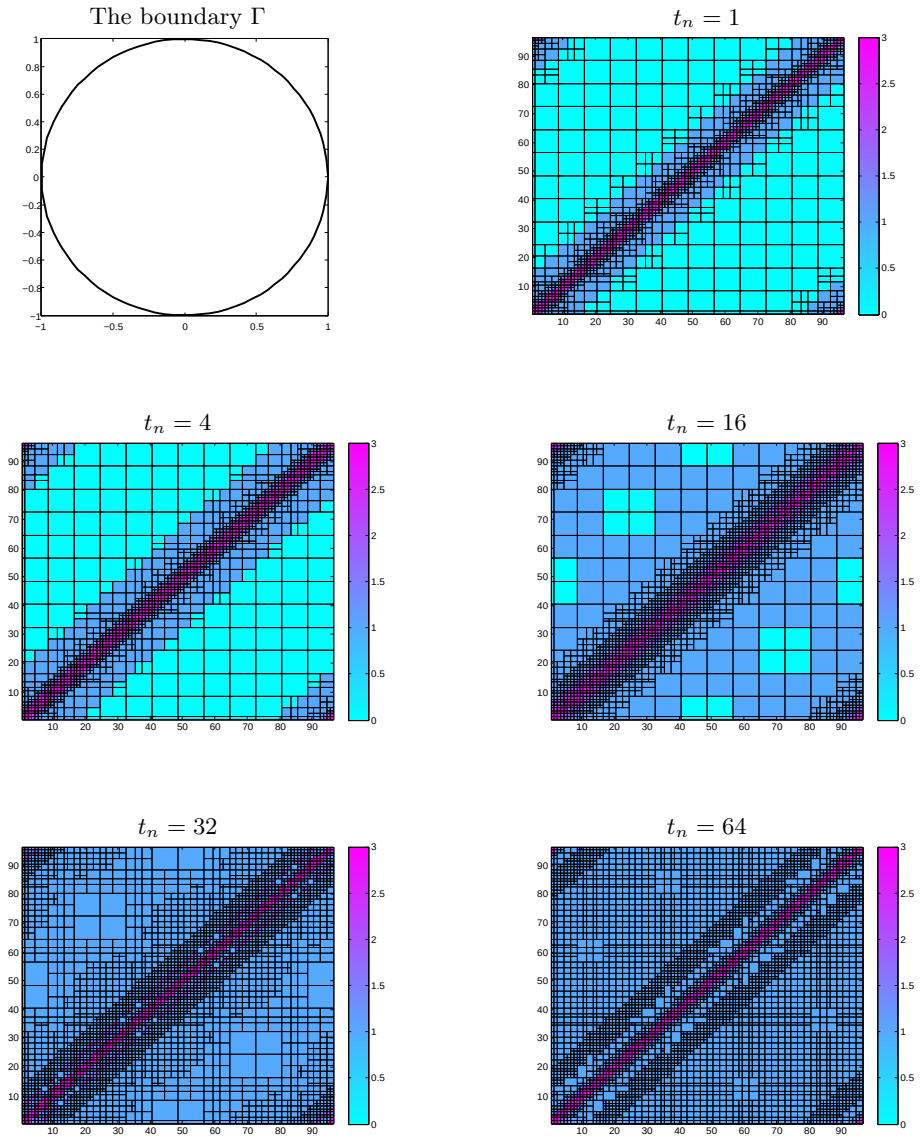

Figure 1: Example 1. Structure of the matrices $\mathbf{K}_n^{\Delta x}$, for $n = 1, 4, 16, 32, 64$, with $M = 96$ and $N = 64$. The admissible blocks belonging to to $\mathcal{C}_n^{\text{adm,far}}$ are colored turquoise, those belonging to $\mathcal{C}_n^{\text{adm,near}}$ are colored blue, the non admissible blocks belonging to $\mathcal{C}_n^{\text{nonadm,FFT}}$ are colored pink and those belonging to $\mathcal{C}_n^{\text{nonadm,direct}}$ are colored violet.

