A fast sparse grid based space-time boundary element method for the nonstationary heat equation

Helmut Harbrecht · Johannes Tausch

Abstract This article presents a fast sparse grid based space-time boundary element method for the solution of the nonstationary heat equation. We make an indirect ansatz based on the thermal single layer potential which yields a first kind integral equation. This integral equation is discretized by Galerkin's method with respect to the sparse tensor product of the spatial and temporal ansatz spaces. By employing the $\mathcal{H}$-matrix and Toeplitz structure of the resulting discretized operators, we arrive at an algorithm which computes the approximate solution in a complexity that essentially corresponds to that of the spatial discretization. Nevertheless, the convergence rate is nearly the same as in case of a traditional discretization in full tensor product spaces.

Mathematics Subject Classification (2000) 35K20, 65F50, 65M38

Keywords Boundary element methods, heat equation, sparse grids

1 Introduction

The numerical solution of parabolic evolution problems arises in many applications. In case of the nonstationary heat equation, a boundary reduction by means of boundary integral equations is possible. Provided that the heat equation is homogeneous, only the n-dimensional surface $\Gamma := \partial\Omega$ needs to be discretized instead of the spatial domain $\Omega \subset \mathbb{R}^{n+1}$, $n = 1, 2$. If one uses N^Γ degrees of freedom for discretizing functions on the surface Γ and N^I degrees of freedom for discretizing functions on the time interval I, then a traditional Galerkin discretization would have $N^\Gamma \cdot N^I$ degrees of freedom. By "traditional" we mean the discretization of functions on $\Gamma \times I$ in the *full tensor product space*. On the other hand, by using the *sparse tensor product* between the spatial and temporal ansatz space, this

H. Harbrecht
Departement Mathematik und Informatik,
Universität Basel,
Basel, Schweiz.
E-mail: helmut.harbrecht@unibas.ch

Johannes Tausch
Department of Mathematics,
Southern Methodist University,
Dallas, Texas, USA.
E-mail: tausch@smu.edu

number of the degrees of freedom can be considerably reduced to essentially $\max\{N^\Gamma, N^I\}$ degrees of freedom, see e.g. [3,7,23]. Here and in the sequel, essentially means that the complexity estimate may be multiplied by (poly-) logarithmic factors. In the context of space-time discretizations, this fact has been exploited in e.g. [8,18] for finite element methods and in [5] for boundary element methods.

The nonlocality of boundary integral operators results in densely populated system matrices and algorithms that scale at least quadratically in the number of degrees of freedom, unless fast methods are used. Such methods have been developed recently for the layer potentials of the heat equation when using the full tensor product space, see e.g. [19,20], but for sparse tensor product spaces this is still an open problem.

This article presents a fast algorithm which scales essentially linearly in the number of degrees of freedom of the sparse tensor product space. Consequently, we are able to take full advantage of the reduction of the degrees of freedom. For further literature on boundary element methods for sparse grid discretizations, we refer the reader to e.g. [4,9,17,21].

The rest of the article is organized as follows. Section 2 introduces the Dirichlet problem for the heat equation and the indirect boundary integral reformulation using the thermal single layer operator. The traditional Galerkin discretization in full tensor product spaces is discussed in Section 3. The sparse tensor product discretization is then considered in Section 4. In particular, we show that the convergence rate is nearly the same as for the traditional Galerkin discretization provided that the solution offers enough smoothness in terms of Sobolev spaces of dominant mixed derivatives. Section 5 describes the numerical realization of a fast boundary element method which scales essentially linear in the dimension of the sparse tensor product space. This algorithm heavily relies upon the fact that the stiffness matrix is Toeplitz in time. It remains to show that the the spatial portion of the system matrix can also be applied efficiently. This is the topic of Section 6 while the related error analysis is derived in Section 7. Finally, numerical results obtained with our impementation of the algorithm is presented in Section 8.

To keep the technical level of the discussion at a minimum, we focus here on the thermal single layer potential operator. The treatment of the double layer, adjoint and hypersingular operators is analogous which permits the solution of a wide range of inital boundary value problems of the Heat equation. The discussion is limited to homogenous initial conditions and source terms. While the efficient treatment of the corresponding integral operators with sparse grids is conceivable, the implementation would likely require significant modifications of the methodology.

2 Problem formulation

Let $\Omega \subset \mathbb{R}^{n+1}$, $n = 1, 2$, be a simply connected domain with piecewise smooth boundary $\Gamma := \partial\Omega$ and let $I = (0, T)$ be a time interval for for a given $T > 0$. We consider the following Dirichlet boundary problem for the heat equation: Seek $u \in H^1(\Omega) \otimes L^2(I) \cap H^{-1}(\Omega) \otimes H^1(I)$, such that

$$\partial_t u - \Delta u = 0 \quad \text{in } \Omega \times I \tag{2.1}$$

with boundary condition

$$u = f \quad \text{on } \Gamma \times I \tag{2.2}$$

and initial condition

$$u = 0 \quad \text{on } \Omega \times \{0\}. \tag{2.3}$$

To solve the problem (2.1)–(2.3), we introduce the thermal single layer operator

$$\mathcal{V}g(\mathbf{x}, t) = \int_0^t \int_\Gamma G(\|\mathbf{x} - \mathbf{y}\|, t - \tau) g(\mathbf{y}, \tau) \, d\sigma_{\mathbf{y}} \, d\tau \tag{2.4}$$

where $\mathbf{x} \in \Gamma$ and $G(\cdot, \cdot)$ is the heat kernel, given by

$$G(r, t) = \frac{1}{(4\pi t)^{\frac{n+1}{2}}} \exp\left(-\frac{r^2}{4t}\right), \quad t \geq 0 \tag{2.5}$$

and $G(r, t) = 0$ if $t < 0$.

In view of the continuity of the single layer potential operator at the boundary, the ansatz

$$u(\mathbf{x}, t) = \int_0^t \int_\Gamma G(\|\mathbf{x} - \mathbf{y}\|, t - \tau) q(\mathbf{y}, \tau) \, d\sigma_{\mathbf{y}} \, d\tau \tag{2.6}$$

amounts to the boundary integral equation

$$\mathcal{V}q = f \quad \text{on } \Gamma \times I. \tag{2.7}$$

Once (2.7) has been solved for q, the solution u of the heat equation (2.1)–(2.3) can be computed for all $(\mathbf{x}, t) \in \Omega \times I$ by means of (2.6).

To describe the mapping properties of the boundary integral operator $\mathcal{V}$, let us consider for $r, s \geq 0$ the anisotropic Sobolev spaces of the following form

$$\mathcal{H}^{r,s}(\Gamma \times I) := H^r(\Gamma) \otimes L^2(I) \cap L^2(\Gamma) \otimes H_0^s(I),$$

where the index 0 indicates that zero initial conditions at $t = 0$ are incorporated. The norm of $\mathcal{H}^{r,s}(\Gamma \times I)$ is

$$\|u\|_{\mathcal{H}^{r,s}(\Gamma \times I)} = \|u\|_{H^r(\Gamma) \otimes L^2(I)} + \|u\|_{L^2(\Gamma) \otimes H^s(I)}.$$

Moreover, if $r, s < 0$, the space $\mathcal{H}^{r,s}(\Gamma \times I)$ is defined by duality, i.e., $\mathcal{H}^{r,s}(\Gamma \times I) := \left(\mathcal{H}^{-r,-s}(\Gamma \times I)\right)'$. Then, in accordance with [6,16], the operator $\mathcal{V}$ defines a bilinear form on $\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)$ which is continuous

$$\langle \mathcal{V}p, q \rangle_{L^2(\Gamma \times I)} \lesssim \|p\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \|q\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \quad \text{for all } p, q \in \mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)$$

and elliptic

$$\langle \mathcal{V}p, p \rangle_{L^2(\Gamma \times I)} \gtrsim \|p\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)}^2 \quad \text{for all } p \in \mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I).$$

Consequently, the boundary integral equation (2.7) is uniquely solvable provided that the right hand side satisfies $f \in \mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)$.

Note that here and in the following $\lesssim$ and $\gtrsim$ indicate that the inequalities hold up to positive multiplicative constants. Further, $a \sim b$ means that $a \lesssim b \lesssim a$.

3 Galerkin discretization

For the Galerkin discretization, we consider two sequences of nested spaces

$$V_0^\Gamma \subset V_1^\Gamma \subset \cdots \subset V_{\ell_s}^\Gamma \subset \cdots \subset L^2(\Gamma), \qquad V_0^I \subset V_1^I \subset \cdots \subset V_{\ell_t}^I \subset \cdots \subset L^2(I).$$

We shall assume that these ansatz spaces are generated by *single-scale bases* $\Phi_{\ell_s}^\Gamma = \{\varphi_{\ell_s,k_s}^\Gamma\}_{k_s \in \Delta_{\ell_s}^\Gamma}$ and $\Phi_{\ell_t}^I = \{\varphi_{\ell_t,k_t}^I\}_{k_t \in \Delta_{\ell_t}^I}$, respectively, that is

$$|\Delta_{\ell_s}^\Gamma| = \dim V_{\ell_s}^\Gamma \sim 2^{\ell_s n}, \quad |\Delta_{\ell_t}^I| = \dim V_{\ell_t}^I \sim 2^{\ell_t}.$$

and

$$V_{\ell_s}^\Gamma = \operatorname{span} \Phi_{\ell_s}^\Gamma, \quad V_{\ell_t}^I = \operatorname{span} \Phi_{\ell_t}^I.$$

We denote the approximation power of the ansatz spaces by d_s and d_t, i.e.,

$$\inf_{v_{\ell_s} \in V_{\ell_s}^\Gamma} \|v - v_{\ell_s}\|_{L^2(\Gamma)} \lesssim 2^{-\ell_s d_s} \|v\|_{H^{d_s}(\Gamma)}, \quad \inf_{v_{\ell_t} \in V_{\ell_t}^I} \|v - v_{\ell_t}\|_{L^2(I)} \lesssim 2^{-\ell_t d_t} \|v\|_{H^{d_t}(I)}.$$

For example, the piecewise constant ($d_s = 1$) or continuous piecewise linear ($d_s = 2$) ansatz functions on a sequence of meshes, obtained by uniform refinement, satisfy our assumptions on the spatial ansatz spaces $V_{\ell_s}^\Gamma$.

We choose a finest level L_s for space and L_t for time and write $\boldsymbol{L} := (L_s, L_t)$. Due to Céa's lemma, a Galerkin scheme for (2.7) in the tensor product space $U_{\boldsymbol{L}}^{\Gamma \times I} := V_{L_s}^\Gamma \otimes V_{L_t}^I$ leads to the error estimate

$$\|q - q_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2},-\frac{1}{4}}(\Gamma \times I)} \lesssim \left(2^{-\frac{L_s}{2}} + 2^{-\frac{L_t}{4}}\right)\left(2^{-L_s d_s} + 2^{-L_t d_t}\right) \|q\|_{\mathcal{H}^{d_s,d_t}(\Gamma \times I)}, \qquad (3.8)$$

provided that the boundary Γ and the given Dirichlet datum f, and thus the solution q, are smooth enough, see [6,16]. As easily seen from (3.8), in case of $d_s = 2d_t$, the optimal choice is $L_t = 2L_s$.

4 Sparse tensor product discretization

The tensor product space $U_{\boldsymbol{L}}^{\Gamma \times I} = V_{L_s}^\Gamma \otimes V_{L_t}^I$ contains $\dim V_{L_s}^\Gamma \cdot \dim V_{L_t}^I \sim 2^{L_s n} \cdot 2^{L_t}$ degrees of freedom. Compared with this, finite element methods which are based on a sparse grid discretization of the space-time cylinder offer essentially the complexity $\mathcal{O}(2^{L_s(n+1)})$, see e.g. [3,8,18] and the references therein. This means, the time discretization comes for free, at least from a complexity point of view. As a consequence, although algorithms are available which solve the heat equation by layer potentials in essentially linear complexity relative to the number of unknowns in the tensor product space $U_{\boldsymbol{L}}^{\Gamma \times I}$ (cf. [14,15,19,20]), there is no gain in the use of boundary integral equations. To overcome this obstruction, as in [5], we shall consider a Galerkin discretization in the *sparse tensor product* of the ansatz spaces $V_{L_s}^\Gamma$ and $V_{L_t}^I$.

