HIERARCHICAL MATRIX APPROXIMATION FOR KERNEL-BASED SCATTERED DATA INTERPOLATION

ARMIN ISKE*, SABINE LE BORNE†, AND MICHAEL WENDE‡

Abstract. Scattered data interpolation by radial kernel functions leads to linear equation systems with large, fully populated, ill-conditioned interpolation matrices. A successful iterative solution of such a system requires an efficient matrix-vector multiplication as well as an efficient preconditioner. While multipole approaches provide a fast matrix-vector multiplication, they avoid the explicit set-up of the system matrix which hinders the construction of preconditioners, such as approximate inverses or factorizations which typically require the explicit system matrix for their construction. In this paper, we propose an approach that allows both an efficient matrix-vector multiplication as well as an explicit matrix representation which can then be used to construct a preconditioner. In particular, the interpolation matrix will be represented in hierarchical matrix format, and several approaches for the blockwise low-rank approximation are proposed and compared, of both analytical nature (separable expansions) and algebraic nature (adaptive cross approximation).

Key words. Radial basis function, scattered data interpolation, hierarchical matrices, data-sparse approximation, adaptive cross approximation

AMS subject classifications. 65D05, 41A05, 41A58, 41A63, 65F30, 15A23

1. Introduction. Radial basis functions (RBF), or, radial kernel functions, are powerful tools for meshfree interpolation from multivariate scattered data [5, 8, 9, 10, 14, 15]. In order to explain RBF interpolation only very briefly, assume we are given function values $f_X = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))^T \in \mathbb{R}^N$ sampled from a target function $f : \mathbb{R}^d \to \mathbb{R}$, for $d \geq 1$, at a set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ of pairwise distinct interpolation points. According to the RBF reconstruction scheme, an interpolant $s : \mathbb{R}^d \to \mathbb{R}$ to f is required to be of the form

$$s(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \Phi(\boldsymbol{x}, \boldsymbol{x}_j) + p(\boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

where Φ is a radially symmetric kernel function, i.e., $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(|\boldsymbol{x} - \boldsymbol{y}|_2)$ for a radial kernel $\phi : [0, \infty) \to \mathbb{R}$, and where $|\cdot|_2$ is the Euclidean norm on $\mathbb{R}^d$. Moreover, p in (1) is assumed to be a d-variate real-valued polynomial, whose degree $m - 1$ is determined by the *order* $m \in \mathbb{N}_0$ of the kernel Φ. To be more precise, we require that Φ is *conditionally positive definite* of order m on $\mathbb{R}^d$, $\Phi \in \mathbf{CPD}(m)$, i.e., for any point set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \subset \mathbb{R}^d$ of size $N = \#X$, the *kernel matrix*

$$B_{\Phi,X} = (\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j))_{1 \leq i,j \leq N} \in \mathbb{R}^{N,N} \tag{2}$$

is positive definite on the linear subspace

$$L_X = \left\{ c = (c_1, \ldots, c_N)^T \in \mathbb{R}^N : \sum_{j=1}^{N} c_j p(\boldsymbol{x}_j) = 0 \text{ for all } p \in \pi_{m-1}^d \right\} \subset \mathbb{R}^N, \tag{3}$$

*University of Hamburg, Bundesstr. 55, 20146 Hamburg, Germany (iske@math.uni-hamburg.de).

† Hamburg University of Technology, Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany (leborne@tuhh.de).

‡ Hamburg University of Technology, Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany (michael.wende@tuhh.de).

where

$$\pi_{m-1}^d := \left\{ p(\boldsymbol{x}) = \sum_{|\boldsymbol{n}|_1 < m} c_{\boldsymbol{n}} \boldsymbol{x}^{\boldsymbol{n}} \;\middle|\; \boldsymbol{n} \in \mathbb{N}_0^d,\ c_{\boldsymbol{n}} \in \mathbb{R} \right\}$$

is the space of all d-variate polynomials of degree at most $m-1$. For $m=0$ we have $L_X = \mathbb{R}^N$, in which case Φ is *positive definite* on $\mathbb{R}^d$, $\Phi \in \mathbf{PD}$.

We remark that for $\Phi \in \mathbf{CPD}(m)$ the interpolation problem $s_X = f_X$ has a unique solution s of the form (1), under constraints $c = (c_1, \ldots, c_N)^T \in L_X$ and under the (rather weak) assumption that the interpolation points X are π_{m-1}^d-*unisolvent*, i.e., any polynomial in π_{m-1}^d can uniquely be reconstructed from its values on X. In fact, the interpolation problem $s_X = f_X$ leads us, for a fixed basis $\{p_1, \ldots, p_Q\}$ of π_{m-1}^d, to the linear system

$$\begin{aligned} \sum_{j=1}^{N} c_j \Phi(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{\ell=1}^{Q} b_\ell p_\ell(\boldsymbol{x}_i) &= f(\boldsymbol{x}_i) \quad && \text{for } i = 1, \ldots, N, \\ \sum_{j=1}^{N} c_j p_i(\boldsymbol{x}_j) &= 0 && \text{for } i = 1, \ldots, Q, \end{aligned} \tag{4}$$

with unknown coefficients $c = (c_1, \ldots, c_N)^T \in \mathbb{R}^N$ for the major part of s in (1) and coefficients $b = (b_1, \ldots, b_Q)^T \in \mathbb{R}^Q$ for its polynomial part. We can rewrite the system (4) in matrix form as

$$\begin{pmatrix} B_{\Phi,X} & P_X \\ P_X^T & 0 \end{pmatrix} \begin{pmatrix} c \\ b \end{pmatrix} = \begin{pmatrix} f_X \\ 0 \end{pmatrix}, \tag{5}$$

where the *polynomial matrix*

$$P_X = (p_\ell(\boldsymbol{x}_i))_{1 \le i \le N; 1 \le \ell \le Q} \in \mathbb{R}^{N,Q}$$

is injective, due to the assumed π_{m-1}^d-unisolvence of the interpolation points X.

To goal of this paper is the approximation of the interpolation matrix $B_{\Phi,X}$ in (5) in a way that requires almost optimal storage, allows a matrix-vector multiplication of almost optimal complexity, and can be used for subsequent (approximate) matrix factorization to construct a preconditioner. But we will neither discuss the condition number of the interpolation matrix (cf. [7] for a very recent account on this problem), the actual construction of a preconditioner nor the system's iterative solution.

The remainder of this paper is organized as follows: In Section 2, we introduce the interpolation problem using radial kernels, the key ideas of hierarchical matrices, and the connection between these two topics. In Section 3, we describe (geometric and algebraic) hierarchical clustering algorithms for meshless methods which yield the hierarchical block structure of an $\mathcal{H}$-matrix. In Section 5, we show various approaches to fill the matrix blocks with low rank approximations in factored form. Approaches include analytical methods based on separable expansions of the kernel function, either tailored to a particular kernel function or kernel-independent such as interpolation or Taylor expansion, as well as an algebraic approach by the name of adaptive cross approximation. Section 6 provides a numerical illustration and comparison of the discussed methods, and finally Section 7 concludes the paper with a summary and outlook to future work.

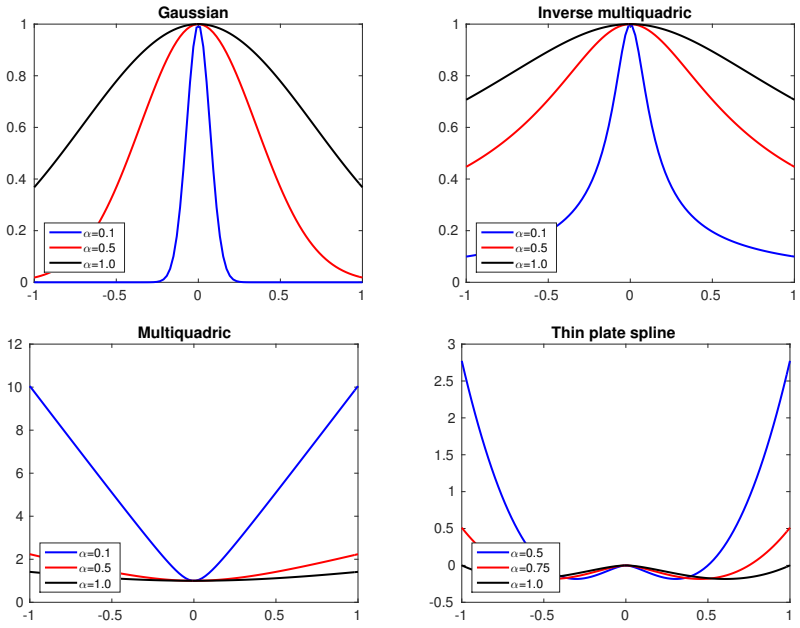


FIG. 1. *Generating functions* $\phi_{\mathrm{GAU},\alpha}, \phi_{\mathrm{IMQ},\alpha}, \phi_{\mathrm{MQ},\alpha}, \phi_{\mathrm{TPS},\alpha}$.

2. Preliminaries.

2.1. Scattered data interpolation using radial kernels. According to the formulation of the interpolation problem in (4) and the resulting matrix form in (5), we consider working with radially symmetric (multivariate) kernel functions

$$\Phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, \ (\boldsymbol{x}, \boldsymbol{y}) \mapsto \Phi(\boldsymbol{x}, \boldsymbol{y}) := \phi(|\boldsymbol{x} - \boldsymbol{y}|_2)$$

being generated by a (univariate) function $\phi : \mathbb{R}_+ \to \mathbb{R}$. We will focus on four radial kernels $\phi = \phi_{Z,\alpha}$ with *scaling parameter* $\alpha \in \mathbb{R}_+$ and $Z \in \{\mathrm{GAU}, \mathrm{IMQ}, \mathrm{MQ}, \mathrm{TPS}\}$,

$$\begin{aligned}
\phi_{\mathrm{GAU},\alpha}(r) &= \exp\left(-\left(\tfrac{r}{\alpha}\right)^2\right) && \text{(Gaussian, GAU)},\\
\phi_{\mathrm{IMQ},\alpha}(r) &= \left(1+\left(\tfrac{r}{\alpha}\right)^2\right)^{-\frac{1}{2}} && \text{(Inverse Multiquadric, IMQ)},\\
\phi_{\mathrm{MQ},\alpha}(r) &= \left(1+\left(\tfrac{r}{\alpha}\right)^2\right)^{\frac{1}{2}} && \text{(Multiquadric, MQ)},\\
\phi_{\mathrm{TPS},\alpha}(r) &= \left(\tfrac{r}{\alpha}\right)^2 \log\left(\tfrac{r}{\alpha}\right) && \text{(Thin Plate Spline, TPS)}.
\end{aligned}$$

as shown in Figure 1. The positive definite $\phi_{\mathrm{GAU},\alpha} \in \mathbf{PD}$ and $\phi_{\mathrm{IMQ},\alpha} \in \mathbf{PD}$ generate symmetric positive definite interpolation matrices $B \equiv B_{\Phi,X}$ in (2), regardless of the choice of (pairwise distinct) data sites $\boldsymbol{x}_i$. The kernels $\phi_{\mathrm{MQ},\alpha} \in \mathbf{CPD}(1)$ and $\phi_{\mathrm{TPS},\alpha} \in \mathbf{CPD}(2)$, however, are conditionally positive definite of order $m = 1$ (for MQ) and $m = 2$ (for TPS), so that the symmetric kernel matrix $B \equiv B_{\Phi,X}$ is positive definite on the linear subspace L_X in (3) for π^d_{m-1}-unisolvent interpolation points X.

A common feature of the four resulting interpolation matrices B is that they are dense and, moreover, their condition numbers depend on the shape parameter $\alpha > 0$, deteriorating with increasing α which "flattens" the generating functions $\phi_{Z,\alpha}$, see Figure 1.

In the following section, we will introduce the technique of hierarchical ($\mathcal{H}$-) matrices which will offer a (highly accurate) approximation $B_{\mathcal{H}} \approx B$ to the interpolation matrix B which features an (almost) linear complexity for storage and matrix-vector multiplication and can be used for subsequent matrix factorization.

