

STRUCTURED DATA-SPARSE APPROXIMATION TO HIGH ORDER TENSORS ARISING FROM THE DETERMINISTIC BOLTZMANN EQUATION

BORIS N. KHOROMSKIJ

ABSTRACT. We develop efficient data-sparse representations to a class of high order tensors via a block many-fold Kronecker product decomposition. Such a decomposition is based on low separation-rank approximations of the corresponding multivariate generating function. We combine the *Sinc* interpolation and a quadrature-based approximation with hierarchically organised block tensor-product formats. Different matrix and tensor operations in the generalised Kronecker tensor-product format including the Hadamard-type product can be implemented with the low cost. An application to the collision integral from the deterministic Boltzmann equation leads to an asymptotical cost $O(n^4 \log^\beta n) - O(n^5 \log^\beta n)$ in the one-dimensional problem size n (depending on the model kernel function), which noticeably improves the complexity $O(n^6 \log^\beta n)$ of the full matrix representation.

1. INTRODUCTION

In large-scale applications one deals with algebraic operations on high-dimensional, densely populated matrices or tensors which require considerable computational resources. For example, we mention the numerical approximation to multi-dimensional integral operators, the Lyapunov matrix equation in control theory, density matrix calculation for solving the Schrödinger equation for many-particle systems, deterministic numerical methods for the Boltzmann equation, as well as various applications in chemometrics, psychometrics and stochastic models.

This paper is motivated by the problem of extensive matrix calculations arising in computational dilute gas dynamics. The bottleneck of the modern numerical methods based on the deterministic Boltzmann equation is the expensive computation of the collision integral defined on the six-dimensional rectangular grid in the coordinate-velocity space [4, 19]. On the algebraic level the problem is equivalent to the evaluation of certain matrix operations including standard and Hadamard-type products of large fully populated tensors. Our prime interest here is the efficient data-sparse representation of arising tensor operations. We focus on the case when the corresponding tensors can be obtained as traces of an explicitly given multivariate function on a tensor-product lattice (see Definition 2.1 of a function-generated tensor). The important feature of the generating function is its good approximability by a separable expansion.

Received by the editor February 22, 2005 and, in revised form, October 4, 2005.

2000 *Mathematics Subject Classification.* Primary 65F50, 65F30; Secondary 15A24, 15A99.

Key words and phrases. Boltzmann equation, hierarchical matrices, Kronecker tensor product, high order tensors, *sinc* interpolation and quadratures.

Our approach is based on generalized three- or multifold Kronecker tensor-product decompositions¹ of a high order tensor A (see definitions in §2) that include data-sparse hierarchically classified matrix blocks of the Kronecker tensor-product format (cf. (1.1)). Specifically, given $p, q, n \in \mathbb{N}$, we approximate $A \in \mathbb{R}^{n^{pq}}$ (or certain blocks in A) by a q th order tensor A_r of the Kronecker product form

$$(1.1) \quad A_r = \sum_{k=1}^r c_k V_k^1 \otimes \cdots \otimes V_k^q \approx A, \quad c_k \in \mathbb{R},$$

where the low-dimensional components $V_k^\ell \in \mathbb{R}^{n^p}$ can be further represented in a structured data-sparse format (say, in the \mathcal{H} -matrix, Kronecker product or Toeplitz format). Here and in the following \otimes denotes the Kronecker product operation. The *Kronecker rank* r , the number of products in (1.1), is supposed to be small. Therefore, A_r can be represented with the low cost qrn^p compared with n^{pq} . The tensor-product format (1.1) has plenty of other merits.

In general, the fully populated target tensor A has n^{pq} nonzero entries, which require $O(n^{pq})$ arithmetical operations (at least) to perform the corresponding tensor-tensor arithmetics. However, in many applications the representation by a fully populated tensor A appears to be highly redundant. Hence we are interested in accurate decompositions (1.1) with possibly small Kronecker rank r which depends only logarithmically on both the tolerance $\varepsilon > 0$ and the problem size $N = n^p$. This will reduce the complexity of tensor operations to $O(n^p \log^\beta n)$ or even to $O(n \log^\beta n)$, but inheriting the important features of the original tensor A . The Kronecker product approximation to a certain class of *function-generated matrices* was introduced in [18, 23] (specifically, for translation-invariant functions). Several methods and numerical algorithms for the hierarchical tensor-product approximation to the multi-dimensional nonlocal operators (dense matrices) are described in [2, 18, 11, 10, 12, 14, 16, 15]. Tensor-product approximations were shown as a promising tool in many-body system calculations [1, 8, 9, 20].

In this paper, we apply the format (1.1) as well as its block version to treat the special class of *high-order function-generated tensors* (cf. §2). We describe certain tensor-tensor operations and present the complexity analysis for the low-rank tensors. In §3 this data-sparse tensor format is applied for numerical calculations of the collision integral from the discrete Boltzmann equation (particularly, in the case $q = 2$, $p = 3$). We make use of the discretization scheme for the collision integral described in [19]. The proposed method is based on a low separation rank approximation to the non-shift-invariant kernel functions of the form $g_1(\|u\|, \|v\|)$ or $g_2(\|u\|, \|v\|, |\langle u, v \rangle|)$, $u, v \in \mathbb{R}^p$. Here and in the following, $\langle \cdot, \cdot \rangle$ denotes the scalar product in \mathbb{R}^p and $\|u\| := \sqrt{\langle u, u \rangle}$. For a function of the type g_1 corresponding to the so-called *variable hard spheres* model, our algorithm is proved to have the computational cost $O(n^{p+1} \log^\beta n)$, which improves drastically the complexity bound $O(n^{2p} \log^\beta n)$ of the full matrix arithmetics. For the more general kernel function of the type g_2 (cf. §3.1), we are able to reduce the asymptotical complexity to $O(n^{2p-1} \log^\beta n)$. Note that our algorithms are applicable in the case of nonuniform grids. In Appendix A, we address the error analysis and discuss some numerical methods for the separable approximation to multivariate functions.

¹In traditional literature on chemometrics, psychometrics and data mining, such representations are known as three- and multiway decompositions.

2. ARITHMETICS OF TENSOR-PRODUCT MATRIX FORMATS

The matrix arithmetic in the tensor-product format is well presented in the literature (cf. [24] and references therein). In this section we recall the properties of standard matrix/tensor operations and then consider in more detail some special topics related to the Hadamard tensor product. The latter will be used in our particular application in §3.

2.1. Definitions and examples. First, we define the function-generated tensors. Let us introduce the product index set $\mathcal{I}^\ell = \mathcal{I}_1^\ell \otimes \cdots \otimes \mathcal{I}_p^\ell$, where we use multi-indices $\mathbf{i}_\ell = (i_{\ell,1}, \dots, i_{\ell,p}) \in \mathcal{I}^\ell$, $\ell = 1, \dots, q$, with the components $i_{\ell,m} \in \{1, \dots, N_{\mathcal{I}^\ell}\}$, for $m = 1, \dots, p$. In the following we simplify the considerations and set $N_{\mathcal{I}^\ell} = n$, $\ell = 1, \dots, q$, which implies $\#\mathcal{I}^\ell = n^p$, where $\#$ denotes the cardinality of an index set and $N_{\mathcal{I}^\ell} = n$ is the one-dimensional problem size.

Let $\{\zeta_{\mathbf{i}_1}^1, \dots, \zeta_{\mathbf{i}_q}^q\}$ with $\mathbf{i}_\ell \in \mathcal{I}^\ell$, $\ell = 1, \dots, q$, be a set of collocation points living on the uniform tensor-product lattice $\omega_{\mathbf{d}} = \omega_1 \times \cdots \times \omega_q$, where ω_ℓ , $\ell = 1, \dots, q$, is the uniform rectangular grid on $[-L, L]^p$ indexed by \mathcal{I}^ℓ . We also define the index set $\mathcal{I}^{\mathbf{d}} = \mathcal{I}^1 \otimes \cdots \otimes \mathcal{I}^q$. For the ease of presentation, we consider a *uniform* lattice, however, all the constructions are applicable for a quasi-uniform distribution of collocation points.

Definition 2.1. Given the multivariate function

$$(2.1) \quad g : \mathbb{R}^{\mathbf{d}} \rightarrow \mathbb{R} \quad \text{with } \mathbf{d} = qp, \quad p \in \mathbb{N}, \quad q \geq 2,$$

defined in a hypercube $\Omega = \{(\zeta^1, \dots, \zeta^q) \in \mathbb{R}^{\mathbf{d}} : \|\zeta^\ell\|_\infty \leq L, \ell = 1, \dots, q\} \in \mathbb{R}^{\mathbf{d}}$, $L > 0$, where $\|\cdot\|_\infty$ means the ℓ_∞ -norm of $\zeta^\ell \in \mathbb{R}^p$, $\mathbb{R}^{\mathbf{d}}$, on the index set $\mathcal{I}^{\mathbf{d}}$, we introduce the function-generated q th order tensor

$$(2.2) \quad A \equiv A(g) := [a_{\mathbf{i}_1 \dots \mathbf{i}_q}] \in \mathbb{R}^{\mathcal{I}^{\mathbf{d}}} \quad \text{with} \quad a_{\mathbf{i}_1 \dots \mathbf{i}_q} := g(\zeta_{\mathbf{i}_1}^1, \dots, \zeta_{\mathbf{i}_q}^q).$$

In various applications, the function g is analytic in all variables except the “small” set of singularity points given by a hyperplane $\mathcal{S}(g) := \{\zeta \in \Omega : \zeta^1 = \zeta^2 = \cdots = \zeta^q\}$ or by a single point $\mathcal{S}(g) := \{\zeta \in \Omega : \zeta^1 = \zeta^2 = \cdots = \zeta^q = 0\}$.

In numerical calculations for the Boltzmann equation, n may vary from several tens to several hundreds, therefore, one arrives at a challenging computational problem (cf. [4, 19]). Indeed, the storage required for a naive “pointwise” representation to the tensor A in (2.2) amounts to $O(n^{qp})$, which for $p = 3$, $q \geq 2$, and $n \geq 10^2$ is no longer tractable. We are interested in the efficient storage of A and in a fast calculation of sums involving the tensor A , which is algorithmically equivalent to the evaluation of an associated bilinear form $\langle A, \cdot \rangle$ and certain tensor-tensor products.

A multifold decomposition (1.1) can be derived by using a corresponding separable expansion of the generating function g (see the Appendix for more details). In this section the existence of such an expansion will be postulated.

Assumption 2.2. Suppose that a multivariate function $g : \mathbb{R}^{\mathbf{d}} \rightarrow \mathbb{R}$ can be approximated by a separable expansion

$$(2.3) \quad g_r(\zeta^1, \dots, \zeta^q) := \sum_{k=1}^r c_k \Phi_k^1(\zeta^1) \cdots \Phi_k^q(\zeta^q) \approx g, \quad \zeta^\ell \in \mathbb{R}^p, \quad \ell = 1, \dots, q,$$

where $c_k \in \mathbb{R}$ and with a given set of functions $\{\Phi_k^\ell : \mathbb{R}^p \rightarrow \mathbb{R}\}$.

In computationally efficient algorithms a separation rank r is supposed to be as small as possible, while the set of functions $\{\Phi_k^\ell : \mathbb{R}^p \rightarrow \mathbb{R}\}$ can be fixed or chosen adaptively to the problem.

Under Assumption 2.2, we can introduce the multifold decomposition (1.1) generated by g_r via $A_r := A(g_r)$, which corresponds to

$$V_k^\ell = \{\Phi_k^\ell(\zeta_{i_\ell}^\ell)\}_{i_\ell \in \mathcal{I}^\ell} \in \mathbb{R}^{n^p}, \quad \ell = 1, \dots, q, \quad k = 1, \dots, r,$$

where $\zeta_{i_\ell}^\ell$ belongs to the set of collocation points in variable ζ^ℓ . It can be proven that the accuracy of such a decomposition can be estimated by the approximation error $g - g_r$ (cf. §2.3 and the discussion in [18]).

Though in general the construction of an approximation (2.3) with small separation rank r is a complicated numerical task, in many interesting applications efficient and elegant algorithms are available (cf. §3 and the Appendix).

2.2. Matrix and tensor operations. We recall that the Kronecker product operation $A \otimes B$ of two matrices $A = [a_{ij}] \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{h \times g}$ is an $mh \times ng$ matrix that has the block-representation $[a_{ij}B]$ (corresponding to $p = 2$).