The sparse space-time tensor Galerkin discretization is based on multilevel decompositions of the ansatz spaces. To that end, we set

$$W_{\ell_s}^\Gamma := V_{\ell_s}^\Gamma \ominus V_{\ell_s-1}^\Gamma, \quad W_{\ell_s}^\Gamma = \operatorname{span} \Psi_{\ell_s}^\Gamma,$$
$$W_{\ell_t}^I := V_{\ell_t}^I \ominus V_{\ell_t-1}^I, \quad W_{\ell_t}^I = \operatorname{span} \Psi_{\ell_t}^I.$$

The basis functions $\Psi_{\ell_s}^\Gamma = \{\psi_{\ell_s,k_s}^\Gamma\}_{k_s \in \nabla_{\ell_s}^\Gamma}$ and $\Psi_{\ell_t}^I = \{\psi_{\ell_t,k_t}^I\}_{k_t \in \nabla_{\ell_t}^I}$ are hierarchical bases or wavelets. Instead of a discretization in the full tensor product space

$$U_{\boldsymbol{L}} := V_{L_s}^\Gamma \otimes V_{L_t}^I = \bigoplus_{\frac{\ell_s}{L_s}, \frac{\ell_t}{L_t} \leq 1} W_{\ell_s}^\Gamma \otimes W_{\ell_t}^I,$$

we will consider a discretization in the sparse tensor product space

$$\widehat{U}_{\boldsymbol{L}} := V_{L_s}^{\Gamma} \widehat{\otimes} V_{L_t}^{I} = \bigoplus_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1} W_{\ell_s}^{\Gamma} \otimes W_{\ell_t}^{I}. \tag{4.9}$$

The following lemma has been proven in [7, 8]. It states that the time discretization is essentially free provided that $2L_s \gtrsim L_t$.

Lemma 1 *For $L_s = \sigma L_t \to \infty$, where $\sigma > 0$ is fixed, the sparse tensor product space (4.9) satisfies*

$$\dim \widehat{U}_{\boldsymbol{L}} \sim \begin{cases} 2^{L_s n} + 2^{L_t}, & \text{if } L_s n \neq L_t, \\ L_s 2^{L_s n}, & \text{if } L_s n = L_t. \end{cases}$$

On the other hand, the approximation property in the sparse tensor product space is essentially the same as in the full analogue, provided that we spend some extra smoothness in terms of the mixed Sobolev spaces

$$\mathcal{H}_{\text{mix}}^{r,s}(\Gamma \times I) := H^r(\Gamma) \otimes H_0^s(I).$$

In particular, we find the following result for the best approximation in the energy space under consideration.

Lemma 2 *For $L_s = \sigma L_t \to \infty$, where $\sigma > 0$ is fixed, there holds*

$$\inf_{\widehat{v}_{\boldsymbol{L}} \in \widehat{U}_{\boldsymbol{L}}^{\Gamma \times I}} \|v - \widehat{v}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \lesssim \sqrt{L_s} 2^{-\frac{L_s L_t}{4L_s + 2L_t}} (2^{-L_s d_s} + 2^{-L_t d_t}) \|v\|_{\mathcal{H}_{\text{mix}}^{d_s, d_t}(\Gamma \times I)}$$

provided that $L_s d_s \neq L_t d_t$. In case of equality, i.e., $L_s d_s = L_t d_t$, an additional logarithmic factor appears:

$$\inf_{\widehat{v}_{\boldsymbol{L}} \in \widehat{U}_{\boldsymbol{L}}^{\Gamma \times I}} \|v - \widehat{v}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \lesssim L_s 2^{-\frac{L_s L_t}{4L_s + 2L_t}} 2^{-L_s d_s} \|v\|_{\mathcal{H}_{\text{mix}}^{d_s, d_t}(\Gamma \times I)}.$$

Proof We shall denote the L^2-orthogonal projection onto the sparse tensor product space $\widehat{U}_{\boldsymbol{L}}$ by $\widehat{\Pi}_{\boldsymbol{L}} : L^2(\Gamma \times I) \to \widehat{U}_{\boldsymbol{L}}$. Then, by a standard duality argument, we obtain

$$\inf_{\widehat{v}_{\boldsymbol{L}} \in \widehat{U}_{\boldsymbol{L}}^{\Gamma \times I}} \|v - \widehat{v}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \leq \sup_{u \in \mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)} \frac{\langle v - \widehat{\Pi}_{\boldsymbol{L}}v, u \rangle_{L^2(\Gamma \times I)}}{\|u\|_{\mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)}}$$

$$= \sup_{u \in \mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)} \frac{\langle v - \widehat{\Pi}_{\boldsymbol{L}}v, u - \widehat{\Pi}_{\boldsymbol{L}}u \rangle_{L^2(\Gamma \times I)}}{\|u\|_{\mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)}}$$

$$\leq \|v - \widehat{\Pi}_{\boldsymbol{L}}v\|_{L^2(\Gamma \times I)} \sup_{u \in \mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)} \frac{\|u - \widehat{\Pi}_{\boldsymbol{L}}u\|_{L^2(\Gamma \times I)}}{\|u\|_{\mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I)}}.$$

From [5, Lemma 5.2] it follows that

$$\mathcal{H}^{\frac{1}{2}, \frac{1}{4}}(\Gamma \times I) \subset \mathcal{H}_{\text{mix}}^{\frac{\lambda}{2}, \frac{1-\lambda}{4}}(\Gamma \times I) \text{ for all } \lambda \in [0, 1].$$

Hence, we conclude

$$\inf_{\widehat{v}_{\boldsymbol{L}} \in \widehat{U}_{\boldsymbol{L}}^{\Gamma \times I}} \|v - \widehat{v}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2}, -\frac{1}{4}}(\Gamma \times I)} \lesssim \|v - \widehat{\Pi}_{\boldsymbol{L}}v\|_{L^2(\Gamma \times I)} \sup_{u \in \mathcal{H}_{\text{mix}}^{\frac{\lambda}{2}, \frac{1-\lambda}{4}}(\Gamma \times I)} \frac{\|u - \widehat{\Pi}_{\boldsymbol{L}}u\|_{L^2(\Gamma \times I)}}{\|u\|_{\mathcal{H}_{\text{mix}}^{\frac{\lambda}{2}, \frac{1-\lambda}{4}}(\Gamma \times I)}}. \tag{4.10}$$

To bound the first term on the right hand side of (4.10), we use the error estimate

$$\|v - \widehat{\Pi}_{\boldsymbol{L}}v\|_{L^2(\Gamma \times I)} \lesssim \begin{cases} (2^{-L_s d_s} + 2^{-L_t d_t}) \|v\|_{\mathcal{H}_{\text{mix}}^{d_s, d_t}(\Gamma \times I)}, & \text{if } L_s d_s \neq L_t d_t, \\ \sqrt{L_s} 2^{-L_s d_s} \|v\|_{\mathcal{H}_{\text{mix}}^{d_s, d_t}(\Gamma \times I)}, & \text{if } L_s d_s = L_t d_t, \end{cases} \tag{4.11}$$

which has been shown in [7].

To bound the second term on the right hand side of (4.10), we use again the error estimate of [7] to arrive at

$$\|u - \widehat{\Pi}_{\boldsymbol{L}} u\|_{L^2(\Gamma \times I)} \lesssim (2^{-\frac{\lambda}{2}L_s} + 2^{-\frac{1-\lambda}{4}L_t})\|u\|_{\mathcal{H}_{\mathrm{mix}}^{\frac{\lambda}{2},\frac{1-\lambda}{4}}(\Gamma \times I)}$$

for all $u \in \mathcal{H}_{\mathrm{mix}}^{\frac{\lambda}{2},\frac{1-\lambda}{4}}(\Gamma \times I)$, provided that $2\lambda L_s \neq (1-\lambda)L_t$. In the case $2\lambda L_s = (1-\lambda)L_t$, which means that

$$\lambda = \frac{L_t}{2L_s + L_t} = \frac{1}{2\sigma + 1},$$

an additional logarithmic factor shows up:

$$\|u - \widehat{\Pi}_{\boldsymbol{L}} u\|_{L^2(\Gamma \times I)} \lesssim \sqrt{L_s} 2^{-\frac{\lambda}{2}L_s}\|u\|_{\mathcal{H}_{\mathrm{mix}}^{\frac{\lambda}{2},\frac{1-\lambda}{4}}(\Gamma \times I)} = \sqrt{L_s} 2^{-\frac{L_s L_t}{4L_s + 2L_t}}\|u\|_{\mathcal{H}_{\mathrm{mix}}^{\frac{\lambda}{2},\frac{1-\lambda}{4}}(\Gamma \times I)}. \qquad (4.12)$$

This choice yields the best attainable rate since the two terms $2^{-\frac{\lambda}{2}L_s}$ and $2^{-\frac{1-\lambda}{4}L_t}$ are balanced[1].

Plugging the estimates (4.11) and (4.12) into the duality argument (4.10) yields finally the desired result.   □

Remark 1 Along the lines of [5–7], we can determine the best cost complexity of the tensor product approximation and the sparse tensor product approximation, respectively, as $L_s = \sigma L_t \to \infty$. If we consider piecewise linear ansatz function in space, i.e., $d_s = 2$, and piecewise constant ansatz function in time, i.e., $d_t = 1$, we obtain the best cost complexity for the discretization in the tensor product space $U_{\boldsymbol{L}}$ for the choice $L_s = 2L_t$: When using N degrees of freedom for the discretization, it follows

$$\|q - q_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2},-\frac{1}{4}}(\Gamma \times I)} \lesssim \begin{cases} N^{-\frac{5}{6}}\|q\|_{\mathcal{H}^{2,1}(\Gamma \times I)}, & \text{if } n = 1, \\ N^{-\frac{5}{8}}\|q\|_{\mathcal{H}^{2,1}(\Gamma \times I)}, & \text{if } n = 2. \end{cases}$$

Compared with this, the best cost complexity for the Galerkin discretization with respect to the sparse tensor product space $\widehat{U}_{\boldsymbol{L}}$ is given by equilibrating the degrees of freedom in $V_{L_s}^{\Gamma}$ and $V_{L_t}^{I}$. For N degrees of freedom, we find then the estimate

$$\|q - q_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2},-\frac{1}{4}}(\Gamma \times I)} \lesssim \begin{cases} N^{-\frac{7}{6}}(\log N)^{\frac{7}{6}+\frac{1}{2}}\|q\|_{\mathcal{H}_{\mathrm{mix}}^{2,1}(\Gamma \times I)}, & \text{if } n = 1 \text{ and } L_s = L_t, \\ N^{-\frac{9}{8}}(\log N)^{\frac{9}{8}+1}\|q\|_{\mathcal{H}_{\mathrm{mix}}^{2,1}(\Gamma \times I)}, & \text{if } n = 2 \text{ and } 2L_s = L_t. \end{cases}$$

We see that the cost complexity is nearly doubled when using the sparse tensor product discretization in $n = 2$ dimensions. Moreover, for $n = 1$ dimensions, the piecewise linear discretization in space does not pay off since the choice $d_s = 1$ would essentially give the same cost complexity.

5 Algorithms

5.1 Fast matrix-vector multiplication

Throughout the article, the basis in $\widehat{U}_{\boldsymbol{L}}$ will be denoted by

$$\widehat{\Psi}_{\boldsymbol{L}} := \left\{ \widehat{\psi}_{\boldsymbol{\ell},\mathbf{k}} = \psi_{\ell_s,k_s}^{\Gamma} \otimes \psi_{\ell_t,k_t}^{I} : \mathbf{k} = (k_s, k_t) \in \nabla_{\boldsymbol{\ell}} := \nabla_{\ell_s}^{\Gamma} \times \nabla_{\ell_t}^{I}, \ \frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1 \right\}.$$

---

[1] By balancing these terms, we obtain an improvement of the results in [5].