2.2. $\mathcal{H}$-matrices. An $\mathcal{H}$-matrix is, in short, a matrix whose block index set has been hierarchically partitioned and whose resulting matrix blocks are given in factored form whenever the rank of such a matrix block is (significantly) smaller than its size (i.e., the minimum of its numbers of rows and columns). We will now provide brief formal definitions of these concepts, for more detailed discussions we refer to the textbooks [2, 12] and the survey [13].

Let $I, J \subset \mathbb{N}$ be (finite) row/column index sets, and let $B = (b_{ij})_{i\in I, j\in J} \in \mathbb{R}^{I\times J} \cong \mathbb{R}^{\#I,\#J}$. Here, $\#I$ denotes number of elements (indices) of the set I. In the following definition, a *partition* of I is a set of nonempty, disjoint subsets whose union is the entire set I, whereas a *hierarchical partitioning* will be defined as a certain set of several nested partitions and finally, an $\mathcal{H}$-block partition is a partition consisting of subsets of a hierarchical partitioning. An actual construction of an $\mathcal{H}$-block partition as defined here will follow in Section 3.

Definition 1 (hierarchical partitioning, $\mathcal{H}$-block partition). *The sequence $P_I^\ell = \{I_1^\ell, \dots, I_{p^\ell}^\ell\}$, $\ell = 0, ..., L$, of partitions of an index set I is called a hierarchical partitioning of I of depth L if it holds that*

- $P_I^0 = \{I\}$, *(root)*
- $I_k^\ell = \bigcup_{j\in I_{sub}} I_j^{\ell+1}$ *for a subset $I_{sub} \subset \{1, \dots, p^{\ell+1}\}$, (hierarchical nestedness, i.e., an index set on level ℓ is the union of index sets on the finer level $\ell+1$).*

Given hierarchical partitionings P_I^ℓ, P_J^ℓ, $\ell = 1, \dots, \min\{L_I, L_J\}$ of index sets I, J with respective depths L_I, L_J, the sequence $P_{I\times J}^\ell = \{b_1^\ell, \dots, b_{q^\ell}^\ell\}$, $\ell = 0, ..., L$, with blocks $b_k^\ell \subset I \times J$ is called a hierarchical block partitioning of $I \times J$ of depth $L := \min\{L_I, L_J\}$ based on hierarchical (index) partitionings P_I^ℓ, P_J^ℓ if it holds that

- *$P_{I\times J}^\ell$ is a hierarchical partitioning of $I \times J$,*
- *for every b_k^ℓ there exist $\sigma \in P_I^\ell$, $\tau \in P_J^\ell$ such that $b_k^\ell = \sigma \times \tau$.*

Finally, a partition $P_{I\times J}^{\mathcal{H}} = \{b_1, \dots, b_n\}$ of $I \times J$ is called an $\mathcal{H}$-block partition if $b_i \in \bigcup_{\ell\in\{1,\dots,L\}} P_{I\times J}^\ell$ for all $i = 1, \dots, n$.

In an $\mathcal{H}$-matrix, some of its matrix blocks are represented in factored form, others as a full matrix. The distinction is based on an admissibility condition defined next.

Definition 2 (admissibility condition). *Let I, J be two index sets, and let $P_{I\times J}$ be a partition of $I \times J$. An admissibility condition is a function*

$$Adm : P_{I\times J} \to \{true, false\}.$$

We call $b \in P_{I\times J}$ admissible if $Adm(b) =$true and inadmissible otherwise.

Definition 3 ($\mathcal{H}$-matrix). *Let $P_{I\times J}^{\mathcal{H}}$ be an $\mathcal{H}$-block partition of $I \times J$, let Adm be an admissibility condition and let $k \in \mathbb{N}$. Then, the set of $\mathcal{H}$-matrices (with respect to $P_{I\times J}^{\mathcal{H}}$, Adm and k) is defined as*

$$\mathcal{H}(P_{I\times J}^{\mathcal{H}}, k, Adm) := \{B \in \mathbb{R}^{I\times J} \mid rank(B|_b) \le k \text{ for all admissible blocks } b \in P_{I\times J}^{\mathcal{H}}\}.$$

We call matrix blocks $B|_b$ admissible if $b \in P_{I\times J}^{\mathcal{H}}$ is admissible. Admissible matrix blocks can be represented in factored form, i.e., $B|_b = UV^T$ for $b = \sigma \times \tau$, $U \in \mathbb{R}^{\#\sigma,k'}$, $V \in \mathbb{R}^{\#\tau,k'}$ for some $k' \le k$.

The savings in storage and cost for matrix-vector multiplication of an $\mathcal{H}$-matrix $B_{\mathcal{H}}$ compared to a full matrix B originate in the factored low-rank representations of admissible matrix blocks. The actual amount of savings, i.e., the hopefully reduced, almost linear complexity estimate $\mathcal{O}(kN \log N)$ depends on the particular $\mathcal{H}$-block partition and admissibility condition used in the $\mathcal{H}$-matrix construction. It is fairly straightforward to see that large, admissible matrix blocks are favorable when it comes to storage and computational complexity.

The following subsection will motivate why scattered data interpolation matrices of Subsection 2.1 fit into the framework of hierarchical matrices introduced in this subsection.

2.3. Basic connection: low rank approximation of matrix blocks. The explicit construction of an interpolation matrix in $\mathcal{H}$-matrix format is simply an extension of the well-known multipole approach used for the fast matrix-vector multiplication with (scattered data) interpolation matrices B (see [1] and reference therein for a more comprehensive account on this subject). If the underlying kernel function $\Phi(\boldsymbol{x}, \boldsymbol{y})$ allows for a separable expansion

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{n=1}^{k} u_n(\boldsymbol{x}) \cdot v_n(\boldsymbol{y}) \qquad \text{for } k \in \mathbb{N} \tag{6}$$

on a subset $Q_x \times Q_y \subset \mathbb{R}^d \times \mathbb{R}^d$, then the matrix block

$$(b_{ij})_{\substack{x_i \in Q_x \\ x_j \in Q_y}} = (\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j))_{\substack{x_i \in Q_x \\ x_j \in Q_y}} = (\phi(|\boldsymbol{x}_i - \boldsymbol{x}_j|_2))_{\substack{x_i \in Q_x \\ x_j \in Q_y}}$$

of the interpolation matrix B has at most rank k, independent of its number of rows (number of data sites $\boldsymbol{x}_i$ in Q_x) and columns (number of data sites $\boldsymbol{x}_j$ in Q_y). Typically, $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is not separable but can be approximated by a separable expansion

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) \approx \Phi^k(\boldsymbol{x}, \boldsymbol{y}) := \sum_{n=1}^{k} u_n(\boldsymbol{x}) \cdot v_n(\boldsymbol{y})$$

which yields a low rank approximation

$$(b_{ij})_{\substack{x_i \in Q_x \\ x_j \in Q_y}} \approx (\Phi^k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{\substack{x_i \in Q_x \\ x_j \in Q_y}},$$

once again with its rank bounded by k independently of the size of the block. The (entrywise) error between the exact matrix block and its low rank approximation is bounded by the approximation error $|\Phi - \Phi^k|_{\infty, Q_x \times Q_y}$.

If it is known that $\Phi(\boldsymbol{x}, \boldsymbol{y})$ allows for a separable expansion, or rather, if error bounds of an expansion are known, this yields an upper bound for the rank of the respective matrix block. If the expansion is explicitly available, it can be used for the construction of the low rank factors of the matrix blocks, i.e.,

$$(\Phi^k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{\substack{x_i \in Q_x \\ x_j \in Q_y}} = UV^T \tag{7}$$

with

$$U = (u_{ij})_{\substack{x_i \in Q_x \\ j \in \{1, \ldots, k\}}}, \; u_{ij} = u_j(\boldsymbol{x}_i), \qquad V = (v_{ij})_{\substack{x_i \in Q_y \\ j \in \{1, \ldots, k\}}}, \; v_{ij} = v_j(\boldsymbol{x}_i).$$

The kernel functions $\Phi(\boldsymbol{x},\boldsymbol{y}) = \phi(|\boldsymbol{x}-\boldsymbol{y}|_2)$ which we introduced in Subsection 2.1 have in common that they (unfortunately) are not separable but allow for exponentially convergent separable approximations as long as $\boldsymbol{x}$ and $\boldsymbol{y}$ are well separated, i.e., their distance $r = |\boldsymbol{x}-\boldsymbol{y}|_2$ is "large enough" as formalized in the following definition.

DEFINITION 4 (standard admissibility condition). *Let $\{\boldsymbol{x}_i = (x_{i,1},\ldots,x_{i,d}) \in \mathbb{R}^d \mid i \in I\}$ be a set of data sites for an index set $I := \{1,\ldots,N\}$ ($N \in \mathbb{N}$). For a subset $\sigma \subset I$, we define the bounding box Q_σ as the smallest axis-parallel box containing all data sites $\{\boldsymbol{x}_i \mid i \in \sigma\}$, i.e.,*

$$Q_\sigma := [a_1,b_1]\times\ldots\times[a_d,b_d] \quad \text{with} \quad a_j := \min_{i\in\sigma}\{x_{i,j}\},\quad b_j := \max_{i\in\sigma}\{x_{i,j}\},\ (j\in\{1,\ldots,d\}). \tag{8}$$

Then, given a partition $P_{I\times I}$ and an $\eta \in \mathbb{R}_+$, the standard admissibility condition is defined by

$$Adm : P_{I\times I} \to \{true,\, false\},\qquad \sigma\times\tau \mapsto \begin{cases} true & : \ \min\{\operatorname{diam}(Q_\sigma),\operatorname{diam}(Q_\tau)\} \le \eta\ \operatorname{dist}(Q_\sigma,Q_\tau), \\ false & : \ else \end{cases} \tag{9}$$

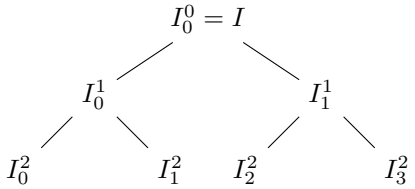
where the diameter and distance of bounding boxes are computed with respect to the Euklidean norm.

The following Section 3 introduces a (hierarchical clustering) algorithm to construct an $\mathcal{H}$-block partition suitable for an $\mathcal{H}$-matrix approximation of the interpolation matrix B. Details on the construction and convergence of separable expansions on admissible matrix blocks of this $\mathcal{H}$-block partition will follow in Section 4.

3. Construction of an $\mathcal{H}$-block partitions for scattered point sets. The hierarchical partitioning of the index set I (see Definition 1) associated with data sites $\{\boldsymbol{x}_i\}$ may be constructed through recursive (geometric) bisection. Beginning with the root $I_0^0 := I$, an index set I_k^ℓ (containing more than $n_{\min} \in \mathbb{N}$ indices, i.e., $\#I_k^\ell > n_{\min}$) is partitioned into sets $I_{2k}^{\ell+1}, I_{2k+1}^{\ell+1}$ as follows:

1. Determine the spatial direction along which the index set I_k^ℓ will be divided: $r := \operatorname{arg\,max}_{j\in\{1,\ldots,d\}}\{b_j - a_j\}$ where a_j, b_j define the boundig box $Q_{I_k^\ell}$ of I_k^ℓ in (8).
2. Set $\operatorname{mid}_r := (a_r + b_r)/2$.
3. Set $I_{2k}^{\ell+1} := \{i \in I_k^\ell \mid x_{i,r} < \operatorname{mid}_r\}$, $I_{2k+1}^{\ell+1} := I_k^\ell \setminus I_{2k}^{\ell+1}$.