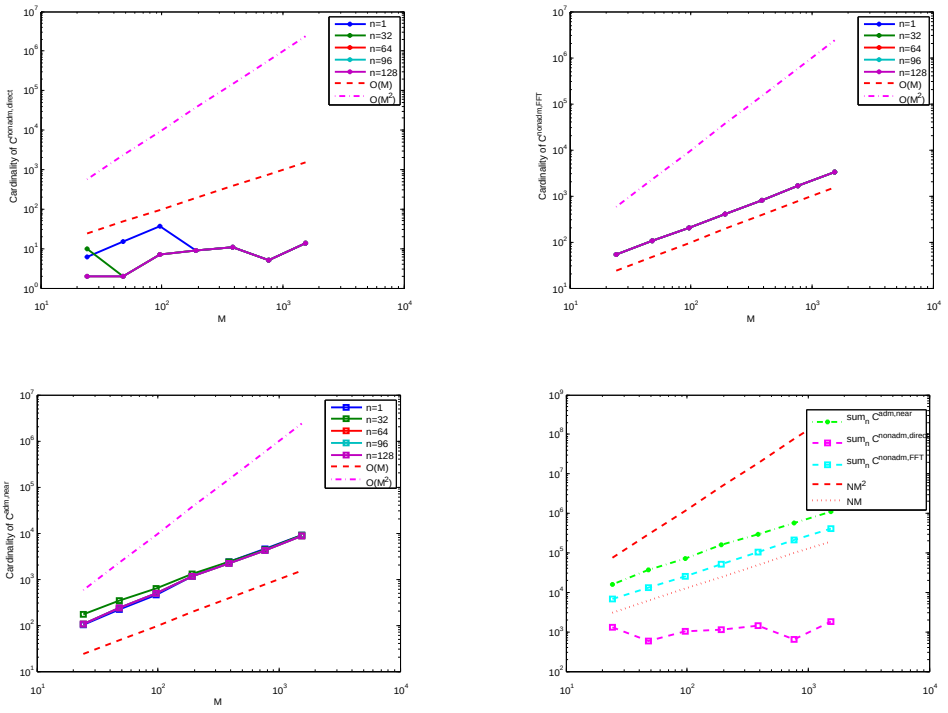

Figure 2: Example 1. Storage comparison for $T = 10$, $N = 128$ and increasing values of M: the $\mathcal{C}^{\text{nonadm,direct}}$ fields (top-left), the $\mathcal{C}^{\text{nonadm,FFT}}$ fields (top-right), the $\mathcal{C}^{\text{adm,near}}$ fields (bottom-left) for different time steps, and the total storage requirement (bottom-right). In this setting the $\mathcal{C}^{\text{adm,far}}$ field is empty.

again on the unit circle Γ. Hence, we conclude that

$$\begin{aligned}\mathcal{K}_0\phi_m(x) &= \frac{1}{2\pi}\int_\Gamma K_0\left(\frac{\|x-y\|}{\Delta t}\right)\phi_m(y)\,d\Gamma_y = \int_\Gamma \frac{\mathrm{i}}{4}H_0^{(1)}\left(\frac{\mathrm{i}}{\Delta t}\|x-y\|\right)\phi_m(y)\,d\Gamma_y \\ &= \lambda_m\left(\frac{\mathrm{i}}{\Delta t}\right)\phi_m(x).\end{aligned}$$

Note that

$$\lambda_m\left(\frac{\mathrm{i}}{\Delta t}\right) = \frac{\pi\,\mathrm{i}}{2}J_m\left(\frac{\mathrm{i}}{\Delta t}\right)H_m^{(1)}\left(\frac{\mathrm{i}}{\Delta t}\right) = I_m\left(\frac{1}{\Delta t}\right)K_m\left(\frac{1}{\Delta t}\right).$$

For $n>0$, we use Cauchy's integral representation

$$\omega_n(d) = \frac{1}{2\pi n!}\left.\frac{\partial^n K_0\left(\frac{\gamma(\zeta)}{\Delta t}d\right)}{\partial\zeta^n}\right|_{\zeta=0} = \frac{1}{(2\pi)^2\,\mathrm{i}}\int_{\mathcal{C}}\frac{K_0\left(\frac{\gamma(z)}{\Delta t}d\right)}{z^{n+1}}dz$$

with $\gamma(\zeta)=1-\zeta$ for the BDF1 method and $\mathcal{C}$ is a circle around $0\in\mathbb{C}$ with radius <1. By interchanging the ordering of integration we obtain

$$\mathcal{K}_n\phi(x) = \frac{1}{(2\pi)^2\,\mathrm{i}}\int_{\mathcal{C}}\int_\Gamma\frac{K_0\left(\frac{\gamma(z)}{\Delta t}\|x-y\|\right)}{z^{n+1}}\phi(y)\,d\Gamma_y dz.$$