Then, the Galerkin matrix $\widehat{\mathbf{V}}_{\boldsymbol{L}} = \langle \mathcal{V}\widehat{\boldsymbol{\Psi}}_{\boldsymbol{L}}, \widehat{\boldsymbol{\Psi}}_{\boldsymbol{L}}\rangle_{L^2(\Gamma \times I)}$ consists of the block matrices

$$\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'} := \langle \mathcal{V}(\Psi^{\Gamma}_{\ell'_s} \otimes \Psi^I_{\ell'_t}), \Psi^{\Gamma}_{\ell_s} \otimes \Psi^I_{\ell_t}\rangle_{L^2(\Gamma \times I)} \tag{5.13}$$

where $\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t}, \frac{\ell'_s}{L_s} + \frac{\ell'_t}{L_t} \leq 1$. Here, the block $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$ has asymptotically the dimension $2^{\ell_s n + \ell_t} \times 2^{\ell'_s n + \ell'_t}$. Obviously, by writing $\widehat{\mathbf{u}}_{\boldsymbol{L}} = [\mathbf{u}_{\boldsymbol{\ell}}]_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1}$, the matrix-vector multiplication $\widehat{\mathbf{w}}_{\boldsymbol{L}} = \widehat{\mathbf{V}}_{\boldsymbol{L}}\widehat{\mathbf{u}}_{\boldsymbol{L}}$ can be block wise computed by

$$\widehat{\mathbf{w}}_{\boldsymbol{L}} = [\mathbf{w}_{\boldsymbol{\ell}}]_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1} = \left[\sum_{\frac{\ell'_s}{L_s} + \frac{\ell'_t}{L_t} \leq 1} \mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\mathbf{u}_{\boldsymbol{\ell}'} \right]_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1} = \widehat{\mathbf{V}}_{\boldsymbol{L}}\widehat{\mathbf{u}}_{\boldsymbol{L}}. \tag{5.14}$$

Lemma 3 *Assume that the block matrix-vector product $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\mathbf{u}_{\boldsymbol{\ell}'}$ is computable in complexity $\mathcal{O}\big(M \cdot 2^{\max\{\ell_s n + \ell_t, \ell'_s n + \ell'_t\}}\big)$. Then, the matrix-vector product $\widehat{\mathbf{w}}_{\boldsymbol{L}} = \widehat{\mathbf{V}}_{\boldsymbol{L}}\widehat{\mathbf{u}}_{\boldsymbol{L}}$ is of complexity $\mathcal{O}\big(ML_sL_t \dim(\widehat{U}_{\boldsymbol{L}})\big)$.*

Proof The assertion follows immediately from (5.14) and

$$\sum_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t}, \frac{\ell'_s}{L_s} + \frac{\ell'_t}{L_t} \leq 1} M \cdot 2^{\max\{\ell_s n + \ell_t, \ell'_s n + \ell'_t\}}$$

$$= \sum_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1} M \cdot \left(\sum_{\substack{\frac{\ell'_s}{L_s} + \frac{\ell'_t}{L_t} \leq 1 \\ \ell_s n + \ell_t \leq \ell'_s n + \ell'_t}} 2^{\ell'_s n + \ell'_t} + \sum_{\substack{\frac{\ell'_s}{L_s} + \frac{\ell'_t}{L_t} \leq 1 \\ \ell_s n + \ell_t > \ell'_s n + \ell'_t}} 2^{\ell_s n + \ell_t} \right)$$

$$\lesssim \sum_{\frac{\ell_s}{L_s} + \frac{\ell_t}{L_t} \leq 1} M \cdot \left(\dim(\widehat{U}_{\boldsymbol{L}}) + 2^{\ell_s n + \ell_t} L_s L_t \right)$$

$$\lesssim ML_sL_t \dim(\widehat{U}_{\boldsymbol{L}}).$$

$\square$

5.2 Restrictions and prolongations

Since it is algorithmically difficult to compute matrices in wavelet coordinates and with ansatz and test functions on different levels, we use restrictions and prolongations to realize matrix vector products with $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$ in single-scale spaces.

Because $W^{\Gamma}_{\ell_s} \subset V^{\Gamma}_{\ell'_s}$ for any $\ell_s \leq \ell'_s$, we can represent a given function $u_{\ell_s} \in W^{\Gamma}_{\ell_s}$ in the space $V^{\Gamma}_{\ell'_s}$. Such a prolongation will be denoted by $J^{\ell'_s}_{\ell_s}$. Its discrete counterpart $\mathbf{J}^{\ell'_s}_{\ell_s}$ can obviously be applied to a given vector $\mathbf{u}_{\ell_s}$ in complexity $\mathcal{O}\big(2^{\ell'_s n}\big)$. Vice versa, a function $u_{\ell'_s}$ in $V^{\Gamma}_{\ell'_s}$ can be restricted to the space $W^{\Gamma}_{\ell_s}$ which we denote by $J^{\ell_s}_{\ell'_s}$. The cost of the corresponding discrete operation $\mathbf{J}^{\ell_s}_{\ell'_s}\mathbf{u}_{\ell'_s}$ is of the order $\mathcal{O}\big(2^{\ell'_s n}\big)$. Note that $\big(\mathbf{J}^{\ell_s}_{\ell'_s}\big)^T = \mathbf{J}^{\ell'_s}_{\ell_s}$.

Likewise, due to $W^I_{\ell_t} \subset V^I_{\ell'_t}$ for any $\ell_t \leq \ell'_t$, corresponding operators $I^{\ell'_t}_{\ell_t}$ and $I^{\ell_t}_{\ell'_t}$ exist with respect to the time. Their discrete counterparts are denoted by $\mathbf{I}^{\ell'_t}_{\ell_t}$ and $\mathbf{I}^{\ell_t}_{\ell'_t}$, where the application to a vector costs $\mathcal{O}\big(2^{\ell'_t}\big)$ operations.

In the following, we will use the notational convention

$$\tilde{\ell}_s := \max\{\ell_s, \ell'_s\} \quad \text{and} \quad \tilde{\ell}_t := \max\{\ell_t, \ell'_t\}.$$
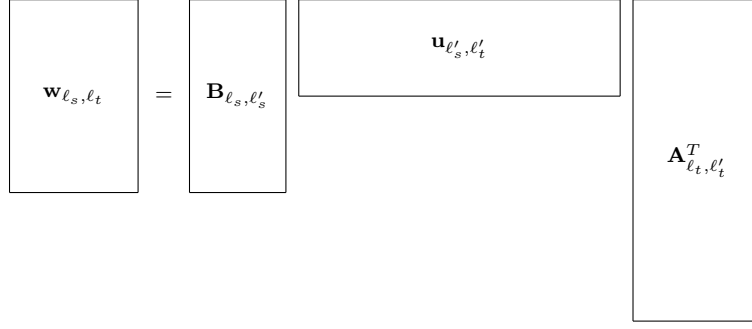
Fig. 5.1 Visualization of the matrix-vector product in (5.18). Here, it is cheaper to perform first the multiplication $\mathbf{B}_{\ell_s,\ell_s'}\mathbf{u}_{\ell_s',\ell_t'}$ and then the multiplication of the result with $\mathbf{A}_{\ell_t,\ell_t'}^T$.

Thus, we obtain the representation in the single-scale spaces

$$\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'} = \left(\mathbf{I}_{\tilde{\ell}_t}^{\ell_t} \otimes \mathbf{J}_{\tilde{\ell}_s}^{\ell_s}\right)\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}\left(\mathbf{I}_{\ell_t'}^{\tilde{\ell}_t} \otimes \mathbf{J}_{\ell_s'}^{\tilde{\ell}_s}\right) \tag{5.15}$$

where $\tilde{\boldsymbol{\ell}} = (\tilde{\ell}_s, \tilde{\ell}_t)$ and

$$\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}} := \langle \mathcal{V}(\Phi_{\tilde{\ell}_s}^{\Gamma} \otimes \Phi_{\tilde{\ell}_t}^{I}), \Phi_{\tilde{\ell}_s}^{\Gamma} \otimes \Phi_{\tilde{\ell}_t}^{I}\rangle_{L^2(\Gamma \times I)} \tag{5.16}$$

Remark 2 The dimension of the matrix $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ is asymptotically $2^{\max\{\ell_t,\ell_t'\}n+\max\{\ell_s,\ell_s'\}}$ which is, in general, larger than the dimensions of $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$. In fact, it turns out that it is not possible to compute a matrix-vector product with $\widehat{\mathbf{V}}_{\boldsymbol{L}}$ in the desired $\mathcal{O}\big(ML_sL_t\dim(\widehat{U}_{\boldsymbol{L}})\big)$ complexity, if the factors are evaluated in the sequence suggested by (5.15), even if the application of $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ has linear complexity. However, we will show below that $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ can be approximated by a sum of Kronecker products, which will lead to an algorithm with log-linear complexity in $\dim(\widehat{U}_{\boldsymbol{L}})$.

5.3 Block matrix-vector multiplication

To get a guideline for the realization of an essentially optimal block matrix-vector multiplication, let us assume from now on that $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$ is approximated by a sum of tensor products

$$\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'} \approx \sum_{i=1}^{M} \mathbf{A}_{\ell_t,\ell_t'}^{(i)} \otimes \mathbf{B}_{\ell_s,\ell_s'}^{(i)}. \tag{5.17}$$

Such a representation is also called *low-rank approximation*. Provided that for all $i = 1, \ldots, M$ the application of the matrices $\mathbf{A}_{\ell_t,\ell_t'}^{(i)}$ and $\mathbf{B}_{\ell_s,\ell_s'}^{(i)}$ to a vector can be evaluated in $\mathcal{O}\big(2^{\max\{\ell_t,\ell_t'\}}\big)$ and $\mathcal{O}\big(2^{\max\{\ell_s,\ell_s'\}n}\big)$ operations, respectively, then the matrix-vector product

$$\mathbf{w}_{\boldsymbol{\ell}} = \mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}\mathbf{u}_{\boldsymbol{\ell}'} \approx \sum_{i=1}^{M} \big(\mathbf{A}_{\ell_t,\ell_t'}^{(i)} \otimes \mathbf{B}_{\ell_s,\ell_s'}^{(i)}\big)\mathbf{u}_{\boldsymbol{\ell}'}$$

is computable within the complexity $\mathcal{O}\big(M \cdot 2^{\max\{\ell_s n+\ell_t, \ell_s' n+\ell_t'\}}\big)$. We will show this in the remainder of this section.

For a matrix $X \in \mathbb{R}^{m \times n}$, $\text{vec}(X) \in \mathbb{R}^{nm}$ is the vector that is obtained by stacking the columns of X. From the identity

$$\text{vec}(\mathbf{w}_{\boldsymbol{\ell}}^{(i)}) = (\mathbf{A}_{\ell_t,\ell_t'}^{(i)} \otimes \mathbf{B}_{\ell_s,\ell_s'}^{(i)})\text{vec}(\mathbf{u}_{\boldsymbol{\ell}'}) \iff \mathbf{w}_{\boldsymbol{\ell}}^{(i)} = \mathbf{B}_{\ell_s,\ell_s'}^{(i)}\mathbf{u}_{\boldsymbol{\ell}'}\big(\mathbf{A}_{\ell_t,\ell_t'}^{(i)}\big)^T \tag{5.18}$$

we conclude that, for $\ell_s n + \ell_t' \leq \ell_s' n + \ell_t$, it is cheaper to compute the vector $\mathbf{w}_{\boldsymbol{\ell}}^{(i)}$ in the order

$$\mathbf{z} = \mathbf{B}_{\ell_s, \ell_s'}^{(i)} \mathbf{u}_{\boldsymbol{\ell}'}, \quad \mathbf{w}_{\boldsymbol{\ell}}^{(i)} = \left(\mathbf{A}_{\ell_t, \ell_t'}^{(i)} \mathbf{z}^T \right)^T. \tag{5.19}$$

(we refer to Fig. 5.1 for a corresponding visualization). Here, the evaluation of $\mathbf{z}$ is of complexity $\mathcal{O}\big(2^{\ell_t'} \cdot 2^{\max\{\ell_s, \ell_s'\} n}\big)$ and thus the complexity for computing $\mathbf{w}_{\boldsymbol{\ell}}^{(i)}$ via (5.19) is

$$\mathcal{O}\big(2^{\ell_t'} \cdot 2^{\max\{\ell_s, \ell_s'\} n} + 2^{\ell_s n} \cdot 2^{\max\{\ell_t, \ell_t'\}}\big) = \mathcal{O}\big(2^{\max\{\ell_s n + \ell_t, \ell_s' n + \ell_t', \ell_s n + \ell_t'\}}\big).$$

Due to the supposition $\ell_s n + \ell_t' \leq \ell_s' n + \ell_t$, we have

$$\ell_s n + \ell_t' \leq (\ell_s' n + \ell_t') - \ell_t' + (\ell_s n + \ell_t) - \ell_s n$$

and thus

$$2(\ell_s n + \ell_t') \leq (\ell_s' n + \ell_t') + (\ell_s n + \ell_t) \leq 2 \max\{\ell_s n + \ell_t, \ell_s' n + \ell_t'\}.$$

Therefore, the complexity for the matrix-vector multiplication (5.19) is $\mathcal{O}\big(2^{\max\{\ell_s n + \ell_t, \ell_s' n + \ell_t'\}}\big)$ which is order optimal.