The index sets I_k^ℓ are commonly associated with the nodes of a tree:



The resulting hierarchical partitioning of I and the (standard) admissibility condition (see Definition 4) are used in the following construction of a hierarchical block partitioning. Beginning with the root $I_0^0 \times I_0^0$, only blocks $I_k^\ell \times I_{k'}^\ell$ that are inadmissible and not smaller than $n_{\min}$ are recursively partitioned, i.e.,

$$\begin{aligned} &\text{if} \quad (Adm(I_k^\ell \times I_{k'}^\ell) = \text{false} \ \wedge\ \min\{\#I_k^\ell, \#I_{k'}^\ell\} > n_{\min}) \\ &\text{then partition } I_k^\ell \times I_{k'}^\ell \text{ into} \\ &\qquad \{I_{2k}^{\ell+1}\times I_{2k'}^{\ell+1},\ I_{2k+1}^{\ell+1}\times I_{2k'}^{\ell+1},\ I_{2k}^{\ell+1}\times I_{2k'+1}^{\ell+1},\ I_{2k+1}^{\ell+1}\times I_{2k'+1}^{\ell+1}\}. \end{aligned}$$

Finally, an $\mathcal{H}$-block partition $P_{I\times I}$ is obtained as the set of all blocks $I_k^\ell \times I_{k'}^\ell$ that have not been further partitioned. By construction, these blocks must either be admissible or small (with respect to $n_{\min}$).

In order to be able to associate the index blocks in the $\mathcal{H}$-block partition $P_{I\times I}$ with a matrix block of consecutive rows and columns, we define a permutation $ind2dof : I \to I$ (index-to-dof) that satisfies $ind2dof(i) < ind2dof(j)$ for any two indices i, j that have been assigned to disjoint index sets $i \in I_{2k}^{\ell+1}$ and $j \in I_{2k+1}^{\ell+1}$ in step 3. of the index partitioning described above. Such a permutation yields a relabeling of data sites and a resulting row and column permutation of the interpolation matrix.

4. Low rank approximation through separable expansions. In this part we will discuss three analytical methods to construct separable expansions Φ^k of Φ: (1) Polynomial interpolation (Section 4.1), (2) Taylor expansion (Section 4.2) and (3) a special expansion designed for multiquadric basis functions (Section 4.5). In all cases, $\Phi(\boldsymbol{x}, \boldsymbol{y})$ will be considered as a function of $\boldsymbol{x} \in Q_x$ for a fixed $\boldsymbol{y} \in Q_y$. The length k of the expansions is determined by a parameter $m \in \mathbb{N}_0$ such that $k = 0$ when $m = 0$. In $d > 1$ dimensions the rank k increases rapidly with m.

4.1. Polynomial interpolation. Given a multiindex $\boldsymbol{n} = (n_1, \ldots, n_d) \in \mathbb{N}_0^d$ and $\boldsymbol{x} \in \mathbb{R}^d$, we define

$$|\boldsymbol{n}|_1 = n_1 + \cdots + n_d, \quad |\boldsymbol{n}|_\infty = \max\{|n_1|, \ldots, |n_d|\}, \quad \boldsymbol{n}! = n_1! \cdots n_d!,$$
$$\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} \ldots x_d^{n_d}, \quad \partial^{\boldsymbol{n}}\Phi(\boldsymbol{x}) = \partial_{x_1}^{n_1} \ldots \partial_{x_d}^{n_d}\Phi(\boldsymbol{x}).$$

The space of polynomials of degree less than $m \in \mathbb{N}_0$ *per spatial dimension*,

$$\Pi_{m-1}^d := \{p(\boldsymbol{x}) = \sum_{|\boldsymbol{n}|_\infty < m} c_{\boldsymbol{n}} \boldsymbol{x}^{\boldsymbol{n}} \mid \boldsymbol{n} \in \mathbb{N}_0^d,\ c_{\boldsymbol{n}} \in \mathbb{R}\}$$

has dimension $\dim(\Pi_{m-1}^d) = m^d$.

We will use a Lagrange basis of Π_{m-1}^d consisting of polynomials $L_{\boldsymbol{n}}$ for $|\boldsymbol{n}|_\infty < m$ constructed as follows: Let $\widehat{\boldsymbol{x}}_{\boldsymbol{n}} = (\widehat{x}_{n_1,1}, \ldots, \widehat{x}_{n_d,d}) \in \mathbb{R}^d$, $\boldsymbol{n} = (n_1, \ldots, n_d) \in \{1, \ldots, m\}^d$, be m^d points on a tensor grid that is obtained through transformation of the zeros of the Chebyshev polynomials T_m of degree m to the intervals $[a_i, b_i]$ of the bounding box $Q_\sigma = [a_1, b_1] \times \ldots \times [a_d, b_d]$ (8), i.e.,

$$\widehat{x}_{i,j} = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} \cos\left(\frac{2j-1}{2m}\pi\right), \quad i \in \{1, \ldots, d\},\ j \in \{1, \ldots, m\}.$$

The univariate Lagrange polynomials in $[a_i, b_i]$ are given by

$$L_{i,j}(x) = \prod_{\ell \in \{1,\ldots,m\}\setminus\{j\}} \frac{x - \widehat{x}_{i,\ell}}{\widehat{x}_{i,j} - \widehat{x}_{i,\ell}}, \quad i \in \{1, \ldots, d\}, \quad j \in \{1, \ldots, m\},$$

and their multivariate counterparts are obtained by multiplication

$$L_{\boldsymbol{n}}(\boldsymbol{x}) = L_{1,n_1}(x_1) \cdot \ldots \cdot L_{d,n_d}(x_d)$$

and satisfy $L_{\boldsymbol{n}}(\widehat{\boldsymbol{x}}_{\boldsymbol{\ell}}) = \delta_{\boldsymbol{n},\boldsymbol{\ell}}$ for all $|\boldsymbol{\ell}|_\infty < m$.

A separable approximation $\Phi^k(\boldsymbol{x}, \boldsymbol{y})$ of $\Phi(\boldsymbol{x}, \boldsymbol{y})$ is given by the polynomial in $\boldsymbol{x}$ which interpolates $\Phi(\boldsymbol{x}, \boldsymbol{y})$ at the tensor grid points $\widehat{\boldsymbol{x}}_{\boldsymbol{n}}$, i. e. $\Phi^k(\widehat{\boldsymbol{x}}_{\boldsymbol{n}}, \boldsymbol{y}) = \Phi(\widehat{\boldsymbol{x}}_{\boldsymbol{n}}, \boldsymbol{y})$

for all $|\boldsymbol{n}|_\infty < m$. The interpolating polynomial $\Phi^k(\boldsymbol{x},\boldsymbol{y})$ can be represented using the Lagrange basis,

$$\Phi^k(\boldsymbol{x},\boldsymbol{y}) = \sum_{|\boldsymbol{n}|_\infty<m} \underbrace{L_{\boldsymbol{n}}(\boldsymbol{x})}_{u_{\boldsymbol{n}}(\boldsymbol{x})} \underbrace{\Phi(\widehat{\boldsymbol{x}}_{\boldsymbol{n}},\boldsymbol{y})}_{v_{\boldsymbol{n}}(\boldsymbol{y})}, \tag{10}$$

the representation rank is $k = m^d$.

4.2. Taylor expansion. A Taylor expansion of $\Phi(\boldsymbol{x},\boldsymbol{y})$ with respect to $\boldsymbol{x}$ about a point $\boldsymbol{x}_0 \in Q_\sigma$ yields the separable expansion

$$\Phi^k(\boldsymbol{x},\boldsymbol{y}) = \sum_{|\boldsymbol{n}|_1<m} \underbrace{\frac{(\boldsymbol{x}-\boldsymbol{x}_0)^{\boldsymbol{n}}}{\boldsymbol{n}!}}_{u_{\boldsymbol{n}}(\boldsymbol{x})} \underbrace{\partial_{\boldsymbol{x}}^{\boldsymbol{n}}\Phi(\boldsymbol{x}_0,\boldsymbol{y})}_{v_{\boldsymbol{n}}(\boldsymbol{y})}. \tag{11}$$

We choose the center of the expansion $\boldsymbol{x}_0$ to be the midpoint of the cuboid Q_σ. The rank of such an expansion is $k = \#\{\boldsymbol{n}\in\mathbb{N}_0^d \mid |\boldsymbol{n}|_1 < m\} = \binom{m+d-1}{d}$.

In order to facilitate the implementation of a separable approximation through Taylor expansion, we now derive recursion formulas for the multivariate derivatives $\partial_{\mathbf{x}}^{\boldsymbol{n}}\Phi(\boldsymbol{x},\boldsymbol{y})$, $\boldsymbol{n}\in\mathbb{N}_0^d$. In order to simplify notation, we define $\boldsymbol{z} := \boldsymbol{x}-\boldsymbol{y}$ as well as $\Phi(\boldsymbol{z}) := \phi(|\boldsymbol{z}|_2)$ and write $\partial^{\boldsymbol{n}}\Phi(\boldsymbol{z})$ instead of $\partial_{\mathbf{x}}^{\boldsymbol{n}}\Phi(\boldsymbol{x},\boldsymbol{y})$ in view of $\mathbf{y}$ being constant and

$$\partial_{\mathbf{x}}^{\boldsymbol{n}}\Phi(\boldsymbol{x},\boldsymbol{y}) = \partial_{\mathbf{x}}^{\boldsymbol{n}}\phi(|\boldsymbol{x}-\boldsymbol{y}|_2) = \partial_{\mathbf{z}}^{\boldsymbol{n}}\phi(|\mathbf{z}|_2)|_{\boldsymbol{z}:=\boldsymbol{x}-\boldsymbol{y}}.$$

The $C^\infty(0,\infty)$-functions

$$\begin{aligned} \psi_{\mathrm{GAU},\alpha}(r) &= e^{-\frac{r}{\alpha^2}}, \quad \psi_{\mathrm{IMQ},\alpha}(r) = \left(1+\frac{r}{\alpha^2}\right)^{-\frac{1}{2}}, \\ \psi_{\mathrm{MQ},\alpha}(r) &= \left(1+\frac{r}{\alpha^2}\right)^{\frac{1}{2}}, \quad \psi_{\mathrm{TPS},\alpha}(r) = \frac{r}{2\alpha^2}\log\frac{r}{\alpha^2} \end{aligned} \tag{12}$$

all satisfy $\psi_{Z,\alpha}(r^2) = \phi_{Z,\alpha}(r)$ for $Z\in\{\mathrm{GAU},\mathrm{IMQ},\mathrm{MQ},\mathrm{TPS}\}$. In fact, according to Schoenberg (see [15, Theorem 7.13]) there exists for every positive definite radial function $\phi\in\mathbf{PD}$ an associated completely monotone function $\psi\in C^\infty(0,\infty)$, $\psi\not\equiv 0$, satisfying $\psi(r^2)=\phi(r)$. We remark that Schoenberg's characterization has been extended to conditionally positive definite radial functions $\phi\in\mathbf{CPD}(m)$ by Micchelli (see [15, Theorem 8.19]). Based on ψ, we now define

$$\Psi_n(\boldsymbol{z}) := \psi^{(n)}(|\boldsymbol{z}|_2^2), \quad n\in\mathbb{N}_0,$$

where $\psi^{(n)} = \partial^n\psi$ denotes the n-th derivative of $\psi:\mathbb{R}_+\to\mathbb{R}$.

