
A time-adaptive space-time FMM for the heat
equation

R. Watschinger, G. Of

**Berichte aus dem
Institut für Angewandte Mathematik**

Technische Universität Graz

A time-adaptive space-time FMM for the heat
equation

R. Watschinger, G. Of

**Berichte aus dem
Institut für Angewandte Mathematik**

Bericht 2022/6

Technische Universität Graz
Institut für Angewandte Mathematik
Steyrergasse 30
A 8010 Graz

WWW: <http://www.applied.math.tugraz.at>

© Alle Rechte vorbehalten. Nachdruck nur mit Genehmigung des Autors.

A time-adaptive space-time FMM for the heat equation

Raphael Watschinger, Günther Of

November 4, 2022

We present a new time-adaptive FMM for a space-time boundary element method for the heat equation. The method extends the existing parabolic FMM by adding new operations that allow for an efficient treatment of tensor product meshes which are adaptive in time. We analyze the efficiency of the new operations and the approximation quality of the related kernel expansions and present numerical experiments that reveal the benefits of the new method.

1 Introduction

For the solution of initial boundary value problems of the transient heat equation or rather their numerical approximation there exists a variety of different methods. Among them are boundary element methods, see, e.g. [1, 5, 14]. Boundary element methods rely on a reformulation of the initial boundary value problem as an integral equation on the boundary of the space-time domain, which is then discretized. Typically tensor product discretizations with a fixed spatial mesh and uniform time steps are considered. This is advantageous for several aspects. Due to the tensor product structure of the mesh and the causality of the problem, the BEM matrices are lower triangular block matrices. The uniformity in time leads to a block Toeplitz structure. Thus, only one block per time step has to be computed and stored. These properties allow to use an efficient blockwise forward elimination algorithm for the solution of the BEM systems.

However, the restriction to uniform meshes and the described solution procedure limit the range of applications. Firstly, the level of parallelism of the time-sequential forward elimination procedure is limited by the spatial components. This may be a bottleneck with modern hardware. In [7, 25] the authors have presented BEM implementations for the heat equation, which are parallel in space and time. Secondly, the concept prevents adaptivity. A huge number of uniform time steps might be necessary to resolve some local aspects with respect to time. Here a discretization which is adaptive in time can be advantageous. While local temporal aspects can be resolved much better, the total number of time steps may be much lower. This reduces the total number of degrees of freedom and the size of the system of linear equations. However, the block Toeplitz structure is lost for varying time step sizes and more blocks have to be computed and stored. This can efficiently be handled by a distributed parallelization.

A general difficulty when using standard BEM is that the occurring system matrices are dense. Several fast and data-sparse algorithms have been developed to overcome this problem. Typically these methods provide an efficient compression of the matrices and a fast realization of matrix vector products, which leads to efficient solvers. Some examples for the heat equation

are algorithms based on Fourier series and FFT [9, 10], the parabolic fast multipole method (pFMM) [18, 19, 22, 23], or a fast sparse grid method [12]. In this work we present a time-adaptive FMM which stems from the pFMM. In its original form the pFMM combines the standard ideas and principles of FMM algorithms, like clustering the computational domain and approximating interactions in well-separated clusters by suitable truncated series expansions, with an efficient forward elimination algorithm.

Just like most other fast BEM heat solvers, the pFMM was developed for uniform time steps. While it can be applied for meshes with non-uniform time steps, the overall compression and performance of the pFMM will be worse in this setting than for a uniform time discretization with the same number of time intervals. In fact, there might be many blocks which are not compressed. Here we will present a new variant of the parabolic FMM which performs well for widely varying time step sizes. Similarly to the adaptive version [3, 4, 20] of the FMM for the Laplace kernel, we will introduce certain one-sided expansions with respect to time and related FMM operations. These operations are in the spirit of S2L (source to local, also called Q2L) and M2T (moment to target, also called M2P) operations, which turned out to be efficient in cases where only a few elements are contained in a cluster. The new operations enable the compression of additional matrix blocks and improve the performance of the method. The new method is described without the original forward elimination scheme to allow for parallelism in space and time as in [25].

The paper is organized as follows: In Section 2 we briefly describe the considered boundary element method for the heat equation. In Section 3 we discuss the kernel expansions of the time-adaptive FMM. Besides the original expansion used in the pFMM, which consists of a two-sided interpolation in time and an additional truncated Chebyshev expansion in space, we introduce new temporally one-sided expansions for a more efficient algorithm. We analyze the approximation errors of the different expansions and point out the advantages of the new temporally one-sided space-time expansions in adaptive situations. In Section 4 we present the new time-adaptive space-time FMM for the heat equation. It is based on the original FMM algorithm [22] for the heat equation and enhanced by new operations. To enable these new operations we modify the generation of the cluster tree in Section 4.1 by introducing additional clusters and identify the related matrix blocks in the hierarchical structure in Section 4.2. The original and new FMM operations are introduced in Sections 4.3 and 4.4, and the new time-adaptive space-time FMM algorithm is presented in Section 4.5. In Sections 4.6 and 4.7 we comment on the runtime complexity and storage requirements of its operations. Finally, we present two numerical examples which are well-suited for adaptive temporal meshes and demonstrate the gains of the new time-adaptive FMM for the heat equation.

2 Boundary integral equations and discretization

We consider the initial Dirichlet boundary value problem of the transient heat equation with the heat capacity constant $\alpha > 0$ for a bounded Lipschitz domain $\Omega \subset \mathbb{R}^3$ with a boundary $\Gamma = \partial\Omega$ as a model problem:

$$\begin{aligned} \frac{\partial}{\partial t} u(\mathbf{x}, t) - \alpha \Delta u(\mathbf{x}, t) &= 0 && \text{for } (\mathbf{x}, t) \in \Omega \times (0, T), \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) && \text{for } \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t) && \text{for } (\mathbf{x}, t) \in \Sigma := \Gamma \times (0, T). \end{aligned} \tag{1}$$

Detailed descriptions on how to solve such a problem by using boundary integral equations and related boundary element methods are provided in [5, 6]. Common approaches (direct via the

first boundary integral equation or indirect) require the solution of a boundary integral equation of the form

$$Vq = f \quad \text{on } \Sigma. \quad (2)$$

Here q is the unknown quantity of interest, and V is the single layer boundary integral operator that has the integral representation

$$Vq(\mathbf{x}, t) = \int_0^t \int_{\Gamma} G_{\alpha}(\mathbf{x} - \mathbf{y}, t - \tau) q(\mathbf{y}, \tau) d\mathbf{s}_y d\tau \quad \text{for } (\mathbf{x}, t) \in \Sigma$$

with the fundamental solution of the heat equation

$$G_{\alpha}(\mathbf{x} - \mathbf{y}, t - \tau) = \begin{cases} \frac{1}{(4\pi\alpha(t - \tau))^{3/2}} \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{4\alpha(t - \tau)}\right) & \text{for } \tau < t, \\ 0 & \text{for } \tau > t. \end{cases} \quad (3)$$

The right-hand side f in (2) depends on the chosen approach. In case of a direct approach

$$f = \left(\frac{1}{2}\text{Id} + K\right)g - M_0 u_0, \quad (4)$$

where K is the double layer boundary integral operator, see e.g. [6, Section 4.6], and M_0 is the trace of the initial potential, see [6, Section 4.2]. In case of an indirect approach $f = g - M_0 u_0$.

For the numerical approximation of (2) we consider a Galerkin method. As a first step we discretize the lateral boundary Σ by a space-time tensor product mesh $\Sigma_h = \Gamma_h \otimes \mathcal{I}_h$. Here, $\Gamma_h = \{\gamma_{j_x}\}_{j_x=1}^{E_x}$ is an admissible triangular surface mesh of Γ and \mathcal{I}_h is a potentially adaptive mesh for the time interval $(0, T)$ with grid points t_{j_t} , $j_t = 0, 1, \dots, E_t$, sorted in ascending order such that the j_t^{th} temporal element is (t_{j_t-1}, t_{j_t}) . The elements in the space-time mesh Σ_h are tensor products of the form $\sigma_{j_t, j_x} = \gamma_{j_x} \times (t_{j_t-1}, t_{j_t})$.

We want to find a piecewise constant approximation

$$q_h(\mathbf{y}, \tau) = \sum_{j_t=1}^{E_t} \sum_{j_x=1}^{E_x} q_{j_t, j_x} \varphi_{j_t, j_x}^{0,0}(\mathbf{y}, \tau)$$

of q in (2) with basis functions $\varphi_{j_t, j_x}^{0,0}$ which are one on a space-time boundary element σ_{j_t, j_x} and zero otherwise. The system of linear equations of the considered Galerkin method for this purpose is

$$\mathbf{V}_h \mathbf{q} = \mathbf{f}, \quad (5)$$

where \mathbf{q} is the vector of the coefficients q_{j_t, j_x} and \mathbf{f} is the corresponding vector of the right-hand side. We assume that the coefficients are ordered by the temporal index j_t first and then by the spatial index j_x . The entries of the Galerkin matrix \mathbf{V}_h are given by

$$\mathbf{V}_h[(k_t - 1)E_x + k_x, (j_t - 1)E_x + j_x] = \int_{t_{k_t-1}}^{t_{k_t}} \int_{\gamma_{k_x}} \int_{t_{j_t-1}}^{t_{j_t}} \int_{\gamma_{j_x}} G_{\alpha}(\mathbf{x} - \mathbf{y}, t - \tau) d\mathbf{s}_y d\tau d\mathbf{s}_x dt$$

for $k_x, j_x = 1, \dots, E_x$ and $k_t, j_t = 1, \dots, E_t$. Due to the causality of the heat kernel this matrix has lower triangular block structure. More details on the discretization and computation of the matrix entries of \mathbf{V}_h are given in [26].

Note that system (5) is huge and that the matrix \mathbf{V}_h is dense except for its lower triangular block structure. Thus, a standard BEM is limited to small problems. In general a data-sparse method such as the FMM is necessary to solve large-scale problems. In the following sections we will introduce a new variant of the FMM which is well suited for the treatment of meshes that are adaptive in time.

3 Standard and temporally one-sided kernel expansions

The FMM presented in this paper is based on suitable separable approximations of the heat kernel (3) that combine an interpolation in time with a truncated Chebyshev expansion in space. In addition to the expansion described in [17, 23] we consider two new temporally one-sided expansions. All three expansions are introduced and analyzed in the following.

We start by fixing the notation. $T_m(x) = \cos(m \arccos(x))$ are the Chebyshev polynomials of order m on $[-1, 1]$. The Chebyshev nodes $\{\xi_k^{(m)}\}_{k=0}^m$ of order $m + 1$ on $[-1, 1]$ are the roots of T_{m+1} . For an interval $I = [a, b]$ we define the transformed Chebyshev nodes $\xi_{I,k}^{(m)} := \varphi_I(\xi_k^{(m)})$ with the affine map $\varphi_I(x) := (a(1-x) + b(1+x))/2$ from $[-1, 1]$ to I . Let $\mathcal{P}_m(I)$ be the space of polynomials on I whose degrees are less than or equal to m . When interpolating a function $f \in C(I)$ by polynomials in $\mathcal{P}_m(I)$ we use the transformed Chebyshev nodes as interpolation points and obtain

$$\mathcal{I}_I^{(m)}[f] = \sum_{k=0}^m f(\xi_{I,k}^{(m)}) L_{I,k},$$

where $\mathcal{I}_I^{(m)} : C(I) \rightarrow \mathcal{P}_m(I)$ denotes the interpolation operator and $\{L_{I,k}\}_{k=0}^m$ are the Lagrange polynomials

$$L_{I,k}(t) := \prod_{\substack{j=0 \\ j \neq k}}^m \frac{t - \xi_{I,j}^{(m)}}{\xi_{I,j}^{(m)} - \xi_{I,k}^{(m)}}.$$

When considering two intervals I_1 and I_2 and a function $f \in C(I_1 \times I_2)$ we define the one-sided interpolation operator $\mathcal{I}_{\text{dir}, I_1}^{(m)}$ as the operator that interpolates f only in I_1 , i.e.

$$\mathcal{I}_{\text{dir}, I_1}^{(m)} := \mathcal{I}_{I_1}^{(m)} \otimes \text{Id}.$$

Similarly we set $\mathcal{I}_{\text{dir}, I_2}^{(m)} := \text{Id} \otimes \mathcal{I}_{I_2}^{(m)}$. In addition, we consider the two-sided interpolation operator $\mathcal{I}_{I_1 \times I_2}^{(m)} : C(I_1 \times I_2) \rightarrow \mathcal{P}_m(I_1) \otimes \mathcal{P}_m(I_2)$ which interpolates f in both variables and is defined by $\mathcal{I}_{I_1 \times I_2}^{(m)} := \mathcal{I}_{I_1}^{(m)} \otimes \mathcal{I}_{I_2}^{(m)}$. We will use interpolation for the approximation of the heat kernel (3) in the temporal variables t and τ .

In the spatial variables \mathbf{x} and \mathbf{y} we will use a truncated Chebyshev expansion as in [17, 23]. For this purpose we define the transformed Chebyshev polynomials $T_{[a,b],k} := T_k \circ \varphi_{[a,b]}^{-1}$ for an interval $[a, b]$. For a box $[\mathbf{a}, \mathbf{b}] = \prod_{j=1}^3 [a_j, b_j] \subset \mathbb{R}^3$ and a multi-index $\boldsymbol{\kappa} \in \mathbb{N}_0^3$ we define in addition the tensor products

$$T_{[\mathbf{a}, \mathbf{b}], \boldsymbol{\kappa}}(\mathbf{x}) := \prod_{j=1}^3 T_{[a_j, b_j], \kappa_j}(x_j). \quad (6)$$

Let $X_1 = [\mathbf{a}, \mathbf{b}]$ and $X_2 = [\mathbf{c}, \mathbf{d}]$ be two boxes in \mathbb{R}^3 . A function $f \in C(X_1 \times X_2)$ can be represented by the Chebyshev series

$$f(\mathbf{x}, \mathbf{y}) = \sum_{\boldsymbol{\kappa} \in \mathbb{N}_0^3} \sum_{\boldsymbol{\nu} \in \mathbb{N}_0^3} f_{\boldsymbol{\kappa}, \boldsymbol{\nu}} T_{X_1, \boldsymbol{\nu}}(\mathbf{x}) T_{X_2, \boldsymbol{\kappa}}(\mathbf{y}), \quad (7)$$

where $f_{\boldsymbol{\kappa}, \boldsymbol{\nu}}$ are the appropriate expansion coefficients that can be computed by using the orthogonality of the Chebyshev polynomials in a properly weighted inner product as in the 1D case, see

e.g. [21, Section 1.5]. We will specify these coefficients in the later application. By truncating the Chebyshev series we get the approximation

$$\mathcal{S}_{X_1 \times X_2}^{(m)}[f](\mathbf{x}, \mathbf{y}) = \sum_{\kappa: |\kappa| \leq m} \sum_{\nu: |\kappa+\nu| \leq m} f_{\kappa, \nu} T_{X_1, \nu}(\mathbf{x}) T_{X_2, \kappa}(\mathbf{y}).$$

For the approximation of the heat kernel (3) in space and time we distinguish three cases. In all three cases we consider 4D boxes $Z_{\text{tar}} = X_1 \times I_1$ and $Z_{\text{src}} = X_2 \times I_2$, where $X_1 = [\mathbf{c}, \mathbf{c} + 2\tilde{h}_x \mathbf{1}]$ and $X_2 = [\mathbf{d}, \mathbf{d} + 2\tilde{h}_x \mathbf{1}]$ are two fixed cubes in space with edge length $2\tilde{h}_x$. We distinguish three different configurations of the time intervals I_1 and I_2 :