Hence,

$$\begin{aligned}(\mathcal{K}_n\phi_m)(x) &= \frac{1}{(2\pi)^2\,\mathrm{i}}\int_{\mathcal{C}}z^{-n-1}\int_\Gamma K_0\left(\frac{\gamma(z)}{\Delta t}\|x-y\|\right)\phi_m(y)\,d\Gamma_y dz \\ &= \frac{1}{n!}\left(\frac{n!}{2\pi\,\mathrm{i}}\int_{\mathcal{C}}z^{-n-1}I_m\left(\frac{\gamma(z)}{\Delta t}\right)K_m\left(\frac{\gamma(z)}{\Delta t}\right)dz\right)\phi_m(x) \\ &= \frac{1}{n!}\,\partial_\zeta^n\left(I_m\left(\frac{\gamma(\zeta)}{\Delta t}\right)K_m\left(\frac{\gamma(\zeta)}{\Delta t}\right)\right)\Big|_{\zeta=0}\phi_m(x)\end{aligned}$$

By setting $\lambda_{n,m} := \frac{1}{n!}\,\partial_\zeta^n\left(I_m\left(\frac{\gamma(\zeta)}{\Delta t}\right)K_m\left(\frac{\gamma(\zeta)}{\Delta t}\right)\right)\Big|_{\zeta=0}$, the relation (52) holds true. Note that the evaluation of $\lambda_{n,m}$ in this form is numerically very unstable and we recommend to use a representation of the derivative ∂_ζ^n as a contour integral and its approximation by the trapezoidal rule.

Next, we employ the eigenpairs $(\lambda_{n,m},\phi_m)$ to construct an exact solution of the time-discrete problem (8) for a right-hand side of the form

$$g_m(t,x) := \alpha(t)\,\phi_m(x), \quad \text{i.e.,} \quad g_{n,m}(x) = \alpha_n\phi_m(x) \quad \text{with} \quad \alpha_n := \alpha(t_n).$$

We choose some coefficient vector $\boldsymbol{\beta} = (\beta_j)_{j=0}^N \in \mathbb{R}^{N+1}$ which satisfies $\sum_{j=0}^n \lambda_{n-j,m}\beta_j = \alpha_n$ for $0\le n\le N$. Then, it is easy to verify that

$$\varphi_{m,j} := \beta_j\phi_m, \quad j=0,\dots,N \tag{53}$$

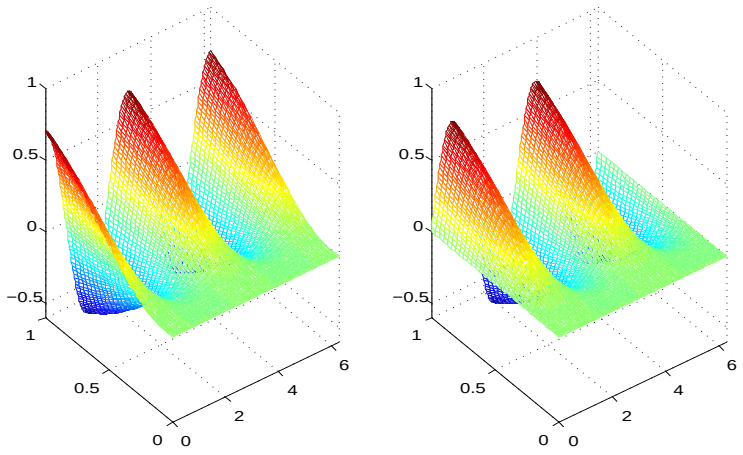

Figure 3: Real(left) and imaginary(right) parts of the exact solution of the semidiscrete problem for $m = 2$, $T = 1$ and $N = 32$.