THEOREM 5. *Let $\boldsymbol{n}\in\mathbb{N}_0^d$ be decomposed into $\boldsymbol{n} = 2\boldsymbol{\ell}+\boldsymbol{m}$ with $\boldsymbol{\ell} = \lfloor \boldsymbol{n}/2\rfloor \in \mathbb{N}_0^d$ and $\boldsymbol{m}\in\{0,1\}^d$. Then the multivariate derivative $\partial^{\boldsymbol{n}}\Phi$ has a representation in the form*

$$\partial^{\boldsymbol{n}}\Phi(\boldsymbol{z}) = \partial_{z_1}^{n_1}\dots\partial_{z_d}^{n_d}\Phi(\boldsymbol{z}) = \sum_{\boldsymbol{j}\le\boldsymbol{\ell}} a_{j_1}^{(n_1)}\dots a_{j_d}^{(n_d)} \boldsymbol{z}^{2\boldsymbol{j}+\boldsymbol{m}} \Psi_{|\boldsymbol{\ell}|_1+|\boldsymbol{j}|_1+|\boldsymbol{m}|_1}(\boldsymbol{z}) \tag{13}$$

with coefficients $\{a_j^{(n)} \mid n\in\mathbb{N}_0,\ j\in\{0,\dots,\lfloor n/2\rfloor\}\}$. These can be arranged in rows $n=0,1,\dots$ and computed recursively, starting from $a_0^{(0)}=1$. Row $2\ell+1$ can be computed from row 2ℓ through

$$\begin{aligned} a_j^{(2\ell+1)} &= 2a_j^{(2\ell)} + 2(j+1)a_{j+1}^{(2\ell)}, \quad j=0,\dots,\ell-1, \\ a_\ell^{(2\ell+1)} &= 2a_\ell^{(2\ell)} \end{aligned} \tag{14}$$

and row 2ℓ can be computed from row $2\ell-1$ through

$$
\begin{aligned}
a_\ell^{(2\ell)} &= 2a_{\ell-1}^{(2\ell-1)}\,, \\
a_j^{(2\ell)} &= 2a_{j-1}^{(2\ell-1)} + (2j+1)a_j^{(2\ell-1)}, \quad j=\ell-1,\dots,1\,, \\
a_0^{(2\ell)} &= a_0^{(2\ell-1)}\,.
\end{aligned} \tag{15}
$$

Proof. We prove the representation (13) beginning with the observation that $\Phi = \Psi_0$ and the relationship

$$\partial_{z_i}\Psi_n(\boldsymbol{z}) = 2z_i\Psi_{n+1}(\boldsymbol{z}). \tag{16}$$

Repeated application of (16) will eventually result in (13). In order to derive explicit recursion formulas, we will begin with the one-dimensional case where $\boldsymbol{z} = z \in \mathbb{R}$ is a scalar. In this case, the first few derivatives are computed to be

$$
\begin{aligned}
\partial^0\Phi(z) &= \Psi_0(z), \\
\partial^1\Phi(z) &= 2\,z\,\Psi_1(z), \\
\partial^2\Phi(z) &= 2\;\Psi_1(z) + 4\,z^2\,\Psi_2(z), \\
\partial^3\Phi(z) &= 12\,z\;\Psi_2(z) + 8\,z^3\,\Psi_3(z), \\
\partial^4\Phi(z) &= 12\;\Psi_2(z) + 48\,z^2\,\Psi_3(z) + 16\,z^4\,\Psi_4(z).
\end{aligned} \tag{17}
$$

In order to derive a general formula for all $\partial^n\Phi(z)$, we distinguish between even and odd n and write n as $n = 2\ell+m$ with $\ell \in \mathbb{N}_0$ and $m \in \{0,1\}$. Denoting the coefficients in the right hand side of $\partial^n\Phi(z)$ in (17) by $a_0^{(n)},\dots,a_\ell^{(n)}$ suggests that there holds

$$\partial^{2\ell}\Phi(z) = \sum_{j=0}^{\ell} a_j^{(2\ell)}\, z^{2j}\, \Psi_{\ell+j}(z) \quad \text{for} \quad n = 2\ell, \tag{18a}$$

$$\text{and} \quad \partial^{2\ell+1}\Phi(z) = \sum_{j=0}^{\ell} a_j^{(2\ell+1)} z^{2j+1}\Psi_{\ell+j+1}(z) \quad \text{for} \quad n = 2\ell+1. \tag{18b}$$

The difference between (18a) and (18b) is minor and merely consists of adding an $m \in \{0,1\}$ in appropriate places, i.e., using $n = 2\ell + m$ with $m \in \{0,1\}$, (18a) and (18b) are combined into

$$\partial^n\Phi(z) = \sum_{j=0}^{\ell} a_j^{(n)} z^{2j+m}\Psi_{\ell+j+m}(z) \quad \text{for} \quad n \in \mathbb{N}_0. \tag{19}$$

We have provided the even and odd cases separately because this allows one to confirm (18) by checking that differentiating (18a) once produces (18b) and vice versa (with ℓ replaced by $\ell+1$). In this way we also see how the coefficients appearing in (17) can be computed one row at time as stated in (14), (15). For the multivariate form of (13), i.e., for $\boldsymbol{z} \in \mathbb{R}^d$, we use (19) for every coordinate and obtain

$$
\begin{aligned}
\partial^{\boldsymbol{n}}\Phi(\boldsymbol{z}) &= \partial_{z_1}^{n_1}\cdots\partial_{z_d}^{n_d}\Phi(\boldsymbol{z}) \\
&= \sum_{j_1=0}^{\ell_1}\cdots\sum_{j_d=0}^{\ell_d} a_{j_1}^{(n_1)}\cdots a_{j_d}^{(n_d)} z_1^{2j_1+m_1}\cdots z_d^{2j_d+m_d}\Psi_{\ell_1+j_1+m_1+\cdots+\ell_d+j_d+m_d}(\boldsymbol{z}) \\
&= \sum_{\boldsymbol{j}\le\boldsymbol{\ell}} a_{j_1}^{(n_1)}\cdots a_{j_d}^{(n_d)}\boldsymbol{z}^{2\boldsymbol{j}+\boldsymbol{m}}\Psi_{|\boldsymbol{\ell}|_1+|\boldsymbol{j}|_1+|\boldsymbol{m}|_1}(\boldsymbol{z}).
\end{aligned}
$$

□

REMARK 6. *The monomial factors appearing in* (19) *have the following connection to the Hermite polynomials*

$$H_n(z) = (-1)^n e^{z^2} \partial^n e^{-z^2}$$

which satisfy the recursion formulas

$$\begin{aligned} H_{-1}(z) &:= 0, \quad H_0(z) = 1, \quad H_{n+1}(z) = 2zH_n(z) - 2nH_{n-1}(z), \\ H_n'(z) &= 2nH_{n-1}(z). \end{aligned}$$

For the Gaussian $\Phi(z) = e^{-z^2}$, *there holds* $\Psi_{\ell+j+m}(z) = (-1)^{\ell+j+m} e^{-z^2}$ *so that* (19) *turns into* $\partial^n \Phi(z) = \sum_{j=0}^{\ell} a_j^{(n)} z^{2j+m} (-1)^{\ell+j+m} e^{-z^2}$. *Multiplication of both sides with* $(-1)^n e^{z^2}$ *leads to*

$$H_n(z) = (-1)^n \sum_{j=0}^{\ell} a_j^{(n)} z^{2j+m} (-1)^{\ell+j+m} = \sum_{j=0}^{\ell} (-1)^{\ell+j} a_j^{(n)} z^{2j+m}.$$

Hence, besides the sign, the coefficients $a_j^{(n)}$ *coincide with the coefficients of the Hermite polynomials* H_n *which can be computed through the above recursion formulas.*

4.3. Asymptotic smoothness. A convergence result for the separable approximations based on polynomial interpolation and Taylor expansion (Subsection 4.4) follows if the kernel Φ is asymptotically smooth. In [4], a similar is obtained for the ACA approach that will be introduced in Section 5.

DEFINITION 7. *The kernel* $\Phi : \mathbb{R}^d \to \mathbb{R}$ *is called* asymptotically smooth *if there exist constants* $c, \beta, \gamma, s \in \mathbb{R}$ *such that for all* $\boldsymbol{n} \in \mathbb{N}_0^d$

$$|\partial^{\boldsymbol{n}} \Phi(\boldsymbol{z})| \leq c |\boldsymbol{n}|_1^{\beta} \gamma^{|\boldsymbol{n}|_1} \boldsymbol{n}! |\boldsymbol{z}|_2^{-|\boldsymbol{n}|_1 - s} .$$

According to [12, Theorem E.8], the kernel $\Phi(\boldsymbol{z}) = \phi(|\boldsymbol{z}|_2)$ is asymptotically smooth if the corresponding radial kernel function ϕ is asymptotically smooth. In principle, this provides a way to check the asymptotic smoothness of Φ by computing derivatives of ϕ analytically. This leads to quite lengthy expressions, however. In the following, we will show that $\phi(r) = \psi(r^2)$ is asymptotically smooth if ψ is asymptotically smooth. This will enable us to confirm for $Z \in \{\mathrm{GAU}, \mathrm{IMQ}, \mathrm{MQ}, \mathrm{TPS}\}$ that ϕ_Z is asymptotically smooth by computing derivatives of ψ_Z, which is rather easy.

THEOREM 8. *The asymptotic smoothness of* $\psi : \mathbb{R}_+ \to \mathbb{R}$ *implies the asymptotic smoothness of* $\phi(r) = \psi(r^2)$. *If* c_2, β_2, γ_2, $s_2 \in \mathbb{R}$ *are the constants in the asymptotic smoothness inequality for* ψ, *then* ϕ *satisfies the asymptotic smoothness inequality with constants* $c = c_2$, $\beta = \beta_2$, $\gamma = 4\gamma_2$, $s = 2s_2$.

Proof. We begin by scaling the coefficients $a_j^{(n)}$ in (13) to produce a transformed set of coefficients

$$\widehat{a}_j^{(n)} = (\ell + m + j)!\, 2^{-(\ell+m+j)}\, a_j^{(n)}, \qquad (n = 2\ell + m,\ m \in \{0, 1\}).$$

The scaling factors were chosen such that these new $\widehat{a}_j^{(n)}$ can still be computed recursively. Instead of (14) we now have

$$\begin{aligned} \widehat{a}_j^{(2\ell+1)} &= (\ell + 1 + j)\widehat{a}_j^{(2\ell)} + 2(j+1)\widehat{a}_{j+1}^{(2\ell)}, \quad j = 0, \ldots, \ell - 1, \\ \widehat{a}_{\ell}^{(2\ell+1)} &= (2\ell + 1)\widehat{a}_{\ell}^{(2\ell)} \end{aligned} \tag{20}$$

and (15) translates into

$$
\begin{aligned}
\widehat{a}_\ell^{(2\ell)} &= \quad (2\ell)\widehat{a}_{\ell-1}^{(2\ell-1)}, \\
\widehat{a}_j^{(2\ell)} &= (\ell+j)\widehat{a}_{j-1}^{(2\ell-1)} + (2j+1)\widehat{a}_j^{(2\ell-1)}, \quad j = \ell-1,\dots,1, \\
\widehat{a}_0^{(2\ell)} &= \qquad\qquad\qquad\quad \widehat{a}_0^{(2\ell-1)}.
\end{aligned} \tag{21}
$$

According to (20) and (21), the transformed coefficients are still integers. Based on these formulas we are now going to derive an upper bound for

$$
\sum_{j=0}^{\ell} a_j^{(n)}(\ell+m+j)! = \sum_{j=0}^{\ell} 2^{\ell+m+j}\widehat{a}_j^{(n)} \le 2^n \sum_{j=0}^{\ell} \widehat{a}_j^{(n)}. \tag{22}
$$

In the sum to the far right of (22) each $\widehat{a}_j^{(n)}$ is the sum of one or two $\widehat{a}_j^{(n-1)}$ weighted by a factor $\le n$. Since in this way each $\widehat{a}_j^{(n-1)}$ is used at most twice, we can replace the sum over row n with the sum over row $n-1$ multiplied with $2n$. Starting from (22) we apply this step n times until we arrive at row 0 where $a_0^{(0)} = 1$:

$$
\begin{aligned}
\sum_{j=0}^{\ell=\lfloor \frac{n}{2} \rfloor} a_j^{(n)}(\ell+m+j)! &\le 2^n 2n \sum_{j=0}^{\lfloor \frac{n-1}{2} \rfloor} \widehat{a}_j^{(n-1)} \\
&\le 2^n (2n)(2(n-1)) \sum_{j=0}^{\lfloor \frac{n-2}{2} \rfloor} \widehat{a}_j^{(n-2)} \le \cdots \le 2^{2n} n!
\end{aligned} \tag{23}
$$