If $\ell_s n + \ell_t' > \ell_s' n + \ell_t$, we change the order of multiplication in (5.18) and compute

$$\mathbf{z} = \mathbf{A}_{\ell_t, \ell_t'}^{(i)} \mathbf{u}_{\boldsymbol{\ell}'}^T, \quad \mathbf{w}_{\boldsymbol{\ell}}^{(i)} = \mathbf{B}_{\ell_s, \ell_s'}^{(i)} \mathbf{z}^T \tag{5.20}$$

By using arguments analogous to above, one readily infers that the complexity of computing $\mathbf{w}_{\boldsymbol{\ell}}^{(i)}$ via (5.20) is also of order optimal complexity $\mathcal{O}\big(2^{\max\{\ell_s n + \ell_t, \ell_s' n + \ell_t'\}}\big)$.

Remark 3 One logarithmic factor in the cost complexity of the matrix-vector product described here can be removed by using the unidirectional principle, see e.g. [1,2,22]. Nevertheless, we have not exploited this approach for sake of simplicity in representation.

5.4 Tensor product representation of $\mathbf{V}_{\boldsymbol{\ell}, \boldsymbol{\ell}'}$

In this section we show how to compute the approximation (5.17) using the factorization in (5.15). To keep the technical level of the discussion at a minimum, we assume that the temporal spaces $V_{\ell_t}^I$ consist of piecewise constant ansatz functions on a uniform subdivision of $I = [0, T]$ into $2^{\ell_t} n_t$ intervals, where n_t is a small integer. Thus, the temporal basis functions φ_{ℓ_t, k_t}^I are scaled and translated versions of the box function.

The coefficients of the matrix $\mathbf{V}_{\tilde{\boldsymbol{\ell}}, \tilde{\boldsymbol{\ell}}}$ in (5.16) are given by

$$\left[\mathbf{V}_{\tilde{\boldsymbol{\ell}}, \tilde{\boldsymbol{\ell}}} \right]_{(k_s, k_t)(k_s', k_t')} = \int_0^T \int_0^T \left\{ \int_\Gamma \int_\Gamma G(\|\mathbf{x} - \mathbf{y}\|, t - \tau) \varphi_{\tilde{\ell}_s, k_s}^\Gamma (\mathbf{x}) \varphi_{\tilde{\ell}_s, k_s'}^\Gamma (\mathbf{y}) \, \mathrm{d}\sigma_{\mathbf{y}} \, \mathrm{d}\sigma_{\mathbf{x}} \right\}$$
$$\times \varphi_{\tilde{\ell}_t, k_t}^I (t) \varphi_{\tilde{\ell}_t, k_t'}^I (\tau) \, \mathrm{d}\tau \, \mathrm{d}t,$$

An ordering of the indices (k_s, k_t), where k_s runs fast and k_t runs slow, results in a block-Toeplitz structure of $\mathbf{V}_{\tilde{\boldsymbol{\ell}}, \tilde{\boldsymbol{\ell}}}$, because the integrals only depend on the difference of the indices k_t and k_t'. In addition, the matrix is block lower triangular because the kernel vanishes for $\tau > t$.

We now generate an $\mathcal{H}$-matrix pattern of the matrix $\mathbf{V}_{\tilde{\boldsymbol{\ell}}, \tilde{\boldsymbol{\ell}}}$, see, [10,11]. To this end, we subdivde the interval $I = [0, T]$ into 2^m equal length sub-intervals

$$I_{m, k} := \big[k 2^{-m} T, (k + 1) 2^{-m} T \big], \quad k = 0, 1, \ldots, 2^m - 1, \quad m = 0, 1, \ldots, \tilde{\ell}_t,$$
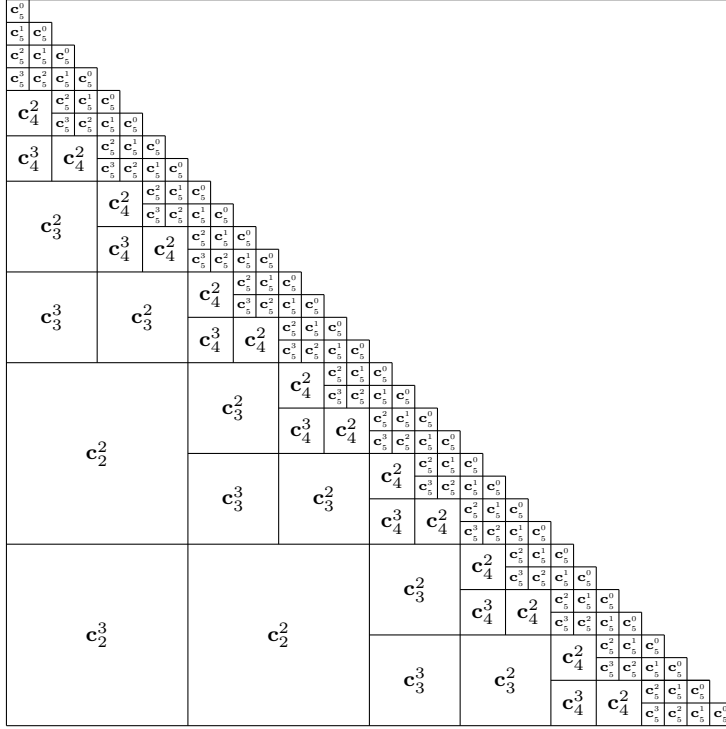
Fig. 5.2 Partitioning of $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ for the case that $\tilde{\ell}_t = 5$.

and define the index sets of level-$\tilde{\ell}_t$ temporal basis functions with support in $I_{m,k}$

$$\mathcal{I}_{m,k} := \left\{ k_t \in \Delta_{\tilde{\ell}_t}^I \ : \ \mathrm{supp} \ \varphi_{\tilde{\ell}_t,k_t}^I \subset I_{m,k} \right\}.$$

These sets are larger for smaller values of m. Moreover, for a given m they form a disjoint union of $\Delta_{\tilde{\ell}_t}^I$ because each $\varphi_{\tilde{\ell}_t,k_t}^I$ is supported in exactly one $I_{m,k}$.

Since the kernel is less peaked for a larger separation of the time variables we call $\mathcal{I}_{m,k} \times \mathcal{I}_{m,k'}$ *admissible* if $d := k - k' \geq 2$. Mind that $k \geq k'$, because the matrix is block lower triangular. The $\mathcal{H}$-matrix pattern is obtained by dividing the matrix into admissible blocks where the size of the blocks increases with the distance to the diagonal, see Fig. 5.2.

Here, the blocks $\mathbf{c}_m^d$ are given by

$$\mathbf{c}_m^d = \left[\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}\right]_{\substack{(k_s,k_t)\in\Delta_{\tilde{\ell}_s}^\Gamma \times \mathcal{I}_{md} \\ (k_s',k_t')\in\Delta_{\tilde{\ell}_s}^\Gamma \times \mathcal{I}_{m0}}}. \tag{5.21}$$

Since $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ is block-Toeplitz, it suffices to consider the cases $d = 2$ and $d = 3$ for the off-diagonal blocks where $2 \leq m \leq \tilde{\ell}_t$. These correspond to admissible index sets and will be referred to as the temporal far field. In addition, the blocks $\mathbf{c}_{\tilde{\ell}_t}^0$ and $\mathbf{c}_{\tilde{\ell}_t}^1$ appear on and near the diagonal, and will be referred to as the temporal nearfield.

The $\mathcal{H}$ pattern suggests to write $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$ as a sum of $2\tilde{\ell}_t$ block-Toeplitz matrices that contain the identical blocks. To that end, define the $(2^m \times 2^m)$-matrices

$$
\mathbf{H}_m^0 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}, \quad
\mathbf{H}_m^1 = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \quad
\mathbf{H}_m^2 = \begin{bmatrix} 0 & & & \\ 0 & 0 & & \\ 1 & 0 & 0 & \\ & \ddots & \ddots & \ddots \\ & & 1 & 0 & 0 \end{bmatrix}
\tag{5.22}
$$

and

$$
\mathbf{H}_m^3 = \begin{bmatrix} 0 & & & & \\ 0 & 0 & & & \\ 0 & 0 & 0 & & \\ 1 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 \\ & & & & \ddots & \ddots & \ddots & \ddots \\ & & & & & 1 & 0 & 0 & 0 \end{bmatrix}.
\tag{5.23}
$$

Note that the ones and zeros in the third subdiagonal of $\mathbf{H}_m^3$ alternate because of the pattern in which the blocks $\mathbf{c}_m^3$ appear in the matrix $\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}}$.

With these notations, one obtains the decomposition

$$
\mathbf{V}_{\tilde{\boldsymbol{\ell}},\tilde{\boldsymbol{\ell}}} = \sum_{d \in \{0,1\}} \mathbf{H}_{\tilde{\ell}_t}^d \otimes \mathbf{c}_{\tilde{\ell}_t}^d \; + \sum_{\substack{m \in \{2,\ldots,\tilde{\ell}_t\} \\ d \in \{2,3\}}} \mathbf{H}_m^d \otimes \mathbf{c}_m^d \, .
\tag{5.24}
$$

Temporal far-field.

Consider the block $\mathbf{c}_m^d$ in the temporal far-field where the ansatz- and test functions $\varphi_{\tilde{\ell}_t,k_t}^I$ and $\varphi_{\tilde{\ell}_t,k_t'}^I$ have support inside $I_{m,d}$ and $I_{m,0}$, respectively. Since $d \in \{2,3\}$, the kernel is smooth and can be well approximated by a degenerate kernel expansion. Such an expansion can be obtained, for instance, by interpolation. This is most conveniently achieved in the local coordinates t', τ' of the respective intervals. For $t \in I_{m,d}$ and $\tau \in I_{m,0}$ they are given by

$$
t = T2^{-m}\left(d + \frac{t'}{2} + \frac{1}{2}\right), \quad \tau = T2^{-m}\left(\frac{\tau'}{2} + \frac{1}{2}\right). \quad -1 \le \tau', t' \le 1,
\tag{5.25}
$$

Thus, setting $\widetilde{\mathbf{r}} = \mathbf{r}/\sqrt{T2^{-m}}$, it follows that

$$
G\left(\|\mathbf{r}\|, t - \tau\right) = (T2^{-m})^{-\frac{n+1}{2}} G\left(\|\widetilde{\mathbf{r}}\|, d + \frac{1}{2}(t' - \tau')\right),
$$

$$
= (T2^{-m})^{-\frac{n+1}{2}} \left\{ \sum_{i,i'=0}^{p_t} G\left(\|\widetilde{\mathbf{r}}\|, d + \frac{1}{2}(\omega^{(i)} - \omega^{(i')})\right) L_i(t') L_{i'}(\tau') + E_{p_t}\left(\|\widetilde{\mathbf{r}}\|\right) \right\}
$$

$$
= \sum_{i,i'=0}^{p_t} G\left(\|\mathbf{r}\|, t^{(i)} - \tau^{(i')}\right) L_i(t') L_{i'}(\tau') + (T2^{-m})^{-\frac{n+1}{2}} E_{p_t}\left(\|\widetilde{\mathbf{r}}\|\right).
\tag{5.26}
$$

Here, $\omega^{(i)}$ and $\omega^{(i')}$ are interpolation nodes in $(-1,1)$, $t^{(i)}$, $\tau^{(i')}$ are their images under the transformation (5.25), L_i are Lagrange polynomials and p_t is the interpolation order. The error $E_{p_t}(r)$ will be analyzed in section 7.