is the exact solution of

$$\sum_{j=0}^{n} \mathcal{K}_{n-j}\varphi_{m,j} = g_{n,m}, \qquad n = 0, \dots, N.$$

We discretize this equation by the Galerkin boundary element method and compare the performance of the original Galerkin BEM with its sparse approximation by panel clustering. We replace the right-hand side $g_{n,m}$ by its interpolant

$$g_{n,m}(x) = \alpha_n \phi_m(x) \approx \alpha_n \sum_{i=1}^{M} \phi_{m,i} b_i(x) \qquad \text{with } \boldsymbol{\phi}_m := (\phi_{m,i})_{i=1}^{M} = (\phi_m(x_i))_{i=1}^{M}$$

and b_i denoting the Lagrange basis for the boundary element space. Then, the Galerkin system has the following block Toeplitz form:

$$\sum_{j=0}^{n} \mathbf{K}_{n-j}^{\Delta x} \varphi_{m,j}^{\Delta x} = \alpha_n \mathbf{B} \boldsymbol{\phi}_m$$

with the mass matrix $\mathbf{B} = \Big((b_i, b_j)_{L^2(\Gamma)} \Big)_{i,j=1}^{M}$.

In Figure 3 we show the interpolation of the time-discrete solution corresponding to the choice $\alpha(t) = t^4 e^{-2t}$, $m = 2$, for $T = 1$ and $N = 32$.

In the following numerical tests we will construct the approximate solution by using the proposed panel-clustering algorithm. We restrict to the BDF1 convolution quadrature and Galerkin BEM with piecewise constant ansatz functions. We will compare this solution with the one obtained by applying the original, unperturbed Galerkin approach described in Section 2.

All the numerical computation has been performed on a PC with Intel Core® i3-3217U CPU (1.80 GHz). To perform our numerical testing we have written standard (i.e., sequential) Matlab® codes.

Example 3. Let Γ be the unit circle where we prescribe the Dirichlet data by $g_m(t,x) = \alpha(t)\phi_m(x)$ defined in Section 7.1, with $\alpha(t) = t^4 e^{-2t}$. We consider a uniform subdivision of Γ into M panels. First, we fix the number of time steps $N = 8$ and study the convergence of the method with respect to the spatial refinement. We denote by $\varphi_{\Delta t,j}^{\Delta x}$ the solution obtained by applying the full Galerkin scheme in space and by $\varphi_{\Delta t,j}^{\mathrm{pc},\Delta x}$ the one obtained by applying the panel clustering method. Since the exact solution is known, we can compute the relative errors defined by (cf. (53))

$$\mathrm{Err}^{\mathrm{Gal}} = \max_{0 \le j \le N} \frac{\|\varphi_{j,m} - \varphi_{\Delta t,j}^{\Delta x}\|_{L^2(\Gamma)}}{\|\varphi_{j,m}\|_{L^2(\Gamma)}}, \tag{54}$$

$$\mathrm{Err}^{\mathrm{PC}} = \max_{0 \le j \le N} \frac{\|\varphi_{j,m} - \varphi_{\Delta t,j}^{\mathrm{pc},\Delta x}\|_{L^2(\Gamma)}}{\|\varphi_{j,m}\|_{L^2(\Gamma)}}. \tag{55}$$

In Table 1 we report the behavior of the errors defined in (54) and (55) for $m = 2$, $T = 1$, and $N = 8$, and choose the control parameter for the panel clustering by $\varepsilon = 1E-08$ and $\eta = 1$, i.e., constant for each time step t_n. Moreover, we report the memory storage, expressed in Kilobytes (Kb) and the CPU time (in seconds) for the full Galerkin and the panel clustering method (these quantities are denoted in the table by the acronym $\mathrm{Mem}^{\mathrm{Gal}}$ and $\mathrm{Mem}^{\mathrm{PC}}$, and $\mathrm{CPU}^{\mathrm{Gal}}$ and $\mathrm{CPU}^{\mathrm{PC}}$, respectively) We vary the degree q of the polynomial approximation in order to show the rapid convergence of the panel-clustering method to the unperturbed Galerkin method. In the same setting, in Table 2 we report the errors, the memory storage and the CPU time with respect to both space and time refinements.