Using the derivative from (13) and the inequality (23), we calculate

$$
\begin{aligned}
|\phi^{(n)}(r)| &\le \sum_{j=0}^{\ell} a_j^{(n)} |r|^{2j+m} |\psi^{(\ell+m+j)}(r^2)| \\
&\le \sum_{j=0}^{\ell} a_j^{(n)} |r|^{2j+m} c_2 (\ell+m+j)^{\beta_2} \gamma_2^{\ell+m+j} (\ell+m+j)! |r|^{-2(\ell+m+j)-2s_2} \\
&\le c_2 n^{\beta_2} \gamma_2^n \sum_{j=0}^{\ell} a_j^{(n)} (\ell+m+j)! |r|^{-2\ell-m-2s_2} \\
&\le c_2 n^{\beta_2} \gamma_2^n 2^{2n} n! |r|^{-n-2s_2} \\
&\le c n^{\beta} \gamma^n n! |r|^{-n-s},
\end{aligned}
$$

implying that ϕ is asymptotically smooth with the stated constants. ∎

The derivatives of the functions $\psi_Z := \psi_{Z,\alpha=1}$ in (12) are given by

$$\psi^{(k)}_{\mathrm{GAU}}(r) = (-1)^k \exp(-r),$$
$$\psi^{(k)}_{\mathrm{IMQ}}(r) = \left(\prod_{j=0}^{k-1}(-0.5-j)\right)(1+r)^{-0.5-k},$$
$$\psi^{(1)}_{\mathrm{TPS}}(r) = 0.5\log(r)+0.5, \quad \psi^{(k)}_{\mathrm{TPS}}(r) = \left(0.5\prod_{j=1}^{k-2}(-j)\right) r^{-(k-1)}, \quad k \geq 2,$$
$$\psi^{(k)}_{\mathrm{MQ}}(r) = \left(\prod_{j=0}^{k-1}(0.5-j)\right)(1+r)^{0.5-k}.$$

All of these functions are asymptotically smooth with constants listed in Table 1. For $\alpha > 0$ we can choose β_2, γ_2 and s_2 in the same way when c_2 is replaced with $\alpha^s c_2$. This is because

$$\psi^{(n)}_\alpha(r) = \frac{1}{\alpha^n}\psi^{(n)}(r/\alpha) \leq \frac{1}{\alpha^n}\, c_2\, n^{\beta_2}\, \gamma_2^n\, (r/\alpha)^{-n-s_2} \leq (\alpha^s c_2)\, n^{\beta_2}\, \gamma_2^n\, r^{-n-s_2}\,.$$

Table 1
Constants of asymptotic smoothness.

	c_2	β_2	γ_2	s_2
ψ_G	1	0	1	0
ψ_I	1	0	1	0.5
ψ_T	0.5	0	1	-1
ψ_M	1	0	1	-0.5

4.4. Convergence of polynomial interpolation and Taylor expansion. Error estimates for polynomial interpolation and Taylor expansion can be expressed involving derivatives of Φ. Convergence $\Phi^k(\boldsymbol{x},\boldsymbol{y}) \to \Phi(\boldsymbol{x},\boldsymbol{y})$ can be proved if Φ is asymptotically smooth and if the standard admissibility condition (Definition 4) is satisfied with η chosen sufficiently small. In this case, the speed of convergence will be exponential.

Definition 9 (Exponential convergence [12, Definiton 4.5]). *The separable expansion (6) is called* exponentially convergent *if there are constants $c_0 \geq 0$, $c_1 > 0$ and $\beta > 0$ such that*

$$|\Phi(\boldsymbol{x},\boldsymbol{y}) - \Phi^k(\boldsymbol{x},\boldsymbol{y})| \leq c_0 \exp(-c_1 k^\beta)$$

for all $\boldsymbol{x} \in Q_\sigma$, $\boldsymbol{y} \in Q_\tau$.

Remark 10. *The constant β will often be $\beta = 1/d$ in $\mathbb{R}^d$.*

According to [12, Theorems 4.17,4.22] polynomial interpolation and Taylor expansion require $\eta < 1/\gamma$ where γ comes from Definition 7.

4.5. Specialized expansions. Polynomial interpolation and Taylor expansion appear in the literature of $\mathcal{H}$-matrices as methods which can be used whenever the kernel is asymptotically smooth. Besides these generic methods, there exist separable expansions tailored to particular kernels. One example for a radial kernel function admitting such an expansion are the thin plate splines, as described in [15, Proposition 15.2]. These expansions are often used in conjunction with fast multipole methods but can also be used for $\mathcal{H}$-matrices.

We will compare the generic aproximation methods to one of these specialized expansions developed for generalized multiquadrics (GMQ) in [6]. The radial kernel functions considered in this paper are

$$\phi_{\mathrm{GMQ},\alpha}(r) = \left(1 + \left(\frac{r}{\alpha}\right)^2\right)^{p/2} \qquad \text{for } \alpha > 0$$

for odd $p \in \mathbb{Z}$. Note that for $p = \pm 1$ this yields the kernels $\phi_{\mathrm{MQ},\alpha}$ and $\phi_{\mathrm{IMQ},\alpha}$.

The separable approximation detailed in [6] is of the form

$$\Phi(\boldsymbol{x},\boldsymbol{y}) \approx \Phi^k(\boldsymbol{x},\boldsymbol{y}) = \sum_{\ell=0}^{m-1} \sum_{|\boldsymbol{n}|_1=\ell} u_{\boldsymbol{n}}^{(\ell)}(\boldsymbol{x}) \cdot v_{\boldsymbol{n}}^{(\ell)}(\boldsymbol{y}), \tag{24}$$

where $m \in \mathbb{N}_0$ controls the length of the expansion. The part depending on $\boldsymbol{x}$ is

$$u_{\boldsymbol{n}}^{(\ell)}(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\boldsymbol{n}}}{\alpha^p \, |\boldsymbol{x}|_2^{2\ell-p}}.$$

For the part depending on $\boldsymbol{y}$ there holds a recurrence which uses coefficients

$$a_\ell = \frac{2\ell - p - 2}{\ell}, \quad b_\ell = \frac{p - \ell + 2}{\ell}.$$

Denoting by $\boldsymbol{e}_i$ the i-th unit vector in $\mathbb{R}^d$ the computation of $v_{\boldsymbol{n}}^{(\ell)}(\boldsymbol{y})$ then proceeds as

$$\begin{aligned} &v_{\boldsymbol{0}}^{(0)}(\boldsymbol{y}) = 1, \qquad v_{\boldsymbol{e}_i}^{(1)}(\boldsymbol{y}) = -p\, y_i, \\ &v_{\boldsymbol{n}}^{(\ell)}(\boldsymbol{y}) = a_\ell \sum_{i=1}^{d} y_i v_{\boldsymbol{n}-\boldsymbol{e}_i}^{(\ell-1)}(\boldsymbol{y}) + b_\ell(|\boldsymbol{y}|_2^2 + \alpha^2) \sum_{i=1}^{d} v_{\boldsymbol{n}-2\boldsymbol{e}_i}^{(\ell-2)}, \quad \ell \geq 2. \end{aligned} \tag{25}$$

In the notation $v_{\boldsymbol{n}}^{(\ell)}$ from [6] it is always the case that $\ell = |\boldsymbol{n}|_1$. The superscript is not really necessary, but it is helpful in understanding the definition since it emphasizes the level-wise computation in (25). The number of terms in (24) depends on m in the same way as for Taylor expansion, i.e. the representation rank is $k = \binom{m+d-1}{d}$.

According to [6, Lemma 3.1] the convergence of $\Phi^k(\boldsymbol{x},\boldsymbol{y})$ to $\Phi(\boldsymbol{x},\boldsymbol{y})$ now requires

$$|\boldsymbol{x}|_2 > \sqrt{|\boldsymbol{y}|_2^2 + \alpha^2} \tag{26}$$

which is also termed as $\boldsymbol{x}$ lying in the farfield of $\boldsymbol{y}$. It is important to note that we can use a shifted version of (26) where $\boldsymbol{x}$ and $\boldsymbol{y}$ are replaced by $\boldsymbol{x} - \boldsymbol{z}$ and $\boldsymbol{y} - \boldsymbol{z}$ for some $\boldsymbol{z} \in \mathbb{R}^d$. Convergence will then be guaranteed if the same shift is applied to (24) as well. This is possible since we have the translation invariance $\Phi(\boldsymbol{x},\boldsymbol{y}) = \Phi(\boldsymbol{x}-\boldsymbol{z},\boldsymbol{y}-\boldsymbol{z})$.

In the context of $\mathcal{H}$-matrices we have to derive separable expansions not for individual pairs $\boldsymbol{x}$ and $\boldsymbol{y}$ but for all points from cuboids Q_σ and Q_τ. Therefore, we need

to formulate the admissibility condition in terms of Q_σ and Q_τ. The admissibility of Q_σ and Q_τ should imply that all $\boldsymbol{x} \in Q_\sigma$ are in the farfield of all $\boldsymbol{y} \in Q_\tau$. This is the case if we use

$$\sqrt{\left(\frac{\operatorname{diam}(Q_\tau)}{2}\right)^2 + \alpha^2} < \eta \operatorname{dist}(Q_\sigma, \boldsymbol{y}_0) \tag{27}$$

as an admissibility condition where the shift $\boldsymbol{y}_0$ is taken as the centre of Q_τ and $\eta \leq 1$ is used to sharpen the condition to obtain faster convergence. In the right hand side of (27) we can also replace $\operatorname{dist}(Q_\sigma, \boldsymbol{y}_0)$ by the smaller value of $\operatorname{dist}(Q_\sigma, Q_\tau)$.

Referring again to [6, Lemma 3.1] the error of the expansion (24) satisfies

$$|\Phi(\boldsymbol{x}, \boldsymbol{y}) - \Phi^k(\boldsymbol{x}, \boldsymbol{y})| \leq \begin{cases} C\, \eta^m & \text{for } p > 0, \\ C\, m^{p+1}\, \eta^m & \text{for } p < 0, \end{cases} \tag{28}$$

where C is a constant depending η, α, $\operatorname{diam}(Q_\tau) \leq \operatorname{diam}(X)$ and p but not on m. The error bound (28) corresponds to an exponential speed of convergence.

5. Low rank approximation through cross approximation. Let $A \in \mathbb{R}^{m,n}$. The cross approximation algorithm may be viewed as a prematurely aborted Gaussian elimination and computes a factored approximation $A \approx UV^T$ using "thin" rectangular matrices U, V. In particular, the entire matrix A is not required as input but only access to individual entries a_{ij} (typically $\mathcal{O}(k)$ rows and columns of A where k is the number of columns in U, V). The computation of a_{ij} typically consists of the evaluation of the kernel function itself and does not require any separable expansions as was the case in the construction of low rank approximations described in Section 4.

Algorithm 1 Cross approximation

Input: method to access individual matrix entries a_{ij} of $A \in \mathbb{R}^{m,n}$
Output: factors $U = (u_1 \dots u_k) \in \mathbb{R}^{m,k}$, $V = (v_1 \dots v_k) \in \mathbb{R}^{n,k}$ that yield $A \approx UV^T$

1: $\ell = 1$.
2: **while** stopping criterion not satisfied **do**
3: Choose an (arbitrary) pivot element $a_{i_\ell j_\ell} \neq 0$.
4: Compute the i_ℓ'th row $a_{i_\ell,:}$ and the j_ℓ'th column $a_{:,j_\ell}$ of A.
5: Set $u_\ell := a_{:,j_\ell}$ and $v_\ell = (a_{i_\ell j_\ell})^{-1} a_{i_\ell,:}^T$.
6: Compute $A \leftarrow A - u_\ell v_\ell^T$.
7: $\ell = \ell + 1$.
8: **end while**

The generic version of this procedure is provided as Algorithm 1. Line 6 might be considered as its most important one: it can be shown that subtraction of the outer product $u_\ell v_\ell^T$ reduces the rank of A by one. In a practical implementation, however, this update is not computed for the entire matrix A (which would imply complexity $\mathcal{O}(nm)$). Instead, all updates are only computed for the row and column selected in line 4, i.e.,

$$\begin{aligned} a_{i_\ell,:} &\leftarrow a_{i,:} - u_{i,1} v_1^T - \dots - u_{i,\ell} v_\ell^T = a_{i_\ell,:} - u_{i,1:\ell} \cdot (v_1 \dots v_\ell)^T, \\ a_{:,j_\ell} &\leftarrow a_{:,j} - u_1 v_{j,1} - \dots - u_\ell v_{j,\ell} = a_{:,j_\ell} - (u_1 \dots u_\ell) \cdot (v_{j,1:\ell})^T. \end{aligned}$$

This reduces the algorithm's complexity to $\mathcal{O}(k^2(n+m))$.