Neglecting $E_{p_t}(r)$ and substituting the series of (5.26) in (5.21) results in a decomposition into Kronecker products. It follows that

$$\mathbf{c}_m^d \approx \sum_{i,i'=0}^{p_t} \mathbf{a}^{(m,i)} \left(\mathbf{a}^{(m,i')}\right)^T \otimes \mathbf{b}_{\ell_s}^{(m,d,i,i')},$$ (5.27)

where

$$\left[\mathbf{a}^{(m,i)}\right]_{k_t} = \int_{I_{m,0}} L_i(\tau')\Phi_{\ell_t,k_t}^I(\tau)d\tau\,,$$

$$\left[\mathbf{b}_{\tilde{\ell}_s}^{(m,d,i,i')}\right]_{k_s,k'_s} = \int_\Gamma \int_\Gamma G(\|\mathbf{x}-\mathbf{y}\|,t^{(i)}-\tau^{(i')})\Phi_{\tilde{\ell}_s,k_s}^\Gamma(\mathbf{x})\Phi_{\tilde{\ell}_s,k'_s}^\Gamma(\mathbf{y})\,\mathrm{d}\sigma_\mathbf{y}\,\mathrm{d}\sigma_\mathbf{x}\,.$$

Note that $\mathbf{a}^{(m,i)}$ is a vector of length $2^{-m}\left|\Delta_{\tilde{\ell}_t}\right|$ and $\mathbf{b}_{\tilde{\ell}_s}^{(m,d,i,i')}$ is a square matrix of size $\left|\Delta_{\tilde{\ell}_s}\right|$. Since the interpolation points and ansatz functions in $I_{m,d}$ are obtained by shifting $2^{-m}Td$ units from the interval I_m^0 the vector $\mathbf{a}^{(m,i)}$ is the same for t- and the τ-variable.

Temporal near-field.

Because of the uniform time discretization, the matrices $\mathbf{c}_{\tilde{\ell}_t}^d$, $d \in \{0,1\}$, in (5.24) have the block-Toeplitz structure

$$\mathbf{c}_{\tilde{\ell}_t}^0 = \begin{bmatrix} \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,0)} & & & \\ \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,1)} & \ddots & & \\ \vdots & \ddots & \ddots & \\ \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t-1)} & \cdots & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,1)} & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,0)} \end{bmatrix} \quad \text{and} \quad \mathbf{c}_{\tilde{\ell}_t}^1 = \begin{bmatrix} \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t)} & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t-1)} & \cdots & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,1)} \\ \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t+1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t+1)} \\ \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,2n_t-1)} & \cdots & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t+1)} & \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t,n_t)} \end{bmatrix},$$

where $n_t = \dim V_0^I$ and

$$\left[\mathbf{b}_{\tilde{\ell}_s}^{(\tilde{l}_t,i)}\right]_{k_s,k'_s} = \int_\Gamma \int_\Gamma G_{\tilde{\ell}_t,i}(\|\mathbf{x}-\mathbf{y}\|)\Phi_{\tilde{\ell}_s,k_s}^\Gamma(\mathbf{x})\Phi_{\tilde{\ell}_s,k'_s}^\Gamma(\mathbf{y})\,\mathrm{d}\sigma_\mathbf{y}\,\mathrm{d}\sigma_\mathbf{x}\,.$$ (5.28)

Here, the kernel contains integration with the ansatz functions in time

$$G_{\tilde{\ell}_t,i}(\|\mathbf{r}\|) = \int_0^T \int_0^T G(\|\mathbf{r}\|,t-\tau)\Phi_{\tilde{\ell}_t,0}^I(\tau)\Phi_{\tilde{\ell}_t,i}^I(t)\,\mathrm{d}\tau\,\mathrm{d}t.$$ (5.29)

The kernel can be expressed in closed form. For the case $i = 0$, the kernel has a $\mathcal{O}(1/\|\mathbf{r}\|)$ singularity, for $i = 1$ the singularity is $\mathcal{O}(\|\mathbf{r}\|)$, and for $i \geq 2$ the kernel is smooth. For the singular cases the spatial integration of the coefficients of (5.28) can be computed with generalized Duffy transforms, similar to the those used for elliptic boundary integral operators, see [14].

Define the shift-matrices

$$\mathbf{s}_n^{(i)} = \begin{bmatrix} 0 & & & & & \\ & \ddots & & & & \\ 1 & & \ddots & & & \\ & \ddots & & \ddots & & \\ & & 1 & & 0 & \end{bmatrix},$$

where n indicates the dimension and i the position of the sub-diagonal. Moreover, define

$$\mathbf{S}_{\tilde{\ell}_t}^{(i)} = \begin{cases} \mathbf{s}_{n_t 2^{\tilde{\ell}_t}}^{(i)}, & 0 \leq i \leq n_t - 1, \\ \mathbf{H}_{\tilde{\ell}_t}^1 \otimes \mathbf{s}_{n_t}^{(i-n_t)}, & n_t \leq i \leq 2n_t - 1. \end{cases}$$

Then, the near-field in (5.24) can be written as

$$\sum_{d \in \{0,1\}} \mathbf{H}^d_{\tilde{\ell}_t} \otimes \mathbf{c}^d_{\tilde{\ell}_t} = \sum_{i=0}^{2n_t-1} \mathbf{S}^{(i)}_{\tilde{\ell}_t} \otimes \mathbf{b}^{(\tilde{\ell}_t,i)}_{\tilde{\ell}_s}. \tag{5.30}$$

Tensor product form of $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$.

The approximation of $\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'}$ in the form of (5.17) can now be obtained by combining (5.15), (5.24), (5.27) and (5.30). Using the multiplication rules of the Kronecker product, we conclude that

$$\mathbf{V}_{\boldsymbol{\ell},\boldsymbol{\ell}'} \approx \sum_{i=0}^{2n_t-1} \mathbf{A}^{(i)}_{\ell_t,\ell'_t} \otimes \mathbf{B}^{(\tilde{\ell}_t,i)}_{\ell_s,\ell'_s} + \sum_{\substack{m \in \{2,\ldots,\tilde{\ell}_t\} \\ d \in \{2,3\} \\ i,i' \in \{0,\ldots,p_t\}}} \mathbf{A}^{(m,d,i,i')}_{\ell_t,\ell'_t} \otimes \mathbf{B}^{(m,d,i,i')}_{\ell_s,\ell'_s}, \tag{5.31}$$

where

$$\mathbf{A}^{(i)}_{\ell_t,\ell'_t} = \mathbf{I}^{\ell_t}_{\tilde{\ell}_t} \mathbf{S}^{(i)}_{\tilde{\ell}_t} \mathbf{I}^{\tilde{\ell}_t}_{\ell'_t},$$

$$\mathbf{A}^{(m,d,i,i')}_{\ell_t,\ell'_t} = \mathbf{I}^{\ell_t}_{\tilde{\ell}_t} \left(\mathbf{H}^d_m \otimes \mathbf{a}^{(i)}_m \left(\mathbf{a}^{(i)}_m \right)^T \right) \mathbf{I}^{\tilde{\ell}_t}_{\ell'_t},$$

$$\mathbf{B}^{(\tilde{\ell}_t,i)}_{\ell_s,\ell'_s} = \mathbf{J}^{\ell_s}_{\tilde{\ell}_s} \mathbf{b}^{(\tilde{\ell}_t,i)}_{\tilde{\ell}_s} \mathbf{J}^{\tilde{\ell}_s}_{\ell'_s},$$

$$\mathbf{B}^{(m,d,i,i')}_{\ell_s,\ell'_s} = \mathbf{J}^{\ell_s}_{\tilde{\ell}_s} \mathbf{b}^{(m,d,i,i')}_{\tilde{\ell}_s} \mathbf{J}^{\tilde{\ell}_s}_{\ell'_s}.$$

Clearly, the matrices $\mathbf{A}^{(i)}_{\ell_t,\ell'_t}$ and $\mathbf{A}^{(m,d,i,i')}_{\ell_t,\ell'_t}$ can be applied with order $2^{\tilde{\ell}_t}$ operations. Note that the order in which the Kronecker product in the second matrix is evaluated is irrelevant, because both factors are square. In the following section, we will show that the matrices $\mathbf{b}^{(\ell_t,i)}_{\tilde{\ell}_s}$ and $\mathbf{b}^{(m,d,i,i')}_{\tilde{\ell}_s}$ can be applied with order $L_s^7 2^{n\tilde{\ell}_s}$ complexity. Then it follows easily that $\mathbf{B}^{(i)}_{\ell_s,\ell'_s}$ and $\mathbf{B}^{(m,d,i,i')}_{\ell_s,\ell'_s}$ can be applied with the same order of operations.

This, together with Lemma 3 and the fact that $p_t \sim L_t$ in (5.26) implies that the matrix $\widehat{\mathbf{V}}_{\mathbf{L}}$ can be applied with $\mathcal{O}\big(L_s^8 L_t^3 \dim(\widehat{U}_{\mathbf{L}})\big)$ cost. Thus the complexity of the algorithm described in Section 5 is log-linear in $\dim(\widehat{U}_{\mathbf{L}})$.

6 Fast evaluation of the Matrices $\mathbf{b}^{(\tilde{\ell}_t,i)}_{\tilde{\ell}_s}$ and $\mathbf{b}^{(m,d,i,i')}_{\tilde{\ell}_s}$

In this section, we show that the spatial matrices $\mathbf{b}^{(\tilde{\ell}_t,i)}_{\tilde{\ell}_s}$ and $\mathbf{b}^{(m,d,i,i')}_{\tilde{\ell}_s}$ are $\mathcal{H}$-matrices and describe an algorithm to compute matrix vector products in $\mathcal{O}\big(L_s^7 2^{2\tilde{\ell}_s}\big)$ complexity. To simplify the discussion we restrict ourselves to the more important case of a two dimensional surface in three-space, that is, $n = 2$ in (2.5). The modifications for the case $n = 1$ are trivial and will result in lower powers of L_s in the complexity estimate.

Since the calculus with $\mathcal{H}$-matrices is well known, see [10,11], we only present a high-level description of the algorithm mainly to set the stage for the ensuing error analysis. There, we will show how the parameters of the algorithm can be selected such that error and complexity bounds can be obtained that are independent of the parameters $\tilde{\ell}_t$, m and d,i,i'.

We first give more detail on how the spatial finite element spaces $V^{\Gamma}_{\ell_s}$ are generated. To that end, assume that the surface Γ is given by a number of parameterizations of the reference triangle $\hat{\sigma} = \{(\hat{x}_1, \hat{x}_2) : 0 \leq \hat{x}_2 \leq \hat{x}_1 \leq 1\}$

$$x_\nu : \hat{\sigma} \to \Gamma_\nu, \quad \nu \in \mathcal{P}(0),$$

where $\mathcal{P}(0)$ is an index set for the initial triangular patches. We assume that the interiors of Γ_ν are disjoint and that common sides of two adjacent Γ_ν's are parametrized in a consistent manner.

The coarsest space V_0^Γ consists of functions whose preimage in $\hat{\sigma}$ is a polynomial. The spaces $V_{\ell_s}^\Gamma$ consist of functions whose preimages are piecewise polynomials on the ℓ_s-th uniform refinement of $\hat{\sigma}$. Every ℓ_s-th level refined triangle parameterizes a triangular patch Γ_ν, $\nu \in \mathcal{P}(\ell_s)$ which in turn generates a sequence of triangularizations of Γ

$$\Gamma = \bigcup_{\nu \in \mathcal{P}(\ell_s)} \Gamma_\nu.$$

The uniform refinement implies a tree structure in the sense that every triangular patch Γ_ν, $\nu \in \mathcal{P}(\ell_s)$ is the union of four triangular patches in level $\ell_s + 1$, denoted as the four children $\mathcal{K}(\nu)$ of ν

$$\Gamma_\nu = \bigcup_{\nu' \in \mathcal{K}(\nu)} \Gamma_{\nu'}.$$

Moreover, every patch ν in level $\ell_s > 0$ has a parent $\pi(\nu)$ in level $\ell_s - 1$.