Case 1: I_1 and I_2 are intervals of similar length and $\text{dist}(I_1, I_2) \gtrsim |I_1|$. We interpolate the heat kernel in both temporal variables t and τ and expand it in both spatial variables to get the approximation

$$\mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{I_1 \times I_2}^{(m_t)}[G_\alpha](\mathbf{x}, t, \mathbf{y}, \tau) = \sum_{a, b=0}^{m_t} \sum_{|\kappa+\nu| \leq m_x} E_{\kappa, \nu}^{a, b} T_{X_1, \nu}(\mathbf{x}) T_{X_2, \kappa}(\mathbf{y}) L_{I_1, b}(t) L_{I_2, a}(\tau) \quad (8)$$

for $(\mathbf{x}, t), (\mathbf{y}, \tau) \in Z_{\text{tar}} \times Z_{\text{src}}$. The coefficients $E_{\kappa, \nu}^{a, b}$ correspond to the expansion coefficients in (7) when expanding the function $(\mathbf{x}, \mathbf{y}) \mapsto G_\alpha(\mathbf{x} - \mathbf{y}, \xi_{I_1, b}^{(m_t)} - \xi_{I_2, a}^{(m_t)})$. They are given by

$$E_{\kappa, \nu}^{a, b} := \frac{1}{(4\pi\alpha(\xi_{I_1, b}^{(m_t)} - \xi_{I_2, a}^{(m_t)}))^{3/2}} \prod_{j=1}^3 E_{\kappa_j, \nu_j}(r_j, d_{a, b}), \text{ with} \quad (9a)$$

$$r_j := (c_j - d_j)/\tilde{h}_x, \quad d_{a, b} := 4\alpha(\xi_{I_1, b}^{(m_t)} - \xi_{I_2, a}^{(m_t)})/\tilde{h}_x^2, \quad (9b)$$

$$E_{k, \ell}(r, d_{a, b}) := \frac{\lambda_k \lambda_l}{\pi^2} \int_{-1}^1 \int_{-1}^1 \exp\left(-\frac{|r + \hat{x} - \hat{y}|^2}{d_{a, b}}\right) T_\ell(\hat{x}) T_k(\hat{y}) w(\hat{x}) w(\hat{y}) d\hat{x} d\hat{y}, \quad (9c)$$

where $\lambda_0 = 1$, $\lambda_k = 2$ for all $k > 0$ and $w(x) := (1-x)^{-1/2}$, cf. [23, Section 5.3, p. 209]. Note that there is no closed formula known for the evaluation of the integrals in (9c). Therefore, we approximate them by a Gauß-Chebyshev quadrature rule in the application as proposed in [24, p. 3563] which yields

$$E_{k, \ell}(r, d_{a, b}) \approx \frac{\lambda_k \lambda_l}{(m_x + 1)^2} \sum_{n, m=0}^{m_x} \exp\left(-\frac{|r + \xi_n^{(m_x)} - \xi_m^{(m_x)}|^2}{d_{a, b}}\right) T_\ell(\xi_n^{(m_x)}) T_k(\xi_m^{(m_x)}).$$

Case 2: The source interval I_2 is smaller than the target interval I_1 and $\text{dist}(I_1, I_2) \gtrsim |I_2|$. In this case we interpolate the heat kernel only in the variable $\tau \in I_2$, but still expand it in both spatial variables to get

$$\mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{\text{dir}, I_2}^{(m_t)}[G_\alpha](\mathbf{x}, t, \mathbf{y}, \tau) = \sum_{a=0}^{m_t} \sum_{|\kappa+\nu| \leq m_x} E_{\kappa, \nu}^a(t) T_{X_1, \nu}(\mathbf{x}) T_{X_2, \kappa}(\mathbf{y}) L_{I_2, a}(\tau) \quad (10)$$

for $(\mathbf{x}, t), (\mathbf{y}, \tau) \in Z_{\text{tar}} \times Z_{\text{src}}$. The expansion coefficients $E_{\kappa, \nu}^a(t)$ that depend on $t \in I_1$ are given by

$$E_{\kappa, \nu}^a(t) = \frac{1}{(4\pi\alpha(t - \xi_{I_2, a}^{(m_t)}))^{3/2}} \prod_{j=1}^3 E_{\kappa_j, \nu_j}(r_j, 4\alpha(t - \xi_{I_2, a}^{(m_t)})/\tilde{h}_x^2),$$

where r_j and E_{κ_j, ν_j} are the same as in (9). The superscript of $E_{\kappa, \nu}^a$ indicates that the corresponding approximation involves an interpolation in the source interval, but does not involve an interpolation in the target interval.

Case 3: The target interval I_1 is smaller than the source interval I_2 and $\text{dist}(I_1, I_2) \gtrsim |I_1|$. Then we interpolate the heat kernel in $t \in I_1$ and expand it in both spatial variables to get

$$\begin{aligned} \mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{\text{dir}, I_1}^{(m_t)} [G_\alpha](\mathbf{x}, t, \mathbf{y}, \tau) &= \sum_{b=0}^{m_t} \sum_{|\kappa + \nu| \leq m_x} E_{\kappa, \nu}^b(\tau) T_{X_1, \nu}(\mathbf{x}) T_{X_2, \kappa}(\mathbf{y}) L_{I_1, b}(t), \\ E_{\kappa, \nu}^b(\tau) &= \frac{1}{(4\pi\alpha(\xi_{I_1, b}^{(m_t)} - \tau))^{3/2}} \prod_{j=1}^3 E_{\kappa_j, \nu_j}(r_j, 4\alpha(\xi_{I_1, b}^{(m_t)} - \tau)/\tilde{h}_x^2) \end{aligned} \quad (11)$$

for $(\mathbf{x}, t), (\mathbf{y}, \tau) \in Z_{\text{tar}} \times Z_{\text{src}}$, where the expansion coefficients $E_{\kappa, \nu}^b$ depend on $\tau \in I_2$.

In the following subsections we investigate the one- and two-sided interpolation error in time and the Chebyshev series truncation error in space for the heat kernel. These results will reveal how the edge length of the spatial boxes X_1 and X_2 and the temporal configuration of I_1 and I_2 influence the approximation error in the three cases described above. We will use this knowledge in the construction of the new time-adaptive FMM.

3.1 Analysis of the interpolation in time

The approximation error of an interpolation of the heat kernel $((\mathbf{x}, t), (\mathbf{y}, \tau)) \mapsto G_\alpha(\mathbf{x} - \mathbf{y}, t - \tau)$ in two separated time intervals decays exponentially with increasing interpolation degree, see e.g. [22, Section 3.1] or [17, p. 73]. We formulate this result in Theorem 1. In Theorem 2 we consider the case of two separated time intervals of different lengths and investigate the error of an interpolation in the shorter time interval only. We will compare the resulting estimates and use them to motivate the one-sided temporal expansions in cases 2 and 3 above.

To simplify the notation we fix the heat capacity constant α and the spatial points \mathbf{x} and \mathbf{y} , and set

$$g(t_1, t_2) := G_\alpha(\mathbf{x} - \mathbf{y}, t_1 - t_2) = \frac{1}{(4\pi\alpha(t_1 - t_2))^{3/2}} \exp\left(-\frac{r^2}{4\alpha(t_1 - t_2)}\right) \quad (12)$$

for all $t_1 > t_2$ and $r^2 = |\mathbf{x} - \mathbf{y}|^2$. The following property of g , which is often denoted as asymptotic smoothness (see e.g. [11, Definition 4.14]), will play a central role in the proofs of the interpolation error estimates.

Proposition 1 (cf. [17, p. 73]). *Let g be the kernel defined in (12). There exists a constant c_{as} such that*

$$|\partial_{t_1}^k \partial_{t_2}^\ell g(t_1, t_2)| \leq \frac{c_{\text{as}} (k + \ell)!(k + \ell + 1)^{3/2}}{\alpha^{3/2} (t_1 - t_2)^{k+\ell+3/2}}, \quad \forall k, \ell \in \mathbb{N}_0 \text{ and all } t_1 > t_2. \quad (13)$$

Proof. The proof follows the lines of [17, proof of Lemma 4.1] where an analogous result is shown for a similar function. Since g depends only on the difference of the variables t_1 and t_2 it suffices to bound $|\partial_{t_1}^n g(t_1, 0)|$ for all $t_1 > 0$ to show (13). For $n = 0$ the estimate in (13) is clearly satisfied for all constants $c_{\text{as}} \geq 1/(4\pi)^{3/2}$. For the case $n \geq 1$ we consider the function $z \mapsto \tilde{g}(z) := g(z, 0)$, which is holomorphic on $\mathbb{C} \setminus \mathbb{R}_{\leq 0}$. Hence, we can use Cauchy's integral formula to get

$$\frac{d^n}{dz^n} \tilde{g}(t_1) = \frac{n!}{2\pi i} \int_{\partial B(t_1, \delta)} \frac{\tilde{g}(\zeta)}{(\zeta - t_1)^{(n+1)}} d\zeta$$

for all $t_1 \in \mathbb{R}_{>0}$ and $\delta < t_1$, where $B(t_1, \delta)$ is the disc with center t_1 and radius δ in \mathbb{C} . For all $\zeta \in \partial B(t_1, \delta)$ there holds $|\zeta| \geq \Re(\zeta) > (t_1 - \delta)$ and $\Re(-r^2/(4\alpha\zeta)) < 0$, so it follows that

$$|\tilde{g}(\zeta)| = \frac{1}{(4\pi\alpha)^{3/2}} \left| \frac{1}{\zeta^{3/2}} \exp\left(-\frac{r^2}{4\alpha\zeta}\right) \right| \leq \frac{1}{(4\pi\alpha)^{3/2}} \frac{1}{(t_1 - \delta)^{3/2}}.$$

Therefore, we can estimate

$$\begin{aligned} \left| \frac{d^n}{dz^n} \tilde{g}(t_1) \right| &\leq \frac{n!}{2\pi} \int_{\partial B(t_1, \delta)} \frac{|\tilde{g}(\zeta)|}{|\zeta - t_1|^{(n+1)}} d\zeta \\ &\leq \frac{n!}{2\pi} \int_{\partial B(t_1, \delta)} \frac{1}{(4\pi\alpha(t_1 - \delta))^{3/2} \delta^{(n+1)}} d\zeta = \frac{n!}{(4\pi\alpha(t_1 - \delta))^{3/2} \delta^n}. \end{aligned}$$

The optimal radius δ^* to minimize this bound is $\delta^* = 2nt_1/(2n + 3)$, for which we get

$$\left| \frac{d^n}{dz^n} \tilde{g}(t_1) \right| \leq \frac{n!}{(4\pi\alpha)^{3/2}} \left(\frac{2n+3}{3} \right)^{3/2} \left(1 + \frac{3}{2n} \right)^n \frac{1}{t_1^{n+3/2}} \leq \frac{e^{3/2}}{(4\pi)^{3/2}} \frac{n!(n+1)^{3/2}}{\alpha^{3/2} t_1^{n+3/2}}.$$

This is the desired estimate for all $n \geq 1$ with $c_{as} = e^{3/2}/(4\pi)^{3/2}$. \square

Before we proceed with the main results of this section, we recall the definition of the Lebesgue constant

$$\Lambda_m := \sup_{u \in C(J) \setminus \{0\}} \frac{\|\mathcal{I}_J^{(m)}[u]\|_{\infty, J}}{\|u\|_{\infty, J}}, \quad (14)$$

i.e. the operator norm of the interpolation operator $\mathcal{I}_J^{(m)} : C(J) \rightarrow \mathcal{P}_m(J)$. Note that Λ_m does not depend on the particular interval J as long as the same interpolation points – up to an affine transformation – are used. In this work we always use transformed Chebyshev nodes as interpolation points. Therefore, we can estimate [21, cf. Theorem 1.2]

$$\Lambda_m \leq \frac{2}{\pi} \log(m+1) + 1. \quad (15)$$

Theorem 1 (Two-sided interpolation error, cf. [17, p. 73]). *Let $\eta_2 > 0$ and $q_2 := 1 + 3/(2\eta_2)$. Let $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$ be two non-empty intervals such that $a_1 > b_2$. Let the admissibility criterion*

$$\eta_2 \text{dist}(I_1, I_2) \geq \max\{|I_1|, |I_2|\} \quad (16)$$

be satisfied. Then there exists a constant $c_2 > 0$ such that

$$\|g - \mathcal{I}_{I_1 \times I_2}^{(m)}[g]\|_{\infty, I_1 \times I_2} \leq \frac{c_2}{(\alpha \text{dist}(I_1, I_2))^{3/2}} q_2^{-(m+1)}. \quad (17)$$

Proof. According to [2, Corollary 4.21] it suffices to show that there exist $C_g \in \mathbb{R}_{\geq 0}$, $\gamma_g \in \mathbb{R}_{>0}$ and $\sigma \in \mathbb{N}$ such that

$$\|\partial_{t_k}^n g\|_{\infty, I_1 \times I_2} \leq \frac{C_g}{\gamma_g^n} \frac{(n+\sigma-1)!}{(\sigma-1)!} \quad \forall n \in \mathbb{N}_0 \text{ and } k \in \{1, 2\} \quad (18)$$

to conclude that

$$\begin{aligned} \|g - \mathcal{I}_{I_1 \times I_2}^{(m)}[g]\|_{\infty, I_1 \times I_2} &\leq 4eC_g(\Lambda_m + 1)^2(m+2)^\sigma \left(1 + \frac{\max\{|I_1|, |I_2|\}}{\gamma_g} \right) \varrho \left(\frac{2\gamma_g}{\max\{|I_1|, |I_2|\}} \right)^{-(m+1)}, \end{aligned} \quad (19)$$

where $\varrho(r) := r + \sqrt{1 + r^2}$. From the estimate (13) and the admissibility criterion (16) it follows that

$$\begin{aligned} \|\partial_{t_k}^n g\|_{\infty, I_1 \times I_2} &\leq \max_{(t_1, t_2) \in I_1 \times I_2} \frac{c_{\text{as}} n!(n+1)^{3/2}}{\alpha^{3/2}(t_1 - t_2)^{n+3/2}} = \frac{c_{\text{as}} n!(n+1)^{3/2}}{\alpha^{3/2} \text{dist}(I_1, I_2)^{n+3/2}} \\ &\leq \frac{c_{\text{as}} n! c_\beta \beta^n}{\alpha^{3/2} \text{dist}(I_1, I_2)^{3/2}} \left(\frac{\eta_2}{\max\{|I_1|, |I_2|\}} \right)^n, \end{aligned}$$

where $\beta > 1$ is arbitrary and c_β is chosen such that $(n+1)^{3/2} \leq c_\beta \beta^n$ for all $n \geq 0$. Thus, (18) is satisfied for the choice $\sigma = 1$, $\gamma_g = \max\{|I_1|, |I_2|\}/(\eta_2 \beta)$ and $C_g = c_{\text{as}} c_\beta / (\alpha \text{dist}(I_1, I_2))^{3/2}$. Since $c_\beta \rightarrow \infty$ as $\beta \rightarrow 1$ we fix $\beta = 4/3$ and a proper constant c_β . Inserting these choices of σ , γ_g and C_g into (19) yields

$$\|g - \mathcal{I}_{I_1 \times I_2}^{(m)}[g]\|_{\infty, I_1 \times I_2} \leq \frac{4e c_{\text{as}} c_\beta (\Lambda_m + 1)^2 (m+2)}{(\alpha \text{dist}(I_1, I_2))^{3/2}} \left(1 + \frac{4}{3} \eta_2\right) \varrho \left(\frac{3}{2\eta_2}\right)^{-(m+1)}.$$

Similarly as in [2, Remark 4.23] we want to simplify this estimate. We set $q_2 = 1 + 3/(2\eta_2)$, $\varrho_2 = \varrho(3/(2\eta_2))$ and $\mu_2 = \varrho_2/q_2$. Since $\varrho(r) > 1 + r$ for all $r \in \mathbb{R}$ we see that $\mu_2 > 1$. Thus, we can rewrite the right-hand side of the above estimate as

$$\frac{4e c_{\text{as}} c_\beta (\Lambda_m + 1)^2 (m+2) (1 + 4\eta_2/3)}{\mu_2^{m+1}} \frac{1}{(\alpha \text{dist}(I_1, I_2))^{3/2}} q_2^{-(m+1)}.$$

The first fraction is a zero sequence in m and can thus be bounded by a constant c_2 . In fact, its numerator grows like $\log(m+1)^2(m+2)$ due to (15), while its denominator grows exponentially with increasing m . Therefore, we end up with the estimate in (17). \square

Theorem 2 (One-sided interpolation error). *Let $\eta_1 > 0$ and $q_1 := 1 + 3/(2\eta_1)$. Let $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$ be two non-empty intervals such that $a_1 > b_2$. Let the admissibility criterion*

$$\eta_1 \text{dist}(I_1, I_2) \geq \min\{|I_1|, |I_2|\} \quad (20)$$

be satisfied. Let k be such that I_k is the shorter time interval. Then there exists a constant $c_1 > 0$ such that

$$\|g - \mathcal{I}_{\text{dir}, I_k}^{(m)}[g]\|_{\infty, I_1 \times I_2} \leq \frac{c_1}{(\alpha \text{dist}(I_1, I_2))^{3/2}} q_1^{-(m+1)}. \quad (21)$$

Proof. The proof is very similar to the proof of Theorem 1. This time we apply [2, Lemma 4.19] which leads to the estimate

$$\|g - \mathcal{I}_{\text{dir}, I_k}^{(m)}[g]\|_{\infty, I_1 \times I_2} \leq 2e C_g (\Lambda_m + 1) (m+2)^\sigma \left(1 + \frac{|I_k|}{\gamma_g}\right) \varrho \left(\frac{2\gamma_g}{|I_k|}\right)^{-(m+1)},$$

with the same constants σ , γ_g and C_g and function ϱ as in the proof of Theorem 1. With the admissibility criterion (20) we get

$$\|g - \mathcal{I}_{\text{dir}, I_k}^{(m)}[g]\|_{\infty, I_1 \times I_2} \leq 2e C_g (\Lambda_m + 1) (m+2)^\sigma \left(1 + \frac{4}{3} \eta_1\right) \varrho \left(\frac{3}{2\eta_1}\right)^{-(m+1)},$$

which can be simplified in the same way as the corresponding result in the proof of Theorem 1 to show (21). \square

Remark 1. The results in Theorems 1 and 2 regarding the two- and one-sided interpolation errors are very similar. In both situations the error decreases exponentially with increasing interpolation degree m . The convergence rate depends on the constants η_2 and η_1 from the admissibility criteria (16) and (20), respectively, and suffers if they become large. For two fixed, separate intervals I_1 and I_2 we can choose $\eta_2 = \max\{|I_1|, |I_2|\}/\text{dist}(I_1, I_2)$ and $\eta_1 = \min\{|I_1|, |I_2|\}/\text{dist}(I_1, I_2)$ as the smallest constants for which (16) and (20) hold. Hence, in all situations where $\max\{|I_1|, |I_2|\}/\text{dist}(I_1, I_2)$ is considerably larger than $\min\{|I_1|, |I_2|\}/\text{dist}(I_1, I_2)$ the one-sided interpolation in the shorter time interval is expected to perform significantly better than the two-sided interpolation.