The computed factorisation $A \approx UV^T$ will depend on the selection of the pivot element (line 3) as well as the stopping criterion (line 2). These two topics will be discussed in the following two subsections.

5.1. Pivot selection. The pivot element in line 3 must be nonzero and preferably not even close to zero. In our implementation we select the first column index $j_1 \in \{1, \dots, m\}$ at random which enables us to compute u_1. All the remaining row and column indices are determined by

$$i_\ell = \operatorname*{arg\,max}_{i=1,\dots,n} (u_\ell)_i, \quad \ell = 1 \dots, k, \quad \text{and} \quad j_\ell = \operatorname*{arg\,max}_{j=1,\dots,m} (v_{\ell-1})_j, \quad \ell = 2 \dots, k.$$

This strategy from [12] always uses the newest available vector u_ℓ or v_ℓ to compute the next pivot. Since in this paper we consider interpolation matrices where each matrix row and column is associated with a particular data site, one could use this information to select pivots apriori and without the need to determine maximum row or column entries. In particular, one might interpret the selected pivots as a coarser set of data sites, and in order to sustain as much information as possible, the data sites corresponding to the pivot rows and columns should be selected along the lines of typical coarsening algorithms. Such a strategy is also supported by the following representation of the factorization computed in Algorithm 1: If $I_r = \{i_1, \dots, i_k\}$ and $I_c = \{j_1, \dots, j_k\}$ collect the respective row and column indices of rows and columns selected as pivots in line 3 of Algorithm 1, then the computed matrix factors U, V satisfy

$$UV^T = \underbrace{A|_{\{1,\dots,m\}\times I_c}}_{\mathbb{R}^{m,k}} \underbrace{(A|_{I_r\times I_c})^{-1}}_{\mathbb{R}^{k,k}} \underbrace{A|_{I_r\times\{1,\dots,n\}}}_{\mathbb{R}^{k,n}}. \tag{29}$$

Hence, pivots should be selected attempting to obtain a well-conditioned $A_{I_r\times I_c}$. The following theorem ensures that there exists a quasioptimal rank-k-approximant in the form of (29).

THEOREM 11 ([11]). *Let $A \in \mathbb{R}^{m,n}$ and $k \in \mathbb{N}$ with $k \le \operatorname{rank}(A)$. Let subsets $I_r \subset \{1, \dots, m\}, I_c \subset \{1, \dots, n\}$ with $\#I_r = \#I_c = k$ be chosen such that*

$$|\det A|_{I_r\times I_c}| = \max\{|\det A|_{I_{r'}\times I_{c'}}| \mid I_{r'} \subset \{1, \dots, m\}, I_{c'} \subset \{1, \dots, n\},\ \#I_{r'} = \#I_{c'} = k\}.$$

Using the "maximum matrix entry" norm $\|A\|_C := \max\{|a_{ij}| \mid 1 \le i \le m,\ 1 \le j \le n\}$, it holds that (29) is quasioptimal, in particular

$$\|A - A|_{\{1,\dots,m\}\times I_c}(A|_{I_r\times I_c})^{-1}A|_{I_r\times\{1,\dots,n\}}\|_C \le (k+1)\ \|M - R_{best}\|_2$$

where R_{best} is the best rank-k-approximation to A (with respect to $\|\cdot\|_2$.)

5.2. Stopping criterion. In line 2 of Algorithm 1 the number of steps k to perform is determined by a stopping criterion. The simplest choice would be to decide on the rank k in advance, which can be useful if we know that this is sufficient to achieve a desired accuracy or if we want to limit the amount of storage. Cross approximation with a fixed k will be referred to as CA from here on.

Strict error estimates for $A - UV^T$ with ACA-computed U, V require assumptions on the underlying kernel function generating the matrix data of A. Here we review (and later use in the numerical tests) a heuristic stopping criterion from the literature (e.g., [12], §9.4.3). It is based on the following assumptions.

ASSUMPTION 12. *Let $R_\ell := U_\ell V_\ell^T$ denote the ACA-approximation of a matrix A after ℓ steps of Algorithm 1. We assume that*

1. $\|A\|_2 \approx \|R_1\|_2$,
2. $\|A - R_\ell\|_2 \leq \|A - R_{\ell-1}\|_2$,
3. $\|A - R_{\ell-1}\|_2 \approx \|R_\ell - R_{\ell-1}\|_2$.

If the conditions of Assumption 12 are satisfied, we conclude that (again using the notation $R_\ell := U_\ell V_\ell^T$)

$$\|A - R_\ell\|_2 \leq \|A - R_{\ell-1}\|_2 \approx \|R_\ell - R_{\ell-1}\|_2 = \|u_\ell v_\ell^T\|_2 = \|u_\ell\|_2 \|v_\ell\|_2.$$

Given a tolerance ϵ, this yields the heuristic stopping criteria

$$\|u_\ell\|_2\|v_\ell\|_2 \leq \epsilon \text{ (absolute)}, \qquad \frac{\|u_\ell\|_2\|v_\ell\|_2}{\|u_1\|_2\|v_1\|_2} \leq \epsilon \text{ (relative)}. \tag{30}$$

We refer to Algorithm 1 implemented with a stopping criterion as in (30) as adaptive cross approximation (ACA).

5.3. $\mathcal{H}$-matrix approximation with constraints. If the generic ACA Algorithm 1 (or, as a matter of fact, a separable approximation of Section 4) is applied to obtain factorizations of all admissible blocks of an $\mathcal{H}$-matrix, symmetry might be lost, i.e., a symmetric interpolation matrix B will be approximated by a non-symmetric $\mathcal{H}$-matrix $B_\mathcal{H}$. However, this "problem" is easily removed by computing only the upper triangular part of an $\mathcal{H}$-matrix $B_\mathcal{H}$ and using its transpose for the respective blocks in the lower triangular part.

While preservation of symmetry poses no problem, the situation is different for positive (semi-) definiteness. If B is positive definite, then $B_\mathcal{H}$ will eventually be positive definite since $B_\mathcal{H} \overset{\epsilon\to 0}{\longrightarrow} B$ when $B_\mathcal{H}$ is constructed using an error tolerance ϵ for low rank approximations in its admissible blocks. However, a small enough ϵ might be impractical for the application at hand, especially if B's smallest eigenvalue is close to zero. If eigenvectors v of small eigenvalues of B are known, those could be used to construct an $\mathcal{H}$-matrix $B_\mathcal{H}$ with blockwise constraints, enforcing $B_\mathcal{H} v = Bv$ [3]. Alternatively, along the lines of regularization, one could simply add a small constant to the diagonal of $B_\mathcal{H}$, albeit sacrificing approximation accuracy for the sake of positive definiteness. In the end, it needs to be decided whether it is worth to pursue positive definiteness when it comes at the expense of accuracy and/or computational efficiency.

6. Numerical results. We implemented and compared the construction of $\mathcal{H}$-matrices with low rank approximations based on

- polynomial interpolation (INT), see §4.1, (10),
- Taylor expansion (TAY), see §4.2, (11),
- cross approximation with fixed rank k (CA) or stopping criterion (30) (ACA), see §5, Algorithm 1,
- and the expansion for generalized multiquadrics (GMQ), see §4.5, (24),

where we measure the accuracy of $B_\mathcal{H}$ as an approximation for B in terms of the relative error in the Frobenius norm, i.e. $\|B - B_\mathcal{H}\|_F / \|B\|_F$. Since we want to include the GMQ expansion (§4.5) in our comparison, several of the tests involve the inverse multiquadric ϕ_{IMQ} as a radial basis function. We illustrate the convergence speed in §6.1.1 and the memory and computational time in §6.1.2. In §6.2 we demonstrate that the generic approximation methods INT, TAY and CA can be applied successfully to

the radial kernels introduced in §6. The approximation accuracy of CA can be seen to be close to optimal regardless of ϕ and d. The scaling behaviour of ACA for increasing values of N is treated in §6.3. We again measure storage requirements and computational time and find that the claim of an almost linear computational complexity is supported by our results.

In principle, our programs work for any spatial dimension. Due to the excessive growth of the representation rank, however, it was not feasible to use any other value except $d = 1, 2, 3$. Even in this range our available resources did not allow us to obtain approximations with an accuracy in the order of machine precision in all cases.

6.1. Inverse multiquadric. In our tests for ϕ_{IMQ}, we use $N = 10,000$ points and a shape factor $\alpha = N^{1/d}$. We choose different values of η in the admissibility conditions (9) and (27) in order to study its effect on the speed of convergence. These values were $\eta = 1.0, 0.5, 0.25$ for CA, INT and TAY and $\eta = 0.5, 0.25, 0.125$ for the GMQ expansion.

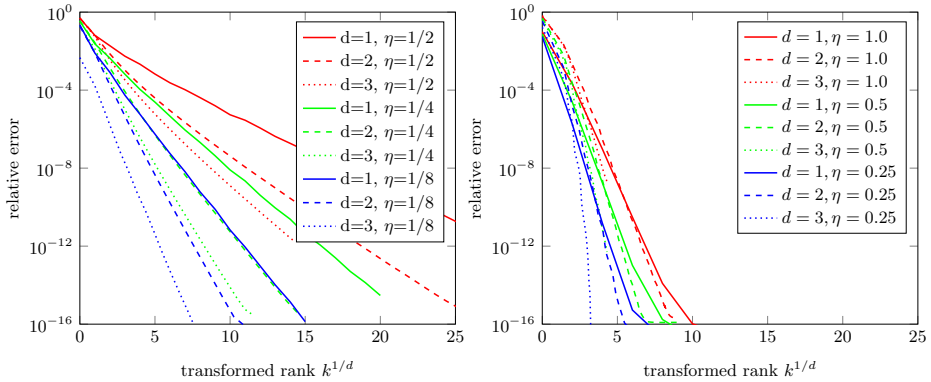


FIG. 2. $N = 10,000$, *inverse multiquadric* ϕ_{IMQ}, $\alpha = N^{-1/d}$, *Left: GMQ expansion (§4.5). Right: cross approximation at fixed rank (CA).*

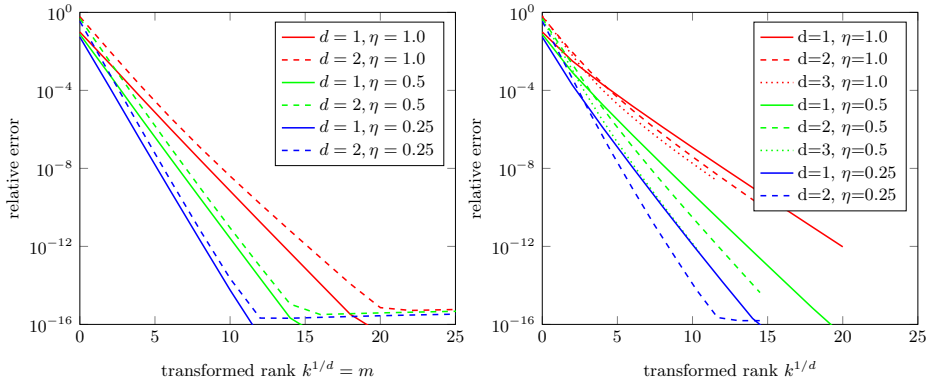


FIG. 3. $N = 10,000$, *inverse multiquadric* ϕ_{IMQ}, $\alpha = N^{-1/d}$, *Left: interpolation (INT), Right: Taylor expansion (TAY).*

6.1.1. Accuracy. The accuracy depending on the transformed rank $k^{1/d}$ is shown in Figures 2 (GMQ, CA), and 3 (INT, TAY). All the approximation methods show exponential convergence in the sense of Definition 9 where smaller values of η accelerate the error decay. Best results are obtained for ACA, followed by INT, TAY and the specialized GMQ expansion in this order.

