CSIAM Trans. Appl. Math.
doi:

Vol. **x**, No. 1, pp. 1-34
February 202x

Multi-layer hierarchical structures

Jianlin Xia*

Abstract. In structured matrix computations, existing rank structures such as hierarchically semiseparable (HSS) forms admit fast and stable factorizations. However, for discretized problems, such forms are restricted to 1D cases. In this work, we propose a framework to break such a 1D barrier. We study the feasibility of designing multilayer hierarchically semiseparable (MHS) structures for the approximation of dense matrices arising from multi-dimensional discretized problems such as certain integral operators. The MHS framework extends HSS forms to higher dimensions via the integration of multiple layers of structures, i.e., structures within the dense generator representations of HSS forms. Specifically, in the 2D case, we lay theoretical foundations and justify the existence of MHS structures based on the fast multipole method (FMM) and algebraic techniques such as representative subset selection. Rigorous numerical rank bounds and conditions for the structures are given. Representative subsets of points and a multi-layer tree are used to intuitively illustrate the structures. The MHS framework makes it convenient to explore multidimensional FMM structures. MHS representations are suitable for stable direct factorizations and can take advantage of existing methods and analysis well developed for simple HSS methods. Numerical tests for some discretized operators show that the appropriate inner-layer numerical ranks are significantly smaller than the off-diagonal numerical ranks used in standard HSS approximations.

1 Introduction

Rank structured matrices have been widely used for the fast direct solution of some integral and differential equations, especially elliptic problems. See [1, 5, 8, 9, 12, 15, 29, 32, 37, 38, 41] for a partial list of references. A basic idea of these methods is to approximate certain dense (intermediate) matrices or fill-in by rank structured forms. Such dense matrices include discretized integral operators, inverses of discretized PDEs, and Schur complements in the direct factorizations of some sparse matrices. PDE/integral equation theories together with linear algebra techniques have been used to show the existence of the rank structures. That is, appropriate off-diagonal blocks of these dense matrices have small numerical ranks.

*Corresponding author.

Several rank structured representations have been designed for the approximation of these dense matrices. Among the most widely used ones are hierarchical structured representations such as $\mathcal{H}$ [4], $\mathcal{H}^2$ [5], and hierarchically semiseparable (HSS) matrices [8,42]. $\mathcal{H}/\mathcal{H}^2$ matrices and matrices based on the fast multipole method (FMM) are applicable to 2D and 3D cases. However, factorizations of such matrices usually involve recursion or inversion [6, 15] and their numerical behaviors such as stability are unclear. The HSS form mainly aims at 1D cases, and is essentially a special $\mathcal{H}^2$ form that explores the weak admissibility [19]. It is based on simple domain bisections and is thus easier to implement and analyze. It is also widely accessible to the general scientific computing community. In particular, efficient, stable, and scalable HSS operations (especially ULV-type factorizations [8, 42]) are available. Moreover, the hierarchical approximation accuracy and backward stability of HSS methods are well studied [33, 34]. For more general sparse problems, the applicability of HSS matrices can be extended via the integration into sparse matrix techniques such as nested dissection [11] and the multifrontal method [10].

Existing HSS-based structured direct solvers work well for 1D discretized integral equations and 2D discretized elliptic PDEs. However, the efficiency is usually less satisfactory for higher dimensions. It is possible to still approximate the dense matrices corresponding to two dimensions by HSS forms. Although this simplifies the implementation, the performance of the relevant matrix operations is far from optimal for large sizes due to the large off-diagonal numerical ranks.

In some recent studies, additional structures within some HSS approximations have been explored. In fact, it has been noticed that, in some applications, the dense blocks (called generators) that define the HSS forms are also structured [9, 18, 39, 43, 45]. By taking advantage of such structures, it is possible to design multi-dimensional structured algorithms for dense discretized matrices just based on simple HSS methods. For example, in a fast selected inversion algorithm [44], some diagonal and off-diagonal blocks of the inverse of a sparse matrix are approximated by HSS and low-rank forms, respectively. For some 2D discretized integral operators, the method in [9] exploits the inner structures with the aid of some integral equation techniques.

In this work, we lay theoretical foundations for a multi-layer hierarchically semiseparable (MHS) structure for multiple dimensions and design an MHS representation. We show the feasibility of using MHS forms to approximate some dense discretized matrices. We exploit multiple layers of hierarchial or tree structures under the general FMM framework. For some discretized matrices on 2D domains, if HSS forms are used for the approximation, we show that the dense generators have inner HSS or low-rank structures similarly to the work in [9]. Unlike the method in [9], we consider general FMM interactions between all the subdomains resulting from nested bisection based on algebraic methods. A structure-preserving rank-revealing factorization [17, 45] is used to produce subsets of representative points in the mesh during off-diagonal compression. These representative points facilitate the study of inner structures. In [9], some boundary mesh points essentially play the role of representative points. Here, more general represen-

tative points at multiple hierarchical levels are used and help to justify the existence of inner structures following the FMM ideas.