3.2 Analysis of the expansion in space

The approximation error for the truncated Chebyshev expansion of the heat kernel for a fixed temporal setting has been investigated in [17, 23]. Here we collect and slightly improve those results. We start by citing an estimate of the absolute value of the expansion coefficients defined in (9c).

Lemma 1 (cf. [23, Lemma 1]). *For the coefficients $E_{k,\ell}(r, \delta)$ in (9c) there holds*

$$|E_{k,\ell}(r, \delta)| \leq \frac{\lambda_k \lambda_\ell}{a^{k+\ell}} \exp\left(\frac{1}{\delta} \left(a - \frac{1}{a}\right)^2\right), \quad (22)$$

where $a \in \mathbb{R}_{>0}$ can be chosen arbitrarily.

Remark 2. The estimate in (22) is slightly sharper than the one in the referenced paper where an additional multiplicative factor 4 is included in the argument of the exponential function. This factor can be dropped in the last step of the proof in [23], see [15, Satz 4.13].

Theorem 3 (Chebyshev series truncation error, cf. [23, p. 209] and [17, Corollary 4.1]). *Let $\tilde{c}_{\text{st}} \in \mathbb{R}_{>0}$. Let X_1 and X_2 be two cubes with edge length $2\tilde{h}_x$. Let $t, \tau \in \mathbb{R}$ such that $t > \tau$ and*

$$\frac{4\alpha(t-\tau)}{\tilde{h}_x^2} \geq \tilde{c}_{\text{st}}. \quad (23)$$

Let the function $\sigma : \mathbb{R}_{>0} \rightarrow \mathbb{R}$ be defined by

$$\sigma(s) := \frac{1}{2} \left(\ln(s + \sqrt{1+s^2}) - \frac{\sqrt{s^2+1}-1}{s} \right). \quad (24)$$

Then there exists a constant $c_x > 0$ such that

$$\begin{aligned} & \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha(\cdot_x - \cdot_y, t - \tau)]\|_{\infty, X_1 \times X_2} \\ & \leq c_x \frac{(m_x + 2)^5}{(\alpha(t - \tau))^{3/2}} \exp\left(- (m_x + 1) \sigma((m_x + 1)\tilde{c}_{\text{st}}/12)\right). \end{aligned} \quad (25)$$

Proof. The proof is based on the proof of [17, Corollary 4.1] and the idea to minimize the right-hand side of (22) in [24, p. 3551], where the function κ corresponds to σ here. To estimate the error we represent it as the remainder of the Chebyshev series, namely

$$(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha(\cdot_x - \cdot_y, t - \tau)](\mathbf{x}, \mathbf{y}) = \sum_{n=m_x+1}^{\infty} \sum_{|\boldsymbol{\kappa}+\boldsymbol{\nu}|=n} E_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^-(t, \tau) T_{X_1, \boldsymbol{\nu}}(\mathbf{x}) T_{X_2, \boldsymbol{\kappa}}(\mathbf{y}),$$

where $E_{\kappa, \nu}^-(t, \tau) = (4\pi\alpha(t - \tau))^{-3/2} \prod_{j=1}^3 E_{\kappa_j, \nu_j}(r_j, \delta)$ with $\delta = 4\alpha(t - \tau)/\tilde{h}_x^2$, r_j and $E_{\kappa_j, \nu_j}(r_j, \delta)$ from (9). The polynomials $T_{X_1, \nu}$ and $T_{X_2, \kappa}$ are tensor products of transformed Chebyshev polynomials as defined in (6) and, thus, there holds $|T_{X_1, \nu}(\mathbf{x})| \leq 1$ for all $\mathbf{x} \in X_1$ and all $\nu \in \mathbb{N}_0^3$ and the same for $T_{X_2, \kappa}$ in X_2 . Hence, we can estimate

$$\|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha(\cdot \mathbf{x} - \cdot \mathbf{y}, t - \tau)]\|_{\infty, X_1 \times X_2} \leq \sum_{n=m_x+1}^{\infty} \sum_{|\kappa+\nu|=n} |E_{\kappa, \nu}^-(t, \tau)|. \quad (26)$$

For a given n and $\kappa, \nu \in \mathbb{N}_0^3$ such that $|\kappa + \nu| = n$ the estimate in (22) yields

$$(4\pi\alpha(t - \tau))^{3/2} |E_{\kappa, \nu}^-(t, \tau)| \leq \exp\left(\frac{1}{\delta}\left(a_n - \frac{1}{a_n}\right)^2\right)^3 \prod_{j=1}^3 \frac{\lambda_{\kappa_j} \lambda_{\nu_j}}{a_n^{\kappa_j + \nu_j}} \leq \frac{64}{a_n^n} \exp\left(\frac{3}{\delta}\left(a_n - \frac{1}{a_n}\right)^2\right),$$

where $a_n \in \mathbb{R}_{>0}$ can be chosen arbitrarily. To find the optimal a_n we minimize the function f_n on the right-hand side, which can be rewritten in the form

$$f_n(a_n) = 64 \exp\left(-n \ln(a_n) + \frac{3}{\delta}\left(a_n - \frac{1}{a_n}\right)^2\right).$$

From the necessary condition $f'_n(a_n^*) = 0$ we obtain $a_n^* = \sqrt{n\delta/12 + \sqrt{1 + (n\delta/12)^2}}$, which is indeed a minimizer of f_n in $\mathbb{R}_{>0}$ since f_n becomes unbounded for a_n tending to zero or infinity. Therefore, the minimum of f_n is given by

$$f_n(a_n^*) = 64 \exp\left(-n \left(\ln(a_n^*) - \frac{12}{4n\delta}\left(a_n^* - \frac{1}{a_n^*}\right)^2\right)\right) = 64 \exp(-n\sigma(n\delta/12))$$

with the function σ from (24) and we obtain

$$|E_{\kappa, \nu}^-(t, \tau)| \leq \frac{64}{(4\pi\alpha(t - \tau))^{3/2}} \exp(-n\sigma(n\delta/12)).$$

By using this estimate and $\#\{(\kappa, \nu) \in \mathbb{N}_0^3 \times \mathbb{N}_0^3 : |\kappa + \nu| = n\} = \binom{n+5}{5}$ we can further estimate the approximation error in (26) by

$$\begin{aligned} \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha(\cdot \mathbf{x} - \cdot \mathbf{y}, t - \tau)]\|_{\infty, X_1 \times X_2} \\ \leq \frac{64}{(4\pi\alpha(t - \tau))^{3/2}} \sum_{n=m_x+1}^{\infty} \binom{n+5}{5} \exp(-n\sigma(n\delta/12)). \end{aligned}$$

Standard arguments of calculus show that the function σ is monotonically increasing. Hence, we can use (23), which corresponds to the estimate $\delta = 4\alpha(t - \tau)/\tilde{h}_x^2 \geq \tilde{c}_{\text{st}}$, to get for all $n > m_x$

$$\exp(-n\sigma(n\delta/12)) \leq \exp(-n\sigma((m_x + 1)\tilde{c}_{\text{st}}/12)) = q^n,$$

where $q := \exp(-\sigma((m_x + 1)\tilde{c}_{\text{st}}/12))$. The series $\sum_{n=m_x+1}^{\infty} 5! \binom{n+5}{5} q^n$ is the remainder of the fifth derivative of the geometric series $\sum_{n=0}^{\infty} q^n$ truncated after $(m_x + 1)$ terms. It can be estimated by

$$\sum_{n=m_x+1}^{\infty} 5! \binom{n+5}{5} q^n \leq \frac{5! (m_x + 2)^5}{(1 - q)^6} q^{m_x+1}.$$

The variable q converges monotonically to 0 as $m_x \rightarrow \infty$ so we can bound $(1-q)^{-6}$ by a constant c for all $m_x \geq 0$. We conclude

$$\|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha(\cdot_x - \cdot_y, t - \tau)]\|_{\infty, X_1 \times X_2} \leq \frac{64}{(4\pi\alpha(t - \tau))^{3/2}} c (m_x + 2)^5 q^{m_x+1}$$

which is (25) with $c_x = (64c)/((4\pi)^{3/2})$. \square

Remark 3. The behavior of σ in (24) determines the approximation error in (25). In the proof of Theorem 3 we have already pointed out that σ is monotonically increasing. Furthermore there holds $\sigma(s) \sim s/4$ for $s \rightarrow 0$ and $\sigma(s) \sim \ln(2s)/2$ for $s \rightarrow \infty$ [24, p. 3551]. Thus, the convergence in (25) is super-exponential in m_x . Note that the approximation quality depends on the constant \tilde{c}_{st} in (23), which appears in the argument of σ . For small values of \tilde{c}_{st} the estimate suffers. Thus, we have to ensure that (23) holds for a reasonably large constant \tilde{c}_{st} in the later application. This means that we have to adapt the size of the cubes to the temporal configuration.

3.3 The space-time approximation error

By combining the results from the previous two sections we can estimate the approximation error of the expansions (8), (10) and (11) in $Z_{\text{tar}} \times Z_{\text{src}}$.

Theorem 4 (Full space-time expansion error, [17, cf. Lemma 7.4]). *Let $\tilde{c}_{\text{st}}, \eta_2 \in \mathbb{R}_{>0}$ and $q_2 := 1 + 3/(2\eta_2)$. Let Λ_{m_t} be the Lebesgue constant defined in (14) and σ be the function in (24). Let $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$ be two non-empty intervals with $a_1 > b_2$ that satisfy the admissibility criterion (16). Let X_1 and X_2 be two cubes in \mathbb{R}^3 with edge length $2\tilde{h}_x$ such that*

$$\frac{4\alpha \text{dist}(I_1, I_2)}{\tilde{h}_x^2} \geq \tilde{c}_{\text{st}}. \quad (27)$$

Let $Z_{\text{tar}} = X_1 \times I_1$ and $Z_{\text{src}} = X_2 \times I_2$. Then there exist constants $c_2, c_x \in \mathbb{R}_{>0}$ such that

$$\begin{aligned} & \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{I_1 \times I_2}^{(m_t)})[G_\alpha]\|_{\infty, Z_{\text{tar}} \times Z_{\text{src}}} \\ & \leq \frac{1}{(\alpha \text{dist}(I_1, I_2))^{3/2}} \left(c_2 q_2^{-(m_t+1)} + c_x \Lambda_{m_t}^2 (m_x + 2)^5 \exp\left(- (m_x + 1)\sigma((m_x + 1)\tilde{c}_{\text{st}}/12)\right) \right). \end{aligned} \quad (28)$$

Proof. Throughout this proof we use $\|\cdot\|_\infty$ to denote $\|\cdot\|_{\infty, Z_{\text{tar}} \times Z_{\text{src}}}$ to shorten the notation. We start to estimate the approximation error by adding and subtracting $\mathcal{I}_{I_1 \times I_2}^{(m_t)}[G_\alpha]$ and using the triangle inequality to get

$$\|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{I_1 \times I_2}^{(m_t)})[G_\alpha]\|_\infty \leq \|(\text{Id} - \mathcal{I}_{I_1 \times I_2}^{(m_t)})[G_\alpha]\|_\infty + \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)}) \mathcal{I}_{I_1 \times I_2}^{(m_t)}[G_\alpha]\|_\infty.$$

The first term can be further estimated by using Theorem 1. In fact, the function g defined in (12) which is considered in that theorem is just the heat kernel for fixed spatial points \mathbf{x} and \mathbf{y} and the estimate does not depend on these points. Thus,

$$\|(\text{Id} - \mathcal{I}_{I_1 \times I_2}^{(m_t)})[G_\alpha]\|_\infty \leq \frac{c_2}{(\alpha \text{dist}(I_1, I_2))^{3/2}} q_2^{-(m_t+1)}$$

with c_2 from Theorem 1. For the second term we observe that

$$\|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{I_1 \times I_2}^{(m_t)})[G_\alpha]\|_\infty \leq \sup_{f \in C(Z_{\text{tar}} \times Z_{\text{src}}) \setminus \{0\}} \frac{\|\mathcal{I}_{I_1 \times I_2}^{(m_t)}[f]\|_\infty}{\|f\|_\infty} \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha]\|_\infty,$$

where we used that the operators $\mathcal{S}_{X_1 \times X_2}^{(m_x)}$ and $\mathcal{I}_{I_1 \times I_2}^{(m_t)}$ commute. The operator norm of $\mathcal{I}_{I_1 \times I_2}^{(m_t)}$ is bounded by $\Lambda_{m_t}^2$, which follows from (14). For the other term we use assumption (27) and apply Theorem 3 to get

$$\begin{aligned} \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)})[G_\alpha]\|_\infty &\leq \sup_{(t,\tau) \in I_1 \times I_2} \left\{ c_x \frac{(m_x + 2)^5}{(\alpha(t - \tau))^{3/2}} \exp\left(-(m_x + 1)\sigma((m_x + 1)\tilde{c}_{\text{st}}/12)\right) \right\} \\ &\leq c_x \frac{(m_x + 2)^5}{(\alpha \text{dist}(I_1, I_2))^{3/2}} \exp\left(-(m_x + 1)\sigma((m_x + 1)\tilde{c}_{\text{st}}/12)\right). \end{aligned}$$

Combining all these estimates completes the proof. \square

By using the result from Theorem 2 one can prove the following approximation error estimates for the expansions (10) and (11) in an analogous way.

Theorem 5 (Temporally one-sided space-time expansion errors). *Let \tilde{c}_{st} , $\eta_1 \in \mathbb{R}_{>0}$ and $q_1 := 1 + 3/(2\eta_1)$. Let $I_1 = [a_1, b_1]$ and $I_2 = [a_2, b_2]$ be two non-empty intervals with $a_1 > b_2$ that satisfy the one-sided admissibility criterion (20). Let X_1 and X_2 be two cubes in \mathbb{R}^3 with edge length $2h_x$ such that criterion (27) holds. Let $Z_{\text{tar}} = X_1 \times I_1$ and $Z_{\text{src}} = X_2 \times I_2$. Let $k \in \{1, 2\}$ be such that I_k is the shorter time interval. Then there exist constants $c_1, c_x \in \mathbb{R}_{>0}$ such that*

$$\begin{aligned} \|(\text{Id} - \mathcal{S}_{X_1 \times X_2}^{(m_x)} \mathcal{I}_{\text{dir}, I_k}^{(m_t)})[G_\alpha]\|_{\infty, Z_{\text{tar}} \times Z_{\text{src}}} \\ \leq \frac{1}{(\alpha \text{dist}(I_1, I_2))^{3/2}} \left(c_1 q_1^{-(m_t+1)} + c_x \Lambda_{m_t} (m_x + 2)^5 \exp\left(-(m_x + 1)\sigma(\tilde{c}_{\text{st}}(m_x + 1)/12)\right) \right). \end{aligned} \quad (29)$$

Theorems 4 and 5 show that we can control the approximation error of the expansions in (8), (10) and (11) if we can control the respective errors in time and space separately. The temporally single sided expansions (10) and (11) are favorable in those situations, where the one-sided interpolation performs better than the two-sided interpolation, see Remark 1. The spatial sizes of the boxes Z_{tar} and Z_{src} have to be adapted to the distance of their temporal components according to the criterion (27) for all three expansions.