For the Kronecker product format (1.1), the commonly used tensor-tensor operations can be performed with low cost. For the moment, we assume that the cost related to V_k^ℓ is linear-logarithmic in n^p , namely the required memory and the complexity of tensor-vector multiplication is bounded by $\mathcal{W}(V_k^\ell) = O(n^p \log^\beta n)$. Here and in the following, $\mathcal{W}(B)$ denotes the complexity of tensor-vector multiplication for the tensor B . For the index set $\mathcal{I}^\ell = \mathcal{I}_1^\ell \otimes \dots \otimes \mathcal{I}_p^\ell$ we assume $\#\mathcal{I}_m^\ell = n$ for $m = 1, \dots, p$. Given p_1 such that $1 \leq p_1 < p$, we further use the factorisation $\mathcal{I}^\ell = \mathcal{I}_x^\ell \otimes \mathcal{J}_y^\ell$, where

$$\mathcal{I}_x^\ell = \mathcal{I}_1^\ell \otimes \dots \otimes \mathcal{I}_{p_1}^\ell, \quad \mathcal{J}_y^\ell = \mathcal{I}_{p_1+1}^\ell \otimes \dots \otimes \mathcal{I}_p^\ell.$$

This naturally induces the representation $\mathcal{I}^\mathbf{d} = \mathcal{I}_x \otimes \mathcal{J}_y$, where $\mathcal{I}_x = \mathcal{I}_x^1 \otimes \dots \otimes \mathcal{I}_x^q$, $\mathcal{J}_y = \mathcal{J}_y^1 \otimes \dots \otimes \mathcal{J}_y^q$. One can write $A = [a_{ij}]_{i \in \mathcal{I}_x, j \in \mathcal{J}_y}$; then the elements in $\mathbb{R}^{\mathcal{I}_x}$ and in $\mathbb{R}^{\mathcal{J}_y}$ can be interpreted as vectors. In the following, if this does not lead to ambiguity, we shall omit indices x, y .

Given vectors $x \in \mathbb{R}^{\mathcal{I}_x}$, $y \in \mathbb{R}^{\mathcal{J}_y}$, one can introduce the products $x^T A \in \mathbb{R}^{\mathcal{J}_y}$ and $Ay \in \mathbb{R}^{\mathcal{I}_x}$. The associated bilinear form $\langle x, Ay \rangle$ is approximated by $\langle x, A_r y \rangle$, which still amounts to $O(rN^{1+1/q})$ arithmetic operations with $N = n^p$. A simplification is possible if both x and y are, in turn, of tensor form, i.e.,

$$(2.4) \quad \begin{aligned} x &= \sum_{m=1}^{r_x} x_m^{(1)} \otimes \dots \otimes x_m^{(q)}, \quad x_m^{(\ell)} \in \mathbb{R}^{\mathcal{I}_x^\ell}, \\ y &= \sum_{s=1}^{r_y} y_s^{(1)} \otimes \dots \otimes y_s^{(q)}, \quad y_s^{(\ell)} \in \mathbb{R}^{\mathcal{J}_y^\ell}, \quad \ell = 1, \dots, q. \end{aligned}$$

Now one can calculate the products $x^T A_r$ and $A_r y$ by

$$(2.5) \quad x^T A_r = \sum_{k=1}^r \sum_{m=1}^{r_x} (x_m^{(1)})^T V_k^1 \otimes \dots \otimes (x_m^{(q)})^T V_k^q,$$

$$(2.6) \quad A_r y = \sum_{k=1}^r \sum_{s=1}^{r_y} V_k^1 y_s^{(1)} \otimes \dots \otimes V_k^q y_s^{(q)},$$

which requires $O(qr \max\{r_x, r_y\} \max_{k,\ell} \mathcal{W}(V_k^\ell))$ operations. Computing the bilinear form, we replace the general expression by

$$(2.7) \quad \langle x, A_r y \rangle = \sum_{k=1}^r \sum_{s=1}^{r_y} \sum_{m=1}^{r_x} \left\langle x_m^{(1)}, V_k^1 y_s^{(1)} \right\rangle \cdots \left\langle x_m^{(q)}, V_k^q y_s^{(q)} \right\rangle$$

involving only $qr_x r_y r$ terms to be calculated. This requires only

$$O(qr_x r_y r \max_{k,\ell} \mathcal{W}(V_k^\ell))$$

operations.

The following lemma indicates the simple (but important) duality between multiplication of functions and the Hadamard product² of the corresponding function generated tensors.

Lemma 2.3. *Given tensors $A, B \in \mathbb{R}^{\mathcal{I}^d}$ generated by the multivariate functions g_1 and g_2 , respectively, then the function generated tensor $G_{12} \in \mathbb{R}^{\mathcal{I}^d}$ corresponding to the product function $g_{12} = g_1 \cdot g_2$, equals the Hadamard product $G_{12} = A \odot B$.*

Let both A and B be represented in the form (1.1) with the Kronecker rank r_A , r_B and with V_k^ℓ substituted by $A_k^\ell \in \mathbb{R}^{\mathcal{I}^\ell}$ and $B_k^\ell \in \mathbb{R}^{\mathcal{I}^\ell}$, respectively. Then $A \odot B$ is a tensor with the Kronecker rank $r_A r_B$ given by

$$A \odot B = \sum_{k=1}^{r_A} \sum_{m=1}^{r_B} c_k c_m (A_k^1 \odot B_m^1) \otimes \cdots \otimes (A_k^q \odot B_m^q).$$

Proof. By definition we have

$$G_{12} := [g_{12}(\zeta_1^1, \dots, \zeta_{i_q}^q)] = [g_1(\zeta_1^1, \dots, \zeta_{i_q}^q) \cdot g_2(\zeta_1^1, \dots, \zeta_{i_q}^q)] = A \odot B.$$

Furthermore, it is easy to check that

$$(A_1 \otimes B_1) \odot (A_2 \otimes B_2) = (A_1 \odot A_2) \otimes (B_1 \odot B_2),$$

and similar for q -term products. Applying the above relations, we obtain

$$\begin{aligned} A \odot B &= \left(\sum_{k=1}^{r_A} c_k \bigotimes_{\ell=1}^q A_k^\ell \right) \odot \left(\sum_{m=1}^{r_B} c_m \bigotimes_{\ell=1}^q B_m^\ell \right) \\ &= \sum_{k=1}^{r_A} \sum_{m=1}^{r_B} c_k c_m \left(\bigotimes_{\ell=1}^q A_k^\ell \right) \odot \left(\bigotimes_{\ell=1}^q B_m^\ell \right), \end{aligned}$$

and the second assertion follows. \square

Next we introduce and analyse the more complicated Hadamard-type tensor operations. We consider tensors defined on the index sets \mathcal{I} , \mathcal{J} , \mathcal{L} , as well as on certain of their products, where each of the above-mentioned index sets again has an intrinsic p -fold tensor structure. Let us be given tensors $U \otimes Y \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ with $U \in \mathbb{R}^{\mathcal{I}}$, $Y \in \mathbb{R}^{\mathcal{J}}$, and $B \in \mathbb{R}^{\mathcal{I} \times \mathcal{L}}$, and let $T : \mathbb{R}^{\mathcal{L}} \rightarrow \mathbb{R}^{\mathcal{J}}$ be the linear operator (which, in turn, admits a certain tensor representation) that maps tensors defined on the index set \mathcal{L} into those defined on \mathcal{J} . We introduce the Hadamard “scalar”

²We define the Hadamard product $C = A \odot B = \{c_{i_1 \dots i_q}\}_{(i_1 \dots i_q) \in \mathcal{I}^d}$ of two tensors $A, B \in \mathbb{R}^{\mathcal{I}^d}$ by the entry-wise multiplication $c_{i_1 \dots i_q} = a_{i_1 \dots i_q} \cdot b_{i_1 \dots i_q}$.

product $[D, C]_{\mathcal{I}} \in \mathbb{R}^{\mathcal{K}}$ of two tensors $D := [D_{\mathbf{i}, \mathbf{k}}] \in \mathbb{R}^{\mathcal{I} \times \mathcal{K}}$ and $C := [C_{\mathbf{i}, \mathbf{k}}] \in \mathbb{R}^{\mathcal{I} \times \mathcal{K}}$ with $\mathcal{K} \in \{\mathcal{I}, \mathcal{J}, \mathcal{L}\}$ by

$$[D, C]_{\mathcal{I}} := \sum_{\mathbf{i} \in \mathcal{I}} [D_{\mathbf{i}, \mathcal{K}}] \odot [C_{\mathbf{i}, \mathcal{K}}],$$

where \odot denotes the Hadamard product on the index set \mathcal{K} and $[D_{\mathbf{i}, \mathcal{K}}] := [D_{\mathbf{i}, \mathbf{k}}]_{\mathbf{k} \in \mathcal{K}}$. Now we are able to prove the following lemma.

Lemma 2.4. *Let U, Y, B and T be given as above. Then, with $\mathcal{K} = \mathcal{J}$, the following identity is valid:*

$$(2.8) \quad [U \otimes Y, T \cdot B]_{\mathcal{I}} = Y \odot (T \cdot [U, B]_{\mathcal{I}}) \in \mathbb{R}^{\mathcal{J}}.$$

Proof. By definition of the Hadamard scalar product we have

$$\begin{aligned} [U \otimes Y, T \cdot B]_{\mathcal{I}} &= \sum_{\mathbf{i} \in \mathcal{I}} [U \otimes Y]_{\mathbf{i}, \mathcal{J}} \odot [T \cdot B]_{\mathbf{i}, \mathcal{J}} \\ &= \sum_{\mathbf{i} \in \mathcal{I}} [[U]_{\mathbf{i}} \cdot Y]_{\mathbf{i}, \mathcal{J}} \odot [T \cdot B]_{\mathbf{i}, \mathcal{J}} \\ &= Y \odot \left(\sum_{\mathbf{i} \in \mathcal{I}} [U]_{\mathbf{i}} [T \cdot B]_{\mathbf{i}, \mathcal{J}} \right) \\ &= Y \odot \left(T \cdot \sum_{\mathbf{i} \in \mathcal{I}} [U]_{\mathbf{i}} [B]_{\mathbf{i}, \mathcal{L}} \right); \end{aligned}$$

then the assertion follows. \square

Identity (2.8) is of great importance in the forthcoming applications, since on the right-hand side the operator T is removed from the scalar product and, hence, it applies only once.

We summarize that the Kronecker tensor-product format possesses the following numerical complexity:

- *Data compression.* The storage for the V_k^ℓ matrices of (1.1) is only $\mathcal{O}(qrn^p)$ with $r = \mathcal{O}(\log^\alpha n)$ for some $\alpha > 0$, while that for the original (dense) tensor A is $\mathcal{O}(n^{qp})$.
- *Matrix-by-vector complexity.* Instead of $\mathcal{O}(n^{qp})$ operations to compute Ay , $y \in \mathbb{R}^{\mathcal{I}_y}$, we now need only $\mathcal{O}(qrn^{p+1})$ operations. If the vector can be represented in a tensor-product form (cf. (2.4)), the corresponding cost is reduced to $\mathcal{O}(qrr_y n^p)$ operations, while the calculation of the bilinear form $\langle x, A_r y \rangle$ requires $\mathcal{O}(qrr_x r_y n^p)$ flops.
- *Matrix-by-matrix complexity.* To compute AB , we need only $\mathcal{O}(qr^2 n^{p+1})$ operations.
- *The Hadamard matrix product.* In general, the Hadamard product $A \odot B$ of two q th order tensors $A, B \in \mathbb{R}^{\mathcal{I}^d}$ requires $\mathcal{O}(n^{qp})$ multiplications, while for tensors A, B represented by the Kronecker product ansatz (1.1) we need only $\mathcal{O}(qr^2 n^p)$ arithmetical operations (cf. Lemma 2.3).

Note that if V_k^ℓ allows a certain data-sparse representation (say, the \mathcal{H} -matrix format), then all the complexity estimates might be correspondingly improved.