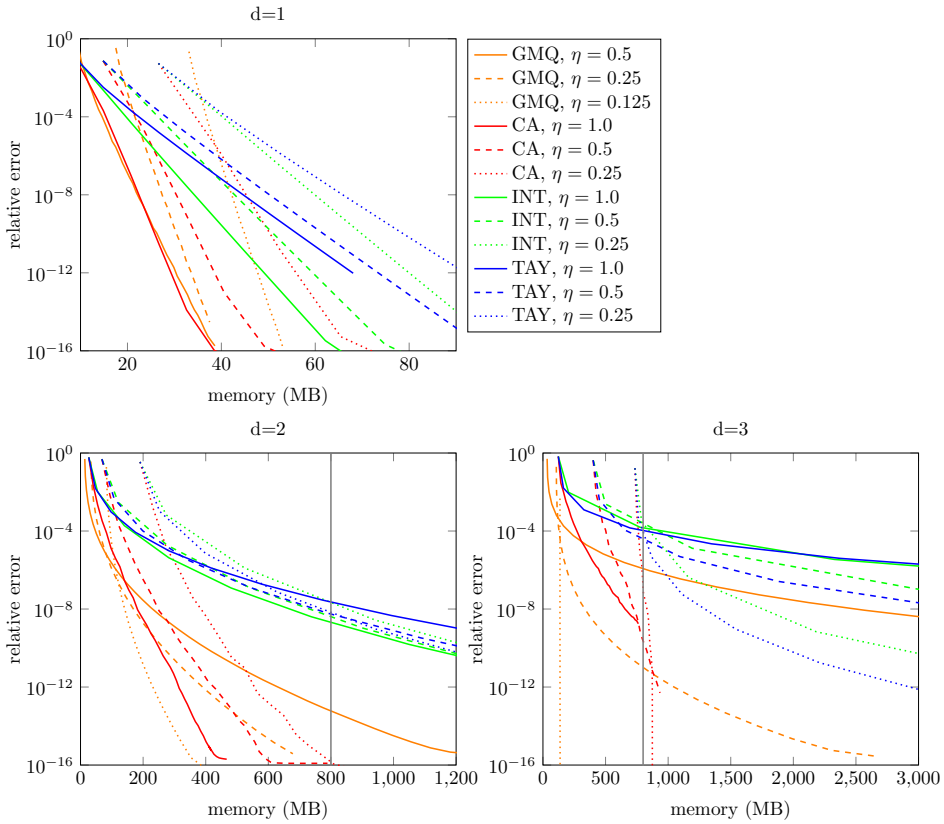


FIG. 4. *Amount of memory required to store $B_{\mathcal{H}}$ for $N = 10,000$, $\phi = \phi_{IMQ,\alpha}$, $\alpha = N^{-1/d}$, $d = 1$, $d = 2$, $d = 3$. Gray vertical line shows storage of full matrix (800 MB).*

6.1.2. Storage and computational time. Storage requirements and computational time depend not only on the rank but also on the $\mathcal{H}$-block partition which is the same for the approaches ACA, INT and TAY but differs for the specialized GMQ-expansion in view of the different admissibility condition. The required storage for different spatial dimensions $d = 1, 2, 3$ is shown in Figure 4. GMQ and CA require the smallest amount of storage whereas INT and TAY consume much more memory. The full matrix B requires about 800 MB of memory in this case - this is included in the plots for $d = 2, 3$ as a vertical gray line. When using INT or TAY and a high accuracy is desired, an advantage in using an $\mathcal{H}$-approximation $B_{\mathcal{H}}$ is only given for $d = 1$. For $d = 2, 3$, these methods require an unacceptable amount of memory. This conclusion, however, depends on the size N of the matrix. As N increases, storage of the full matrix is N^2 whereas the storage on the $\mathcal{H}$-matrix $B_{\mathcal{H}}$ is of order $\mathcal{O}(N \log N)$, also for INT and TAY. Regarding the choice of η, the results are mixed. For CA, the smallest amount of memory is obtained with the largest value of $\eta = 1$. For GMQ, a

larger value of η is advantageous in $d = 1$ but the situation is reversed for $d = 2, 3$. This is in contrast to what was seen in §6.1.1 and reflects the fact that the amount of storage required does not only depend on the speed of convergence of the separable expansion but also on the admissibility condition.

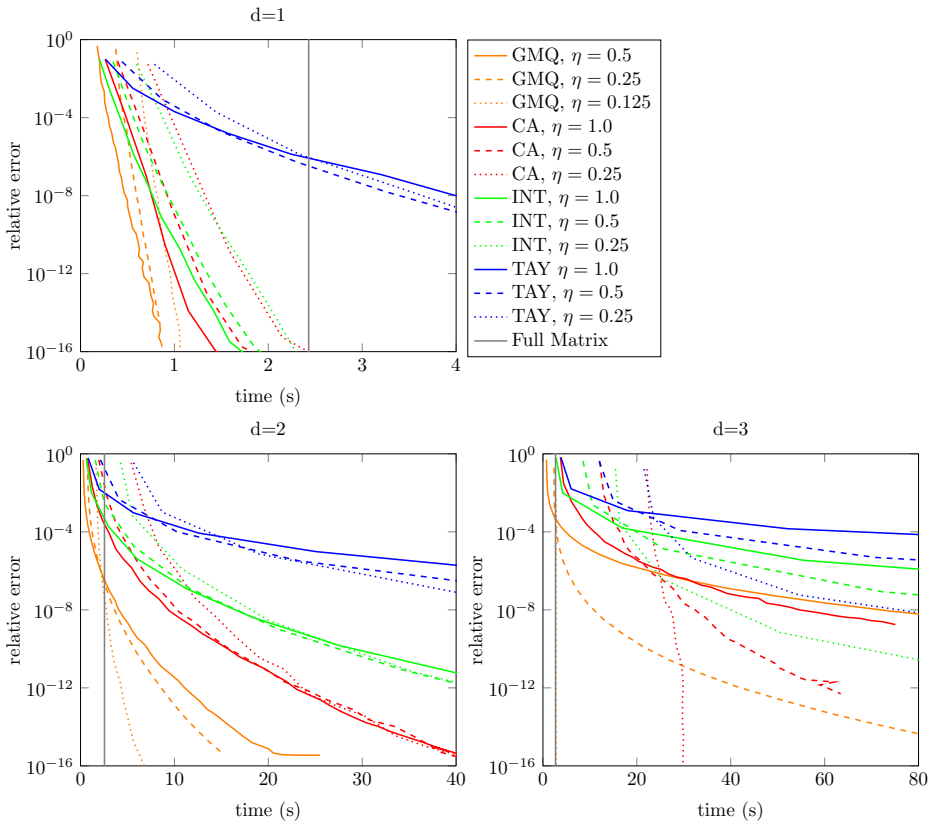


FIG. 5. *Time required to compute $B_{\mathcal{H}}$ for $N = 10,000$, $\phi = \phi_{IMQ,\alpha}$, $\alpha = N^{-1/d}$, for $d = 1, 2, 3$.*

The computational time required to set up $B_{\mathcal{H}}$ depending on the approximation accuracy is shown Figure 5. For the computational time similar observations as for the memory consumption can be made. The CA and GMQ approaches are in general faster than INT and TAY.

6.2. Radial kernel functions and dimensions. In order to show that the generic approximation methods are applicable to all radial kernels and dimensions, we measure the relative error for every combination of $d = 1, 2, 3$ and the four example functions $\phi_{\text{GAU}}, \phi_{\text{IMQ}}, \phi_{\text{TPS}}, \phi_{\text{MQ}}$. The remaining parameters were chosen as $N = 4900$, $\eta = 1$, $\alpha = 1$. Besides cross approximation, interpolation and Taylor expansion, we also included the best approximation through low-rank blocks based on singular value decomposition. In this approach, titled SVD in the diagrams, the approximation of admissible blocks of $B_{\mathcal{H}}$ is such that the columns of U and V from (7) are the left and right singular vectors belonging to the largest singular values of this block. This procedure minimizes the error and serves as a reference for the other methods. The results for $d = 1, 2, 3$ dimensions are presented in Figures 6 to 8. CA with a fixed rank shows almost optimal performance across all ϕ and d that we considered.

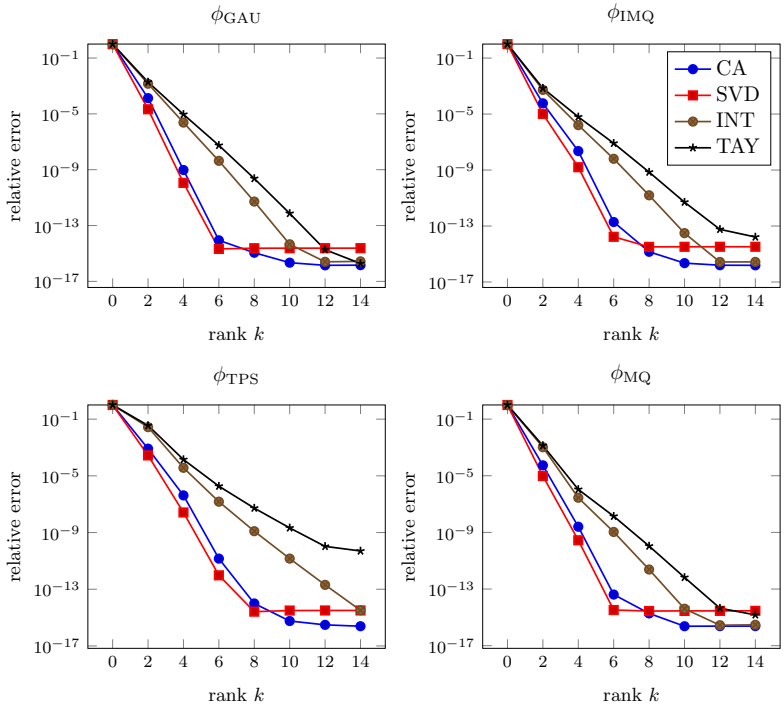


FIG. 6. *Comparsion of basis functions and approximation methods for* $N = 4900$ *and* $d = 1$.

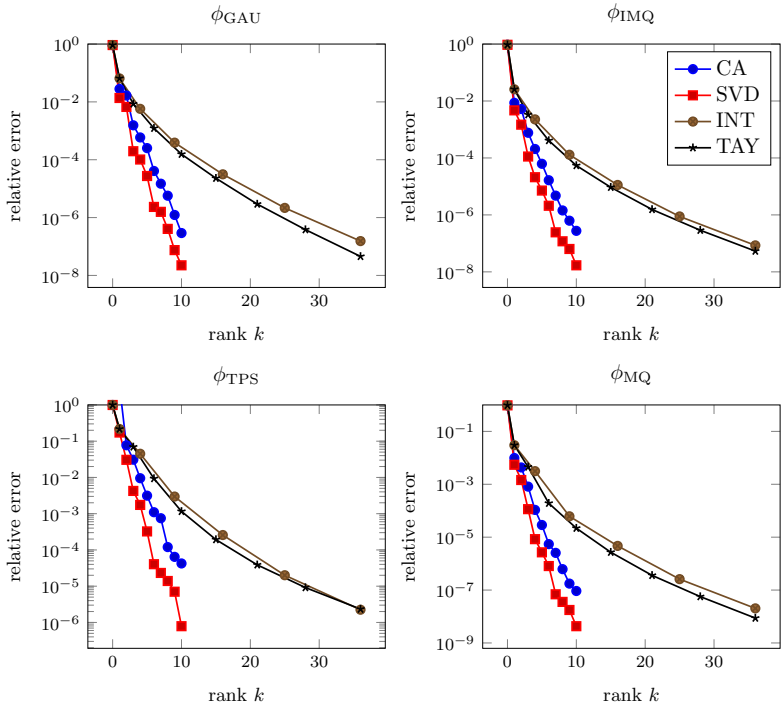


FIG. 7. *Comparsion of basis functions and approximation methods for* $d = 2$.