The neighbors $\mathcal{N}(\nu)$ of a patch $\nu \in \mathcal{P}(\ell_s)$ are given by

$$\mathcal{N}(\nu) = \left\{ \nu' \in \mathcal{P}(\ell_s) : \min_{\substack{\mathbf{x} \in \Gamma_\nu \\ \mathbf{y} \in \Gamma_{\nu'}}} \|\mathbf{x} - \mathbf{y}\| \le S L_s^{\frac{1}{2}} 2^{-\ell_s} \right\}. \tag{6.32}$$

Here, $S > 0$ is a predetermined constant. The factor $L_s^{\frac{1}{2}}$ implies that the neighbor list is expanded as the mesh is refined and is necessary to ensure convergence of the method. We assume that the constants are such that all patches in level zero are neighbors of each other. The interaction list $\mathcal{I}(\nu)$ of a patch $\nu \in \mathcal{P}(\ell_s)$ is the set of patches whose parents are neighbors, but who are not neighbors themselves:

$$\mathcal{I}(\nu) = \left\{ \nu' \in \mathcal{P}(\ell_s) : \pi(\nu') \in \mathcal{N}\big(\pi(\nu)\big) \text{ and } \nu' \notin \mathcal{N}(\nu) \right\}.$$

Because of the uniform subdivision, the number of neighbors and the number of patches in interactions list are $\mathcal{O}(L_s)$.

The definition of neighbors and interaction lists implies the subdivision

$$\Gamma \times \Gamma = \bigcup_{\substack{\nu \in \mathcal{P}(\tilde{\ell}_s) \\ \nu' \in \mathcal{N}(\nu)}} \Gamma_\nu \times \Gamma_{\nu'} \quad \cup \quad \bigcup_{\ell_s=0}^{\tilde{\ell}_s} \bigcup_{\substack{\nu \in \mathcal{P}(\ell_s) \\ \nu' \in \mathcal{I}(\nu)}} \Gamma_\nu \times \Gamma_{\nu'}, \tag{6.33}$$

where the number of terms is $\mathcal{O}\big(L_s 2^{2\tilde{\ell}_s}\big)$.

Let $\mathbf{b}_{\tilde{\ell}_s}$ be one of the spatial matrices $\mathbf{b}_{\tilde{\ell}_s}^{(\ell_t, i)}$ or $\mathbf{b}_{\tilde{\ell}_s}^{(m,d,i,i')}$ and let $G(\cdot)$ denote its kernel. Since we will introduce additional superscipts below, we omit the kernel identifying superscripts for notational convenience. From the subdivision (6.33), we obtain the decomposition

$$\mathbf{b}_{\tilde{\ell}_s} = \mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}} + \sum_{\ell_s=0}^{\tilde{\ell}_s} \mathbf{b}_{\tilde{\ell}_s}^{(\ell_s)}, \tag{6.34}$$

where $k_s, k_s' \in \Delta_{\tilde{\ell}_s}$ and

$$\left[\mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}}\right]_{k_s, k_s'} = \sum_{\substack{\nu \in \mathcal{P}(\tilde{\ell}_s) \\ \nu \in \mathcal{N}(\nu)}} \int_{\Gamma_\nu} \int_{\Gamma_{\nu'}} G(\|\mathbf{x} - \mathbf{y}\|) \varphi_{\tilde{\ell}_s, k_s}^\Gamma(\mathbf{x}) \varphi_{\tilde{\ell}_s, k_s'}^\Gamma(\mathbf{y}) \, \mathrm{d}\sigma_{\mathbf{y}} \, \mathrm{d}\sigma_{\mathbf{x}},$$

$$\left[\mathbf{b}_{\tilde{\ell}_s}^{(\ell_s)}\right]_{k_s, k_s'} = \sum_{\substack{\nu \in \mathcal{P}(\ell_s) \\ \nu' \in \mathcal{I}(\nu)}} \int_{\Gamma_\nu} \int_{\Gamma_{\nu'}} G(\|\mathbf{x} - \mathbf{y}\|) \varphi_{\tilde{\ell}_s, k_s}^\Gamma(\mathbf{x}) \varphi_{\tilde{\ell}_s, k_s'}^\Gamma(\mathbf{y}) \, \mathrm{d}\sigma_{\mathbf{y}} \, \mathrm{d}\sigma_{\mathbf{x}}.$$

Since the number of basis functions in level $\tilde{\ell}_s$ that overlap with a patch in level $\tilde{\ell}_s$ are bounded, the matrix $\mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}}$ has $\mathcal{O}(L_s 2^{2\tilde{\ell}_s})$ nonvanishing entries. Of course, the matrices $\mathbf{b}_{\tilde{\ell}_s}^{(\ell_s)}$ become increasingly populated as the level ℓ_s decreases, but since the integrals are over patches in interaction lists, the kernels are smooth functions. Thus, we can approximate the kernel by a degenerate expansion which will lead to a factorization that can be evaluated with $\mathcal{O}(L_s 2^{2\tilde{\ell}_s})$ complexity.

To that end, we enclose every patch Γ_ν in $\mathcal{P}(\ell_s)$ by an axiparallel cube with sidelength $S_1 2^{-\ell_s}$ and center $\mathbf{x}_\nu$. The constant S_1 is chosen such that the cubes will contain the patch Γ_ν tightly which is possible for all ℓ_s with the same S_1 because of the uniform refinement scheme. Then any point in the enclosing cube has local coordinates in $[-1, 1]^3$, that is,

$$\mathbf{x} = 2^{-\ell_s} S_1 \left(\mathbf{x}_\nu + \frac{1}{2} \hat{\mathbf{x}} \right), \quad \text{where} \quad \hat{\mathbf{x}} \in [-1, 1]^3. \tag{6.35}$$

For two points $\mathbf{x} \in \Gamma_\nu$, $\mathbf{y} \in \Gamma_{\nu'}$, where $\nu \in \mathcal{P}(\ell_s)$ and $\nu' \in \mathcal{I}(\nu)$, the kernel can be expanded into a Chebyshev series in the local coordiates, that is,

$$G(\|\mathbf{x} - \mathbf{y}\|) \approx \sum_{\substack{|\boldsymbol{\alpha}| \leq p_s \\ |\boldsymbol{\beta}| \leq p_s}} E_{\boldsymbol{\alpha}, \boldsymbol{\beta}}^{\nu, \nu'} T_{\boldsymbol{\alpha}}(\hat{\mathbf{x}}) T_{\boldsymbol{\beta}}(\hat{\mathbf{y}}) \tag{6.36}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are multiindices and $T_{\boldsymbol{\alpha}}(\cdot)$ are the Chebyshev polynomials. In section 7 we will show that for suitable expansion order the error can be neglected. Then replacing the kernel by the expansion leads to

$$\left[\mathbf{b}_{\tilde{\ell}_s}^{(\ell_s)} \right]_{k_s, k_s'} \approx \sum_{\substack{\nu \in \mathcal{P}(\ell_s) \\ \nu' \in \mathcal{I}(\nu)}} \sum_{\substack{|\boldsymbol{\alpha}| \leq p_s \\ |\boldsymbol{\beta}| \leq p_s}} E_{\boldsymbol{\alpha}, \boldsymbol{\beta}}^{\nu, \nu'} \int_{\Gamma_\nu} T_{\boldsymbol{\alpha}}(\hat{\mathbf{x}}) \varphi_{\tilde{\ell}_s, k_s}^{\Gamma}(\mathbf{x}) \, d\sigma_{\mathbf{x}} \int_{\Gamma_{\nu'}} T_{\boldsymbol{\beta}}(\hat{\mathbf{y}}) \varphi_{\tilde{\ell}_s, k_s'}^{\Gamma}(\mathbf{y}) \, d\sigma_{\mathbf{y}}. \tag{6.37}$$

In matrix form, this can be expressed as the factorization

$$\mathbf{b}_{\tilde{\ell}_s}^{(\ell_s)} \approx \widetilde{\mathbf{b}}_{\tilde{\ell}_s}^{(\ell_s)} = \left(\mathbf{M}_{\tilde{\ell}_s}^{(\ell_s)} \right)^T \mathbf{E}^{(\ell_s)} \mathbf{M}_{\tilde{\ell}_s}^{(\ell_s)}$$

where the the matrices $\mathbf{M}_{\tilde{\ell}_s}^{(\ell_s)}$ contain the moments, i.e., the integrals in (6.37), and the matrices $\mathbf{E}^{(\ell_s)}$ contain the expansion coefficients $E_{\boldsymbol{\alpha}, \boldsymbol{\beta}}^{\nu, \nu'}$. It is not hard to see that these matrices can be evaluated with $\mathcal{O}(L_s p_s^3 2^{2\tilde{\ell}_s})$ and $\mathcal{O}(L_s p_s^6 2^{2\ell_s})$ complexity.

Finally, we note that all kernels $G(\cdot)$ decay exponentially at infinity. Since interaction lists in the coarser levels contain increasingly distant pairs of patches, it is not necessary to evaluate all terms in the sum (6.34). Instead, we select a minimal level $\bar{\ell}_s$ and evaluate the approximation

$$\mathbf{b}_{\tilde{\ell}_s} \approx \widetilde{\mathbf{b}}_{\tilde{\ell}_s} = \mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}} + \sum_{\ell_s = \bar{\ell}_s}^{\tilde{\ell}_s} \widetilde{\mathbf{b}}_{\tilde{\ell}_s}^{(\ell_s)}. \tag{6.38}$$

In the following section we will show that the choice of parameters

$$p_s \sim L_s \quad \text{and} \quad \bar{\ell}_s = \begin{cases} \frac{\tilde{\ell}_t}{2} & \text{when } \mathbf{b}_{\tilde{\ell}_s} = \mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t, i)}, \\ \frac{m}{2} & \text{when } \mathbf{b}_{\tilde{\ell}_s} = \mathbf{b}_{\tilde{\ell}_s}^{(m, d, i, i')}, \end{cases} \tag{6.39}$$

will be sufficent to ensure that the approximation error does not affect the asymptotic convergence of the discretization error. Thus the complexity of a matrix vector product of $\mathbf{b}_{\tilde{\ell}_s}$ using the approximation (6.38) is $\mathcal{O}(L_s^7 2^{n\tilde{\ell}_s})$.

Note that the introduction of the minimal level $\bar{\ell}_s$ does not reduce the asymptotic cost of the matrix-vector product, but ensures the accuracy of the degenerate kernel expansion (6.36). This will become clear in the following error analysis.

7 Error Analysis

We analyse the errors introduced by the low-rank approximation in (5.26) and the fast evaluation of the spatial matrices in (6.38). In both cases, numerical efficiency is achieved by replacing the heat kernel with its interpolate at the Chebyshev nodes. For an analytic function, the interpolation error exhibits exponential convergence and with the Strang lemma it can be concluded that this approximation results in an exponentially small error of the Galerkin solution. Since the convergence of the discretization method is algebraic, the discretization error dominates the error of the fast method as long as the interpolation order grows linearly with the refinement level. This kind of argument is commonly used in the analysis of fast methods for integral equations, see, e.g. [15] in the context of the heat equation and the space-time fast multipole method. For the sparse grid method of this article it remains to ensure that the interpolations have convergence rates that are independent of the various space time levels. This will be discussed below. Our argument is based on the following approximation result for Chebyshev interpolation, which is well known for single-variate functions, see, e.g. [13, Lemma 6.6], but appears to be hard to find for the multivariate case.