2.3. Error analysis. We consider a low Kronecker rank approximation A_r to $A = A(g)$ defined by $A_r := A(g_r)$. To address the approximation issue for $A - A_r$, we assume that the error $g - g_r$ can be estimated in the $L^\infty(\Omega)$ - or $L^2(\Omega)$ -norm (see the Appendix). We also apply the weighted L^2 -norm defined by

$$\|u\|_{L^2, w} := \sqrt{\int_{\Omega} w(\zeta) u^2(\zeta) d\zeta}, \quad w(\zeta) > 0.$$

For the error analysis we make use of the Euclidean and $\|\cdot\|_\infty$ tensor norms

$$\|x\|_2 := \sqrt{\sum_{\mathbf{i} \in \mathcal{I}} x_{\mathbf{i}}^2}, \quad \|x\|_\infty := \max_{\mathbf{i} \in \mathcal{I}} |x_{\mathbf{i}}|, \quad x \in \mathbb{R}^{\mathcal{I}},$$

respectively. Let us use the abbreviations $\mathcal{I} = \mathcal{I}_x$, $\mathcal{J} = \mathcal{J}_y$, $\mathcal{I}^{\mathbf{d}} = \mathcal{I} \times \mathcal{J}$, if this does not lead to ambiguity. Let $g - g_r$ be smooth enough. For the above-defined tensor norms and for a quasi-uniform distribution of collocation points we have

$$(2.9) \quad \|A(g) - A(g_r)\|_2 \leq C \frac{N_{\mathcal{I}}^{1/2} N_{\mathcal{J}}^{1/2}}{L^{q/2}} \|g - g_r\|_{L^2}.$$

The following lemma describes relations between the approximation error $\|g - g_r\|$ evaluated in different norms and the corresponding error of the Kronecker product representation.

Lemma 2.5. *We have $\|A - A_r\|_\infty \leq \|g - g_r\|_{L^\infty(\Omega)}$. For any vectors $x \in \mathbb{R}^{\mathcal{I}}$, $y \in \mathbb{R}^{\mathcal{J}}$, the following bounds on the consistency error $A - A_r$ hold:*

$$(2.10) \quad \begin{aligned} |\langle (A - A_r)x, y \rangle| &\leq \|g - g_r\|_{L^\infty(\Omega)} \|x\|_1 \|y\|_1 \\ &\leq N_{\mathcal{I}}^{1/2} N_{\mathcal{J}}^{1/2} \|g - g_r\|_{L^\infty(\Omega)} \|x\|_2 \|y\|_2, \\ |\langle (A - A_r)x, y \rangle| &\leq C \frac{N_{\mathcal{I}}^{1/2} N_{\mathcal{J}}^{1/2}}{L^{q/2}} \|g - g_r\|_{L^2(\Omega)} \|x\|_2 \|y\|_2, \\ |\langle (A - A_r)x, y \rangle| &\leq C \frac{N_{\mathcal{I}}^{1/2} N_{\mathcal{J}}^{1/2}}{L^{q/2}} \|g - g_r\|_{L^2(\Omega), w} \|\bar{w}_x^{-1/4} x\|_2 \|\bar{w}_y^{-1/4} y\|_2, \end{aligned}$$

where \bar{w}_x , \bar{w}_y are traces of $w(\zeta) > 0$ on the corresponding grids.

Proof. The first assertion follows by the construction of A_r . Indeed,

$$\begin{aligned} \|A - A_r\|_\infty &= \max\{|g(\zeta_{\mathbf{i}_1}^1, \dots, \zeta_{\mathbf{i}_q}^q) - \sum_{k=1}^r \Phi_k^1(\zeta_{\mathbf{i}_1}^1) \cdots \Phi_k^q(\zeta_{\mathbf{i}_q}^q)| : (\mathbf{i}_1, \dots, \mathbf{i}_d) \in \mathcal{I}^{\mathbf{d}}\} \\ &\leq \|g - g_r\|_{L^\infty(\Omega)}. \end{aligned}$$

Now we readily obtain

$$|\langle (A - A_r)x, y \rangle| \leq \|g - g_r\|_{L^\infty(\Omega)} \sum_{\mathbf{i} \in \mathcal{I}, \mathbf{j} \in \mathcal{J}} |x_{\mathbf{i}} y_{\mathbf{j}}| \leq \|g - g_r\|_{L^\infty(\Omega)} \|x\|_1 \|y\|_1,$$

which proves (2.10) since $\|x\|_1 \leq N_{\mathcal{I}}^{1/2} \|x\|_2$ and $\|y\|_1 \leq N_{\mathcal{J}}^{1/2} \|y\|_2$. On the other hand, applying the Cauchy-Schwarz inequality we have

$$|\langle (A - A_r)x, y \rangle| \leq \sum_{\mathbf{i} \in \mathcal{I}, \mathbf{j} \in \mathcal{J}} |(a_{\mathbf{ij}} - a_{r, \mathbf{ij}}) x_{\mathbf{i}} y_{\mathbf{j}}| \leq \|A - A_r\|_2 \|x\|_2 \|y\|_2.$$

Then the second bound follows from (2.9). The rest of the proof is based on similar arguments. \square

In many applications the generating function $g(\zeta)$ actually depends on a few scalar variables which are functionals of ζ (see examples in §3). As a simple example, one might have a function depending on only one scalar parameter, $g(\zeta) = G(\rho(\zeta))$, where $G : [0, a] \rightarrow \mathbb{R}$ with $\rho : [-L, L]^p \rightarrow [0, a]$, $a = a(L) > 0$. In the following, we focus on the case $\rho(\zeta) = \|\zeta\|_2$, where $a = \sqrt{p}L$. The desired separable approximation $g_r(\zeta)$ can be derived from a proper approximation G_r to the univariate function $G(\rho)$, $\rho \in [0, a]$, by exponential sums (cf. [5, 16] for more details). It is easy to see that the approximation error $g - g_r$ arising in Lemma 2.5 (in general, measured in different norms on Ω) can be estimated via the corresponding error $G - G_r$. Concerning the weight function, we further assume that $w(\zeta) = w(\|\zeta\|_2)$.

Lemma 2.6. *The following estimates are valid:*

$$(2.11) \quad \|g - g_r\|_{L^\infty} = \|G - G_r\|_{L^\infty},$$

$$(2.12) \quad \|g - g_r\|_{L^2(\Omega)} \leq CL^{\frac{q-1}{2}} \|G - G_r\|_{L^2[0,a]}.$$

Proof. The first bound is trivial, while the second bound is obtained by passing to integration in the q -dimensional spherical coordinates. \square

3. COLLISION INTEGRAL FROM THE BOLTZMANN EQUATION

3.1. Setting the problem. In the case of simple, dilute gas [7], the particle density $f(t, x, v)$ satisfies the Boltzmann equation

$$f_t + (v, \text{grad}_x f) = \mathcal{Q}(f, f),$$

which describes the time evolution of $f : \mathbb{R}_+ \times \Omega \times \mathbb{R}^3 \rightarrow \mathbb{R}_+$, where $\Omega \subset \mathbb{R}^3$. The deterministic modelling of the Boltzmann equation is limited by the high numerical cost to evaluate the integral term (the Boltzmann collision integral) involved in this equation. With fixed t, x , the Boltzmann collision integral can be split into

$$\mathcal{Q}(f, f)(v) = \mathcal{Q}_+(f, f)(v) + \mathcal{Q}_-(f, f)(v), \quad f(t, x, v) = f(v),$$

where the *loss part* \mathcal{Q}_- has the simple form

$$(3.1) \quad \mathcal{Q}_-(f, f)(v) = f(v) \int_{\mathbb{R}^3} B_{\text{tot}}(\|u\|) f(w) dw$$

with $u = v - w$ being the relative velocity, and the *gain part* can be represented by the double integral (cf. [19])

$$(3.2) \quad \mathcal{Q}_+(f, f)(v) = \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} B(\|u\|, \mu) f(v') f(w') d\mu dw,$$

where $v' = \frac{1}{2}(v + w + \|u\|e) \in \mathbb{R}^3$, $w' = \frac{1}{2}(v + w - \|u\|e) \in \mathbb{R}^3$ and $e \in \mathbb{S}^2 \subset \mathbb{R}^3$ is the unit vector. In the case of the *inverse power cut-off potential*, we have

$$B(\|u\|, \mu) = \|u\|^{1-4/\nu} g_\nu(\mu), \quad \nu > 1, \quad \mu = \cos(\theta) = \frac{\langle u, e \rangle}{\|u\|},$$

with g_ν being a given function of the scattering angle only, such that $g_\nu \in L^1([-1, 1])$ holds.

The integral (3.1) can be represented by a block-Toeplitz matrix, which can be implemented in linear-logarithmic cost in n^p (cf. [19]). Hence, in the following

we focus on the efficient approximation to the integral (3.2). Let \mathcal{F} be the p -dimensional Fourier transform in variables (v, ζ) defined by

$$\hat{h}(\zeta) = \mathcal{F}_{v \rightarrow \zeta}[h(\cdot)](\zeta) := \int_{\mathbb{R}^3} h(v) e^{i\langle v, \zeta \rangle} dv.$$

The function $h(v)$ can be reconstructed by

$$h(v) = \mathcal{F}_{\zeta \rightarrow v}^{-1}[\hat{h}(\cdot)](v) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \hat{h}(\zeta) e^{-i\langle v, \zeta \rangle} d\zeta.$$

Then

$$(3.3) \quad \mathcal{Q}_+(f, f)(v) = \mathcal{F}_{y \rightarrow v} \left[\int_{\mathbb{R}^3} g(u, y) \mathcal{F}_{z \rightarrow y}^{-1}[f(z - u)f(z + u)](u, y) du \right](v),$$

where we have

$$g(u, y) = g(\|u\|, \|y\|, |\langle u, y \rangle|)$$

with

$$g(u, y) = \int_0^\pi g_\nu(\cos \theta) e^{-i\langle u, y \rangle \cos \theta} J_0(\sqrt{\|u\|^2 \|y\|^2 - \langle u, y \rangle^2} \sin \theta) \sin \theta d\theta$$

up to a scaling factor, where $J_0(z)$ is the Bessel function $J_0(z) = \frac{1}{2\pi} \int_0^{2\pi} e^{iz \cos \psi} d\psi$ (cf. [19]).

In this section, we dwell upon the family of kernel functions $g(u, y)$ which depend only on the three scalar variables $\|u\|, \|y\|, \langle u, y \rangle$. As a first example (cf. [19]), we consider the *variable hard spheres* model specified by a function

$$(3.4) \quad g_{1,\lambda}(u, y) := \|u\|^\lambda \operatorname{sinc}\left(\frac{\|u\|\|y\|}{\pi}\right), \quad u, y \in \mathbb{R}^p, \quad \lambda \in (-3, 1],$$

where the sinc-function is defined by

$$(3.5) \quad \operatorname{sinc}(z) = \frac{\sin(\pi z)}{\pi z}, \quad z \in \mathbb{C}.$$

This model corresponds to the case $q = 2$ in (2.1). It is worth noting that $g_{1,\lambda}$ depends solely on two scalar variables $\|u\|, \|y\|$ instead of $2p$ variables in the general case.

The second example to be considered corresponds to the function³ $g_{2,\lambda}$ defined by

$$g_{2,\lambda}(u, y) := \frac{\|u - y\|^\lambda}{\max\{\|u - y\|, \|u + y\|\}}, \quad \lambda \in [0, 1].$$

A direct calculation shows that

$$\max\{\|u - y\|, \|u + y\|\} = \sqrt{\|u\|^2 + \|y\|^2 + 2|\langle u, y \rangle|},$$

hence, finally, we consider the function

$$(3.6) \quad g_{2,\lambda}(u, y) := \frac{\|u - y\|^\lambda}{\sqrt{\|u\|^2 + \|y\|^2 + 2|\langle u, y \rangle|}}, \quad u, y \in \mathbb{R}^p.$$

Our forthcoming analysis shows that the presence of $|\langle u, y \rangle|$ in the arguments of $g_{2,\lambda}(u, y)$ makes the approximation process much more involved compared with the relatively easy construction in the case of the function $g_{1,\lambda}(u, y)$.

³The model problem with $g = g_{2,\lambda}$ was addressed to the author by Professor S. Rjasanow of the University of Saarbrücken.

3.2. Numerical algorithms adapted to some classes of kernels.

3.2.1. Implementation in the general case. The discrete version of the integral (3.3) can be evaluated as a sequence of matrix (tensor) operations (cf. [19] for the detailed description). Given a function $\Psi(u, z) = f(z + u)f(z - u)$, and $g(u, y)$, both represented on the lattice of collocation points $\omega_u \times \omega_z \in \mathbb{R}^{2p}$ (numbered by the index set $\mathcal{I} \times \mathcal{L}$) and $\omega_u \times \omega_y \in \mathbb{R}^{2p}$ (numbered by the index set $\mathcal{I} \times \mathcal{J}$), respectively. Note that we assume $u, z \in [-L, L]^p$ and $y \in [-Y, Y]^p$ with $\#\mathcal{I} = \#\mathcal{J} = \#\mathcal{L} = n^p$ and $Y = \frac{\pi n}{2L}$. Denote by $F_{\mathcal{L} \leftarrow \mathcal{J}} : \mathbb{R}^{n^p} \rightarrow \mathbb{R}^{n^p}$ the p -dimensional FFT matrix.