The following observations can be deduced from the results depicted in Tables 1:

1. Convergence with respect to the polynomial approximation order q
   The convergence of the panel-clustering solution towards the solution of the unperturbed CQ-BEM discretization is rapid. If we choose for this example the polynomial order q according to the function

   $$q(M) = \left\lfloor \frac{1}{4} \log_2 \left(\frac{M}{6} \right) \right\rfloor, \tag{56}$$

   which quite slowly increases with respect to M, the Galerkin error in all cases satisfies

   $$\mathrm{Err}^{\mathrm{PC}} \le \frac{3}{2} \mathrm{Err}^{\mathrm{Gal}}.$$

   We emphasize that there are important practical applications, where the surface Γ is very complicated and requires a large number of panels to be

Table 1: Relative errors defined in (54) and (55), data storage comparison (Kb) and CPU comparison (sec), for $m = 2$, $T = 1$ and $N = 8$.

M	$\mathrm{Err}^{\mathrm{Gal}}$	$\mathrm{Err}^{\mathrm{PC}}$	q	$\mathrm{Mem}^{\mathrm{Gal}}$	$\mathrm{Mem}^{\mathrm{PC}}$	$\frac{\mathrm{Mem}^{\mathrm{Gal}}}{\mathrm{Mem}^{\mathrm{PC}}}$	$\mathrm{CPU}^{\mathrm{Gal}}$	$\mathrm{CPU}^{\mathrm{PC}}$	$\frac{\mathrm{CPU}^{\mathrm{Gal}}}{\mathrm{CPU}^{\mathrm{PC}}}$
24	3.05E-01	3.05E-01	0	3.69E+01	3.22E+01	1.14	2.07E+00	6.24E+00	0.33
48	1.52E-01	1.65E-01	0	1.47E+02	6.79E+01	2.17	8.30E+00	1.45E+01	0.57
		1.52E-01	1		4.36E+02	0.34		1.97E+01	0.44
96	7.56E-02	4.62E-01	0	5.90E+02	1.31E+02	4.51	3.21E+01	2.38E+01	1.35
		7.58E-02	1		8.39E+02	0.70		3.68E+01	0.86
		7.56E-02	2		3.90E+03	0.15		8.22E+01	0.39
192	3.78E-02	7.75E-01	0	2.36E+03	2.67E+02	8.85	1.28E+02	5.24E+01	2.43
		3.92E-02	1		1.73E+03	1.36		6.84E+01	1.87
		3.78E-02	2		8.06E+03	0.29		1.32E+02	0.97
384	1.89E-02	1.26E+00	0	9.44E+03	4.81E+02	19.62	4.89E+02	6.19E+01	6.22
		2.48E-02	1		2.99E+03	3.15		1.30E+02	3.77
		1.90E-02	2		1.39E+04	0.68		2.74E+02	1.78
768	9.45E-03	1.90E+00	0	3.78E+04	9.08E+02	41.58	1.95E+03	1.54E+02	12.65
		2.11E-02	1		5.59E+03	6.75		2.84E+02	6.87
		1.00E-02	2		2.58E+04	1.46		4.99E+02	3.91
		9.45E-03	3		8.03E+04	0.47		9.33E+02	2.09
1536	4.72E-03	2.35E-02	1	1.51E+05	1.11E+04	13.64	7.95E+03	6.03E+02	13.18
		6.70E-03	2		5.11E+04	2.95		1.24E+03	6.41
		4.72E-03	3		1.59E+05	0.95		2.48E+03	3.21
3072	2.36E-03	7.01E-03	2	6.04E+05	1.01E+05	6.01	2.83E+04	2.42e+03	11.69
		2.37E-03	3		3.12E+05	1.93		5.85E+03	4.84

Table 2: Relative errors defined in (54) and (55), data storage comparison (Kb) and CPU comparison (sec), for $m = 2$, $T = 1$.