Rigorous numerical rank bounds for relevant matrix blocks are given. This is based on a generalization of the usual concept of well-separated point sets. Unlike most FMM studies that estimates rank bounds based on the numbers of terms in degenerate series expansions, here we give concrete numerical rank bounds for numerically low-rank blocks. This makes it clear to understand the storage and efficiency of the structured forms.

Following all the rank studies, a systematic definition of the MHS representation is then given. It has a multi-layer tree structured form. We analyze the storage of the MHS form. Under some conditions, the storage is nearly linear in the matrix size N and is lower than a direct HSS approximation. The MHS representation is also more general than a related structured form in [9].

The MHS framework makes it very convenient to explore multi-dimensional FMM structures. There are two major benefits. One is that it enables to take advantage of well-developed HSS methods to design fast and stable MHS operations, especially direct factorizations that are previously difficult with existing multi-dimensional rank structures. The multi-layer framework enables us to repeatedly perform simple structured operations such as HSS or low-rank ones so as to keep MHS algorithms convenient to design and analyze. This makes it feasible to avoid HSS inversion and recompression in [9] and avoid potentially unstable recursive operations in [6,15] when designing multi-dimensional structured factorizations. We mention some hints for designing efficient MHS algorithms.

Another benefit is that existing complexity, accuracy, and stability studies for HSS methods can be readily generalized to MHS methods, which avoids tedious technical analysis and guarantees the efficiency and reliability of MHS methods. For example, it has been shown in [33, 34] that the HSS structure has a natural stability enhancement benefit in the sense that numerical errors propagate only by $O(\log N)$ times along some tree paths. With the use of multi-layer tree structures in MHS forms, this is still the situation so that we can expect nice stability for MHS algorithms.

We verify the feasibility of MHS structures via some numerical tests. For discretized kernel matrices where the kernel functions have degenerate approximations when evaluated at well-separated points in two dimensions, an HSS approximation would have maximum off-diagonal rank (called HSS rank) growing quickly with the matrix size N, while in the MHS approximation, a rank bound for the structure measurement (called MHS rank) stays about the same or only grows very slowly. For reasonable N, the MHS ranks are significantly smaller than the HSS ranks.

The outline of the paper is as follows. We first generalize the concept of separated sets and discuss representative subset selection in Section 2. The design of MHS structures is shown in detail in Section 3, followed by some potential ideas for designing MHS algorithms in Section 4. Section 5 shows the numerical tests. The following is a list of some notation.

- For a set of points Ω, $|\Omega|$ denotes its cardinality.
- For a binary tree $\mathcal{T}$, its nodes are labeled with a single index, say, i. The sibling and parent of i are denoted $\mathrm{sib}(i)$ and $\mathrm{par}(i)$, respectively. $\mathrm{root}(\mathcal{T})$ denotes the root of $\mathcal{T}$.
- For a matrix A and index sets $\mathbf{I}$ and $\mathbf{J}$, $A|_{\mathbf{I}}$ and $A|_{:\times\mathbf{J}}$ denote submatrices of A consisting of its rows and columns selected by $\mathbf{I}$ and $\mathbf{J}$, respectively, and $A|_{\mathbf{I}\times\mathbf{J}}$ corresponds to the selection of row and column entries based on $\mathbf{I}$ and $\mathbf{J}$, respectively.
- Sometimes, the Matlab notation $1:n$ is used to mean 1,2,…,n.
- $\sigma_j(C)$ denotes the jth largest singular value of a matrix C.
- As usual, the notation $O(\cdot)$ is used to avoid extra notation for constants that do not depend on parameters such as matrix sizes.

2 Generalization of separated sets, representative subset selection, and numerical rank estimation

In this section, we generalize the concept of well-separated sets and introduce the notion of representative subset selection in the FMM context. Furthermore, we give relevant numerical rank estimates for discretized kernel matrices where the kernel functions have degenerate approximations when evaluated at well-separated points. These serve as preliminaries for our later design of the MHS structure.

2.1 Generalization of well-separated sets

Consider the discretization of a kernel function ϕ of the form $\phi(|y-z|)$, where y and z are points inside certain domains and $|y-z|$ is the distance between y and z. Some examples of ϕ are $\frac{1}{|y-z|}$, $\frac{1}{|y-z|^2}$, and $\log|y-z|$. Suppose Ω_1 and Ω_2 are two sets of points. We look at the numerical rank of the following discretized matrix:

$$K=\left(\phi(|y_i-z_j|)\right)_{y_i\in\Omega_1,z_j\in\Omega_2}. \tag{2.1}$$

For convenience, we refer to K in (2.1) as an *interaction (matrix)* between Ω_1 and Ω_2.