A practical choice of the parameters L, Y is based on the following observation.

Remark 3.1. It is known that a solution of the Boltzmann equation decays exponentially in the velocity space (cf. asymptotic of the so-called Maxwellian distribution [3], $O(\exp(-|v|^2/C))$).

On the index set $\mathcal{I} \times \mathcal{J} \times \mathcal{L} \in \mathbb{R}^{3p}$, the discrete collision integral $Q_+ \in \mathbb{R}^{\mathcal{L}}$ is represented by the following tensor operations.

Setup step 1: Compute the function generated tensor $B = [b_{\mathbf{il}}] = [\Psi(u_{\mathbf{i}}, z_{\mathbf{l}})] \in \mathbb{R}^{\mathcal{I} \times \mathcal{L}}$ (cost n^{2p}). For any fixed $\mathbf{i} \in \mathcal{I}$, $B_{\mathbf{i}} = \{b_{\mathbf{il}}\}_{\mathbf{l} \in \mathcal{L}} \in \mathbb{R}^{\mathcal{L}}$ is a subtensor of B .

Algorithm 1.

- (a) Compute the function generated tensor $A = [a_{\mathbf{ij}}] := [g(u_{\mathbf{i}}, y_{\mathbf{j}})] \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ (cost $O(n^{2p})$).
- (b) For $\mathbf{i} \in \mathcal{I}$, compute the inverse FFT $W = [w_{\mathbf{ij}}] = [F_{\mathcal{J} \leftarrow \mathcal{L}}^{-1} B_{\mathbf{i}}]_{\mathbf{j}} \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ in the index $\mathbf{l} \in \mathcal{L}$ (cost $O(n^{2p} \log n)$).
- (c) Evaluate the Hadamard scalar product with respect to the index $\mathbf{i} \in \mathcal{I}$ (cost $O(n^{2p})$):

$$V = A \odot W := \left[\sum_{\mathbf{i} \in \mathcal{I}} a_{\mathbf{ij}} w_{\mathbf{ij}} \right]_{\mathbf{j} \in \mathcal{J}} \in \mathbb{R}^{\mathcal{J}}.$$

- (d) Compute the FFT $Q_+ = F_{\mathcal{L} \leftarrow \mathcal{J}} V \in \mathbb{R}^{\mathcal{L}}$ in the index $\mathbf{j} \in \mathcal{J}$ (cost $O(n^p \log n)$).

Usually, the function $g(u, y)$ is defined on a $2p$ -dimensional lattice, thus we conclude that the corresponding function generated tensor A can be represented with the cost $O(n^{2p})$, which is the bottleneck in numerical implementation. All in all, for $p = 3$, Algorithm 1 has a complexity $O(n^6 \log n)$ with a large constant in $O(\cdot)$. This algorithm is awkward to implement, due to the presence of the p -dimensional FFT with the vector-size n^p , applied n^p times. Hence, our main goal is a reduction of the number of FFT calls in the numerical scheme which may reduce its complexity to $O(n^{p+1} \log^\beta n)$.

Note that Setup step 1 has a cost $O(n^{2p})$, which will not be included into the complexity estimate of Algorithm 1 since this calculation has nothing to do with the matrix compression.

3.2.2. Modifications of Algorithm 1 in the case of a low-rank tensor A . In the following we let $p = 3$. To fix the idea, we suppose that the function $g(u, y)$ can be represented (with required tolerance) by the low separation rank expansion

$$(3.7) \quad g(u, y) = \sum_{k=1}^r a_k(u) b_k(y), \quad u \in [-L, L]^3, \quad y \in [-Y, Y]^3,$$

with some continuous functions a_k, b_k and with certain constant $Y > 0$. Then the target tensor A takes the form

$$(3.8) \quad A = \sum_{k=1}^r U_k \otimes Y_k, \quad \text{with } Y_k \in \mathbb{R}^{\mathcal{J}}, \quad U_k \in \mathbb{R}^{\mathcal{I}},$$

where

$$U_k = [a_k(\zeta_{\mathbf{i}}^1)], \quad Y_k = [b_k(\zeta_{\mathbf{j}}^2)], \quad \zeta_{\mathbf{i}}^1 \in \omega_u, \quad \zeta_{\mathbf{j}}^2 \in \omega_y.$$

By a minor modification of the Setup step 1 (cf. Algorithm 1), this assumption now leads to a simplified scheme, which achieves a tremendous speed-up of Algorithm 1, reducing its overall complexity to $O(rn^p \log n)$.

Setup step 1': Evaluate the function generated tensor $B = [b_{\mathbf{il}}] = [\Psi(u_{\mathbf{i}}, z_{\mathbf{l}})] \in \mathbb{R}^{\mathcal{I} \times \mathcal{L}}$ (cost n^{2p}), and for $\mathbf{i} \in \mathcal{I}$, define the restriction $B_{\mathbf{i}} = B_{|(\mathbf{i}, \mathbf{l}), \mathbf{l} \in \mathcal{L}}$ by $B_{\mathbf{i}} = \{b_{\mathbf{il}}\}_{\mathbf{l} \in \mathcal{L}} \in \mathbb{R}^{\mathcal{L}}$. For $k = 1, \dots, r$, precompute the tensors $C_k = \sum_{\mathbf{i} \in \mathcal{I}} U_{k, \mathbf{i}} B_{\mathbf{i}} \in \mathbb{R}^{\mathcal{L}}$ (cost $O(rn^{2p})$).

Algorithm 1'.

- (a') Compute the function generated tensor by (3.8) (cost $O(rn^p)$).
- (b') For $k = 1, \dots, r$, compute the inverse FFT $W_k = F_{\mathcal{J} \leftarrow \mathcal{L}}^{-1} C_k \in \mathbb{R}^{\mathcal{J}}$ in the index \mathbf{l} (cost $O(rn^p \log n)$).
- (c') Evaluate the Hadamard product $V = \sum_{k=1}^r Y_k \odot W_k \in \mathbb{R}^{\mathcal{J}}$ in the index \mathbf{j} (cost $O(rn^p)$).
- (d') Compute the FFT $Q_+ = F_{\mathcal{L} \leftarrow \mathcal{J}} V \in \mathbb{R}^{\mathcal{L}}$ in the index \mathbf{j} , (cost $O(n^p \log n)$).

Lemma 3.2. *Under assumption (3.7), Algorithm 1' is algebraically equivalent to Algorithm 1. The overall cost of Algorithm 1' is estimated by $O(rn^p \log n)$.*

Proof. The equivalence of above algorithms follows from applying Lemma 2.4 with $U = U_k$, $Y = Y_k$, $T = F_{\mathcal{J} \leftarrow \mathcal{L}}^{-1}$. In fact, let W be defined as in Algorithm 1, item (b). Then we have

$$\begin{aligned} V &= \left(\sum_{k=1}^r U_k \otimes Y_k \right) \odot W = \sum_{k=1}^r [U_k \otimes Y_k, T \cdot B]_{\mathcal{I}} \\ &= \sum_{k=1}^r Y_k \odot (T \cdot [U_k, B]_{\mathcal{I}}) = \sum_{k=1}^r Y_k \odot W_k. \end{aligned}$$

The complexity bound is obtained by summing the corresponding cost estimates over all four steps (a')–(d'). \square

To reduce the cost of *Setup step 1'*, we make use of a certain product decomposition of the tensor B when available.

Remark 3.3. The tensor B can be presented as $B = F_+ \odot F_-$, where the Hankel–Toeplitz-type tensors F_{\pm} are generated by functions $f(z \pm u)$, respectively. However, in Algorithm 1 this beneficial property has not been taken into account.

In view of Remark 3.3, any separable expansion of the function $f(z \pm u) \approx \sum_{q=1}^m f_q(\pm u) h_q(z)$ results in a tensor-product representation of B with the Kronecker rank at most $r' = m^2$. The detailed discussion of this issue is beyond the scope of our paper and will be elaborated elsewhere.

Lemma 3.4. *Assume that*

$$(3.9) \quad B = \sum_{q=1}^{r'} D_q \otimes Z_q, \quad D_q \in \mathbb{R}^{\mathcal{I}}, \quad Z_q \in \mathbb{R}^{\mathcal{L}},$$

with small Kronecker rank r' . Then the target tensors C_k , $k = 1, \dots, r$, can be computed in $O(rr'n^p)$ operations (instead of $O(rn^{2p})$).

Proof. Relation (3.9) implies $B_{\mathbf{i}} = \sum_{q=1}^{r'} [D_{q,\mathbf{i}}] Z_q \in \mathbb{R}^{\mathcal{L}}$. Then

$$C_k = \sum_{\mathbf{i} \in \mathcal{I}} U_{k,\mathbf{i}} B_{\mathbf{i}} = \sum_{q=1}^{r'} c_{k,q} Z_q \in \mathbb{R}^{\mathcal{L}}.$$

Since all rr' coefficients $c_{k,q}$ can be computed with cost $O(rr'n^p)$, the assertion follows. \square

The beneficial feature of Algorithm 1' is the reduced number of FFTs (in fact, it requires only $r + 1$ calls of FFT). However, the existence of expansion (3.7) is crucial for the applicability of Algorithm 1'. Next, we extend Algorithm 1' to the case when (3.8) is available only block-wise. For our modified algorithm the number of FFT calls equals the number of blocks in the corresponding block partitioning of the target tensor A .

3.2.3. Modification for the block-low-rank tensors. Let $\mathcal{P}(\mathcal{I}^{\mathbf{d}})$ be a block partitioning of $\mathcal{I}^{\mathbf{d}} = \mathcal{I} \times \mathcal{J}$ with equal-sized blocks $b_{\nu} = \tau_{\nu} \times \sigma_{\nu} \in \mathcal{P}$, $\nu \in \mathcal{I}_{\mathcal{P}}$, and let $N_{\mathcal{P}} = \#\mathcal{I}_{\mathcal{P}}$ be the number of blocks in \mathcal{P} . Moreover, we assume that on each geometrical image $X(b) := \{\zeta_{\alpha} : \alpha \in b\}$ with $b \in \mathcal{P}$ there is a separable approximation

$$(3.10) \quad g(u, y) = \sum_{k=1}^r a_k(u) b_k(y), \quad (u, y) \in X(b),$$

with some continuous functions a_k, b_k depending in general on b . This implies that the target tensor A has a block Kronecker tensor-product representation, i.e.,

$$(3.11) \quad A|_{b_{\nu}} := \sum_{k=1}^r U_{k,\nu} \otimes Y_{k,\nu}, \quad \text{with } Y_{k,\nu} \in \mathbb{R}^{\sigma_{\nu}}, \quad U_{k,\nu} \in \mathbb{R}^{\tau_{\nu}} \quad \forall b_{\nu} \in \mathcal{P},$$

where

$$U_{k,\nu} = [a_k(\zeta_{\mathbf{i}}^1)], \quad Y_{k,\nu} = [b_k(\zeta_{\mathbf{j}}^2)], \quad \zeta_{\mathbf{i}}^1 \in X(\tau_{\nu}), \quad \zeta_{\mathbf{j}}^2 \in X(\sigma_{\nu}).$$

In a natural way, this induces the matrix-to-matrix agglomeration operation $\bigcup_{\nu \in \mathcal{I}_{\mathcal{P}}}$ associated with \mathcal{P} by

$$A := \bigcup_{\nu \in \mathcal{I}_{\mathcal{P}}} B_{\nu} \quad \text{defined by} \quad B_{\nu} = A|_{b_{\nu}}.$$

Furthermore, we also need the matrix-to-vector agglomeration procedure \biguplus defined as follows. For each $\nu \in \mathcal{I}_{\mathcal{P}}$ and any $y_{\nu} \in \mathbb{R}^{\sigma_{\nu}}$ associated with the block $b_{\nu} = \tau_{\nu} \times \sigma_{\nu}$, define

$$y := \biguplus_{\{\nu \in \mathcal{I}_{\mathcal{P}}\} \rightarrow \mathcal{J}} y_{\nu} \in \mathbb{R}^{\mathcal{J}} \quad \text{by} \quad y_{\mathbf{j}} := \sum_{\nu: \mathbf{j} \in \sigma_{\nu}} y_{\nu, \mathbf{j}}.$$

By a proper modification of the Setup step 1' (cf. Algorithm 1'), representation (3.11) now leads to a scheme with the overall complexity $O(rN_{\mathcal{P}}n^p \log n)$.