M	N	$\mathrm{Err}^{\mathrm{Gal}}$	$\mathrm{Err}^{\mathrm{PC}}$	q	$\mathrm{Mem}^{\mathrm{Gal}}$	$\mathrm{Mem}^{\mathrm{PC}}$	$\frac{\mathrm{Mem}^{\mathrm{Gal}}}{\mathrm{Mem}^{\mathrm{PC}}}$	$\mathrm{CPU}^{\mathrm{Gal}}$	$\mathrm{CPU}^{\mathrm{PC}}$	$\frac{\mathrm{CPU}^{\mathrm{Gal}}}{\mathrm{CPU}^{\mathrm{PC}}}$
48	4	1.52E-01	1.96E-01	0	7.37E+01	2.97E+01	2.48	9.34E+00	8.78E+00	1.06
			1.52E-01	1		1.87E+02	0.39		1.18E+01	0.79
96	8	7.56E-02	4.62E-01	0	5.90E+02	1.31E+02	4.51	3.96E+01	3.04E+01	1.30
			7.58E-02	1		8.39E+02	0.70		3.75E+01	1.06
			7.56E-02	2		3.90E+03	0.15		6.23E+01	0.64
192	16	3.78E-02	7.40E-01	0	4.72E+03	6.09E+02	7.74	1.82E+02	1.49E+02	1.22
			3.89E-02	1		4.05E+03	1.16		1.26E+02	1.44
			3.78E-02	2		1.89E+04	0.25		3.21E+02	0.57
384	32	1.89E-02	1.11E+00	0	3.77E+04	2.39E+03	15.78	8.85E+02	3.33E+02	2.66
			3.76E-02	1		1.57E+04	2.40		7.66E+02	1.16
			1.90E-02	2		7.34E+04	0.51		1.39E+03	0.64
768	64	9.45E-03	3.59E-02	1	3.02E+05	5.98E+04	5.05	4.87E+03	1.84E+03	2.65
			1.14E-02	2		2.79E+05	1.08		5.08E+03	0.96
			9.48E-03	3		8.70E+05	0.35		1.25E+04	0.39
1536	128	4.72E-03	7.30E-02	1	2.42E+06	2.40E+05	10.05	3.68E+04	7.27E+03	5.06
			1.78E-02	2		1.12E+06	2.15		1.77E+04	2.08
			5.29E-03	3		3.49E+06	0.69		3.71E+04	0.99

resolved. In this case, the "initial" M is much larger than in our example while the corresponding (unperturbed) Galerkin error still is not small. For such application, the choice of even smaller values of q as compared to (56) is recommended.

2. Storage requirement and CPU time
   (a) Unperturbed CQ-BEM. Table 1 clearly illustrates the sharpness of the theoretically expected quadratic increase of both, the storage and CPU time, for the unperturbed Galerkin method with respect to M.
   (b) CQ-BEM with panel clustering. We have estimated both, the storage complexity and the CPU time (for fixed N) by $\mathcal{O}\left(Mq^4\right)$ (storage) and $\mathcal{O}\left(Mq^5\right)$ (CPU). For fixed $q \in \{0, 1, 2, 3\}$, Table 1 nicely demonstrates the sharpness of the theoretical complexity estimates with respect to M. It also can be observed that the increase $q \to q + 1$, as expected, has a significant effect due to the quartic/quintic scaling with respect to the expansion order q; however the increase of q depending on M and N is at most logarithmically. The memory savings increase from a factor 1.2 for the initial choice of M to a factor $6 - 13$ for the refined meshes while the savings for the CPU time increase up to a factor $5 - 10$ for the refined meshes.

In Table 2 we have doubled both, the number of spatial mesh points *and* the number of time steps in each refinement level. Hence, a "linear increase" with respect to the total number of unknowns from level to level corresponds to $\mathcal{O}\left(N_{\ell+1}M_{\ell+1}\right) = \mathcal{O}\left(4N_\ell M_\ell\right)$, i.e., to a factor 4. Note that for the unperturbed Galerkin method the complexity scales as $\mathcal{O}\left(NM^2\right)$ which corresponds to an increase of a factor 8 from level to level. These theoretically expected growth behavior with increasing refinement levels for the CPU time and memory requirements can clearly be observed in Table 2.

In summary, we have shown that the predicted log-linear complexity (memory and CPU-time) is clearly visible in our numerical examples for the new panel-clustering method.