Lemma 4 *If $f : [-1,1]^d \to \mathbb{R}$ has a complex extension which, for some $\mu > 0$, is analytic in every variable within the ellipse*

$$E_\mu := \left\{ z \in \mathbb{C} : z = \cos\left(\theta + i\mu\right), \theta \in [0, 2\pi] \right\},$$

then the multivariate Chebyshev interpolate of degree $N - 1$ satisfies the error estimate

$$|f(\mathbf{x}) - p(\mathbf{x})| \leq C \max_{z_i \in E_\mu} |f(\mathbf{z})| \sinh^{-1}(N\mu).$$

Proof The proof for the one dimensional case ($d = 1$) is well known and is based on the contour integral representation of the interpolate and the remainder

$$p(x) = \frac{1}{2\pi i} \int\limits_{E_\mu} \left(1 - \frac{T_N(x)}{T_N(z)}\right) \frac{f(z)}{z - x}\, dz,$$

$$f(x) - p(x) = \frac{1}{2\pi i} \int\limits_{E_\mu} \frac{T_N(x)}{T_N(z)} \frac{f(z)}{z - x}\, dz,$$

where $T_N(x) = \cos(N \arccos(x))$ is the N-th Chebyshev polynomial. This implies that the contour integral formula for the multivariate interpolate is

$$p(\mathbf{x}) = \frac{1}{(2\pi i)^d} \int\limits_{E_\mu} \cdots \int\limits_{E_\mu} \left(1 - \frac{T_N(x_1)}{T_N(z_1)}\right) \cdots \left(1 - \frac{T_N(x_d)}{T_N(z_d)}\right) \frac{f(\mathbf{z})}{(z_1 - x_1) \ldots (z_d - x_d)}\, dz_d \ldots dz_1.$$

For $x \in [-1,1]$ real, $|T_N(x)| \leq 1$, whereas $|T_N(z)| \geq \sinh(N\mu)$ for $z \in E_\mu$. Multiplying out the T_N-terms in the above integral gives

$$p(\mathbf{x}) = f(\mathbf{x}) - \frac{1}{(2\pi i)^d} \int\limits_{E_\mu} \cdots \int\limits_{E_\mu} \left(\frac{T_N(x_1)}{T_N(z_1)} + \ldots + \frac{T_N(x_d)}{T_N(z_d)}\right) \frac{f(\mathbf{z})}{(z_1 - x_1) \ldots (z_d - x_d)}\, dz_d \ldots dz_1 + lot(\mathbf{x}).$$

Here, the term $f(\mathbf{x})$ follows from Cauchy's integral formula and $lot(\mathbf{x})$ denotes terms of order $\sinh^{-2}(N\mu)$. Estimating the remaining integral in the obvious way gives the assertion. □

The estimate for the interpolation error is a direct application of the above lemma.

Lemma 5 *For $1 < \eta < d + \sqrt{d^2 - 1}$ there is a constant $C_\eta > 0$, independent of r and m, such that the interpolation error in (5.26) is bounded by*

$$E_{p_t}\left(\|\widetilde{\mathbf{r}}\|\right) \leq C_\eta 2^{\frac{3}{2}L_t} \eta^{-p_t}.$$

Proof Recall that in the local coordinates (5.25) the heat kernel is

$$G(t', \tau') := \frac{2^{\frac{3}{2}m}}{T^{\frac{3}{2}}\left(d + \frac{1}{2}(t' - \tau')\right)} \exp\left(-\frac{\|\widetilde{\mathbf{r}}\|^2}{4\left(d + \frac{1}{2}(t' - \tau')\right)}\right).$$

This function satisfies the assumptions of lemma 4 when $\cosh \mu < d$ which is equivalent to $e^\mu < d + \sqrt{d^2 - 1}$. Since $m \leq L_t$ we see that

$$\max_{t', \tau' \in E_\mu} \left|G(t', \tau')\right| \leq C_\mu 2^{\frac{3}{2}L_t}.$$

This bound is independent of $\|\widetilde{\mathbf{r}}\|$ because $\mathrm{Re}\, d + \frac{1}{2}(t' - \tau') > 0$, so the argument to the exponential function has negative real part. We estimate the hyperbolic sine in lemma 4 by $\sinh(p_t \mu) > (e^\mu)^{p_t}/4$, thus for any $\eta < d + \sqrt{d^2 - 1}$ the assertion follows. $\qquad\square$

Remark 4 The estimate in the above lemma makes clear that any exponential rate of convergence can be achieved if $p_t \sim L_t$ and the proportionality constant is sufficiently large.

We now turn to the error of the fast evaluation method in section 6. There are two error sources in this algorithm, namely replacing the kernel by zero in levels $\ell_s < \bar{\ell}_s$, and the Chebyshev approximation in levels $\bar{\ell}_s \leq \ell_s \leq \tilde{\ell}_s$. For points $\mathbf{x} \in \Gamma_\nu$ and $\mathbf{y} \in \Gamma_{\nu'}$ on the patches in the subdivision (6.33) the kernel of the matrix $\mathbf{b}_{\tilde{\ell}_s}$ in (6.38) is given by

$$\widetilde{G}(\mathbf{x}, \mathbf{y}) = \begin{cases} G(\|\mathbf{x} - \mathbf{y}\|), & \nu \in \mathcal{P}(\tilde{\ell}_s),\ \nu' \in \mathcal{N}(\nu), \\ G_{p_s}(\mathbf{x}, \mathbf{y}), & \nu \in \mathcal{P}(\ell_s),\ \nu' \in \mathcal{I}(\nu),\ \bar{\ell}_s \leq \ell_s \leq \tilde{\ell}_s, \\ 0, & \nu \in \mathcal{P}(\ell_s),\ \nu' \in \mathcal{I}(\nu),\ 0 \leq \ell_s < \bar{\ell}_s, \end{cases}$$

where G_{p_s} is the truncated series expansion in (6.36).

Lemma 6 *For p_s and $\bar{\ell}_s$ given by (6.39), there are constants $C > 0$, $\eta > 1$, independent of $\tilde{\ell}_s$, $\tilde{\ell}_t$, m, d, i and i', such that*

$$\left|G(\|\mathbf{x} - \mathbf{y}\|) - \widetilde{G}(\mathbf{x}, \mathbf{y})\right| \leq C\eta^{-L_s}. \tag{7.40}$$

Proof We begin with the far-field truncation for $\mathbf{b}_{\tilde{\ell}_s}^{(m,d,i,i')}$. The kernel of the matrix is

$$G(\|\mathbf{x} - \mathbf{y}\|) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2^{-m}\delta}\right),$$

where $\delta = d + \frac{1}{2}(\omega^{(i)} - \omega^{(i')})$ is in the interval $[1, 4]$. For the points $\mathbf{x} \in \Gamma_\nu$ and $\mathbf{y} \in \Gamma_{\nu'}$, $\nu \in \mathcal{P}(\ell_s)$ and $\nu' \in \mathcal{I}(\nu)$ the distance satisfies $\|\mathbf{x} - \mathbf{y}\| \geq S L_s^{\frac{1}{2}} 2^{-\ell_s}$ because ν' and ν are not neighbors. Thus the estimate

$$G(\|\mathbf{x} - \mathbf{y}\|) \leq \exp\left(-2^{m-2\ell_s} L_s \frac{S^2}{\delta}\right) \leq \exp\left(-L_s \frac{S^2}{\delta}\right)$$

holds when $\ell_s < \frac{m}{2}$ and the bound in (7.40) is established for $\eta = \exp\left(S^2\right)$.

We now consider the far-field truncation error for the matrices $\mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t, i)}$. A simple change of variables in (5.29) shows that the kernel is

$$G(\|\mathbf{x} - \mathbf{y}\|) = h_t^{\frac{1}{2}} g_i \left(\frac{\|\mathbf{x} - \mathbf{y}\|}{\sqrt{h_t}} \right).$$

where g_i is given by

$$g_i(r) = \begin{cases} \displaystyle\int_0^1 \int_0^t \frac{1}{(t - \tau)^{\frac{3}{2}}} \exp\left(-\frac{r^2}{4(t - \tau)} \right) \mathrm{d}\tau\, \mathrm{d}t, & i = 0, \\ \displaystyle\int_0^1 \int_0^1 \frac{1}{(d + t - \tau)^{\frac{3}{2}}} \exp\left(-\frac{r^2}{4(i + t - \tau)} \right) \mathrm{d}\tau\, \mathrm{d}t, & 0 < i < \dim(V_0^I). \end{cases}$$

From their closed expression in [14] it can be seen that they satisfy the estmate $g_i(r) \leq \frac{C}{r^2} \exp\left(-\frac{r^2}{i+1} \right)$. As before, it follows for $\mathbf{x} \in \Gamma_\nu$ and $\mathbf{y} \in \Gamma_{\nu'}$, where $\nu \in \mathcal{P}(\ell_s)$, $\nu' \in \mathcal{I}(\nu)$ and $\tilde{\ell}_s < \frac{\tilde{\ell}_t}{2}$, that $G(\|\mathbf{x} - \mathbf{y}\|) \leq C \exp\left(-L_s \frac{S^2}{i+1} \right)$ holds. This is the bound in (7.40).

We now turn to the Chebyshev approximation error of the matrices $\mathbf{b}_{\tilde{\ell}_s}^{(m, d, i, i')}$. For $\mathbf{x} \in \Gamma_\nu$, $\mathbf{y} \in \Gamma_{\nu'}$, $\nu \in \mathcal{P}(\ell_s)$, $\nu' \in \mathcal{I}(\nu)$, the kernel is

$$G(\|\mathbf{x} - \mathbf{y}\|) = \exp\left(-\frac{S_1^2 2^{m - 2\ell_s}}{d + \frac{1}{2}(\omega_i - \omega_{i'})} \|\mathbf{r}_{\nu,\nu'} + \hat{\mathbf{x}} - \hat{\mathbf{y}}\|^2 \right),$$

where $\mathbf{r}_{\nu,\nu'} = \mathbf{x}_\nu - \mathbf{x}_{\nu'}$ and $\hat{\mathbf{x}}, \hat{\mathbf{y}} \in [-1, 1]^3$ are the local coordinates defined in (6.35). The scaling of the enclosing cubes and the definition of the neighbors and interaction lists implies that $\|\mathbf{r}_{\nu,\nu'}\| > 3$ if S and L_s are sufficiently large. In view of lemma 4 the function $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \to \|\mathbf{r}_{\nu,\nu'} + \hat{\mathbf{x}} - \hat{\mathbf{y}}\|^2$ is strictly positive in $[-1, 1]^6$ and, for some $\mu > 0$ can be extended to E_μ^6 such that real part of the image remains positive. It follows that the interpolation error decays exponentially at a rate that can be bounded independently of all of the paramenters m, ℓ_s, i, i' and d. If we let $p_s \sim L_s$, we obtain bound (7.40).

It remains to estimate the truncation error in $\mathbf{b}_{\tilde{\ell}_s}^{(\tilde{\ell}_t, d)}$. The argument is based on a similar scaling. In local coordinates, the kernel is

$$G(\|\mathbf{x} - \mathbf{y}\|) = \sqrt{T} 2^{\frac{\tilde{\ell}_t}{2}} g_d \left(\frac{\|\mathbf{x} - \mathbf{y}\|}{\sqrt{T} 2^{\frac{\tilde{\ell}_t}{2}}} \right) = \sqrt{T} 2^{\frac{\tilde{\ell}_t}{2}} g_d \left(2^{\frac{\tilde{\ell}_t}{2} - \ell_s} \frac{S_1}{\sqrt{T}} \|\mathbf{r}_{\nu,\nu'} + \hat{\mathbf{x}} - \hat{\mathbf{y}}\| \right)$$

The above definition implies that the functions $r \to g_i(r)$ are analytic and uniformly bounded for $\mathrm{Re}\{r\} \geq r_0 > 0$. As before, there is $\mu > 0$ such that the function $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \to \|\mathbf{r}_{\nu,\nu'} + \hat{\mathbf{x}} - \hat{\mathbf{y}}\|^2$ has positive real part in E_μ^6. However, to get uniform bounds of G in E_μ^6 the factor $2^{\frac{\tilde{\ell}_t}{2} - \ell_s}$ must be bounded away from zero. Fortunately, this is the case, because in the sum (6.39) we have $\ell_s \geq \frac{1}{2} \tilde{\ell}_t$. This establishes (7.40) for $p_s \sim L_s + L_t$. $\qquad\square$

8 A Numerical Example

To illustrate the theory presented in this work, we discuss numerical results obtained with an implementation of the method. We solve the indirect integral formulation (2.7) where Γ is the unit sphere and $I = [0, 1]$. The right hand side $f(\mathbf{x}, t)$ is chosen such that the solution is given by $g(\mathbf{x}, t) = t^2(3x_3^2 - 1)$. The spaces $V_{\ell_s}^\Gamma$ are the continuous, piecewise linear functions (i.e., $d_s = 2$), subject to a triangulation of the sphere. The coarsest triangulation is obtained by radial projection of the tetrahedron onto the sphere. The spaces $V_{\ell_t}^I$ are the piecewise constants (i.e., $d_t = 1$), subject

L_s	L_t	$\dim U_{\boldsymbol{L}}$	fac	$\dim \widehat{U}_{\boldsymbol{L}}$	fac	error	fac
1	2	2.00e+2		1.10e+2		3.77e−1	
2	4	2.72e+3	13.6	5.60e+2	5.09	2.87e−1	0.762
3	6	4.16e+4	15.2	2.72e+3	4.86	6.96e−2	0.242
4	8	6.58e+5	15.8	1.28e+5	4.71	1.82e−2	0.261
5	10	1.05e+7	16.0	5.89e+5	4.60	4.81e−3	0.264
6	12	1.68e+8	16.0	2.66e+6	4.52	1.38e−3	0.286

Table 8.1 Numerical results obtained with sparse grid implementation.

to a uniform subdivision of the unit interval, where initial space has five intervals. The relationship between the finest spatial and temporal resolution is $L_t = 2L_s$.