Setup step 1'': Given the function-generated tensor $B = [b_{\mathbf{il}}] = [\Psi(u_{\mathbf{i}}, z_1)] \in \mathbb{R}^{\mathcal{I} \times \mathcal{L}}$, we define its restriction in the first index to τ_ν by $B|_{\tau_\nu} := \{b_{\mathbf{il}}\}_{\mathbf{i} \in \tau_\nu, \mathbf{l} \in \mathcal{L}} \in \mathbb{R}^{\tau_\nu \times \mathcal{L}}$. For $k = 1, \dots, r$, $\nu \in \mathcal{I}_{\mathcal{P}}$, precompute the tensors

$$C_{k,\nu} = \sum_{\mathbf{i} \in \tau_\nu} (U_{k,\nu})_{|\mathbf{i}} (B|_{\tau_\nu})_{\mathbf{i}} \in \mathbb{R}^{\mathcal{L}} \quad (\text{cost } O(rN_{\mathcal{P}}n^{2p})).$$

Algorithm 1''.

(a'') Compute (by agglomeration) the function generated tensor

$$A = \bigcup_{\nu \in \mathcal{I}_{\mathcal{P}}} \sum_{k=1}^r U_{k,\nu} \otimes Y_{k,\nu} \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$$

with $U_{k,\nu}, Y_{k,\nu}$ given in (3.11) (cost $O(rN_{\mathcal{P}}n^p)$).

(b'') For $k = 1, \dots, r$, $\nu \in \mathcal{I}_{\mathcal{P}}$, compute the inverse FFT (in the index \mathbf{l}),

$$W_{k,\nu} = (F_{\mathcal{J} \leftarrow \mathcal{L}}^{-1} C_{k,\nu})_{|\sigma_\nu} \in \mathbb{R}^{\sigma_\nu} \quad (\text{cost } O(rN_{\mathcal{P}}n^p \log n)).$$

(c'') Evaluate the agglomerated Hadamard product (in the index \mathbf{j}),

$$V = \biguplus_{\{\nu \in \mathcal{I}_{\mathcal{P}}\} \rightarrow \mathcal{J}} \sum_{k=1}^r Y_{k,\nu} \odot W_{k,\nu} \in \mathbb{R}^{\mathcal{J}} \quad (\text{cost } O(rN_{\mathcal{P}}n^p)).$$

(d'') Compute the FFT $Q_+ = F_{\mathcal{L} \leftarrow \mathcal{J}} V \in \mathbb{R}^{\mathcal{L}}$ in the index \mathbf{j} (cost $O(n^p \log n)$).

Lemma 3.5. *Under assumption (3.7), Algorithm 1'' is algebraically equivalent to Algorithm 1 and has the overall cost $O(rN_{\mathcal{P}}n^p \log n)$.*

Proof. The equivalence of the above algorithms follows from applying Lemma 2.4 with $U = U_{k,\nu}$, $Y = Y_{k,\nu}$, $T = F_{\mathcal{J} \leftarrow \mathcal{L}}^{-1}$. In fact, with W defined as in Algorithm 1, item (b), we have

$$\begin{aligned} V &= \sum_{k=1}^r \left(\bigcup_{\nu \in \mathcal{I}_{\mathcal{P}}} U_{k,\nu} \otimes Y_{k,\nu} \right) \odot W \\ &= \sum_{k=1}^r [\bigcup_{\nu \in \mathcal{I}_{\mathcal{P}}} U_{k,\nu} \otimes Y_{k,\nu}, T \cdot B]_{\mathcal{I}} \\ &= \sum_{k=1}^r \biguplus_{\{\nu \in \mathcal{I}_{\mathcal{P}}\} \rightarrow \mathcal{J}} [U_{k,\nu} \otimes Y_{k,\nu}, (T \cdot B)_{|\tau_\nu}]_{|\tau_\nu \times \sigma_\nu} \\ &= \biguplus_{\{\nu \in \mathcal{I}_{\mathcal{P}}\} \rightarrow \mathcal{J}} \sum_{k=1}^r Y_{k,\nu} \odot (T \cdot [U_{k,\nu}, B|_{\tau_\nu}])_{|\sigma_\nu} = \biguplus_{\{\nu \in \mathcal{I}_{\mathcal{P}}\} \rightarrow \mathcal{J}} \sum_{k=1}^r Y_{k,\nu} \odot W_{k,\nu}. \end{aligned}$$

The complexity bound is obtained by summing the corresponding cost estimates over all four steps (a'')–(d'') above. \square

3.3. Model 1D case. It is instructive to discuss the case $p = 1$. In this situation, the corresponding functions simplify to

$$(3.12) \quad g_{1,\lambda}(u, y) = |u|^\lambda \operatorname{sinc} \left(\frac{|u| |y|}{\pi} \right), \quad u, y \in \mathbb{R},$$

$$(3.13) \quad g_{2,\lambda}(u, y) = \frac{|u - y|^\lambda}{\sqrt{u^2 + y^2 + 2|uy|}}.$$

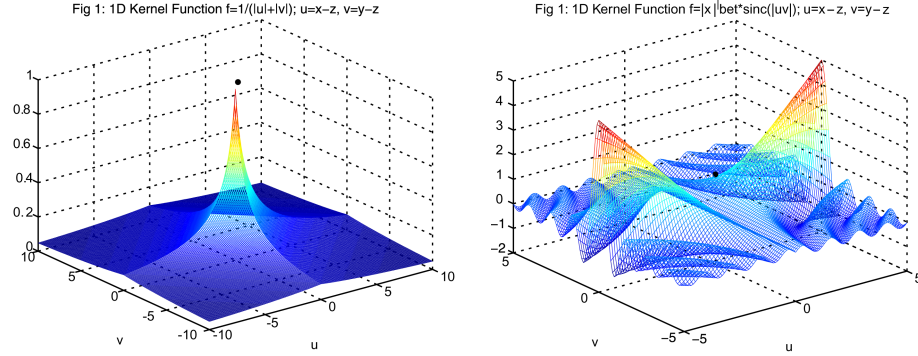


FIGURE 1. Functions $g_{m,\lambda}(u, y)$ for $m = 2, \lambda = 0$ (left) and $m = 1, \lambda = 1$ (right).

Note that in the case $p = 1, 2$, the kernel functions $g_{1,\lambda}(u, y)$, $g_{2,\lambda}(u, y)$ do not correspond to some physically relevant models, however, the corresponding approximation results can be directly adapted to the three-dimensional case.

The full matrix representation leads to $O(n^2)$ -complexity, while our Kronecker tensor-product approximation in the case (3.13) reduces the cost to $O(n \log^\beta n)$. In the case (3.12) the corresponding scheme does not improve the full matrix representation.

3.3.1. Kernel function $g_{1,\lambda}$. First, we consider the family of function $g_{1,\lambda}(u, y)$, $u \in [-1, 1]$, $y \in [-LY, LY]$ with L, Y defined in (3.7). Function $g_{1,\lambda}(u, y)$ can be approximated by sinc interpolation in the variable u (cf. Figure 1, right). We construct a separable approximation to $g_{1,\lambda}(u, y)$ in the upper half-plane $\Omega_+ = [-1, 1] \times \mathbb{R}_+$, and then extend it to the whole plane by symmetry relations. Suppose that $\lambda > 0$. Applying Corollary A.2, we can prove

Lemma 3.6. *For each $\lambda > 0$, the function $g_{1,\lambda}(u, y)$ has a separable approximation which converges exponentially,*

$$(3.14) \quad \left| g_{1,\lambda}(u, y) - \sum_{k=-M}^M g_{1,\lambda}(\phi(k\mathfrak{h}), y) S_{k,\mathfrak{h}}(\phi^{-1}(u)) \right| \leq C \frac{\exp(\delta LY)}{2\pi\delta} e^{-\pi\delta M / \log M},$$

where $S_{k,\mathfrak{h}}$ is the k th sinc function (cf. (A.2)). The tolerance $\varepsilon > 0$ can be achieved with $M = O(LY + |\log \varepsilon|)$, which corresponds to the separation rank $r = M + 1$.

Proof. We check the conditions of Corollary A.2. We observe that the function $g_{1,\lambda}(u, y)$ already has the required form with $\alpha = \lambda > 0$ and $g(x, y) = \text{sinc}(|x|y)$. Since $\text{sinc}(z)$, $z \in \mathbb{C}$, is an entire function, we obtain for the transformed function $f = \phi(x)^\lambda g(x, y)$ that for each fixed $y \in [-LY, LY]$, $f \in H^1(D_\delta)$ with $\delta < \pi$, where the space $H^1(D_\delta)$ is defined in Appendix A. Next, we estimate the constant $N(f, D_\delta)$ depending on L . Due to the trigonometric formula

$$\sin(x + iy) = \sin(x) \cosh(y) + i \cos(x) \sinh(y), \quad x, y \in \mathbb{R},$$

we conclude that for $|y| \leq LY$, the estimate $N(f, D_\delta) \leq C \exp(\delta LY)$ holds. Now we can apply the bound (A.9) to obtain (3.14). Furthermore, since the approximated function is symmetric with respect to $x = 0$, we conclude that the expansion in

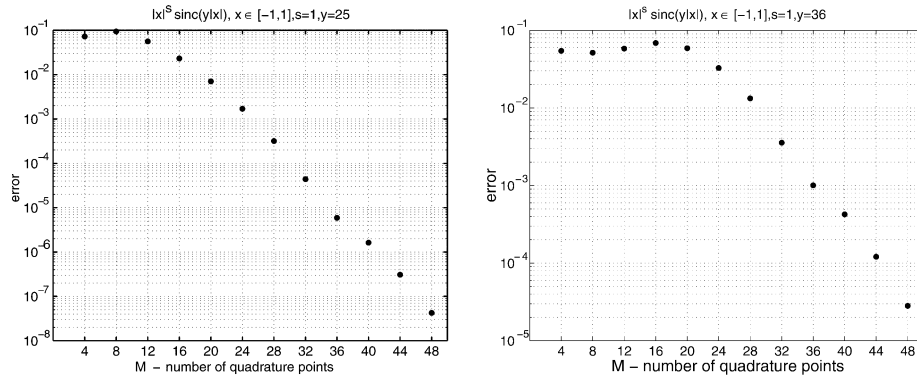


FIGURE 2. L^∞ -error of the *sinc*-interpolation to $|x|^\beta \text{sinc}(|x|y)$, $x \in [-1, 1]$, $y \in [25, 36]$, $\beta = 1$.

(3.14) can be represented by only $r = M + 1$ terms. The bound on M is now straightforward. \square

The exponential convergence in (3.14) is illustrated by numerical example presented in Figure 2 (corresponding to the choice $LY = 4^2, 5^2, 6^2$). One can observe the theoretical behaviour $M = O(LY + |\log \varepsilon|)$. Since $Y = \frac{\pi n}{2L}$, we conclude that $M = O(n)$. Hence for $p = 1$ we arrive at the asymptotical complexity $O(n^2)$.

Note that the general case $\lambda \in (-3, 1]$ can be treated by using a proper weight function (cf. Corollary A.2 under the condition $\lambda + \alpha > 0$), thus all the previous results obtained for $\lambda > 0$ remain valid.

3.3.2. Kernel function $g_{2,\lambda}$. We proceed with the function $g_{2,\lambda}(u, y)$ defined by (3.13). Due to Lemma 2.3 and Corollary A.6 (cf. the Appendix), the desired separable approximation of $g_{2,\lambda}(u, y)$ can be constructed by the Hadamard product of the expansion for $\|u - y\|^\lambda$ and by the corresponding one designed for $g_{2,0}(u, y)$. Hence, without loss of generality, we focus on the case $\lambda = 0$. Denoting $\rho = u^2 + y^2 + 2|uy|$, we can write $g_{2,0}(u, y) = 1/\sqrt{\rho}$. Note that the function $g_{2,0}(u, y)$ has only a point singularity at $u = y = 0$ (see Figure 1, left), while in general $g_{2,\lambda}(u, y)$, $\lambda > 0$, possesses a line singularity at $u = y$.