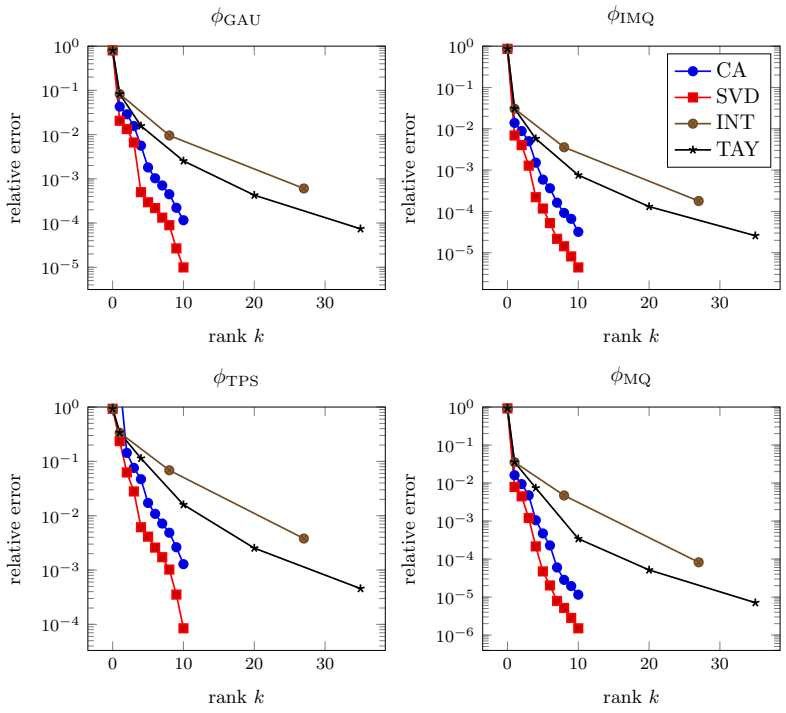


FIG. 8. *Comparsion of basis functions and approximation methods for* $d = 3$.

6.3. Dependence on matrix size N. We next investigate the influence of the number of points N on storage and computational time requirements to set up $B_{\mathcal{H}}$. The construction of $B_{\mathcal{H}}$ is based on ACA using a relative stopping criterion with $\epsilon = 10^{-1}, \ldots, 10^{-12}$ (see §5.2). The tests were carried out for $d = 2$, the Gaussian ϕ_{GAU}, a fixed shape factor $\alpha = 1$ and up to $N = 160,000$ points.

The amount of memory required to store $B_{\mathcal{H}}$ as well as the full matrix B is shown in Figure 9. The results for the size of $B_{\mathcal{H}}$ are compatible with our assumption of an almost linear scaling in terms of N. For a fixed value of N the amount of memory is proportional to $\log(\epsilon)$ corresponding to an exponential speed of convergence of the separable expansion. To give an impression of how the amount of storage relates to a sparse matrix, we note that $B_{\mathcal{H}}$ set up with $\epsilon = 10^{-3}$ consumes about as much memory as a sparse matrix with an average of about 700 non-zero entries per row would. The time required to compute $B_{\mathcal{H}}$ is shown in Figure 10. Again, we observe an almost linear dependence on N.

7. Conclusion. Scattered data interpolation using radial kernels requires the solution of a set of linear equations with a system matrix (5) which is large, dense and often highly ill-conditioned.

In this article, we described and compared various efficient ways to approximate the matrix block B in (5) by an $\mathcal{H}$-matrix $B_{\mathcal{H}}$ with almost linear $\mathcal{O}(N \log N)$ complexity. The $\mathcal{H}$-matrix $B_{\mathcal{H}}$ allows a fast matrix-vector multiplication suitable for iterative solution algorithms. In addition, $\mathcal{H}$-arithmetic is available to compute approximate matrix factorizations which may then serve as preconditioners. The number of columns $Q = \dim(\pi^d_{m-1})$ of the block P_X in (5) corresponding to the polynomial

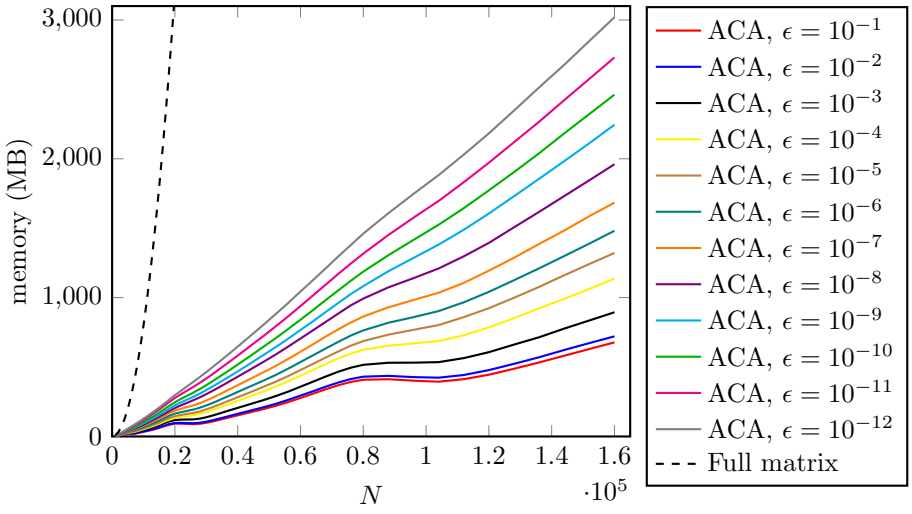


FIG. 9. *Storage requirement of $B_{\mathcal{H}}$ for $d = 2$, $\alpha = 1$, $\phi = \phi_{\mathrm{GAU}}$ depending on N. The memory required for the full matrix B is included for reference.*

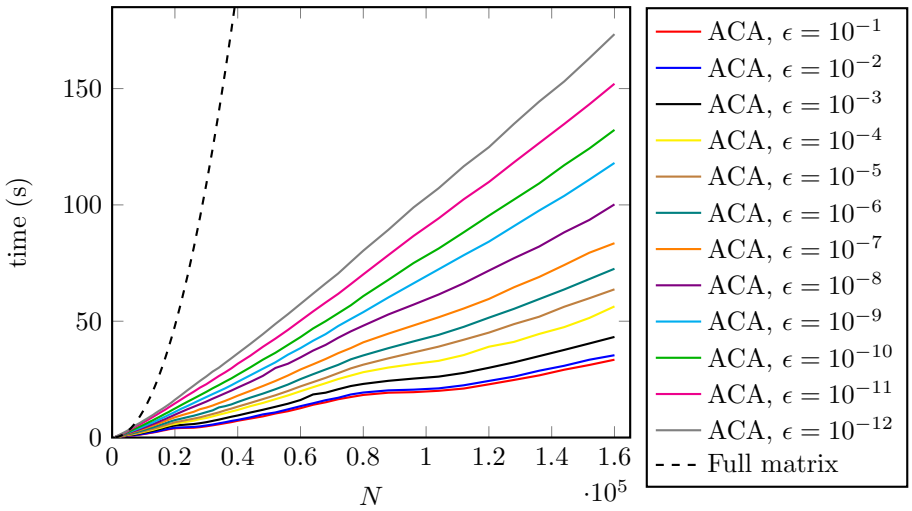


FIG. 10. *Computational time required to build $B_{\mathcal{H}}$ for $d = 2$, $\alpha = 1$, $\phi = \phi_{\mathrm{GAU}}$ depending on N.*

part is considered to be fixed and small compared to the number of points N. Setting up this block as a full matrix therefore only contributes a linear complexity $\mathcal{O}(N)$.

We compared generic approximation methods suitable for general (sufficiently smooth) radial kernels and a separable expansion tailored to multiquadric basis functions [6]. While a Taylor expansion can be expressed generically in the form (11), its implementation would require significant effort since the multivariate derivatives have to be computed for each radial kernel. We produced an implementation which is easily adapted to a wide range of radial kernels ϕ since it only requires a formula for the univariate derivatives $\psi^{(n)}$ of $\psi(r) = \phi(\sqrt{r})$ which we provided for the four radial kernels considered in this paper.

Our numerical tests show that all the considered approximation methods exhibit an exponential convergence with respect to the separation rank. The cross approximation algorithm outperformed the analytical expansions considered here whose rank k increased rapidly (in jumps) when the parameter m in the expansion is incremented. In view of large constants involved in the complexity estimates, the $\mathcal{H}$-matrix approximation $B_{\mathcal{H}}$ of B becomes advantagous only if a certain number N of data sites is exceeded. While we restricted this paper to the construction of an approximation $B_{\mathcal{H}}$ of B, the (iterative) solution of (5), including the construction of $\mathcal{H}$-preconditioners, will be subject of future work.

REFERENCES

[1] R. Beatson and L. Greengard: *A short course on fast multipole methods. Wavelets, Multilevel Methods and Elliptic PDEs* **1** (1997), 1–37.

[2] M. Bebendorf: *Hierarchical Matrices. A Means to Efficiently Solve Elliptic Boundary Value Problems*. Lecture Notes in Computational Science and Engineering, vol. 63, Springer, 2008.

[3] M. Bebendorf, M. Bollhöfer, and M. Bratsch: *Hierarchical matrix approximation with blockwise constraints. BIT Numerical Mathematics* **53** (2013), 311–339.

[4] M. Bebendorf and R. Venn: *Constructing nested bases approximations from the entries of non-local operators. Numerische Mathematik* **121** (2012), 609–635.

[5] M.D. Buhmann: *Radial Basis Functions*, Cambridge University Press, Cambridge, UK, 2003.

[6] J. Cherrie, R. Beatson, and G. Newsam: *Fast evaluation of radial basis functions: Methods for generalized multiquadrics in* $\mathbb{R}^n$. *SIAM J. Sci. Comp.* **23** (2002), 1549–1571.

[7] B. Diederichs and A. Iske: *Improved estimates for condition numbers of radial basis function interpolation matrices*. Preprint # 2016-16, Hamburger Beiträge zur Angewandten Mathematik, University of Hamburg, 2016.

[8] G. Fasshauer: *Meshfree Approximation Methods with Matlab*. World Scientific, Singapore, 2007.

[9] G. Fasshauer and M. McCourt: *Kernel-based Approximation Methods using MATLAB*. Interdisciplinary Mathematical Sciences, vol. 19, World Scientific, 2016.

[10] B. Fornberg and N. Flyer: *A Primer on Radial Basis Functions with Applications to the Geosciences*. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 87, SIAM, 2015.

[11] S. Goreinov and E. Tyrtyshnikov: *The maximal-volume concept in approximation by low-rank matrices*. Contemporary Mathematics **280** (2001), 47–51.

[12] W. Hackbusch: *Hierarchical Matrices: Algorithms and Analysis*. Springer Series in Computational Mathematics, vol. 49, Springer, 2015.

[13] W. Hackbusch: *Survey on the technique of hierarchical matrices*. Vietnam Journal of Mathematics **44** (2016), 71–101.

[14] A. Iske: *Multiresolution Methods in Scattered Data Modelling*. Lecture Notes in Computational Science and Engineering, vol. 37, Springer, 2004.

[15] H. Wendland: *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics, vol. 17, Cambridge University Press, Cambridge, UK, 2005.