Finally, in Figure 4 we compare the exact solution and the approximate ones for $T = 10$ and $m = 0$. Since in this case the solution is a constant function of the space variable for any fixed time t, we report the profiles of the solutions at a fixed point $P = (1, 0) \in \Gamma$.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* Applied Mathematics Series 55. National Bureau of Standards, U.S. Department of Commerce, 1972.

[2] B. Alpert, L. Greengard, and T. Hagstrom. Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation. *SIAM J. Numer. Anal.*, 37(4):1138–1164, 2000.

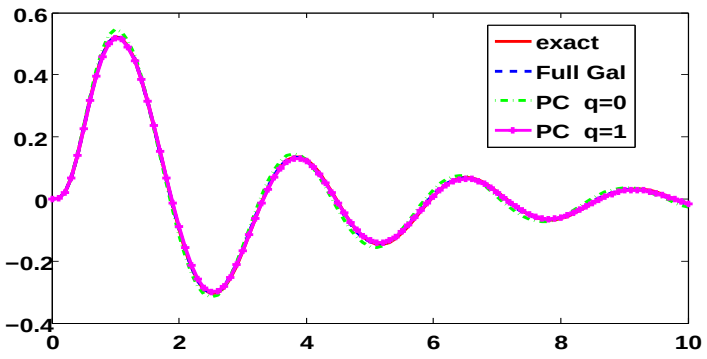

Figure 4: Comparison between the exact solution, $\varphi^{\Delta_x}_{\Delta_t,j}$ (Full Gal) and $\varphi^{\mathrm{pc},\Delta_x}_{\Delta_t,j}$ (PC) at the point $P = (1, 0)$ for $m = 0$, $T = 10$, $M = 24$ and $N = 100$.

[3] S. Amini and S. M. Kirkup. Solution of Helmholtz equation in the exterior domain by elementary boundary integral methods. *J. Comput. Phys.*, 118(2):208–221, 1995.

[4] A. Bamberger and T. Ha-Duong. Formulation variationelle espace-temps pour le calcul par potentiel retardé d'une onde acoustique. *Math. Meth. Appl. Sci.*, 8:405–435 and 598–608, 1986.

[5] L. Banjai, C. Lubich, and J. M. Melenk. Runge-Kutta convolution quadrature for operators arising in wave propagation. *Numer. Math.*, 119(1):1–20, 2011.

[6] L. Banjai and S. Sauter. Rapid solution of the wave equation in unbounded domains. *SIAM J. Numer. Anal.*, 47(1):227–249, 2008.

[7] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[8] S. Börm. *Efficient Numerical Methods for Non-Local Operators*. European Mathematical Society (EMS), Zürich, 2010.

[9] W. Dahmen, B. Faermann, I. Graham, W. Hackbusch, and S. Sauter. Inverse Inequalities on Non-Qasiuniform Meshes and Applications to the Mortar Element Method. *Math. Comp.*, 73(247):1107–1138, 2003.

[10] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31(139):629–651, 1977.

[11] S. Falletta and G. Monegato. An exact non reflecting boundary condition for 2D time-dependent wave equation problems. *Wave Motion*, 51(1):168–192, 2014.

[12] S. Falletta, G. Monegato, and L. Scuderi. A space-time BIE method for nonhomogeneous exterior wave equation problems. The Dirichlet case. *IMA J. Numer. Anal.*, 32(1):202–226, 2012.

[13] S. Falletta, G. Monegato, and L. Scuderi. A space-time BIE method for wave equation problems: the (two-dimensional) Neumann case. *IMA J. Numer. Anal.*, 34(1):390–434, 2014.

[14] S. Falletta and S. A. Sauter. Functional estimates for derivatives of the modified Bessel function K_0 and related exponential functions. *J. Math. Anal. Appl.*, 417(2):559–579, 2014.

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

[16] L. Greengard and V. Rokhlin. A New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions. *Acta Numerica*, 6:229–269, 1997.

[17] M. J. Grote and I. Sim. Local nonreflecting boundary condition for time-dependent multiple scattering. *J. Comput. Phys.*, 230(8):3135–3154, 2011.

[18] T. Ha-Duong. On Retarded Potential Boundary Integral Equations and their Discretization. In M. Ainsworth, P. Davies, D. Duncan, P. Martin, and B. Rynne, editors, *Computational Methods in Wave Propagation*, volume 31, pages 301–336, Heidelberg, 2003. Springer.