Note that in the previous works [17, 22, 23, 25] a criterion different from (27) is used to determine the proper spatial size of the boxes for the expansion (8). There it is required that there exists a constant c_{st} such that

$$\frac{\tilde{h}_x^2}{4\alpha \tilde{h}_t} \leq c_{\text{st}} \quad (30)$$

holds for each space-time box, where \tilde{h}_x denotes the half length of the edges of its spatial component and \tilde{h}_t the half length of its temporal interval. However, the criterion (27) follows from (30) for two boxes $Z_{\text{tar}} = X_1 \times I_1$ and $Z_{\text{src}} = X_2 \times I_2$, if the admissibility criterion (16) holds in addition. Indeed,

$$\frac{4\alpha \text{dist}(I_1, I_2)}{\tilde{h}_x^2} \geq \frac{4\alpha \max\{|I_1|, |I_2|\}}{\tilde{h}_x^2 \eta_2} \geq \frac{2}{c_{\text{st}} \eta_2},$$

which is (27) with $\tilde{c}_{\text{st}} = 2(c_{\text{st}} \eta_2)^{-1}$. In the setting of the temporally one-sided expansions (10) and (11) the situation is similar. If criterion (30) is satisfied for the space-time cluster with the smaller time interval, the spatial sizes of the two boxes coincide, and the admissibility criterion (20) is satisfied, then (27) follows as above. Therefore, we will use the criterion (30) instead of (27) in the following sections to allow for an easier comparison of this paper with previous ones.

4 A time-adaptive space-time FMM algorithm

In this section we describe a new variant of the FMM presented in [25] and [17, 22, 23] which improves the performance for meshes which are adaptive in time. Throughout this section we focus on the matrix V_h , but other BEM matrices of the heat equation like the double layer operator K_h can be handled as well with slight modifications. In a first step, we create a hierarchy of space-time boxes that subdivides the elements in the space-time tensor product mesh Σ_h . This hierarchy, which we denote as box cluster tree, will allow us to find a suitable partition of the BEM matrices, which forms the backbone of our FMM.

4.1 A 4D space-time box cluster tree

The following construction of space-time box cluster trees is similar to the one which we described in [25]. The idea is to create a hierarchy of boxes that partition the space-time tensor product mesh Σ_h by recursively subdividing an initial box $Z^{(0)}$ that contains the whole mesh. Here we introduce new purely spatial subdivisions of certain boxes that will enable adaptive FMM operations in the later algorithm. Note that from here on we consider half-open boxes $Z = (\mathbf{a}, \mathbf{b}] \times (c, d] \subset \mathbb{R}^4$, since they allow for a simple, non-overlapping partition into smaller half-open boxes.

A space-time boundary element $\sigma_{j_t, j_x} = \gamma_{j_x} \times (t_{j_t-1}, t_{j_t})$ of a given mesh Σ_h is assigned to a box $Z = X \times I \subset \mathbb{R}^4$ if its geometrical center is contained in Z . By \hat{Z} we denote the set of all indices (j_t, j_x) corresponding to elements $\sigma_{j_t, j_x} \in \Sigma_h$ that are assigned to Z . We use the symbol $\#\hat{Z}$ for the cardinality of \hat{Z} and additionally introduce $n_t(\hat{Z})$ as the number of all distinct time-indices in \hat{Z} . E.g., if Z is a box with $n_t(\hat{Z}) = 1$, all space-time boundary elements that are assigned to Z share the same temporal component. Such boxes play a central role in the new time-adaptive FMM which motivates the following definition.

Definition 1. A box $Z \in \mathcal{T}_\Sigma$ is called *temporally indivisible* if $n_t(\hat{Z}) = 1$.

Algorithm 1 describes the recursive construction of a space-time box cluster tree \mathcal{T}_Σ . The concrete subdivision steps in line 8, 10 and 12 for a box $Z = (\mathbf{a}, \mathbf{b}] \times (c, d]$ are executed as follows:

- a) Purely temporal subdivision (line 8): We split the time interval $(c, d]$ into $(c, \tilde{c}]$ and $(\tilde{c}, d]$, where \tilde{c} is the grid point t_j of the time partition closest to $(c+d)/2$. Then we subdivide Z into the two boxes $Z_1 = (\mathbf{a}, \mathbf{b}] \times (c, \tilde{c}]$ and $Z_2 = (\mathbf{a}, \mathbf{b}] \times (\tilde{c}, d]$.
- b) Purely spatial subdivision (line 12): The box $(\mathbf{a}, \mathbf{b}]$ is uniformly split into 8 boxes $(\mathbf{a}, \tilde{\mathbf{a}}], \dots, (\tilde{\mathbf{a}}, \mathbf{b}]$, where $\tilde{\mathbf{a}} = 1/2(\mathbf{a} + \mathbf{b})$ is the center of $(\mathbf{a}, \mathbf{b}]$. By combining each of these spatial boxes with the whole time interval $(c, d]$ one obtains a spatial subdivision of Z .
- c) Space-time subdivision (line 10): We subdivide the time interval as in a) and the spatial box as in b). The 16 children of Z are obtained by considering all possible combinations of the resulting time intervals and spatial boxes.

In general, we switch between purely temporal and space-time subdivisions of boxes in the tree construction due to criterion (30), at least after some initial, purely temporal subdivisions. A purely spatial subdivision of a cluster is only executed if it cannot be further subdivided in time. Note that such a cluster is not subdivided anymore in the standard algorithm in [25]. The additional spatial subdivisions here will allow for some new efficient operations in the FMM.

By construction, the temporal part of a space-time boundary element σ_{j_t, j_x} is always fully contained in a box Z , if the element is assigned to this box. To guarantee the same property

Algorithm 1 Construction of a 4D space-time box cluster tree \mathcal{T}_Σ for Σ_h .

Require: Let a space-time tensor mesh Σ_h be given.

Let a bound n_{\max} for the number of elements in a leaf box and $c_{\text{st}} > 0$ for (30) be given.

Let $Z^{(0)} = (\mathbf{a}, \mathbf{a} + 2h_x^{(0)}\mathbf{1}] \times (0, T] \subset \mathbb{R}^4$ be given such that Σ_h is contained in $Z^{(0)}$ and $h_x^{(0)}$ and $h_t^{(0)} := T/2$ satisfy (30).

1: Construct an empty tree \mathcal{T}_Σ and add $Z^{(0)}$ as its root.

2: Call SUBDIVIDECLUSTER($Z^{(0)}$, \mathcal{T}_Σ)

```

3: function SUBDIVIDECLUSTER( $Z$ ,  $\mathcal{T}_\Sigma$ )
4:   if  $\#\hat{Z} \geq n_{\max}$ 
5:     if  $n_t(\hat{Z}) > 1$ 
6:       Let  $\ell = \text{level}(Z)$ ,  $h_t^{(\ell+1)} = 2^{-\ell-1}h_t^{(0)}$  and  $\tilde{h}_x$  be the spatial half-size of  $Z$ .
7:       if  $h_t^{(\ell+1)}$  and  $\tilde{h}_x$  satisfy (30)
8:         Subdivide  $Z$  into  $n_C = 2$  children  $\{Z_k\}_{k=1}^{n_C}$  by a temporal subdivision.
9:       else
10:        Subdivide  $Z$  into  $n_C = 16$  children  $\{Z_k\}_{k=1}^{n_C}$  by a space-time subdivision.
11:     else // new: purely spatial subdivisions for  $Z$  with  $n_t(\hat{Z}) == 1$ 
12:       Subdivide  $Z$  into  $n_C = 8$  children  $\{Z_k\}_{k=1}^{n_C}$  by a spatial subdivision.
13:     for  $k = 1, \dots, n_C$ 
14:       if  $\#\hat{Z}_k \neq 0$ 
15:         Add  $Z_k$  to  $\mathcal{T}_\Sigma$  as child of  $Z$  and call SUBDIVIDECLUSTER( $Z_k$ ,  $\mathcal{T}_\Sigma$ ).

```

in space, we pad the spatial boxes in a uniform way. More details about this padding and the handling of other exceptional situations are given in [25].

In the rest of the paper we will use the terms box and cluster interchangeably when we speak about boxes in the tree \mathcal{T}_Σ . For a box $Z \in \mathcal{T}_\Sigma$ we denote its parent in the tree by $\text{par}(Z)$ and the set of all its children by $\text{child}(Z)$. The temporal level $\ell_t(Z)$ of Z is defined as the number of performed temporal subdivisions starting from the root to obtain Z and the spatial level $\ell_x(Z)$ as the number of spatial subdivisions. Due to our subdivision strategy the temporal level $\ell_t(Z)$ of a box corresponds to its actual level (level(Z)) in the tree, unless Z is obtained from a purely spatial subdivision of its parent. The largest level in the tree \mathcal{T}_Σ is denoted by $\text{depth}(\mathcal{T}_\Sigma)$.

4.2 Matrix partitioning using interaction lists

The boxes in the space-time box cluster tree \mathcal{T}_Σ are used to partition the BEM matrix \mathbf{V}_h . For two boxes $Z_{\text{tar}} = X_{\text{tar}} \times I_{\text{tar}}$ and $Z_{\text{src}} = X_{\text{src}} \times I_{\text{src}}$ in \mathcal{T}_Σ let $\mathbf{V}_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ be the block of \mathbf{V}_h formed by the rows corresponding to indices $(j_t, j_x) \in \hat{Z}_{\text{tar}}$ and the columns corresponding to indices $(k_t, k_x) \in \hat{Z}_{\text{src}}$. Let $Z^{(0)}$ be the root of \mathcal{T}_Σ . The block $\mathbf{V}_h|_{\hat{Z}^{(0)} \times \hat{Z}^{(0)}}$ corresponds to the full matrix \mathbf{V}_h . We can subdivide it into n_c^2 blocks $\mathbf{V}_h|_{\hat{Z}_i^{(1)} \times \hat{Z}_j^{(1)}}$, where $\{Z_j^{(1)}\}_{j=1}^{n_c}$ are the children of $Z^{(0)}$ in \mathcal{T}_Σ . For the FMM we recursively subdivide the matrix into blocks in this manner. The subdivision is stopped for a block $\mathbf{V}_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ if Z_{tar} and Z_{src} allow for one of the kernel approximations in (8), (10) or (11) due to a proper temporal separation. Such a block is called admissible. We will see in Section 4.3 how to approximate admissible blocks of \mathbf{V}_h efficiently. All other blocks are called inadmissible and are further subdivided as long as Z_{tar} and Z_{src} are not both temporally indivisible. The preferred kernel approximation is (8), which is why both the

source and target boxes are subdivided for a typical block refinement, if these box subdivisions include a temporal subdivision. If either the source box or the target box is temporally indivisible (recall Definition 1), the respective other box is further subdivided to detect admissible blocks related to the kernel approximations (10) or (11). These one-sided subdivisions are new compared to the previous works [17, 23, 25] in the setting of the heat equation and inspired by the adaptive FMM in [3, 4, 20]. In the following we describe the full partitioning scheme in more detail.

In Theorem 4 we have shown that the full expansion (8) is well-suited for Z_{tar} and Z_{src} if the temporal intervals I_{tar} and I_{src} satisfy the standard admissibility criterion (16) for a given constant $\eta_2 > 0$. This is checked explicitly during the recursive construction of the blocks. For the temporally one-sided expansions (10) and (11) we have to explicitly check that the criterion (20) is satisfied for a given $\eta_1 > 0$. The criterion (30) will be satisfied in all three cases for a suitable constant $c_{\text{st}} > 0$ due to the construction of boxes of the space-time box cluster tree \mathcal{T}_Σ in Algorithm 1 and the subdivision of blocks, which we will describe in more detail in Algorithm 2.

The partitioning of V_h is further influenced by the causality and the exponential decay of the heat kernel G_α in (3). Due to the causality, $G_\alpha(\mathbf{x} - \mathbf{y}, t - \tau)$ vanishes for all $(\mathbf{x}, t) \in Z_{\text{tar}}$ and $(\mathbf{y}, \tau) \in Z_{\text{src}}$ with $I_{\text{tar}} = (a_1, b_1]$ and $I_{\text{src}} = (a_2, b_2]$ if $a_2 \geq b_1$. We say that I_{src} is *causally relevant* for I_{tar} in the contrary case, i.e. if $a_2 < b_1$. If I_{src} is not causally relevant for I_{tar} , all entries of the block $V_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ are zero and we can ignore it. This is related to the lower triangular block structure of V_h .

The exponential decay of the heat kernel allows to neglect interactions between boxes that are sufficiently separated in space, which we identify as follows. Let Z_{tar} be a box in \mathcal{T}_Σ with spatial level $\ell_x = \ell_x(Z)$. Due to the uniform subdivision strategy employed in the construction of \mathcal{T}_Σ , X_{tar} is one of 8^{ℓ_x} possible boxes in a regular grid \mathcal{G}^{ℓ_x} . We can label the boxes in this grid with multi-indices in $\{0, \dots, 2^{\ell_x} - 1\}^3$ in a regular way (ascending componentwise) and use these indices to measure the distance of boxes. For two spatial boxes X and Y in \mathcal{G}^{ℓ_x} with corresponding multi-indices ξ and ζ we define the *grid distance* of X and Y by

$$\text{griddist}(X, Y) := \max_j \{|\xi_j - \zeta_j|\}. \quad (31)$$

For a fixed truncation parameter n_{tr} we define the *interaction area* $\mathcal{I}_{\mathcal{A}}(X)$ of X in \mathcal{G}^{ℓ_x} by

$$\mathcal{I}_{\mathcal{A}}(X) := \{Y \in \mathcal{G}^{\ell_x} : \text{griddist}(X, Y) \leq n_{\text{tr}}\}. \quad (32)$$

In the partitioning of V_h we will ignore blocks $V_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ if $X_{\text{src}} \notin \mathcal{I}_{\mathcal{A}}(X_{\text{tar}})$. From [23, Section 5.4] it follows that a single, properly chosen truncation parameter n_{tr} can be used for the definition of the interaction area and the related truncation for various pairs of clusters Z_{tar} and Z_{src} . A suitable requirement is that the ratio $\tilde{h}_x^2 / \max\{|I_{\text{tar}}|, |I_{\text{src}}|\}$ of these clusters is similar, where \tilde{h}_x denotes the spatial half length of Z_{tar} and Z_{src} . This will be the case in the partitioning of V_h for clusters Z_{src} and Z_{tar} in the part of \mathcal{T}_Σ where we alternate between purely temporal and space-time refinements.

To keep track of the blocks $V_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ in the final partition of V_h we introduce four different interaction lists for clusters Z_{tar} in \mathcal{T}_Σ . The standard M2L interaction list $\mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$ will include clusters Z_{src} such that the full space-time expansion (8) is admissible for Z_{tar} and Z_{src} but not for their parents. For the new temporally one-sided expansion (10) in the source interval we introduce the M2Lx interaction list $\mathcal{I}_{\text{M2Lx}}(Z_{\text{tar}})$ and for the temporally one-sided expansion (11) in the target interval the Mx2L interaction list $\mathcal{I}_{\text{Mx2L}}(Z_{\text{tar}})$. The nomenclature for these lists is related to the associated FMM operations, which we introduce in Section 4.3. The last list is the nearfield $\mathcal{N}(Z_{\text{tar}})$ which contains clusters Z_{src} corresponding to blocks $V_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ which are inadmissible

and cannot be subdivided further. For a given cluster Z_{tar} some or all of these lists might be empty. The construction of all lists of all clusters in \mathcal{T}_Σ is done during the recursive subdivision of blocks of V_h which we described above and present in full detail in Algorithms 2 and 3.

Algorithm 2 Recursive construction of the interaction lists in \mathcal{T}_Σ .

Require: Let \mathcal{T}_Σ be a space-time cluster tree with root $Z^{(0)}$.

Fix constants $\eta_1, \eta_2 \in \mathbb{R}_{>0}$ for the criteria (16) and (20) and n_{tr} for the definition of (32).

```

1: Call DETERMINEOPERATIONLISTS( $Z^{(0)}, Z^{(0)}$ ).

2: function DETERMINEOPERATIONLISTS( $Z_{\text{src}}, Z_{\text{tar}}$ )
3:   if  $X_{\text{src}} \in \mathcal{I}_A(X_{\text{tar}})$  // Interaction area  $\mathcal{I}_A(X_{\text{tar}})$  defined in (32)
4:     if  $I_{\text{src}}$  and  $I_{\text{tar}}$  satisfy the admissibility criterion (16)
5:       Add  $Z_{\text{src}}$  to  $\mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$ .
6:     else
7:       if  $Z_{\text{tar}}$  is not a leaf and  $Z_{\text{tar}}$  is not temporally indivisible
8:         if  $Z_{\text{src}}$  is not a leaf and  $Z_{\text{src}}$  is not temporally indivisible
9:           for all pairs  $(Z_{\text{src},c}, Z_{\text{tar},c})$  with  $Z_{\text{src},c} \in \text{child}(Z_{\text{src}})$  and  $Z_{\text{tar},c} \in \text{child}(Z_{\text{tar}})$ 
10:          if  $I_{\text{src},c}$  is causally relevant for  $I_{\text{tar},c}$ 
11:            Call DETERMINEOPERATIONLISTS( $Z_{\text{src},c}, Z_{\text{tar},c}$ ).
12:          else //  $Z_{\text{src}}$  is a leaf or a temporally indivisible cluster.
13:            if  $Z_{\text{src}}$  is temporally indivisible
14:              for all  $Z_{\text{tar},c} \in \text{child}(Z_{\text{tar}})$ 
15:                Call DETERMINEM2LANDNEARFIELDLISTS( $Z_{\text{src}}, Z_{\text{tar},c}$ ).
16:              else //  $Z_{\text{src}}$  is not a temporally indivisible cluster but a leaf.
17:                Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
18:            else //  $Z_{\text{tar}}$  is a leaf or a temporally indivisible cluster.
19:              if  $Z_{\text{tar}}$  is temporally indivisible
20:                if  $Z_{\text{src}}$  is not a leaf and  $Z_{\text{src}}$  is not temporally indivisible
21:                  for all  $Z_{\text{src},c} \in \text{child}(Z_{\text{src}})$ 
22:                    Call DETERMINEM2LXANDNEARFIELDLISTS( $Z_{\text{src},c}, Z_{\text{tar}}$ ).
23:                  else
24:                    Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
25:                else //  $Z_{\text{tar}}$  is not a temporally indivisible cluster but a leaf.
26:                  Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .

```

The main routine DETERMINEOPERATIONLISTS in Algorithm 2 realizes the recursive subdivision of blocks. If the spatial component X_{src} is not in the interaction area $\mathcal{I}_A(X_{\text{tar}})$ of X_{tar} , the corresponding block can be neglected and the subdivision is stopped (line 3). Otherwise it is checked whether the admissibility criterion (16) is satisfied (line 4) in which case Z_{src} is added to the proper interaction list $\mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$. In the contrary case, we want to subdivide the block.

The desired subdivision strategy for an inadmissible block is to consider all pairs of children of Z_{src} and Z_{tar} , check whether the corresponding blocks are non-zero due to causality (line 10), and call the routine DETERMINEOPERATIONLISTS for the children if this is the case (line 11). Leaf clusters and temporally indivisible clusters, whose children are obtained by purely spatial subdivisions, are treated separately. If only one of the clusters is temporally indivisible and the other is not a leaf, we can still subdivide the block and check if the resulting blocks are admissible. To motivate this, let e.g. Z_{src} be temporally indivisible and Z_{tar} be neither a

leaf nor temporally indivisible. We can consider the blocks of V_h corresponding to Z_{src} and $Z_{\text{tar},c} \in \text{child}(Z_{\text{tar}})$ for which a temporally one-sided expansion (11) instead of the full expansion (8) may be admissible, since the children of Z_{tar} are subdivided with respect to time. Thus, such clusters are treated separately in the subroutine DETERMINEMx2LANDNEARFIELDLISTS in line 15, which is described in Algorithm 3. Similarly, we treat temporally indivisible clusters Z_{tar} and non-leaf clusters Z_{src} which are not temporally indivisible separately in the subroutine DETERMINEM2LxANDNEARFIELDLISTS in line 22, see Algorithm 3 as well. In all other cases, further subdivisions of the inadmissible block are not possible or would not yield efficiently approximable sub-blocks anymore, so we add Z_{src} to the nearfield $\mathcal{N}(Z_{\text{tar}})$ of Z_{tar} (lines 17, 24 and 26).

Algorithm 3 The subroutines to determine Mx2L and M2Lx interaction lists by asymmetric subdivisions.

```

1: function DETERMINEMx2LANDNEARFIELDLISTS( $Z_{\text{src}}, Z_{\text{tar}}$ )
2:   if  $\ell_x(Z_{\text{src}}) < \ell_x(Z_{\text{tar}})$ 
3:     if  $Z_{\text{src}}$  is a leaf
4:       Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
5:     else // Recursive call for spatially refined children of  $Z_{\text{src}}$ .
6:       for all  $Z_{\text{src},c} \in \text{child}(Z_{\text{src}})$ 
7:         Call DETERMINEMx2LANDNEARFIELDLISTS( $Z_{\text{src},c}, Z_{\text{tar}}$ ).
8:     else
9:       if  $I_{\text{src}}$  and  $I_{\text{tar}}$  satisfy the admissibility criterion (20)
10:        Add  $Z_{\text{src}}$  to  $\mathcal{I}_{\text{Mx2L}}(Z_{\text{tar}})$ .
11:      else
12:        if  $Z_{\text{tar}}$  is temporally indivisible
13:          Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
14:        else // Recursive call for children of  $Z_{\text{tar}}$  which are refined in space and time.
15:          for all  $Z_{\text{tar},c} \in \text{child}(Z_{\text{tar}})$ 
16:            Call DETERMINEMx2LANDNEARFIELDLISTS( $Z_{\text{src}}, Z_{\text{tar},c}$ ).

17: function DETERMINEM2LxANDNEARFIELDLISTS( $Z_{\text{src}}, Z_{\text{tar}}$ )
18:   if  $\ell_x(Z_{\text{tar}}) < \ell_x(Z_{\text{src}})$ 
19:     if  $Z_{\text{tar}}$  is a leaf
20:       Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
21:     else // Recursive call for spatially refined children of  $Z_{\text{tar}}$ .
22:       for all  $Z_{\text{tar},c} \in \text{child}(Z_{\text{tar}})$ 
23:         Call DETERMINEM2LxANDNEARFIELDLISTS( $Z_{\text{src}}, Z_{\text{tar},c}$ ).
24:     else
25:       if  $I_{\text{src}}$  and  $I_{\text{tar}}$  satisfy the admissibility criterion (20)
26:         Add  $Z_{\text{src}}$  to  $\mathcal{I}_{\text{M2Lx}}(Z_{\text{tar}})$ .
27:       else
28:         if  $Z_{\text{src}}$  is temporally indivisible
29:           Add  $Z_{\text{src}}$  to  $\mathcal{N}(Z_{\text{tar}})$ .
30:         else // Recursive call for children of  $Z_{\text{src}}$  which are refined in space and time.
31:           for all  $Z_{\text{src},c} \in \text{child}(Z_{\text{src}})$ 
32:             Call DETERMINEM2LxANDNEARFIELDLISTS( $Z_{\text{src},c}, Z_{\text{tar}}$ ).

```

The routine DETERMINEMx2LANDNEARFIELDLISTS in Algorithm 3 is used to determine pairs of clusters Z_{src} and Z_{tar} and corresponding matrix blocks, for which the approximation (11) is applied. It is initially called in line 15 of Algorithm 2 for a temporally indivisible cluster Z_{src} and a target cluster Z_{tar} such that $\ell_t(Z_{\text{tar}}) > \ell_t(Z_{\text{src}})$. The main goal of the routine is to recursively subdivide the target cluster until the admissibility criterion (20) is satisfied for the temporal components of the resulting clusters or until a final inadmissible block is detected. This is the content of lines 9–16 of Algorithm 3. For the temporally one-sided expansion (11) the spatial lengths of Z_{src} and Z_{tar} have to match, which is the case if their spatial levels coincide. The right spatial level is the one of Z_{tar} . This follows from the last paragraph of Section 3.3 since the spatial and temporal half lengths \tilde{h}_x and \tilde{h}_t of Z_{tar} satisfy (30) due to the construction of \mathcal{T}_Σ . It may be necessary to adapt the spatial level $\ell_x(Z_{\text{src}})$ of Z_{src} , which is done in lines 2–7 of Algorithm 3. If $\ell_x(Z_{\text{src}})$ does not match $\ell_x(Z_{\text{tar}})$ and Z_{src} still has spatially refined children, the routine is called recursively for all these children (line 7). If Z_{src} is a leaf in \mathcal{T}_Σ and its spatial level does not match the one of Z_{tar} , we add it to the nearfield of Z_{tar} (line 4) regardless of whether the temporal components satisfy (20) or not. Note that we do not check for an additional spatial truncation in Algorithm 3. This is motivated by the fact, that $\max\{|I_{\text{src}}|, |I_{\text{tar}}|\}$ does not (or at least not significantly) change when only the target cluster is refined. Therefore, an additional spatial truncation is not reasonable.

The routine DETERMINEM2LXANDNEARFIELDLISTS in Algorithm 3 is the analogue of DETERMINEMx2LANDNEARFIELDLISTS when the roles of source and target clusters are interchanged, i.e. when Z_{tar} is temporally indivisible while Z_{src} is not. In this case we want to find pairs of clusters Z_{src} and Z_{tar} and corresponding matrix blocks, for which the approximation (10) is admissible. The recursion strategy including the choice of spatial levels is completely analogous to the one in DETERMINEM2LXANDNEARFIELDLISTS.

Note that a purely spatially refined cluster Z_{src} in the tree \mathcal{T}_Σ can only be considered as a source cluster in the subroutine DETERMINEMx2LANDNEARFIELDLISTS in Algorithm 3, if the corresponding target cluster Z_{tar} has the same spatial level. In particular, there might exist spatially refined clusters, which are never considered in this routine. For the same reason such clusters might never be considered as target clusters in the subroutine DETERMINEM2LXANDNEARFIELDLISTS in Algorithm 3. Hence, \mathcal{T}_Σ might contain clusters that are never visited in the routine DETERMINEOPERATIONLISTS and its subroutines. These clusters (and all their descendants) are not relevant for the FMM and can be removed from the tree \mathcal{T}_Σ . In the rest of the paper we will assume that such clusters do not exist or have already been removed.

The difference between the standard partitioning scheme in e.g. [17, 23, 25] and the one described here is the treatment of temporally indivisible clusters. In the standard version, such clusters are leaves in the space-time cluster tree and the corresponding inadmissible blocks cannot be subdivided any further. This would correspond to a version of Algorithm 2, where Z_{src} is added to the nearfield $\mathcal{N}(Z_{\text{tar}})$ of Z_{tar} not only in line 23, but also in lines 13 and 20. In this work we allow for a further subdivision and more efficient handling of certain blocks by introducing the purely spatial subdivisions of clusters in the construction of \mathcal{T}_Σ , the temporally one-sided expansions (10) and (11), and the new routines in Algorithm 3. Note that the description of the interaction lists differs in the previous works [17, 23, 25]. Here we use a more algorithmic description.

4.3 Approximation of matrix blocks

In this section we describe how to use the kernel expansions (8), (10) and (11) to efficiently approximate the admissible sub-blocks of V_h from the partition constructed in Section 4.2. This

will lead to the adaptive FMM algorithm for the matrix-vector multiplication $\mathbf{f} = \nabla_h \mathbf{q}$, which we will present in Section 4.5.

Let us consider the application of a block $\nabla_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ to the appropriate part $\mathbf{q}|_{\hat{Z}_{\text{src}}}$ of a vector \mathbf{q} . The result is denoted by $\tilde{\mathbf{f}}|_{\hat{Z}_{\text{tar}}}$ and is given by

$$\tilde{f}_{k_t, k_x} = \sum_{(j_t, j_x) \in \hat{Z}_{\text{src}}} q_{j_t, j_x} \int_{t_{k_t-1}}^{t_{k_t}} \int_{\gamma_{k_x}} \int_{t_{j_t-1}}^{t_{j_t}} \int_{\gamma_{j_x}} G_\alpha(\mathbf{x} - \mathbf{y}, t - \tau) d\mathbf{s}_y d\tau d\mathbf{s}_x dt \quad (33)$$

for all indices $(k_t, k_x) \in \hat{Z}_{\text{tar}}$. Recall that we use pairs (k_t, k_x) as indices for vectors, see Section 2. We can approximate this operation in an efficient way if Z_{src} is contained in the standard interaction list $\mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$ or one of the new lists $\mathcal{I}_{\text{Mx2L}}(Z_{\text{tar}})$ or $\mathcal{I}_{\text{M2Lx}}(Z_{\text{tar}})$. The actual approximations depend on the particular list and are discussed in the following. Most of the occurring quantities in this discussion have been introduced in Section 3.

Case 1: Let $Z_{\text{src}} \in \mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$. We replace the heat kernel in (33) by the full space-time expansion in (8). This is the standard approximation from [17, 22, 23]. As in [25, Section 3.4] we compute the result in three steps:

S2M: For $a \in \{0, \dots, m_t\}$ and a multi-index $\boldsymbol{\kappa} \in \mathbb{N}_0^3$ with $|\boldsymbol{\kappa}| \leq m_x$ compute the *moments* $\boldsymbol{\mu}(Z_{\text{src}})$ by

$$\mu_{a, \boldsymbol{\kappa}}(Z_{\text{src}}) = \sum_{(j_t, j_x) \in \hat{Z}_{\text{src}}} q_{j_t, j_x} \int_{t_{j_t-1}}^{t_{j_t}} \int_{\gamma_{j_x}} T_{X_{\text{src}}, \boldsymbol{\kappa}}(\mathbf{y}) L_{I_{\text{src}}, a}(\tau) d\mathbf{s}_y d\tau. \quad (34)$$

M2L: For $b \in \{0, \dots, m_t\}$ and $\boldsymbol{\nu} \in \mathbb{N}_0^3$ with $|\boldsymbol{\nu}| \leq m_x$ compute the related *local contributions* $\boldsymbol{\lambda}(Z_{\text{tar}})$ by

$$\lambda_{b, \boldsymbol{\nu}}(Z_{\text{tar}}) = \sum_{a=0}^{m_t} \sum_{|\boldsymbol{\kappa} + \boldsymbol{\nu}| \leq m_x} E_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^{a, b} \mu_{a, \boldsymbol{\kappa}}(Z_{\text{src}}). \quad (35)$$

L2T: For all (k_t, k_x) in \hat{Z}_{tar} evaluate

$$\tilde{f}_{k_t, k_x} \approx \sum_{b=0}^{m_t} \sum_{|\boldsymbol{\nu}| \leq m_x} \lambda_{b, \boldsymbol{\nu}}(Z_{\text{tar}}) \int_{t_{k_t-1}}^{t_{k_t}} \int_{\gamma_{k_x}} T_{X_{\text{tar}}, \boldsymbol{\nu}}(\mathbf{x}) L_{I_{\text{tar}}, b}(t) d\mathbf{s}_x dt. \quad (36)$$

Case 2: Let $Z_{\text{src}} \in \mathcal{I}_{\text{M2Lx}}(Z_{\text{tar}})$. By construction, Z_{tar} is temporally indivisible in this case and, therefore, \hat{Z}_{tar} contains only a single time index k_t . We use the temporally one-sided approximation (10) to replace the heat kernel in (33) and get

$$\tilde{f}_{k_t, k_x} \approx \int_{t_{k_t-1}}^{t_{k_t}} \int_{\gamma_{k_x}} \sum_{a=0}^{m_t} \sum_{|\boldsymbol{\kappa} + \boldsymbol{\nu}| \leq m_x} E_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^a(t) T_{X_{\text{tar}}, \boldsymbol{\nu}}(\mathbf{x}) d\mathbf{s}_x dt \mu_{a, \boldsymbol{\kappa}}(Z_{\text{src}}), \quad (37)$$

where the moments $\boldsymbol{\mu}(Z_{\text{src}})$ are the same as in (34). The temporal integral in this equation is evaluated by using a Gauß–Legendre quadrature rule with $\rho_t + 1$ points, i.e.

$$\int_{t_{k_t-1}}^{t_{k_t}} E_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^a(t) dt \approx \sum_{b=0}^{\rho_t} \omega_{k_t, b} E_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^a(\zeta_{k_t, b}), \quad (38)$$

where $\{\zeta_{k_t, b}\}_{b=0}^{\rho_t}$ and $\{\omega_{k_t, b}\}_{b=0}^{\rho_t}$ are the Gauß–Legendre quadrature points and weights on the time interval (t_{k_t-1}, t_{k_t}) . The result in (33) is then approximated in three steps:

S2M: For $a \in \{0, \dots, m_t\}$ and $\kappa \in \mathbb{N}_0^3$ with $|\kappa| \leq m_x$ compute the moments $\mu(Z_{\text{src}})$ as in (34).
M2Lx: For $\nu \in \mathbb{N}_0^3$ with $|\nu| \leq m_x$ compute the related *spatial local contributions* $\lambda^{(x)}(Z_{\text{tar}})$ by

$$\lambda_{\nu}^{(x)}(Z_{\text{tar}}) = \sum_{b=0}^{\rho_t} \omega_{k_t, b} \sum_{|\kappa+\nu| \leq m_x} \sum_{a=0}^{m_t} E_{\kappa, \nu}^a(\zeta_{k_t, b}) \mu_{a, \kappa}(Z_{\text{src}}). \quad (39)$$

Lx2T: For all (k_t, k_x) in \hat{Z}_{tar} evaluate

$$\tilde{f}_{k_t, k_x} \approx \sum_{|\nu| \leq m_x} \int_{\gamma_{k_x}} T_{X_{\text{tar}}, \nu}(\mathbf{x}) d\mathbf{s}_x \lambda_{\nu}^{(x)}(Z_{\text{tar}}). \quad (40)$$

Note that the integral over the temporal target interval (t_{k_t-1}, t_{k_t}) is not part of the Lx2T operation but has moved to the M2Lx operation. The spatial local contributions $\lambda^{(x)}(Z_{\text{tar}})$ in (39) and (40) depend explicitly on the time interval (t_{k_t-1}, t_{k_t}) , but since Z_{tar} contains only a single time interval, we do not indicate this in the notation.