In Section 6 we have described how matrix vector products with the spatial matrices in (6.34) can be evaluated efficiently using $\mathcal{H}$-matrix calculus. For a fully discrete algorithm, the coefficients of the matices $\mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}}$ must be computed by numerical quadrature. Since the kernels have in the worst case a $\mathcal{O}\left(\frac{1}{r}\right)$-singularity, one can use the singularity removing transformations of [12] combined with Gauss quadrature. However, for coarse spatial and fine time scales the kernels can become very peaked and an additional space refinement is necessary to ensure rapid convergence of the Gauss rules. In this process, one can exploit that computations for fixed values of $\tilde{\ell}_t$ and m can be re-used for different values of $\tilde{\ell}_s$. We do not describe this algorithm in detail and only note that the numerical quadrature introduces additional logarithmic factors in the overall complexity estimate of the method.

Table 8.1 displays the dimensions of the full and sparse spaces as well as the L_2-error $\|g-\widehat{g}_{\boldsymbol{L}}\|_{L_2(\Gamma \times I)}$ of the solution. For comparison, we also include results obtained with the full grid method with no fast evaluation techniques in table 8.3.

As apparent from table 8.1 the convergence order is not $\mathcal{O}(2^{-2L_s})$ as in case of the full method. This can be explained as follows. We have $L_s d_s = L_t d_t$, so that in view of Lemma 2 the convergence rate with respect to the energy norm is

$$\|g-\widehat{g}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2},-\frac{1}{4}}(\Gamma \times I)} \lesssim L_s 2^{-\frac{L_s L_t}{4L_s+2L_t}} 2^{-L_s d_s} \|g\|_{\mathcal{H}_{\mathrm{mix}}^{d_s,d_t}(\Gamma \times I)}.$$

Hence, inserting the L^2-orthogonal projection $\widehat{\Pi}_{\boldsymbol{L}}$ onto the space $\widehat{U}_{\boldsymbol{L}}$, we find by the inverse inequality

$$\begin{aligned}
\|g-\widehat{g}_{\boldsymbol{L}}\|_{L_2(\Gamma \times I)} &\le \|(I-\widehat{\Pi}_{\boldsymbol{L}})g\|_{L_2(\Gamma \times I)} + \|\widehat{\Pi}_{\boldsymbol{L}} g - \widehat{g}_{\boldsymbol{L}}\|_{L_2(\Gamma \times I)} \\
&\lesssim \sqrt{L_s} 2^{-L_s d_s} \|g\|_{\mathcal{H}_{\mathrm{mix}}^{d_s,d_t}(\Gamma \times I)} + (2^{L_s/2}+2^{L_t/4})\|\widehat{\Pi}_{\boldsymbol{L}} g - \widehat{g}_{\boldsymbol{L}}\|_{\mathcal{H}^{-\frac{1}{2},-\frac{1}{4}}(\Gamma \times I)} \\
&\lesssim L_s(2^{L_s/2}+2^{L_t/4})2^{-\frac{L_s L_t}{4L_s+2L_t}} 2^{-L_s d_s} \|g\|_{\mathcal{H}_{\mathrm{mix}}^{d_s,d_t}(\Gamma \times I)}.
\end{aligned}$$

If we insert $d_s = 2$, $d_t = 1$, and $2L_s = L_t$, then we obtain

$$\|g-\widehat{g}_{\boldsymbol{L}}\|_{L_2(\Gamma \times I)} \lesssim L_s 2^{-L_s(d_s-1/4)} \|g\|_{\mathcal{H}_{\mathrm{mix}}^{d_s,d_t}(\Gamma \times I)} \sim L_s 2^{-\frac{7}{4}L_s} \|g\|_{\mathcal{H}_{\mathrm{mix}}^{2,1}(\Gamma \times I)}.$$

In Table 8.1, it can be seen that the error indeed closely reproduces the $\mathcal{O}(L_s 2^{-\frac{7}{4}L_s})$ convergence. Also, the dimensions of the sparse tensor product spaces $\dim \widehat{U}_{\boldsymbol{L}}$ reproduce the $\mathcal{O}(L_s 2^{2L_s})$ estimate of Lemma 1 well. Note that for the finer meshes the dimensions of the sparse spaces are dramatically smaller than the full tensor product spaces.

Table 8.2 displays complexity results with our implementation. Our code precomputes the matrices $\mathbf{b}_{\tilde{\ell}_s}^{\mathrm{near}}$ in (6.34) and the coefficients $E_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{\nu,\nu'}$ in (6.36) and store them in memory. We have parallelized this aspect in OpenMP using 16 threads and the timings are reported as setup time. The major cost of the iterative solver is in the computation of the matrix vector product. This aspect of the code is

L_s	setup(s)	fac	apply(s)	fac	coeffs	fac
1			6.00e−4		4.68e+3	
2	3.00e+0		7.99e−3	13.3	1.27e+5	27.1
3	3.60e+1	12.0	1.69e−1	21.2	2.57e+6	20.2
4	5.57e+2	15.5	3.56e+0	21.0	4.61e+7	17.9
5	1.19e+4	21.3	7.81e+1	21.9	8.20e+8	17.8
6	1.48e+5	12.4	1.24e+3	15.9	9.24e+9	11.3

Table 8.2 Timings in seconds and number of stored coefficients for the sparse grid method.

L_s	error	fac	setup(s)	fac	apply(s)	fac	coeffs	fac
1	2.24e−1				1.60e−5		2.00e+3	25.0
2	2.86e−1	1.28	2.00e+0		3.20e−3	200	9.25e+4	46.2
3	5.48e−2	0.191	1.00e+2	50	2.09e+0	654	5.41e+6	58.4
4	1.24e−2	0.226	6.30e+3	63	4.83e+2	231	3.38e+8	62.5

Table 8.3 L^2-errors and timings for the full method.

run in serial on a single thread and reported as the apply time. The table also displays the number of stored matrix- and translation coefficients.

From the shown data it is apparent that in most cases the magnification factors obtained are significantly smaller than 16. This shows that the sparse grid method has an improved complexity over any method that is based on the full grid discretization, even if that method has optimal complexity in $\dim U_{\boldsymbol{L}}$, such as the methods of [19] and [15]. The results displayed in table 8.3 is for the full method, where the theoretical factors are 64 for the setup and 256 for the solution.

For the smaller values of L_s the observed memory allocation and cpu-times for the sparse grid method grow much faster than the theroretical $\dim \widehat{U}_{\boldsymbol{L}}$ rate. The reason is that most of the computing resources are consumed by the many $\mathbf{b}_{\tilde{\ell}_s}$-matrices in (6.34). Since these matrices are relatively small for the values of $\tilde{\ell}_s$ that we computed, the $\mathcal{H}$-format does not yield high compression rates, because the asymptotic rates of Section 6 have not been reached. Only for the largest number of refinements the complexity curves level out and suggest that a nearly $\dim \widehat{U}_{\boldsymbol{L}}$ complexity is indeed possible.

References

1. R. Balder and C. Zenger. The solution of multidimensional real Helmholtz equations on sparse grids. *SIAM J. Sci. Comput.* **17**(3) (1996) 631–646.
2. H.J. Bungartz. A multigrid algorithm for higher order finite elements on sparse grids. *ETNA Electron. Trans. Numer. Anal.* **6** (1997) 63–77.
3. H.J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.* **13** (2004) 147–269.
4. A. Chernov and D. Pham. Sparse spectral BEM for elliptic problems with random input data on a spheroid. *Adv. Comput. Math.* **41**(1) (2015) 77–104.
5. A. Chernov and C. Schwab. Sparse space-time Galerkin BEM for the nonstationary heat equation. *ZAMM Z. Angew. Math. Mech.* **93** (2013) 403–413.
6. M. Costabel. Boundary integral operators for the heat equation. *Integral Equations Operator Theory* **13** (1990) 498–552.
7. M. Griebel and H. Harbrecht. On the construction of sparse tensor product spaces. *Math. Comput.* **82** (2013) 975–994.
8. M. Griebel and D. Oeltz. A sparse grid space-time discretization scheme for parabolic problems. *Computing* **81** (2007) 1–34.
9. M. Griebel, P. Oswald, and T. Schiekofer. Sparse grids for boundary integral equations. *Numer. Mathematik* **83**(2) (1999) 279–312.
10. W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing* **64** (1999) 89–108.

11. W. Hackbusch. *Hierarchical Matrices. Algorithms and Analysis.* Springer, Berlin-Heidelberg, 2015.

12. S. Erichsen and S. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comp. Meth. Appl. Mech. Engrg.* **157** (1998), 215-224.

13. J.C. Mason and D.C. Handscomb Chebyshev Polynomials Chapman & Hall, Boca Raton, 2003

14. M. Messner, M. Schanz, and J. Tausch. A fast Galerkin method for parabolic space-time boundary integral equations. *J. Comp. Phys.* **258** (2014) 15–30.

15. M. Messner, M. Schanz and J. Tausch. An efficient Galerkin boundary element method for the transient heat equation. *SIAM J. Sci. Comput.* 37-3 (2015), pp. A1554-A1576.

16. P. J. Noon. The Single Layer Heat Potential and Galerkin Boundary Element Methods for the Heat Equation. PhD thesis, University of Maryland, 1988.

17. N. Reich. Wavelet compression of anisotropic integrodifferential operators on sparse tensor product spaces. *M2AN Math. Model. Numer. Anal.* **44**(1) (2010) 33–73.

18. R. Stevenson and C. Schwab. Space-time adaptive wavelet methods for parabolic evolution problems. *Math. Comput.* **78** (2009) 1293–1318.

19. J. Tausch. A fast method for solving the heat equation by layer potentials. *J. Comp. Phys.* **224** (2007) 956–969.

20. J. Tausch. Fast Nyström Methods for Parabolic Boundary Integral Equations. In U. Langer, M. Schanz und W.L. Wendland, editors, *Fast Boundary Element Methods in Engineering and Industrial Applications*, volume 63 of Lecture Notes in Computational Mechanics, pages 185–219, Springer, Berlin-Heidelberg, 2012.

21. C. Winter. Wavelet Galerkin schemes for multidimensional anisotropic integrodifferential operators. *SIAM J. Sci. Comput.* **32**(3) (2010) 1545–1566.

22. A. Zeiser. Fast matrix-vector multiplication in the sparse-grid Galerkin method. *J. Sci. Comput.* **47**(3) (2011) 328–346.

23. C. Zenger. Sparse grids. In *Parallel algorithms for partial differential equations (Kiel, 1990)*, volume 31 of *Notes Numer. Fluid Mech.*, pages 241–251. Vieweg, Braunschweig, 1991.