We derive the separable expansion in two steps. First, applying Lemma A.4, we obtain the exponentially convergent quadrature representation

$$\begin{aligned}
 (3.15) \quad g_{2,0}(u, y) &\approx \mathfrak{h} \sum_{k=-M}^M \cosh(k\mathfrak{h}) F(\rho, \sinh(k\mathfrak{h})) \\
 &= \sum_{k=-M}^M c_k \exp(-\mu_k(u^2 + y^2 + 2|uy|))
 \end{aligned}$$

with $F(\rho, u)$ given by (A.12) and with

$$c_k = \frac{2 \cosh(k\mathfrak{h})}{\sqrt{\pi}[1 + \exp(-\sinh(k\mathfrak{h}))]}, \quad \mu_k = \log^2[1 + \exp(\sinh(k\mathfrak{h}))], \quad \mathfrak{h} = \frac{\log M}{M}$$

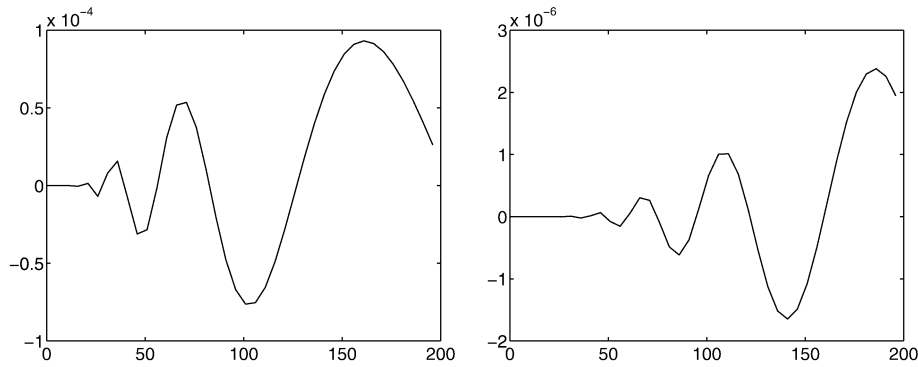


FIGURE 3. The *sinc*-quadrature error for $1/\sqrt{\rho}$, $\rho \in [1, 200]$, where $r = 31$ (left), $r = 61$ (right).

TABLE 1. Best approximation to $1/\sqrt{\rho}$ in L^∞ - and $L^2([1, R])_w$ -norm

R	10	50	100	200	$\ \cdot\ _{L^\infty}$	$w(\rho) = 1/\sqrt{\rho}$
$r = 4$	$3.7_{10}-4$	$9.6_{10}-4$	$1.5_{10}-3$	$2.2_{10}-3$	$1.9_{10}-3$	$4.8_{10}-3$
$r = 5$	$2.8_{10}-4$	$2.8_{10}-4$	$3.7_{10}-4$	$5.8_{10}-4$	$4.2_{10}-4$	$1.2_{10}-3$
$r = 6$	$8.0_{10}-5$	$9.8_{10}-5$	$1.1_{10}-4$	$1.6_{10}-4$	$9.5_{10}-5$	$3.3_{10}-4$
$r = 7$	$3.5_{10}-5$	$3.8_{10}-5$	$3.9_{10}-5$	$4.7_{10}-5$	$2.2_{10}-5$	$8.1_{10}-5$

(cf. Remark A.7). A numerical example confirming the exponential convergence of the quadrature (3.15) is presented in Figure 3. This quadrature is asymptotically optimal, however, the best approximation by exponential sums (cf. [5, 16]) systematically reduces the separation rank (see the discussion in §2.3 and compare Figure 3 with Table 1).

For a comparison with the quadrature approximation, the next table gives errors of the best r -term approximation by exponential sums in a weighted $L^2([1, R])$ -norm for different values of $R = 10, 50, 100, 200$ and with $W(\rho) = 1/\rho$. In fact, we solve the minimization problem (A.17) with $\mu = 1/2$. The last column corresponds to the weight-function $W(\rho) = 1/\sqrt{\rho}$ and $R = 200$, while a column marked by $\|\cdot\|_{L^\infty}$ corresponds to the best approximation in the $\|\cdot\|_{L^\infty([1, 200])}$ -norm.⁴ All calculations have been performed by the MATLAB subroutine FMINS based on the global minimisation by direct search. The initial guess was taken from the $\|\cdot\|_{L^\infty([1, 200])}$ -norm approximation [6] with further extrapolation for the sequence of parameters $R \in [10, 200]$.

Each factor $\exp(-\mu_k(u^2 + y^2))$ in (3.15) is separable. Hence, in a second step, we approximate the function $\exp(-\mu_k|uy|)$ for $u^2 + y^2 \geq h^2$, where h is the mesh parameter of the grid ω_u . By a proper scaling, we reduce this task to the separable approximation of the function $g(x, y) = \exp(-|x|y)$ for $x \in \mathbb{R}_+$, $y \in [1, R]$, where $R \geq 1$ might be a large parameter (in particular, we have $R = O(Y \log^2 M)$).

⁴Best approximation in L^∞ -norm are discussed in D. Braess and W. Hackbusch [6]. A complete list of numerical data can be found in www.mis.mpg.de/sciomp/EXP_SUM/1_x/tabelle.

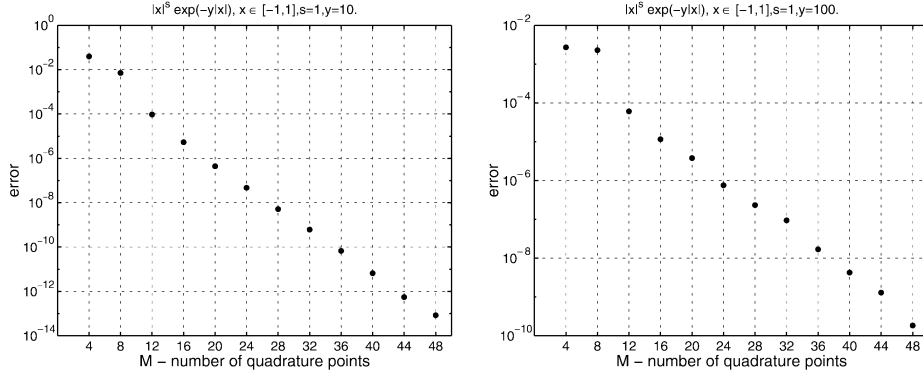


FIGURE 4. L^∞ -error of the *sinc*-interpolation applied to $\exp(-|x|y)$, $x \in [-1, 1]$, $y \in [10, 100]$.

Again, our approach is based on the *sinc*-interpolation (cf. [16, Example A.6.4] for more details). Specifically, we consider the auxiliary function

$$f(x, y) = \frac{x}{1+x} \exp(-xy), \quad x \in \mathbb{R}_+.$$

This function satisfies all the conditions of [21, Example 4.2.11] with $\alpha = \beta = 1$ (see also §2.4.2 in [12]), and hence, with the corresponding choice of interpolation points $x_k := \log[e^{k\mathfrak{h}} + \sqrt{1 + e^{2k\mathfrak{h}}}] \in \mathbb{R}_+$, it can be approximated for $y \in [1, R]$ with exponential convergence

$$(3.16) \quad \sup_{0 < x < \infty} \left| f(x, y) - \sum_{k=-M_1}^{M_1} f(x_k, y) S_{k, \mathfrak{h}}(\log\{\sinh(x)\}) \right| \leq C M_1^{1/2} e^{-c M_1^{1/2}},$$

where $S_{k, \mathfrak{h}}$ is the k th *sinc* function (cf. (A.2)), $\mathfrak{h} = 1/M_1^{1/2}$ and the constant $C = C(R)$ depends on R . The corresponding error estimate for the initial function $g(x, y)$ is given by (A.10) with $\alpha = 1$, while the separation rank is specified by $r_1 = 2M_1 + 1$. A numerical example corresponding to the interval $y \in [1, 10^2]$ is given in Figure 4.

Substituting (3.16) in (3.15) we arrive at a separable approximation to $g_{2,0}$, where the total Kronecker rank is defined by $\mathbf{r} = r r_1$. Finally, we note that the expansion derived for positive parameters $x \in (0, \infty)$, $y \in [1, R]$, can be extended to the region $\Omega_h = [-L, L] \times [-Y, Y] \setminus [-h, h]^2$, which is obtained from the computational domain $[-L, L] \times [-Y, Y]$ by removing a small vicinity of the singularity point located at the origin $u = y = 0$. The resultant expansion can be written in the form

$$(3.17) \quad g_{\mathbf{r}} := \sum_{k=1}^{\mathbf{r}} \Phi_k^1(u) \Phi_k^2(y) \approx g_{2,0}, \quad (u, y) \in \Omega_h,$$

where the set of functions $\{\Phi_k^\ell(\cdot)\}$, $\ell = 1, 2$, is given explicitly by the previous construction.

3.4. Multi-dimensional approximation.

3.4.1. *Application to the kernel $g_{1,\lambda}$ with $p \geq 2$.* Let $p \geq 2$. Then the Kronecker-product approximation A_r to A is based on the results for the 1D case. Due to Lemma 3.6, we obtain the separable expansion (3.7) with $r = 2M + 1$ and with

$$a_k(u) = S_{k,\mathfrak{h}}(\phi^{-1}(\|u\|)), \quad b_k(v) = g_{1,\lambda}(\phi(k\mathfrak{h}), \|v\|).$$

Due to the results in §3.3.1, the choice $r = O(|\log(\varepsilon)| + n)$ ensures that the approximation error is of order $O(\varepsilon)$. Applying the approximation results from Lemma 2.5 we obtain

$$\|A - A_r\|_\infty \leq C\varepsilon$$

for the related Kronecker rank- r tensor A_r . Now we are in the position to apply Algorithm 1', hence the resultant complexity is estimated by $O(rn^p \log n)$ with $r = O(n)$.

3.4.2. *Application to the kernel $g_{2,\lambda}$ with $p \geq 2$.* In contrast to the conventional tensor-product approximation applied to the original tensor A (cf. [18, 16]), in the case of kernel function $g_{2,\lambda}$, the representation of the form (3.8) can be valid only for the matrix blocks corresponding to certain multilevel block decomposition of A . Relying on Corollary A.6, we again consider the case $\lambda = 0$. Compared with the case $p = 1$, the approximation process requires a more complicated hierarchical construction, since the generating function $g_{2,\lambda}$ now depends on the scalar product $\langle u, v \rangle$. Our approximation method includes three steps.

In the *first step* we use the “one-dimensional” result and apply the exponentially convergent expansion (3.15) to the function $g_{2,\lambda}(u, y)$ to obtain

$$(3.18) \quad g_{2,0}(u, y) \approx \sum_{k=-M}^M c_k e^{-\mu_k(\|u\|^2 + \|y\|^2)} e^{-2\mu_k|\langle u, y \rangle|}, \quad u, y \in \mathbb{R}^p,$$

where $c_k, \mu_k \in \mathbb{R}$, are given explicitly (cf. §3.2.2).

In the *second step*, we focus on the separable approximation to the “coupled” term $e_k(x, y) := \exp(-2\mu_k|\langle u, y \rangle|)$, $u, y \in \mathbb{R}^p$. We construct a multilevel admissible block partitioning $\mathcal{P} = \mathcal{P}(\mathcal{I} \otimes \mathcal{J})$ of the index set $\mathcal{I} \otimes \mathcal{J}$ such that $\#(\mathcal{P}) = O(n^{p-1})$. Moreover, on each geometrical image $X(b) := \{\zeta_\alpha \in \omega_d : \alpha \in b\}$ with $b \in \mathcal{P}$, there is a separable approximation

$$(3.19) \quad e_k(x, y) \approx \left(\sum_{k=-M_1}^{M_1} U_{k,1}(u_1) Y_{k,1}(y_1) \right) \cdots \left(\sum_{k=-M_1}^{M_1} U_{k,p}(u_p) Y_{k,p}(y_p) \right),$$

$(u, y) \in X(b)$. To obtain the partitioning \mathcal{P} , we construct a multilevel decomposition of the computational domain $\Omega = [-L, L]^p \times [-Y, Y]^p$. For the current discussion we can fix $L = Y = 1$.