Case 3: Let $Z_{\text{src}} \in \mathcal{I}_{\text{Mx2L}}(Z_{\text{tar}})$. In this case Z_{src} is temporally indivisible and \hat{Z}_{src} contains only a single time index j_t . We approximate (33) in three steps by substituting the heat kernel with the temporally one-sided expansion in (11) and by proceeding similarly as in the last case:
S2Mx: For $\kappa \in \mathbb{N}_0^3$ with $|\kappa| \leq m_x$ compute the spatial moments $\mu^{(x)}(Z_{\text{src}})$ by

$$\mu_{\kappa}^{(x)}(Z_{\text{src}}) = \sum_{(j_t, j_x) \in \hat{Z}_{\text{src}}} q_{j_t, j_x} \int_{\gamma_{j_x}} T_{X_{\text{src}}, \kappa}(\mathbf{y}) d\mathbf{s}_y. \quad (41)$$

Mx2L: For $b \in \{0, \dots, m_t\}$ and $\nu \in \mathbb{N}_0^3$ with $|\nu| \leq m_x$ compute the related local contributions $\lambda_b(Z_{\text{tar}})$

$$\lambda_{b, \nu}(Z_{\text{tar}}) = \sum_{a=0}^{\rho_t} \sum_{|\kappa+\nu| \leq m_x} \omega_{j_t, a} E_{\kappa, \nu}^b(\zeta_{j_t, a}) \mu_{\kappa}^{(x)}(Z_{\text{src}}), \quad (42)$$

where $\{\zeta_{j_t, a}\}_{a=0}^{\rho_t}$ and $\{\omega_{j_t, a}\}_{a=0}^{\rho_t}$ are Gauß-Legendre quadrature points and weights on the time interval (t_{j_t-1}, t_{j_t}) .

L2T: For all $(k_t, k_x) \in \hat{Z}_{\text{tar}}$ evaluate the local contributions as in (36). Note that the integral over the temporal source interval (t_{j_t-1}, t_{j_t}) is not part of the S2Mx operation but is included in the Mx2L operation. The spatial moments $\mu^{(x)}(Z_{\text{src}})$ in (41) and (42) depend explicitly on this time interval (t_{j_t-1}, t_{j_t}) .

The FMM operations above are obtained by replacing the heat kernel in (33) with a suitable expansion and by reordering the sums. The related approximation error can be deduced from the results in Section 3.3, see e.g. [19, Section 4.1]. Operations like S2M in (34) and L2T in (36) involve integrals of polynomials, which can be computed exactly by suitable quadrature formulae. In the cases 2 and 3 above we additionally evaluate the integrals of the expansion coefficients $E_{\kappa, \nu}^a(\cdot_t)$ and $E_{\kappa, \nu}^b(\cdot_\tau)$ over the given time intervals by some quadrature formula, see (38). It is not surprising that the resulting operations are similar to those in case 1, since the additional interpolation in that case can be interpreted as an alternative quadrature formula. However, we can choose the quadrature formula in (38) freely and adapt it for individual time intervals to obtain a better accuracy. The results in Section 3.1 indicate that this is necessary in those situations, where we use the temporally one-sided expansions. Alternatively, one could evaluate the integrals of the expansion coefficients $E_{\kappa, \nu}^a(\cdot_t)$ and $E_{\kappa, \nu}^b(\cdot_\tau)$ analytically. However, this would destroy the product structure of the coefficients and prohibit an efficient handling as in [24, Section 4.3], on which we will further comment in Section 4.6.

4.4 A nested computation of moments and local contributions

To improve the efficiency of the FMM we use a nested computation of moments and local contributions. This is a standard approach in FMM algorithms and was used already in [17, 22, 23, 25] to compute the moments of clusters Z_{src} from the moments of their children and pass local contributions of clusters Z_{tar} down to their children to evaluate them in a single effort. The related operations are based on a suitable change of basis of the temporal Lagrange polynomials and spatial Chebyshev polynomials, so they do not introduce any additional approximation errors. The newly introduced spatial moments and local contributions can be computed in a nested manner as well. In addition, the spatial moments $\mu^{(\mathbf{x})}(Z_{\text{src}})$ can be used to compute standard moments $\mu(Z_{\text{src}})$, if both are needed. Similarly, local contributions $\lambda(Z_{\text{tar}})$ can be transformed into spatial local contributions $\lambda^{(\mathbf{x})}(Z_{\text{tar}})$ and evaluated together. In the following we list all related FMM operations:

Mx2Mx: For a temporally indivisible cluster $Z = X \times I$ with spatially refined children the spatial moments $\mu^{(\mathbf{x})}(Z)$ can be computed from the spatial moments of its children by

$$\mu_{\boldsymbol{\nu}}^{(\mathbf{x})}(Z) = \sum_{\substack{Z_c \in \text{child}(Z) \\ Z_c = X_c \times I}} \sum_{\boldsymbol{\kappa} \leq \boldsymbol{\nu}} q_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^{(\mathbf{x})}(X_c, X) \mu_{\boldsymbol{\kappa}}^{(\mathbf{x})}(Z_c) \quad (43)$$

for all $\boldsymbol{\nu} \in \mathbb{N}_0^3$ with $|\boldsymbol{\nu}| \leq m_x$. Here $\boldsymbol{\kappa} \leq \boldsymbol{\nu}$ is understood componentwise and for the boxes $X = [\mathbf{a}, \mathbf{b}]$ and $X_c = [\mathbf{c}, \mathbf{d}]$ we define $q_{\boldsymbol{\kappa}, \boldsymbol{\nu}}^{(\mathbf{x})}(X_c, X) := \prod_j q_{\boldsymbol{\kappa}_j, \boldsymbol{\nu}_j}^{(\mathbf{x})}((c_j, d_j], (a_j, b_j])$ with

$$q_{\boldsymbol{\kappa}_j, \boldsymbol{\nu}_j}^{(\mathbf{x})}((c_j, d_j], (a_j, b_j]) := \frac{\lambda_{\boldsymbol{\kappa}_j}}{m_x + 1} \sum_{n=0}^{m_x} T_{(a_j, b_j], \boldsymbol{\nu}_j}(\xi_{(c_j, d_j], n}^{(m_x)}) T_{(c_j, d_j], \boldsymbol{\kappa}_j}(\xi_{(c_j, d_j], n}^{(m_x)}),$$

where $\lambda_{\boldsymbol{\kappa}_j}$ is defined as in (9c) and $\{\xi_{(c_j, d_j], n}^{(m_x)}\}_{n=0}^{m_x}$ are Chebyshev nodes on the intervals $(c_j, d_j]$ of X_c .

Mx2M: The spatial moments $\mu^{(\mathbf{x})}(Z)$ of a cluster $Z = X \times I$ given in (41) can be transformed into the standard moments $\mu(Z)$ in (34) via

$$\mu_{a, \boldsymbol{\kappa}}(Z) = \mu_{\boldsymbol{\kappa}}^{(\mathbf{x})}(Z) \int_{t_{j_t-1}}^{t_{j_t}} L_{I, a}(\tau) d\tau \quad (44)$$

for all $a \in \{0, \dots, m_t\}$ and $\boldsymbol{\kappa} \in \mathbb{N}_0^3$ with $|\boldsymbol{\kappa}| \leq m_x$, where (t_{j_t-1}, t_{j_t}) is the only time interval contained in the temporally indivisible cluster Z . Note that the conversion of the spatial moments $\mu^{(\mathbf{x})}(Z)$ of Z to the standard moments $\mu(Z)$ has to be executed only if Z is a temporally indivisible cluster whose parent is not temporally indivisible. The standard moments of all other temporally indivisible clusters are not needed.

M2M: The operations used to compute the moments $\mu(Z)$ of a cluster Z from the moments of its children are the same as in the original works, cf. [22, Sections 4.2 and 4.3]. Based on the subdivision used to construct the children one distinguishes two operations:

Temporal M2M: If the children of $Z = X \times I$ were created by a purely temporal subdivision

$$\mu_{b, \boldsymbol{\kappa}}(Z) = \sum_{\substack{Z_c \in \text{child}(Z) \\ Z_c = X \times I_c}} \sum_{a=0}^{m_t} q_{a, b}^{(t)}(I_c, I) \mu_{a, \boldsymbol{\kappa}}(Z_c) \quad (45)$$

for all $b \in \{0, \dots, m_t\}$ and $\kappa \in \mathbb{N}_0^3$ with $|\kappa| \leq m_x$, where $q_{a,b}^{(t)}(I_c, I) := L_{I,b}(\xi_{I_c,a}^{(m_t)})$.

Space-time M2M: If the children of $Z = X \times I$ were created by a space-time subdivision

$$\mu_{b,\nu}(Z) = \sum_{\substack{Z_c \in \text{child}(Z) \\ Z_c = X_c \times I_c}} \sum_{a=0}^{m_t} \sum_{\kappa \leq \nu} q_{a,b}^{(t)}(I_c, I) q_{\kappa,\nu}^{(x)}(X_c, X) \mu_{a,\kappa}(Z_c) \quad (46)$$

for all $b \in \{0, \dots, m_t\}$ and $\nu \in \mathbb{N}_0^3$ with $|\nu| \leq m_x$, where the coefficients $q_{\kappa,\nu}^{(x)}(X_c, X)$ and $q_{a,b}^{(t)}(I_c, I)$ are the same as in (43) and (45), respectively.

Temporal L2L operations and *space-time L2L* operations are used to pass down local contributions of a cluster Z_{tar} to its children. They are the transposed operations of the corresponding M2M operations. For a temporally indivisible cluster Z_{tar} whose parent is not temporally indivisible the local contributions $\lambda(Z_{\text{tar}})$ are transformed into spatial local contributions $\lambda^{(x)}(Z_{\text{tar}})$ with an $L2Lx$ operation. If Z_{tar} is not a leaf, its spatial local contributions are passed down to its children by $Lx2Lx$ operations. The L2Lx and Lx2Lx operations are the transposed operations of the Mx2M and Mx2Mx operations in (44) and (43), respectively.

4.5 A time-adaptive space-time FMM for the heat equation

The full time-adaptive space-time FMM algorithm is presented in Algorithm 4. It can be described as an efficient approximation of the matrix-vector product $V_h \mathbf{q}$ in a blockwise manner. The blocks are given by the partition constructed by Algorithm 2. Admissible blocks are approximated in the FMM as described in Section 4.3. In addition, the nested computation of moments and local contributions from Section 4.4 is used. Note that the moments $\mu(Z_{\text{src}})$ of a cluster Z_{src} have to be computed only once and can then be used for all M2L or M2Lx operations of this cluster. The same is true for spatial moments $\mu^{(x)}(Z_{\text{src}})$ and Mx2L operations. The moments of all clusters in \mathcal{T}_{Σ} are computed in the first phase of the FMM (lines 2–17), where an explicit distinction and transition between spatial moments and standard moments takes place. After this phase all moments and spatial moments are available, so all M2L, Mx2L and M2Lx operations can be executed (lines 18–25). The resulting local contributions and spatial local contributions of each target cluster are summed up. They are passed downwards and evaluated in a single effort for the leaves of \mathcal{T}_{Σ} (lines 26–39). Again, explicit transitions between local contributions and their spatial counterparts take place when necessary. In the last phase, the inadmissible blocks are applied directly (lines 40–42).

The new time-adaptive space-time FMM algorithm can be seen as an enhancement of the original one in [22], which is presented in a similar form in [25, Algorithm 2]. The additional subdivision of formerly inadmissible blocks by the routines in Algorithm 3 and the new FMM operations related to purely spatial moments and local contributions allow for a more efficient treatment of space-time tensor product meshes which are adaptive in time. In Sections 4.6 and 4.7 we will discuss the costs for the execution of individual FMM operations and, in particular, analyze the compression obtained by the new operations. The benefits of the time-adaptive space-time FMM will be shown in the numerical examples in Section 5.

We suppose that the concept of the temporal nearfield approximation in [18] can be extended to the setting of adaptive meshes in time with some technical difficulties. This would allow for a compression of the formerly inadmissible blocks as well. However, our approach appears to fit into the present concept more easily.

In [25] the authors presented a parallel version of the standard space-time FMM for the heat equation for distributed memory systems. The parallel algorithm exploits the temporal

Algorithm 4 Time-adaptive space-time FMM for the approximate evaluation of $\mathbf{f} = \nabla_h \mathbf{q}$.

Require: Let a space-time box cluster tree \mathcal{T}_Σ as in Algorithm 1 be given.

Let the interaction lists for the matrix partition be constructed by Algorithm 2.

Let expansion degrees m_t and m_x and a parameter ρ_t (for (38)) be given.

```

1: Initialize  $\mathbf{f} = \mathbf{0}$ .
2: for all leaves  $Z \in \mathcal{T}_\Sigma$       // Forward transformation
3:   if  $Z$  is temporally indivisible:
4:     Compute  $\boldsymbol{\mu}^{(x)}(Z)$  by S2Mx (41).
5:   if  $\text{par}(Z)$  is not temporally indivisible:
6:     Compute  $\boldsymbol{\mu}(Z)$  by Mx2M (44).
7:   else: Compute  $\boldsymbol{\mu}(Z)$  by S2M (34).
8: for all levels  $\ell = \text{depth}(\mathcal{T}_\Sigma) - 1, \dots, 2$ 
9:   for all non-leaf boxes  $Z \in \mathcal{T}_\Sigma$  with  $\text{level}(Z) == \ell$ 
10:    if  $Z$  is temporally indivisible:
11:      Compute  $\boldsymbol{\mu}^{(x)}(Z)$  by Mx2Mx (43).
12:    if  $\text{par}(Z)$  is not temporally indivisible:
13:      Compute  $\boldsymbol{\mu}(Z)$  by Mx2M (44).
14:    else
15:      if children of  $Z$  are refined only in time:
16:        Compute  $\boldsymbol{\mu}(Z)$  by temporal M2M (45).
17:      else: Compute  $\boldsymbol{\mu}(Z)$  by space-time M2M (46).
18: for all boxes  $Z_{\text{tar}} \in \mathcal{T}_\Sigma$       // Multiplication phase
19:   Initialize  $\boldsymbol{\lambda}(Z_{\text{tar}}) = \mathbf{0}$  and  $\boldsymbol{\lambda}^{(x)}(Z_{\text{tar}}) = \mathbf{0}$  if necessary.
20:   for all boxes  $Z_{\text{src}} \in \mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$ 
21:     M2L: Update  $\boldsymbol{\lambda}(Z_{\text{tar}})$  by adding the result from (35).
22:   for all boxes  $Z_{\text{src}} \in \mathcal{I}_{\text{Mx2L}}(Z_{\text{tar}})$ 
23:     Mx2L: Update  $\boldsymbol{\lambda}(Z_{\text{tar}})$  by adding the result from (42).
24:   for all boxes  $Z_{\text{src}} \in \mathcal{I}_{\text{M2Lx}}(Z_{\text{tar}})$ 
25:     M2Lx: Update  $\boldsymbol{\lambda}^{(x)}(Z_{\text{tar}})$  by adding the result from (39).
26: for all levels  $\ell = 3, \dots, \text{depth}(\mathcal{T}_\Sigma)$       // Backward transformation
27:   for all boxes  $Z \in \mathcal{T}_\Sigma$  with  $\text{level}(Z) == \ell$ 
28:     if  $\text{par}(Z)$  is temporally indivisible:
29:       Update  $\boldsymbol{\lambda}^{(x)}(Z)$  by Lx2Lx using  $\boldsymbol{\lambda}^{(x)}(\text{par}(Z))$ .
30:     else
31:       if  $Z$  results from  $\text{par}(Z)$  by a temporal refinement:
32:         Update  $\boldsymbol{\lambda}(Z)$  by temporal L2L using  $\boldsymbol{\lambda}(\text{par}(Z))$ .
33:       else: Update  $\boldsymbol{\lambda}(Z)$  by space-time L2L using  $\boldsymbol{\lambda}(\text{par}(Z))$ .
34:       if  $Z$  is temporally indivisible:
35:         Update  $\boldsymbol{\lambda}^{(x)}(Z)$  by L2Lx using  $\boldsymbol{\lambda}(Z)$ .
36: for all leaves  $Z \in \mathcal{T}_\Sigma$ 
37:   if  $Z$  is temporally indivisible:
38:     Update  $\mathbf{f}|_{\hat{Z}}$  by Lx2T (40).
39:   else: Update  $\mathbf{f}|_{\hat{Z}}$  by L2T (36).
40: for all  $Z_{\text{tar}} \in \mathcal{T}_\Sigma$       // Nearfield evaluation
41:   for all  $Z_{\text{src}} \in \mathcal{N}(Z_{\text{tar}})$ 
42:     Nearfield operation: Update  $\mathbf{f}|_{\hat{Z}_{\text{tar}}} += \nabla_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}} \mathbf{q}|_{\hat{Z}_{\text{src}}}$ .