[19] T. Ha-Duong, B. Ludwig, and I. Terrasse. A Galerkin BEM for transient acoustic scattering by an absorbing obstacle. *Int. J. Numer. Meth. Engng*, 57:1845–1882, 2003.

[20] W. Hackbusch and S. Börm. $\mathcal{H}^2$-matrix approximation of integral operators by interpolation. *Appl. Numer. Math.*, 43(1-2):129–143, 2002.

[21] W. Hackbusch, B. Khoromskij, and S. Sauter. On $\mathcal{H}^2$-matrices. In H.-J. Bungartz, R. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–30, Heidelberg, 2000. Springer-Verlag.

[22] W. Hackbusch, W. Kress, and S. Sauter. Sparse convolution quadrature for time domain boundary integral formulations of the wave equation by cutoff and panel-clustering. In M. Schanz and O. Steinbach, editors, *Boundary Element Analysis: Mathematical Aspects and Applications*, volume 18, pages 113–134. Springer Lecture Notes in Applied and Computational Mechanics, 2006.

[23] W. Hackbusch, W. Kress, and S. Sauter. Sparse Convolution Quadrature for Time Domain Boundary Integral Formulations of the Wave Equation. *IMA J.Numer. Anal.*, 29:158–179, 2009.

[24] W. Hackbusch and Z. Nowak. On the Fast Matrix Multiplication in the Boundary Element Method by Panel-Clustering. *Numerische Mathematik*, 54:463–491, 1989.

[25] T. Hagstrom and S. I. Hariharan. A formulation of asymptotic and exact boundary conditions using local operators. *Appl. Numer. Math.*, 27(4):403–416, 1998.

[26] E. Hairer, C. Lubich, and M. Schlichte. Fast numerical solution of nonlinear Volterra convolution equations. *SIAM J. Sci. Stat. Comput.*, 6(3):532–541, 1985.

[27] W. Kress and S. Sauter. Numerical Treatment of Retarded Boundary Integral Equations by Sparse Panel Clustering. *IMA J. Numer. Anal.*, 28:162–185, 2008.

[28] M. Lopez-Fernandez and S. A. Sauter. Generalized Convolution Quadrature with Variable Time Stepping. *IMA J. Numer. Anal.*, 33(4):1156–1175, 2013.

[29] C. Lubich. Convolution quadrature and discretized operational calculus I. *Numer. Math.*, 52:129–145, 1988.

[30] C. Lubich. Convolution quadrature and discretized operational calculus II. *Numer. Math.*, 52:413–425, 1988.

[31] C. Lubich. On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations. *Numer. Math.*, 67:365–389, 1994.

[32] S. Sauter. Variable Order Panel Clustering. *Computing*, 64:223–261, 2000. Extended version:
http://www.mis.mpg.de/preprints/1999/preprint1999_52.pdf.

[33] S. Sauter and C. Schwab. *Boundary Element Methods*. Springer, Heidelberg, 2010.

[34] S. Sauter and A. Veit. Adaptive Time Discretization for Retarded Potentials. Technical Report 06-2013, Universität Zürich, 2013.

[35] S. Sauter and A. Veit. A Galerkin Method for Retarded Boundary Integral Equations with Smooth and Compactly Supported Temporal Basis Functions. *Numer. Math.*, 123:145–176, 2013.

[36] S. A. Sauter and A. Krapp. On the Effect of Numerical Integration in the Galerkin Boundary Element Method. *Numer. Math.*, 74(3):337–360, 1996.

[37] F.-J. Sayas. Retarded potentials and time domain boundary integral equations: a road-map. Technical report, Dept. of Math. Sci., University of Delaware, https://www.math.udel.edu/ fjsayas/TDBIEclassnotes2012.pdf, 2014.

[38] E. P. Stephan, M. Maischak, and E. Ostermann. Transient Boundary Element Method and Numerical Evaluation of Retarded Potentials. In M. Bubak, G. van Albada, J. Dongarra, and P. Sloot, editors, *Computational Science – ICCS 2008, LNCS, 5102*, pages 321–330, Heidelberg, 2008. Springer.