Let us introduce separation variables $s_m = u_m y_m \in [-1, 1]$, $m = 1, \dots, p$. Now we construct a zero-level hierarchical decomposition \mathcal{D}_0 of a domain $S := [-1, 1]^p$ with respect to the separation hyperplane $Q_p := \{(s_1, \dots, s_m) : s_1 + \dots + s_p = 0\}$, which is, in fact, the singularity set for the exponential function of interest $e_k(x, y)$ (cf. (3.18)). The decomposition is defined by a hierarchical partitioning of S using the tensor-product binary tree $\mathcal{T}_p := \underbrace{\mathcal{T} \otimes \dots \otimes \mathcal{T}}_p$ based on a weak admissibility

criteria (cf. [17] describing the corresponding \mathcal{H} -matrix construction). Specifically, we have $\mathcal{T} = \{t_0, t_1, \dots, t_{L_0}\}$ with $t_0 = [-1, 1]$, $t_\ell := [2^{1-\ell}(i-1) - 1, 2^{1-\ell}i - 1]$, $i =$

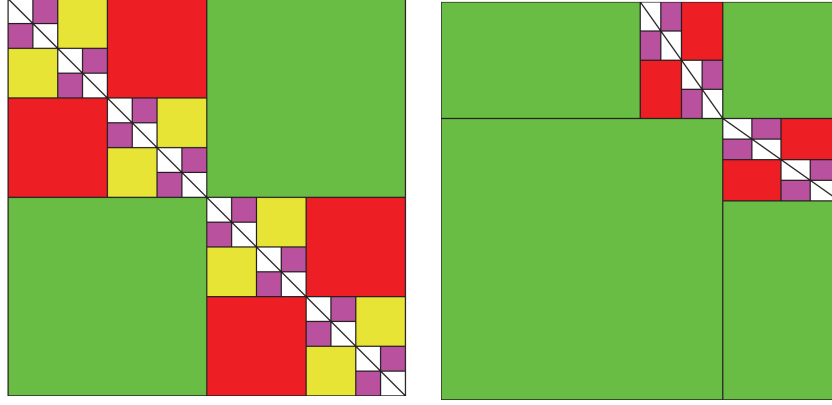


FIGURE 5. Zero-level decomposition with $\omega = (0, 4)$ (left), first-level partitioning with $\omega = (1, 3)$ (right).

$1, 2, \dots, 2^\ell\}$. The levels of the decomposition will be numbered by a double index $\omega = (0, \ell)$, $\ell = 0, 1, \dots, L_0$. A block $b \in \mathcal{D}_0$ on level ℓ is called *admissible* if it lies on one side of the separation hyperplane Q_p , that is, $Q_p \cap (b \setminus \partial b) = \emptyset$ (cf. Figure 5, left, corresponding to $p = 2$, $L_0 = 4$; here the separation hyperplane Q_2 is depicted by a diagonal line). In particular, on each level $\ell = 0, 1, \dots, L_0$, we have 2^ℓ subdomains of the size $2^{1-\ell} \times 2^{1-\ell}$.

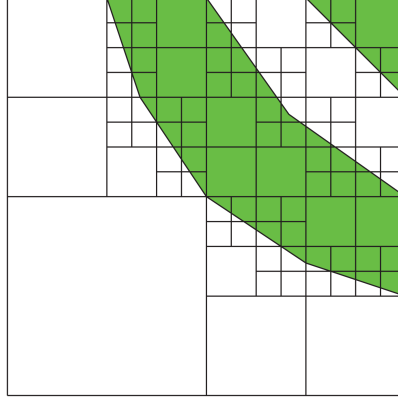
On every subdomain of the decomposition \mathcal{D}_0 , we can fix the sign of s_m , $m = 1, \dots, p$, as well as the sign of the scalar product $\langle u, y \rangle = s_1 + \dots + s_p$, hence the corresponding exponent will be separated.

Note that two subdomains on level $\omega = (0, 1)$ already have a “rectangular shape” in the initial variables, hence each of them results in a separable representation of the generating function $e_k(u, y)$. In turn, each subdomain on level $\ell \geq 2$ defined by the decomposition \mathcal{D}_0 of the parameter domain S will be further represented using a hierarchical decomposition by tensor-product subdomains in the initial variables.

To proceed, we consider subdomains on level $\omega = (0, 2)$. For $p = 2$ we have four regions in parameter domain, each of which has to be further decomposed in terms of the initial variables. For example, the right top subdomain on this level is described by $1/2 \leq u_2 y_2 \leq 1$, $0 \leq -u_1 y_1 \leq 1/2$. The three-level decomposition of this domain is depicted in Figure 5, right. One has to represent each of the two subdomains separated by the hyperbolas $u_1 y_1 = u_2 y_2 = 1/2$ by using only a few rectangular regions (optimal packing problem). The remaining three subdomains in level $\omega = (0, 2)$ are treated completely similar. On level $\omega = (0, 3)$ we have eight regions, each of which is determined by a pair of hyperbolas in the variables u_i, y_i , $i = 1, \dots, p$. For example, the right top subdomain is specified by $3/4 \leq u_2 y_2 \leq 1$, $1/4 \leq -u_1 y_1 \leq 1/2$. The example of a partitioning on level $\omega = (2, 4)$ is given in Figure 6.

In the final *third step*, we represent all blocks in A corresponding to the admissible partitioning \mathcal{P} , in the format (3.8),

$$(3.20) \quad A|_{b_\nu} \approx \sum_{m=-M_0}^{M_0} b_m U_{m,\nu} \otimes Y_{m,\nu}, \quad b_\nu \in \mathcal{P},$$

FIGURE 6. Second-level partitioning with $\omega = (2, 4)$.

with $M_0 = M_1^p$. Since the block-wise representation (3.20) is valid, Algorithm 1'' can be applied (cf. §3.2.3).

The approximation methods proposed in this section yield exponential convergence in r , $r_1 = 2M_1 + 1$, which means that both r and r_1 depend logarithmically on the tolerance ε , i.e., $r = \mathcal{O}(|\log \varepsilon|^q)$, $q \in [1, 2]$ (the same for r_1). It is easy to see that $N_{\mathcal{P}} = \mathcal{O}(n^{p-1})$, hence Lemma 3.5 leads to the overall complexity $\mathcal{O}(n^{2p-1} \log^\beta n)$.

APPENDIX A. SEPARABLE APPROXIMATION TO MULTIVARIATE FUNCTIONS

If a function of $\rho = \sum_{i=1}^d x_i \in [a, b] \subset \mathbb{R}$ can be written as the integral

$$g(\rho) = \int_{\mathbb{R}} e^{\rho F(t)} G(t) dt$$

and if quadrature can be applied, one obtains the separable approximation

$$g(x_1 + \cdots + x_d) \approx \sum_{\nu} c_{\nu} G(x_{\nu}) \prod_{i=1}^d e^{x_i F(x_{\nu})}.$$

Based on the results in [16], in §§A.1, A.2, we discuss the asymptotically optimal *sinc* quadratures. For a class of completely monotone functions one can employ the best approximation by exponential sums. For the general class of analytic functions $g : \mathbb{R}^d \rightarrow \mathbb{R}$, the tensor-product *sinc* interpolation does the job (cf. §A.1).

A.1. Sinc interpolation and quadratures. In this section, we describe *sinc*-quadrature rules to compute the integral

$$(A.1) \quad I(f) = \int_{\mathbb{R}} f(\xi) d\xi,$$

and the *sinc*-interpolation method to represent the function f itself. We introduce the family $H^1(D_{\delta})$ of all complex-valued functions (using the conventional notations from [21]), which are analytic in the strip $D_{\delta} := \{z \in \mathbb{C} : |\Im z| \leq \delta\}$, such that for each $f \in H^1(D_{\delta})$, we have

$$N(f, D_{\delta}) := \int_{\partial D_{\delta}} |f(z)| |dz| = \int_{\mathbb{R}} (|f(x + i\delta)| + |f(x - i\delta)|) dx < \infty.$$

Let

$$(A.2) \quad S_{k,\mathfrak{h}}(x) = \frac{\sin[\pi(x - k\mathfrak{h})/\mathfrak{h}]}{\pi(x - k\mathfrak{h})/\mathfrak{h}} \equiv \text{sinc}\left(\frac{x}{\mathfrak{h}} - k\right) \quad (k \in \mathbb{Z}, \mathfrak{h} > 0, x \in \mathbb{R})$$

be the k th *sinc* function with step size \mathfrak{h} , evaluated at x (cf. (3.5)). Given $f \in H^1(D_\delta)$, $\mathfrak{h} > 0$, and $M \in \mathbb{N}_0$, the corresponding truncated *sinc* interpolant (cardinal series representation) and *sinc* quadrature read as

$$(A.3) \quad C_M(f, \mathfrak{h})(x) = \sum_{\nu=-M}^M f(\nu\mathfrak{h}) S_{\nu,\mathfrak{h}}(x), \quad x \in \mathbb{R}, \quad \text{and} \quad T_M(f, \mathfrak{h}) = \mathfrak{h} \sum_{\nu=-M}^M f(\nu\mathfrak{h}),$$

respectively. The issue is therefore to estimate the interpolation and quadrature errors. Adapting the basic theory from [21], the following error estimates are proven in [16].

Proposition A.1. *Let $f \in H^1(D_\delta)$. If, in addition, f satisfies the condition*

$$(A.4) \quad |f(\xi)| \leq C \exp(-be^{a|\xi|}) \quad \text{for all } \xi \in \mathbb{R} \text{ with } a, b, C > 0,$$

then, under the choice $\mathfrak{h} = \log(\frac{2\pi a M}{b}) / (aM)$, the error of $T_M(f, \mathfrak{h})$ satisfies

$$(A.5) \quad |I(f) - T_M(f, \mathfrak{h})| \leq C N(f, D_\delta) e^{-2\pi\delta a M / \log(2\pi a M/b)}.$$

Moreover, with the same choice of \mathfrak{h} as above, the error of $C_M(f, \mathfrak{h})$ satisfies

$$(A.6) \quad |f - C_M(f, \mathfrak{h})| \leq C \frac{N(f, D_\delta)}{2\pi\delta} e^{-2\pi\delta a M / \log(2\pi a M/b)}.$$

In the case $\omega = \mathbb{R}_+$ one has to substitute the integral (A.1) by $\xi = \varphi(z)$ such that $\varphi : \mathbb{R} \rightarrow \mathbb{R}_+$ is a bijection. This changes f into $f_1 := \varphi' \cdot (f \circ \varphi)$. Assuming $f_1 \in H^1(D_\delta)$, one can apply Proposition A.1 to the transformed function.

For further applications, we reformulate the previous result for parameter dependent functions $g(x, y)$ defined on the reference interval $x \in (0, 1]$. Following [16], we introduce the mapping

$$(A.7) \quad \zeta \in D_\delta \mapsto \phi(\zeta) = \frac{1}{\cosh(\sinh(\zeta))}, \quad \delta < \frac{\pi}{2}.$$

Let $D_\phi(\delta) := \{\phi(\zeta) : \zeta \in D_\delta\} \supset (0, 1]$ be the image of D_δ . One easily checks that if a function g is holomorphic on $D_\phi(\delta)$, then

$$f(\zeta) := \phi^\alpha(\zeta) g(\phi(\zeta)) \quad \text{for any } \alpha > 0$$

is also holomorphic on D_δ .

Note that the finite *sinc* interpolation $C_M(f(\cdot, y), \mathfrak{h}) = \sum_{k=-M}^M f(k\mathfrak{h}, y) S_{k,\mathfrak{h}}$, together with the back-transformation $\zeta = \phi^{-1}(x) = \text{arsinh}(\text{arcosh}(\frac{1}{x}))$ and multiplication by $x^{-\alpha}$, yield the separable approximation g_M of the function $g(x, y)$ we are interested in,

$$(A.8) \quad g_M(x, y) := \sum_{k=-M}^M \phi(k\mathfrak{h})^\alpha g(\phi(k\mathfrak{h}), y) \cdot x^{-\alpha} S_{k,\mathfrak{h}}(\phi^{-1}(x)) \approx g(x, y)$$

with $x \in (0, 1] = \phi(\mathbb{R})$, $y \in Y$. Since $\phi(\zeta)$ in (A.7) is an even function, the separation rank is given by $r = M + 1$ if g is even and by $r = 2M + 1$ in the general case. The error analysis is presented in the following statement.

Corollary A.2. *Let $Y \in \mathbb{R}^m$ be any parameter set and assume that for all $y \in Y$ the functions $g(\cdot, y)$ as well as their transformed counterparts $f(\zeta, y) := \phi^\alpha(\zeta)g(\phi(\zeta), y)$ satisfy the following conditions:*

- (a) $g(\cdot, y)$ is holomorphic on $D_\phi(\delta)$, and $\sup_{y \in Y} N(f(\cdot, y), D_\delta) < \infty$;
- (b) $f(\cdot, y)$ satisfies (A.4) with $a = 1$ and certain C, b for all $y \in Y$.