```

structure of the FMM, which allows to consider groups of operations on a temporal level. These groups of operations do not have to be executed in the strictly separated phases which we see in Algorithm 4, but can be executed efficiently based on individual dependencies. This flexible execution order enables an efficient parallelization in distributed memory. The distribution of work and communication between different processes can be handled on a temporal level, which makes it simple and efficient. In the parallel algorithm, collections of moments or local contributions of clusters are communicated between processes in a non-blocking way when they are ready and required. The new time-adaptive version allows for a similar treatment by introducing new groups of operations related to the new FMM operations (S2Mx, Mx2Mx, Mx2L, M2Lx, ...) and adjusting the dependencies of the grouped operations. The distribution of the work among computing nodes has to be adapted to the time-adaptive mesh.

4.6 Costs of individual FMM operations

In Table 1 we give an overview of the runtime complexity of all FMM operations. The complexity $\mathcal{O}((m_x + 1)^4(m_t + 1)^2)$ of the M2L operations is only achieved if they are executed as proposed in [24, Section 4.3]. A naive implementation would require $\mathcal{O}(\binom{m_x+3}{3}^2(m_t + 1)^2)$ operations instead. The same holds for the M2Lx and Mx2L operations, where a smart implementation requires $\mathcal{O}((m_x + 1)^4(m_t + 1)(\rho_t + 1))$ operations instead of $\mathcal{O}(\binom{m_x+3}{3}^2(m_t + 1)(\rho_t + 1))$ in the naive case. The complexity of all other FMM operations can be estimated by counting the number of multiplications in the respective equations (34)–(46) above, utilizing the tensor product structure of operations whenever possible. E.g., in case of the S2M operations one can use a decomposition into operations similar to S2Mx and Mx2M operations in (41) and (44). We observe that the costs of the newly introduced FMM operations are similar to those of the standard FMM operations.

The numbers in Table 1 reveal why the approximation of admissible blocks of V_h is efficient. Let us consider for example a block $V_h|_{\hat{Z}_{\text{tar}} \times \hat{Z}_{\text{src}}}$ corresponding to clusters Z_{tar} and Z_{src} in \mathcal{T}_Σ with $Z_{\text{src}} \in \mathcal{I}_{\text{M2L}}(Z_{\text{tar}})$. The computation of the moments $\mu(Z_{\text{src}})$ and evaluation of the local contributions $\lambda(Z_{\text{tar}})$ has to be done only once for each cluster, so it does not have to be repeated for every admissible block. Thus, we can focus on the M2L operations when estimating the costs of the application of such an admissible block in the FMM algorithm. These costs are $\mathcal{O}((m_x + 1)^4(m_t + 1)^2)$. In comparison, a direct application of the non-approximated block would

Table 1: Runtime complexities of all FMM operations in the time-adaptive space-time FMM for the heat equation. The cluster Z appearing in the estimated costs for S2M, L2T, S2Mx and Lx2T operations is the related source or target cluster.

Operations	Runtime complexity
S2M and L2T	$\mathcal{O}(\#\hat{Z}\binom{m_x+3}{3} + n_t(\hat{Z})\binom{m_x+3}{3}(m_t + 1))$
Purely temporal M2M and L2L	$\mathcal{O}(\binom{m_x+3}{3}(m_t + 1)^2)$
Space-time M2M and L2L	$\mathcal{O}(\binom{m_x+3}{3}(m_t + 1)(\frac{m_x}{4} + (m_t + 1)))$
M2L	$\mathcal{O}((m_x + 1)^4(m_t + 1)^2)$
S2Mx and Lx2T	$\mathcal{O}(\#\hat{Z}\binom{m_x+3}{3})$
Mx2Mx and Lx2Lx	$\mathcal{O}(\binom{m_x+3}{3}\frac{m_x}{4})$
Mx2M and L2Lx	$\mathcal{O}(\binom{m_x+3}{3}(m_t + 1))$
M2Lx and Mx2L	$\mathcal{O}((m_x + 1)^4(m_t + 1)(\rho_t + 1))$

require $\mathcal{O}(\#\hat{Z}_{\text{tar}} \#\hat{Z}_{\text{src}})$ operations. Hence, the low rank approximation is more efficient in terms of the execution time, if Z_{tar} and Z_{src} contain more than $\mathcal{O}((m_x + 1)^2(m_t + 1))$ elements.

In terms of storage requirements, the approximation of blocks in the FMM is beneficial as well. In fact, we do not store the coefficients of the FMM operations in our implementation, but only the moments and local contributions for each cluster – at most one of each kind per cluster. Since the computation of nearfield entries is rather expensive, we fully assemble and store all inadmissible blocks instead. Therefore, the required storage for admissible blocks is negligibly small compared to inadmissible blocks.

4.7 Complexity analysis for newly approximated blocks

A complete complexity analysis of the presented time-adaptive FMM highly depends on the specific adaptive decomposition of the time interval $(0, T)$ and would require some related restrictive assumptions. Instead we present a comparison of the standard and the time-adaptive FMM when the latter one provides some additional compression.

We consider two time cluster intervals I_{src} and I_{tar} which are inadmissible with respect to the original criterion (16) but admissible with respect to the adaptive criterion (20). We assume that the interval I_{tar} and all related space-time clusters $Z_{\text{tar}} = X_{\text{tar}} \times I_{\text{tar}}$, respectively, are temporally indivisible. Note that the situation related to interchanged roles of I_{src} and I_{tar} is similar to the considered one due to the symmetric character of operations like $Mx2L$ and $M2Lx$. In addition we assume that the spatial mesh Γ_h with E_x similarly sized elements is sufficiently fine and that the spatial clustering resolves the surface.

We start with the complexity analysis of the matrix blocks related to the considered two time intervals for the new time-adaptive version of the FMM. In this setting $M2Lx$ operations are applied for related clusters Z_{tar} and $Z_{\text{src}} = X_{\text{src}} \times I_{\text{src}}$. The combined effort of the two operations $L2Lx$ and $Lx2T$ is the same as the effort of the $L2T$ operation for the same cluster. Therefore, these two operations do not create additional effort in the overall algorithm. Some additional effort is created by the $Lx2Lx$ operations on the spatially refined clusters. A single $Lx2Lx$ operation is rather inexpensive and there is at most one operation per cluster. Compared to the effort of $M2L$ and $M2Lx$ operations the effort of all $Lx2Lx$ operations is small. There are not any related $S2Mx$, $Mx2Mx$, and $Mx2M$ operations in this part of the FMM algorithm. Thus we neglect the effort of all these operations in this analysis and focus on the effort of the related $M2Lx$ operations.

The effort of a $M2Lx$ operation between the considered clusters Z_{src} and Z_{tar} is given in Table 1. This operation is applied for all clusters $Z_{\text{src}} \in \mathcal{I}_{M2Lx}(Z_{\text{tar}})$, see the lines 24f in Algorithm 4. Thus, we have to estimate their number $\#\mathcal{I}_{M2Lx}(Z_{\text{tar}})$. The list $\mathcal{I}_{M2Lx}(Z_{\text{tar}})$ is filled in line 26 of Algorithm 3 and affected by the truncation strategy described in Section 4.2. But the truncation is implemented in line 3 of Algorithm 2. At that point the interaction area \mathcal{I}_A in (32) typically contains $(2n_{\text{tr}} + 1)^2$ non-empty boxes of the regular grid on the surface of the domain. Due to d_x additional spatial descending steps according to lines 18–23 in Algorithm 3, their number increases by the factor 4^{d_x} approximately. Thus,

$$\#\mathcal{I}_{M2Lx}(Z_{\text{tar}}) \approx (2n_{\text{tr}} + 1)^2 4^{d_x}. \quad (47)$$

By taking the sum over all clusters Z_{tar} related to I_{tar} , we can estimate the computational effort of the time-adaptive FMM related to the two time intervals by

$$C_{\text{new}} \approx \#\mathcal{B}^{\ell_x} (2n_{\text{tr}} + 1)^2 4^{d_x} \mathcal{O}((m_x + 1)^4(m_t + 1)(\rho_t + 1)), \quad (48)$$

where $\#\mathcal{B}^{\ell_x}$ is the number of non-empty spatial boxes of the regular grid \mathcal{G}^{ℓ_x} on the spatial level ℓ_x of Z_{tar} and Z_{src} .

In the standard FMM the related blocks are inadmissible and, thus, directly assembled and stored as dense blocks. Again we apply the truncation strategy described in Section 4.2. We have already counted the number of clusters Z_{src} relevant for Z_{tar} in (47). If we take the sum over all clusters Z_{tar} related to I_{tar} , which contain all E_x spatial elements, we can estimate the number of nearfield entries related to the two time cluster intervals as

$$C_{\text{old}} \approx E_x(2n_{\text{tr}} + 1)^2 4^{d_x} n_{\ell_x} n_t, \quad (49)$$

where n_{ℓ_x} is the average number of elements in the non-empty spatial boxes \mathcal{B}^{ℓ_x} of the regular grid \mathcal{G}^{ℓ_x} , and n_t is the number of time intervals in the cluster interval I_{src} and Z_{src} , respectively.

Typically we will have a small integer d_x for locally quasi-uniform temporal meshes. But n_{ℓ_x} may be large for temporally indivisible time intervals. Thus, the number C_{old} of nearfield entries of the considered matrix block may be large or even $n_t E_x^2$. The new version reduces the memory requirements and the computational time significantly. C_{old} nearfield entries do not have to be computed and stored. Instead we only need to store an additional expansion per cluster which typically requires much less memory.

For the execution time of a matrix vector product the comparison is more involved. In the original version the related effort is proportional to C_{old} . The effort C_{new} of the new method in (48) depends on the number of M2Lx operations, i.e. on the specific setting of the two time intervals I_{src} and I_{tar} . Since $\#\mathcal{B}^{\ell_x} = E_x/n_{\ell_x}$, the new method will provide a faster application of the related blocks, if $C_{\text{new}} < C_{\text{old}}$, i.e.

$$\mathcal{O}((m_x + 1)^4(m_t + 1)(\rho_t + 1)) < n_{\ell_x}^2 n_t.$$

This is the case if the number n_t of time intervals in I_{src} and the average number n_{ℓ_x} of spatial elements per cluster on the spatial level ℓ_x of Z_{src} and Z_{tar} are large. Thus, the new method is advantageous for fine spatial meshes, in general. As a rule of thumb: If the operations take place on a relatively large spatial level ℓ_x , n_{ℓ_x} is small and the original nearfield matrix vector product is faster. If the operations take place on a relatively small spatial level ℓ_x , n_{ℓ_x} is large and the new FMM version will be faster. Due to the construction of the cluster tree and the operation lists $\mathcal{I}_{\text{M2Lx}}$, the relevant spatial level ℓ_x is related to the cluster levels of the two time intervals I_{src} and I_{tar} .

5 Numerical experiments

The time-adaptive FMM algorithm presented and analyzed in the previous sections has been implemented in the publicly available C++ library besthea [16]. The implementation provides a hybrid OpenMP-MPI parallelization of Algorithm 4 following the ideas in [25]. In this section we consider numerical experiments to show the benefits of the newly introduced time-adaptive FMM operations. The experiments in Section 5.1 were carried out using a local workstation with 384 GiB of RAM and two 16-core Intel Xeon Gold 5218 processors. For the compilation we used the Intel compiler v2021.2.0. For the experiments in Section 5.2 we used the VSC-4 cluster in Vienna, Austria, and the Intel compiler v19.1.3.304. The nodes of VSC-4 are equipped with two 24-core Intel Skylake Platinum 8174 processors. We used standard nodes providing 96 GB of RAM and fat nodes, providing 384 GB of RAM, when necessary. A guide to reproduce the experiments is given in the repository [16].

5.1 Example 1: Exponential decay in time

In the first example we use a direct boundary integral approach to solve the initial boundary value problem (1) of the heat equation with the heat capacity constant $\alpha = 1$ in the space-time cylinder $Q = \Omega \times (0, T)$, where Ω is the cube $(-0.5, 0.5)^3$ and $T = 0.25$. We choose the exact solution of the heat equation

$$u(\mathbf{x}, t) = \exp(-3\pi^2 t) \prod_{j=1}^3 \sin(\pi(x_j + 0.5)), \quad \text{for all } (\mathbf{x}, t) \in Q \quad (50)$$

with the related initial and Dirichlet data as well as Neumann datum q . We solve the integral equation (2) for q with the right-hand side f in (4), which simplifies to $f = -M_0 u_0$ since $g = 0$. Due to the characteristic exponential decay in time of the solution u in (50) it is reasonable to increase the time step size as time advances. For our numerical experiment we choose the adaptive temporal mesh \mathcal{I}_h illustrated in Figure 1. For a better presentation we depict the lengths of the intervals. This mesh was constructed by an adaptive refinement algorithm that was tailored to reduce the L^2 projection error of the temporal part $q_t(t) = \exp(-3\pi^2 t)$ of the exact solution (50). For the spatial mesh Γ_h we use a uniform triangulation of the cube's surface into 12 288 triangles. The resulting space-time tensor product mesh $\Sigma_h = \Gamma_h \otimes \mathcal{I}_h$ consists of 245 760 space-time boundary elements. In addition, a volume mesh of 196 608 tetrahedra conforming to Γ_h is used for the initial potential.

We want to compare the original FMM as implemented in [25] and our new time-adaptive variant when solving the discrete system $\nabla_h \mathbf{q} = \mathbf{f}$, where \mathbf{f} is a discretization of the right-hand side $-M_0 u_0$. For the evaluation of the initial potential we use analytical integration in time and numerical quadrature formulae for triangles and tetrahedra in space. In singular cases we apply a recursive subdivision routine of volume and surface elements. Our results indicate that we are able to compute the right-hand side \mathbf{f} accurately enough in this way. For our time-adaptive FMM we use the cluster parameters $n_{\max} = 800$ and $c_{\text{st}} = 4.5$, the spatial truncation parameter $n_{\text{tr}} = 2$, the constants $\eta_2 = 1$ and $\eta_1 = 1$ for the admissibility criteria (16) and (20), respectively, and the expansion degrees $m_t = 4$ and $m_x = 12$. For the numerical quadrature in (38) we use $\varrho_t = 3$, i.e. a 4-point rule, which proves to be sufficiently accurate in our examples. For the original version of the FMM we use the same parameters except η_1 and ϱ_t , which are not needed.

In Table 2 we compare the results when solving the system $\nabla_h \mathbf{q} = \mathbf{f}$ using the GMRES method with a diagonal preconditioner for the time-adaptive and the standard FMM. The relative

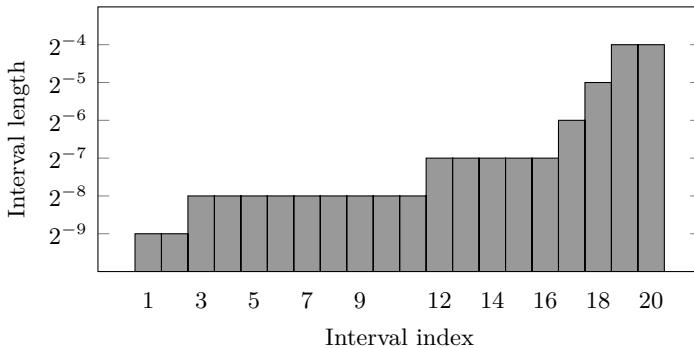


Figure 1: The time mesh \mathcal{I}_h consisting of 20 time steps in the time interval $(0, 0.25)$ adapted to $q_t(t) = \exp(-3\pi^2 t)$. For each interval its length is depicted.

Table 2: Solution of the system $\mathbf{V}_h \mathbf{q} = \mathbf{f}$ for the problem in Section 5.1 using the GMRES method with a diagonal preconditioner and the standard FMM and time-adaptive FMM for the matrix-vector multiplication.

	Std FMM	Time-adapt. FMM
Relative L^2 approximation error	0.0530951	0.0530951
Storage for the nearfield part of \mathbf{V}_h [GiB]	39.92	32.04
Storage for the moments and local contributions [GiB]	0.0211	0.0216
Time to assemble \mathbf{V}_h [s]	307.32	217.56
Time per GMRES iteration [s]	1.36	1.42
Total time (assembly of \mathbf{V}_h and solution) [s]	373.98	286.92

L^2 approximation errors $\|q - q_h\|_{L^2(\Sigma)} / \|q\|_{L^2(\Sigma)}$ in the first line coincide. This shows that the additional kernel approximations do not affect the accuracy of the time-adaptive FMM. In both versions, we store the nearfield matrices since their computation is costly. The required storage of 39.92 GiB for the nearfield matrices of the standard FMM was reduced by roughly 20% by introducing the new FMM operations. The storage required for the additional spatial local contributions is negligibly small as we can see from the values in the third line of the table.

In lines 4-6 of Table 2 we compare the execution times when using the standard and time-adaptive FMM. In both cases the GMRES method required 49 iterations to achieve a relative accuracy of 10^{-8} . Note that the total times do not include the computation of the right-hand side, which takes about 250 s by using an FMM. The time for a single GMRES iteration, which is dominated by the time needed for a matrix-vector multiplication, is slightly lower for the standard FMM in this example. This is not completely unexpected, since the application of nearfield matrix blocks is simple and efficient if they are already computed. The newly introduced Mx2L and M2Lx operations are more efficient than the related nearfield operations for source and target clusters containing many space-time boundary elements, but can be less efficient for clusters containing few elements, as we have seen in Section 4.7. The corresponding gains and losses seem to balance out in this example. The assembly of the nearfield matrices requires considerably more time than the GMRES solver and is significantly faster in the time-adaptive version, due to the reduced number of nearfield entries which have to be computed. Therefore, the new time-adaptive FMM outperforms the standard version in this example.

5.2 Example 2: Rapid change of boundary data

We study an example with boundary data that change rapidly over time. This is another scenario where adaptive temporal meshes are better suited than uniform meshes. We consider the function

$$q_t(t) = \frac{t}{4} + \left(t - \frac{1}{6}\right)^{1/4} \mathbb{1}_{(1/6, \infty)}(t), \quad t \in (0, T), \quad (51)$$

where $T = 0.25$, $\mathbb{1}_{(1/6, \infty)}(t) = 1$ for $t > 1/6$ and $\mathbb{1}_{(1/6, \infty)}(t) = 0$ for $t \leq 1/6$. The function q_t is depicted in Figure 2a. It is non-smooth at time $t = 1/6$ when the part $(t - 1/6)^{1/4}$ is turned on. A suitable adaptive mesh that resolves this event is advantageous for a piecewise constant approximation of this function. The mesh \mathcal{I}_h in Figure 2b was built for that purpose.

We consider a related indirect BEM, i.e. we want to find q such that $Vq = g$ for a given Dirichlet datum g and initial datum $u_0 = 0$ in the space-time cylinder $Q = \Omega \times (0, 0.25)$, where Ω is a crankshaft geometry whose surface is discretized by a mesh Γ_h consisting of 10 722 triangles. We

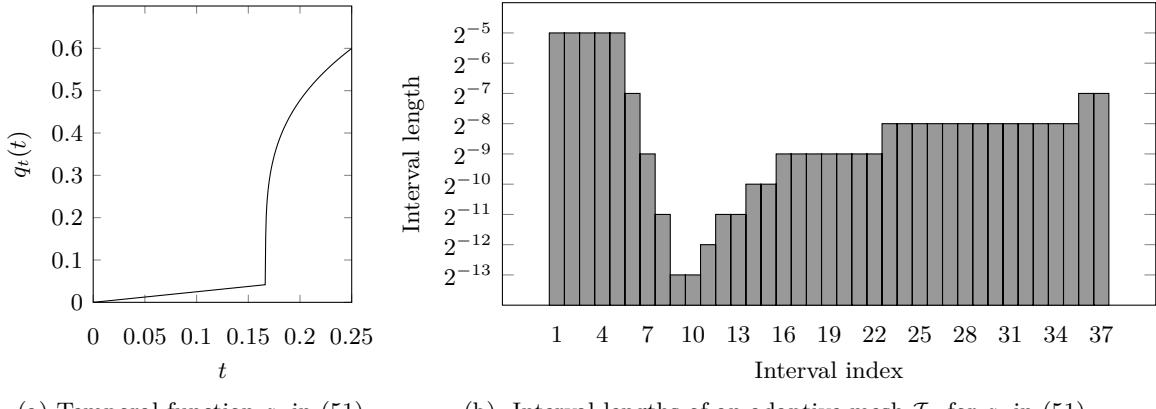


Figure 2: Visualization of the rapidly changing part q_t of q in (52) in the time interval $(0, 0.25)$ and a suitable mesh on this interval that resolves the non-smooth behavior of q_t at time $t = 1/6$ well.

set the exact solution q to

$$q(\mathbf{x}, t) = q_t(t)q_{\mathbf{x}}(\mathbf{x}), \text{ for all } (\mathbf{x}, t) \in \Sigma, \quad \text{with } q_{\mathbf{x}}(\mathbf{x}) = \frac{2n_1(\mathbf{x})x_1 - 3n_2(\mathbf{x})x_2 + n_3(\mathbf{x})x_3}{10} \quad (52)$$

and q_t in (51). The mesh \mathcal{I}_h in Figure 2b and the surface mesh Γ_h of the crankshaft are well-suited to approximate the solution q and we can check the approximation errors to ensure the correctness of our method. Since the right-hand side g is unknown, we will use a piecewise constant approximation \hat{g}_h on a finer mesh $\hat{\Sigma}_h$ (constructed from the original mesh $\Sigma_h = \Gamma_h \otimes \mathcal{I}_h$ by uniform refinements; once in space, twice in time) whose coefficients are computed by $\hat{\mathbf{g}} = \hat{\mathbf{V}}_h \hat{\mathbf{q}}$. Here, $\hat{\mathbf{q}}$ denotes the coefficient vector of the piecewise constant projection of q on $\hat{\Sigma}_h$ and $\hat{\mathbf{V}}_h$ is the single layer operator matrix on the fine mesh.

For all examples in this section, we choose the same FMM parameters as in Section 5.1. We compare the standard FMM and our new time-adaptive variant by solving the system $\mathbf{V}_h \mathbf{q} = \mathbf{g}$ on the space-time mesh $\Sigma_h = \Gamma_h \otimes \mathcal{I}_h$ that contains 396 714 space-time boundary elements. For each of the computations we use a single node of the VSC-4 cluster: a standard node for the time-adaptive FMM and a fat node for the standard FMM due to the higher memory demand. The results of the computations are given in Table 3. They are similar to the results in Section 5.1. The approximation quality does not suffer when introducing the new FMM operations, while the required storage for nearfield matrices and related assembly times are reduced significantly. In this example 98 GMRES iterations were needed in both cases, and a single iteration was faster for the time-adaptive FMM than for the standard FMM.

The level of adaptivity differs for the examples in this section and Section 5.1, as the constant related to locally quasi-uniform meshes differs. We have $\frac{t_{j+1}-t_j}{t_j-t_{j-1}} \leq 2$ for Example 1 and $\frac{t_j-t_{j-1}}{t_{j+1}-t_j} \leq 4$ for Example 2. While the grading of the mesh is stronger in Example 2, the reductions of the storage requirements and the computational times are slightly better. In general, the improvement of the new version highly depends on the specific setting, please see the discussion in Section 4.7.

Finally we compare the computations on the above time-adaptive mesh Σ_h to the computation on a suitable uniform mesh. We consider the space-time mesh $\Sigma_{h,u} = \Gamma_h \otimes \mathcal{I}_{h,u}$, where $\mathcal{I}_{h,u}$ is the uniform time mesh on $(0, T)$ consisting of 512 uniform time steps with length 2^{-11} . Note that $\Sigma_{h,u}$ contains 5 489 664 space-time boundary elements. The large number of uniform time steps in $\mathcal{I}_{h,u}$

is required to resolve the rapid change of q_t around the point $t = 1/6$ reasonably well. While the uniform mesh $\mathcal{I}_{h,u}$ is finer than the adaptive mesh \mathcal{I}_h in Figure 2b in general, it is coarser around $t = 1/6$. When projecting the function q_t to these two meshes the resulting L^2 approximation errors on $(0, T)$ are similar, but the error is slightly larger around time $t = 1/6$ for the uniform mesh. On the uniform mesh $\Sigma_{h,u}$ we solve the system $\nabla_{h,u} \mathbf{q}_u = \mathbf{g}_u$ with the standard FMM. Due to the large size of the problem, we use 16 standard nodes of VSC-4 for the computation and the parallel FMM in [25]. The results are given in Table 4.

When comparing the results in Tables 3 and 4 one notices the superior performance of the time-adaptive method over the standard method with the uniform time steps. Although we use 16 nodes of VSC-4 for the computations on the uniform mesh, the computational time is higher than the computational time of the adaptive time mesh on a single node, even when using the standard FMM on the latter. The higher computation time is caused by the larger number of GMRES iterations (153 instead of 98) required for the solution and the longer duration of a single matrix vector product.

6 Conclusions and Outlook

We have presented temporally one-sided expansions of the heat kernel and related error estimates. Based on these results we have developed new FMM expansions and operations as well as a new time-adaptive version of the FMM for the heat equation. Our theoretical considerations and the numerical examples demonstrate some substantial improvements of fast BEM calculations for meshes which are adaptive with respect to time.

While the new time-adaptive FMM reduces the amount of non-compressed nearfield blocks, some of the nearfield blocks related to the identical and the previous time cluster remain uncompressed. A sophisticated compression technique for such blocks is presented in [18] for uniform time steps. We plan to develop an alternative method for the non-uniform setting. This will significantly reduce the computational times and the memory requirements of our method once more.

To further speed up the nearfield assembly time, which makes up a significant amount of the total time in our numerical experiments, one could think about using GPUs for these computations. With such devices it might even be possible to compute the nearfield entries on the fly instead of storing them, which would lead to a drastic reduction of the storage requirements. Such an approach was considered for a standard BEM (i.e. without a fast method) in [13] and could be extended for the space-time FMM presented in this paper.

Table 3: Solution of the system $\nabla_h \mathbf{q} = \mathbf{g}$ for the problem in Section 5.2 and the time-adaptive mesh Σ_h . The system is solved using the GMRES method with a diagonal preconditioner and the standard FMM and time-adaptive FMM for the matrix-vector multiplication.

	Std FMM	Time-adapt. FMM
Relative L^2 approximation error	0.013129	0.013129
Storage for the nearfield part of ∇_h [GiB]	93.23	66.64
Storage for the moments and local contributions [GiB]	0.0281	0.0305
Time to assemble ∇_h [s]	495.34	302.27
Time per GMRES iteration [s]	1.27	1.16
Total time (assembly of ∇_h and solution) [s]	619.43	415.56

Table 4: Solution of the system $\nabla_{h,u} \mathbf{q}_u = \mathbf{g}_u$ for the problem in Section 5.2 and the mesh $\Sigma_{h,u}$ with 512 uniform time steps. The system is solved using the GMRES method with a diagonal preconditioner and the standard FMM using 16 nodes of VSC-4.

	StdFMM (16 nodes)
Relative L^2 approximation error	0.011851
Average storage for the nearfield part of ∇_h per node [GiB]	79.90
Average storage for moments and local contributions per node [GiB]	0.0467
Time to assemble ∇_h [s]	406.12
Time per GMRES iteration [s]	2.29
Total time (assembly of ∇_h and solution) [s]	756.40

The improvements of the FMM are an important step towards the development of efficient adaptive space-time boundary element methods for the heat equation. First results regarding such an adaptive BEM can be found in [8]. In that work a posteriori error estimates are introduced which drive the non-local mesh refinement of the proposed adaptive algorithm. This adaptive algorithm and the considered numerical examples cover only the spatially two dimensional case. When transferring the ideas to the spatially three dimensional case, a fast method is needed for the solution of the related systems because otherwise the problem size is prohibitively limited. A variant of our time-adaptive FMM adapted to meshes without tensor product structure might be suitable for such an application.

Acknowledgments

The authors acknowledge the support provided by the Austrian Science Fund (FWF) under the project I 4033-N32 in a joint project with the Czech Science Foundation (project 19-29698L). The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

References

- [1] D. N. Arnold and P. J. Noon. Boundary integral equations of the first kind for the heat equation. In C. Brebbia, W. Wendland, and G. Kuhn, editors, *Boundary elements IX, Vol. 3 (Stuttgart, 1987)*, pages 213–229. Comput. Mech., Southampton, 1987.
- [2] S. Börm. *Efficient Numerical Methods for Non-local Operators*. Number 14 in Tracts in Mathematics. European Mathematical Society, Zürich, 2010. doi:10.4171/091.
- [3] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comput.*, 9(4):669–686, 1988. doi:10.1137/0909044.
- [4] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999. doi:10.1006/jcph.1999.6355.
- [5] M. Costabel. Boundary integral operators for the heat equation. *Integral Equations. Oper. Theory*, 13:498–552, 1990. doi:10.1007/BF01210400.

- [6] S. Dohr, K. Niino, and O. Steinbach. Space-time boundary element methods for the heat equation. In U. Langer and O. Steinbach, editors, *Space-time methods – applications to partial differential equations*, number 25 in Radon Series on Computational and Applied Mathematics, pages 1–60. De Gruyter, Berlin, 2019.
- [7] S. Dohr, J. Zapletal, G. Of, M. Merta, and M. Kravčenko. A parallel space-time boundary element method for the heat equation. *Comput. Math. Appl.*, 78(9):2852–2866, 2019. doi:10.1016/j.camwa.2018.12.031.
- [8] G. Gantner and R. van Venetië. Adaptive space-time BEM for the heat equation. *Comput. Math. Appl.*, 107:117–131, 2022. doi:10.1016/j.camwa.2021.12.022.
- [9] L. Greengard and P. Lin. Spectral approximation of the free-space heat kernel. *Appl. Comput. Harmon. Anal.*, 9(1):83–97, 2000. doi:10.1006/acha.2000.0310.
- [10] L. Greengard and J. Strain. A fast algorithm for the evaluation of heat potentials. *Comm. Pure Appl. Math.*, 43(8):949–963, 1990. doi:10.1002/cpa.3160430802.
- [11] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*, volume 49 of *SSCM*. Springer, Heidelberg, 2015. doi:10.1007/978-3-662-47324-5.
- [12] H. Harbrecht and J. Tausch. A fast sparse grid based space-time boundary element method for the nonstationary heat equation. *Numer. Math.*, 140(1):239–264, 2018. doi:10.1007/s00211-018-0963-5.
- [13] J. Homola. Acceleration of the space-time boundary element method using GPUs. Master’s thesis, VŠB Technical University of Ostrava, 2021. URL <https://hdl.handle.net/10084/143855>.
- [14] G. C. Hsiao and J. Saranen. Boundary integral solution of the two-dimensional heat equation. *Math. Methods Appl. Sci.*, 16(2):87–114, 1993. doi:10.1002/mma.1670160203.
- [15] G. Kratochwill. Analyse und Vergleich verschiedener Varianten der schnellen Gauß–Transformation. Master’s thesis, Technische Universität Graz, Graz, 2018. URL <https://permalink.obvsg.at/tug/AC15175955>.
- [16] M. Merta, G. Of, R. Watschinger, and J. Zapletal. besthea. <https://github.com/zap150/besthea>, 2020.
- [17] M. Meßner. *A Fast Multipole Galerkin Boundary Element Method for the Transient Heat Equation*. Number 23 in Monographic Series TU Graz: Computation in Engineering and Science. Verlag der Technischen Universität Graz, Graz, 2014. doi:10.3217/978-3-85125-350-4.
- [18] M. Messner, M. Schanz, and J. Tausch. A fast Galerkin method for parabolic space-time boundary integral equations. *J. Comput. Phys.*, 258:15–30, 2014. doi:10.1016/j.jcp.2013.10.029.
- [19] M. Messner, M. Schanz, and J. Tausch. An efficient Galerkin boundary element method for the transient heat equation. *SIAM J. Sci. Comput.*, 37(3):A1554–A1576, 2015. doi:10.1137/151004422.
- [20] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM J. Sci. Comput.*, 15(3):713–735, 1994. doi:10.1137/0915046.

- [21] T. J. Rivlin. *Chebyshev polynomials: from approximation theory to algebra and number theory*. Wiley, New York, 2nd edition, 1990.
- [22] J. Tausch. A fast method for solving the heat equation by layer potentials. *J. Comput. Phys.*, 224(2):956–969, 2007. doi:10.1016/j.jcp.2006.11.001.
- [23] J. Tausch. Fast Nyström methods for parabolic boundary integral equations. In U. Langer, M. Schanz, O. Steinbach, and W. Wendland, editors, *Fast Boundary Element Methods in Engineering and Industrial Applications*, number 63 in Lecture Notes in Applied and Computational Mechanics, pages 185–219. Springer-Verlag, Berlin Heidelberg, 2012. doi:10.1007/978-3-642-25670-7_6.
- [24] J. Tausch and A. Weckiewicz. Multidimensional fast Gauss transforms by Chebyshev expansions. *SIAM J. Sci. Comput.*, 31:3547–3565, 2009. doi:10.1137/080732729.
- [25] R. Watschinger, M. Merta, G. Of, and J. Zapletal. A parallel fast multipole method for a space-time boundary element method for the heat equation. *SIAM J. Sci. Comput.*, 44(4):C320–C345, 2022. doi:10.1137/21M1430157.
- [26] J. Zapletal, R. Watschinger, G. Of, and M. Merta. Semi-analytic integration for a parallel space-time boundary element method modeling the heat equation. *Comput. Math. Appl.*, 103:156–170, 2021. doi:10.1016/j.camwa.2021.10.025.