Then the optimal choice $\mathfrak{h} := \frac{\log M}{M}$ of the step size yields the pointwise error estimates

$$(A.9) \quad |E_M(f, \mathfrak{h})(\zeta)| := |f(\zeta, y) - C_M(f, \mathfrak{h})(\zeta)| \leq C \frac{N(f, D_\delta)}{2\pi\delta} e^{-\pi\delta M / \log M},$$

$$(A.10) \quad |g(x, y) - g_M(x, y)| \leq |x|^{-\alpha} |E_M(f(\cdot, y), \mathfrak{h})(\phi^{-1}(x))|, \quad x \in (0, 1].$$

Corollary A.2 and, in particular, estimate (A.10) are applied in §3.

Remark A.3. The blow-up at $x = 0$ is avoided by restricting x to $[h, 1]$ ($h > 0$). In applications with a discretisation step size h , it suffices to apply this estimate for $|x| \geq \text{const} \cdot h$. Since usually $1/h = \mathcal{O}(n^\beta)$ for some $\beta > 0$ (and n the problem size), the factor $|x|^{-\alpha}$ is bounded by $\mathcal{O}(n^{\alpha\beta})$ and can be compensated by the exponential decay in (A.9) with respect to M .

A.2. Sinc quadratures for the Gauss and Laplace transforms. In the following, we recall the asymptotically optimal quadrature for the Gauss integral (cf. [16] for more details)

$$\frac{1}{\rho} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-\rho^2 t^2} dt,$$

presented in the form

$$(A.11) \quad \frac{1}{\rho} = \int_{\mathbb{R}} f(w) dw \quad \text{with}$$

$$(A.12) \quad f(w) = \cosh(w) F(\rho, \sinh(w)), \quad F(\rho, u) := \frac{2}{\sqrt{\pi}} \frac{e^{-\rho^2 \log^2(1+e^u)}}{1 + e^{-u}}.$$

Lemma A.4 ([16]). *Let $\delta < \pi/2$. Then for the function f in (A.12) we have $f \in H^1(D_\delta)$, and, in addition, the condition (A.4) is satisfied with $a = 1$. Let $\rho \geq 1$. Then the $(2M+1)$ -point quadrature (cf. Proposition A.1) with the choice $\delta(\rho) = \frac{\pi}{C + \log(\rho)}$, $C \geq 0$, allows the error bound*

$$(A.13) \quad |I(f) - T_M(f, \mathfrak{h})| \leq C_1 \exp\left(-\frac{\pi^2 M}{(C + \log(\rho)) \log M}\right).$$

To treat the case $\lambda \neq 0$ in (3.4), (3.6), first, we consider the *sinc* quadratures for the integral

$$(A.14) \quad \frac{1}{\rho^\mu} = \frac{1}{\Gamma(\mu)} \int_{\mathbb{R}_+} \xi^{\mu-1} e^{-\xi\rho} d\xi, \quad \mu > 0.$$

Note that in the case $\mu = 1$, the problem is solved in [16]. Similarly, here we propose to parametrise (A.14) by using the substitution $\xi = \log(1 + e^u)$ resulting in

$$\frac{1}{\rho^\mu} = \int_{\mathbb{R}} f_1(u) du, \quad f_1(u) := \frac{1}{\Gamma(\mu)} [\log(1 + e^u)]^{\mu-1} \frac{e^{-\rho \log(1+e^u)}}{1 + e^{-u}}.$$

Then a second substitution $u = \sinh(w)$ leads to (with $g(w) = 1 + e^{\sinh(w)}$)

$$(A.15) \quad \frac{1}{\rho^\mu} = \int_{\mathbb{R}} f_2(w) dw, \quad f_2(w) = \frac{1}{\Gamma(\mu)} \frac{[\log(g(w))]^{\mu-1} \cosh(w)}{1 + e^{-\sinh(w)}} e^{-\rho \log(g(w))}.$$

Lemma A.5. *Let $\delta < \pi/2$. Then for the function f_2 defined by (A.15) we have $f_2 \in H^1(D_\delta)$, and, in addition, the condition (A.4) is satisfied with $a = 1$. Let $\rho \geq 1$. Then the $(2M + 1)$ -point quadrature (cf. Proposition A.1) with the choice $\delta(\rho) = \frac{\pi}{C + \log(\rho)}$, $C > 0$, allows the error bound*

$$(A.16) \quad |I(f_2) - T_M(f_2, \mathbf{h})| \leq C_1 \exp\left(-\frac{\pi^2 M}{(C + \log(\rho)) \log M}\right).$$

Proof. We apply Proposition A.1. First, we observe the double exponential decay of f_2 on the real axis

$$f_2(w) \approx \frac{1}{2} (\sinh(w))^{\mu-1} e^{w - \frac{\rho}{2} e^w}, \quad w \rightarrow \infty; \quad f_2(w) \approx \frac{1}{2} e^{w(\mu-1)|w| - \frac{1}{2} e^{|w|}}, \quad w \rightarrow -\infty.$$

We check that $f_2(w) \in H^1(D_\delta)$. Since the zeros of $1 + e^{\sinh(w)}$ are outside of D_δ , we conclude that $f_2(w)$ is analytic in D_δ . Finally, we can prove that the current choice of δ leads to $N(f_2, D_\delta) < \infty$ uniformly in ρ (the proof is similar to those in the case $\mu = 1$ considered in [16]). Finally, due to Proposition A.1, we obtain the error bound (A.5), which leads to (A.16). \square

As a direct consequence of Lemma A.5, we obtain the following result.

Corollary A.6. *The function $\|u - v\|^\lambda$, $\lambda > 0$, allows a low separation rank approximation at the cost $O(n \log^3 n)$ based on the factorization*

$$\|u - v\|^\lambda = \|u - v\|^{2m} \frac{1}{\|u - v\|^{2m-\lambda}} \quad \text{with } m \in \mathbb{N},$$

such that $2m - \lambda > 0$, where the first term on the right-hand side is already separable. The second factor can be approximated by a standard sinc quadrature applied to the integral (A.14). This can be improved by solving the minimisation problem to obtain the best approximation by exponential sums. In particular, in the case $\lambda \in (0, 1]$, we choose $m = 1$.

Remark A.7. The number of terms $r = 2M + 1$ in the quadratures $T_M(f, \mathbf{h})$ and $T_M(f_2, \mathbf{h})$ is asymptotically optimal. However, in large-scale computations it can be optimized by using the best approximation of $1/\rho^\mu$ by exponential sums (cf. [6, 16] for more details).

For example, the best approximation to $1/\rho^\mu$ in the weighted L^2 -norm is reduced to the minimisation of an explicitly given differentiable functional. Given $R > 1$, $\mu > 0$, $N \geq 1$, find $2N$ real parameters $\alpha_1, \omega_1, \dots, \alpha_N, \omega_N \in \mathbb{R}_{>0}$, such that

$$(A.17) \quad F_\mu(R; \alpha_1, \omega_1, \dots, \alpha_N, \omega_N) := \int_1^R W(x) \left(\frac{1}{x^\mu} - \sum_{i=1}^N \omega_i e^{-\alpha_i x} \right)^2 dx \rightarrow \min.$$

Approximating the integral (A.17) by certain quadrature, the minimization problem can be solved by the gradient method or Newton-type methods with a proper choice of the initial guess.

ACKNOWLEDGMENTS

The author would like to thank Professors W. Hackbusch and S. Rjasanow for useful discussions and comments.

REFERENCES

- [1] M. Albrecht: Towards a frequency independent incremental *ab initio* scheme for the self energy. Preprint University Siegen, 2004; accepted by Theor. Chem. Acc. (2005); DOI:10.1007/S00214-006-0085-5.
- [2] G. Beylkin and M.J. Mohlenkamp: *Numerical operator calculus in higher dimensions*. Proc. Natl. Acad. Sci. USA 99 (2002), 10246-10251. MR1918798 (2003h:65071)
- [3] G. Bird: *Molecular Gas Dynamics and the Direct Simulation of Gas Flow*. Clarendon Press, Oxford, 1994. MR1352466 (97e:76078)
- [4] A.V. Bobylev and S. Rjasanow: Fast deterministic method of solving the Boltzmann equation for hard spheres. Eur. J. Mech. B/Fluids 18 (1999) 869-887. MR1728639 (2001c:76109)
- [5] D. Braess: *Nonlinear Approximation Theory*. Springer-Verlag, Berlin, 1986. MR0866667 (88e:41002)
- [6] D. Braess and W. Hackbusch: Approximation of $1/x$ by Exponential Sums in $[1, \infty)$. Preprint 83, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig 2004. IMA J. Numer. Anal. **25** (2005), 685–697. MR2170519
- [7] Cercignani, C., Illner, R., Pulvirenti, M.: *The Mathematical Theory of Dilute Gases*. Springer, New York, 1994. MR1307620 (96g:82046)
- [8] G. Csanyi and T.A. Arias: Tensor product expansions for correlation in quantum many-body systems. Phys. Rev. B, v. 61, n. 11 (2000), 7348-7352.
- [9] H.-J. Flad, W. Hackbusch, B.N. Khoromskij, and R. Schneider: Concept of data-sparse tensor-product approximation in many-particle models. MPI MIS, Leipzig 2005 (in preparation).
- [10] I.P. Gavriluk, W. Hackbusch, and B.N. Khoromskij: *Data-sparse approximation to operator-valued functions of elliptic operators*. Math. Comp. **73** (2004), no. 247, 1297–1324. MR2047088 (2005b:47086)
- [11] I.P. Gavriluk, W. Hackbusch, and B.N. Khoromskij: *Data-sparse approximation to a class of operator-valued functions*. Math. Comp. **74** (2005), 681–708. MR2114643 (2005i:65068)
- [12] I. P. Gavriluk, W. Hackbusch and B. N. Khoromskij: Tensor-Product Approximation to Elliptic and Parabolic Solution Operators in Higher Dimensions. Computing **74** (2005), 131-157. MR2133692
- [13] I.S. Gradshteyn and I.M. Ryzhik: *Table of Integrals, Series and Products* (6th edition), Academic Press, San Diego, 2000. MR1773820 (2001c:00002)
- [14] L. Grasedyck: Existence and computation of a low Kronecker-rank approximation to the solution of a tensor system with tensor right-hand side. Computing **70** (2003), 121-165.
- [15] W. Hackbusch: *Hierarchische Matrizen – Algorithmen und Analysis*. Vorlesungsmanuskript, MPI MIS, Leipzig 2004.
- [16] W. Hackbusch and B.N. Khoromskij: Low-Rank Kronecker Tensor-Product Approximation to Multi-dimensional Nonlocal Operators. Parts I/II. Preprints 29/30, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig 2005 (Computing, to appear).
- [17] W. Hackbusch, B.N. Khoromskij, and R. Kriemann: Hierarchical matrices based on a weak admissibility criterion. Computing **73** (2004), 207-243. MR2106249
- [18] W. Hackbusch, B.N. Khoromskij and E. Tyrtyshnikov: *Hierarchical Kronecker tensor-product approximation*. Preprint 35, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2003; J. Numer. Math. **13** (2005), 119-156. MR2149863 (2006d:65152)
- [19] I. Ibragimov and S. Rjasanow: Numerical Solution of the Boltzmann Equation on the Uniform Grid. Computing 69(2002), 163-186. MR1954793 (2004i:82060)
- [20] H. Koch and A.S. de Merás: Size-intensive decomposition of orbital energy denominators. Journal of Chemical Physics, **113** (2) 508-513 (2000).
- [21] F. Stenger: *Numerical methods based on Sinc and analytic functions*. Springer-Verlag, 1993. MR1226236 (94k:65003)

- [22] E. E. Tyrtyshnikov: Tensor approximations of matrices generated by asymptotically smooth functions. *Sbornik: Mathematics*, Vol. 194, No. 5-6 (2003), 941-954 (translated from *Mat. Sb.*, Vol. 194, No. 6 (2003), 146-160.) MR1992182 (2004d:41036)
- [23] E. E. Tyrtyshnikov: Kronecker-product approximations for some function-related matrices. *Linear Algebra Appl.*, 379 (2004), 423-437. MR2039752 (2005d:15005)
- [24] C. Van Loan: The ubiquitous Kronecker product. *J. of Comp. and Applied Mathematics* **123** (2000), 85-100. MR1798520 (2001i:65055)
- [25] T. Zhang and G.H. Golub: Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. Appl.* **23** (2001), 534-550. MR1871328 (2002j:15039)

MAX-PLANCK-INSTITUTE FOR MATHEMATICS IN THE SCIENCES, INSELSTR. 22-26, D-04103
LEIPZIG, GERMANY

E-mail address: bokh@mis.mpg.de