In the FMM, two sets Ω_1 and Ω_2 are generally considered well separated if their distance is greater than or equal to their diameters so that $\phi(|y-z|)$ for $y\in\Omega_1$ and $z\in\Omega_2$ can be approximated by a series with a finite number of terms to reach any given accuracy [2,16]. Correspondingly, the matrix K in (2.1) has a small numerical rank. More specifically, we define separated sets in the following way, which slightly relaxes the usual concept of separated sets.

Definition 2.1. (*Separated sets*) Use the notation $\delta(y_0,\Omega_1)$ to denote the radius of a set Ω_1 with respect to a center y_0 in the following sense:

$$\delta(y_0,\Omega_1) = \max_{y_i\in\Omega_1} |y_0-y_i|.$$

A set Ω_2 is α-separated from Ω_1 for a constant $\alpha>1$ if there exists a point $y_0\in\Omega_1$ such that for all points $z\in\Omega_2$,

$$|z-y_0| \geq \alpha\cdot\delta(y_0,\Omega_1).$$

Since we assume a fixed constant $\alpha>1$ is used for all our studies, we often just say two sets are separated.

Remark 2.1. (*Kernel expansion assumption*) For convenience, in all the following discussions, we assume $\phi(|y-z|)$ has a *finite-term degenerate expansion* with respect to a relative tolerance ε for y and z respectively in two sets that are α-separated. That is,

$$\phi(|y-z|) = \sum_{i=1}^{r_0} f_i(y)g_i(z)+e, \tag{2.2}$$

where the approximation error satisfies, for an appropriate constant μ,

$$|e| \leq \frac{\mu}{\alpha^{r_0}}|\phi(|y-z|)| \leq \varepsilon|\phi(y-z)|, \tag{2.3}$$

Remark 2.2. Here, we focus on shift-invariant kernel functions $\phi(|y-z|)$ just for convenience. It makes K symmetric so as to eliminate some technical details when we derive the essential ideas. Moreover, such functions frequently arise from some PDE/integral equation problems. For some cases, actual forms of the error bound (2.3) have been derived [2, 16, 28, 30]. Note that our ideas do not rely on specific forms of $\phi(|y-z|)$ and can be extended to the case $\phi(y-z)$ or smooth functions $\phi(y,z)$ that allow degenerate approximations for well-separated y and z variables.

With the kernel expansion assumption, the matrix K in (2.1) corresponding to separated sets Ω_1 and Ω_2 can be approximated by a matrix of small rank, with the entrywise relative error ε. On the other hand, r_0 in (2.2) is not exactly the so-called numerical rank of K. To rigorously consider the numerical rank, we state a precise definition of ε-rank as follows. Some ε-rank estimates will then be given later.

Definition 2.2. The *numerical rank* or *ε-rank* of a matrix K is the number of singular values of K that are greater than $\varepsilon\|K\|_2$ for a tolerance ε, as often used in low-rank approximation methods.

Remark 2.3. Clearly, K has ε-rank at most r if $\sigma_{r+1}(K)\leq\varepsilon\sigma_1(K)$. Thus, if K can be approximated by a rank-r matrix with 2-norm absolute accuracy no larger than $\varepsilon\|K\|_2$, then K has ε-rank at most r according to the Eckart-Young theorem.

Then consider more general cases where Ω_1 and Ω_2 are not well separated. For example, suppose Ω_1 and Ω_2 are 2D sets as in Figure 1(a). (Later, we do not strictly distinguish a domain and a set. In our figures, we usually only draw a domain to indicate a set.) The two sets are not separated. In the FMM, the sets are further partitioned. We can get a subset $\tilde{\Omega}_1$ of Ω_1 as in Figure 1(b) so that we can choose $y_0 \in \tilde{\Omega}_1$ as shown and verify that $\tilde{\Omega}_1$ and Ω_2 are α-separated for a constant $\alpha > 1$. For example, if Ω_1 and Ω_2 are two unit squares, then $\delta(y_0,\Omega_1) \le \frac{\sqrt{10}}{4}$, and for any $z \in \Omega_2$,

$$|z-y_0| \ge \frac{3}{4}\sqrt{2} \ge \alpha \cdot \delta(y_0,\Omega_1), \quad \text{with } \alpha \approx 1.34.$$

(a) Two sets in 2D that are not well separated (b) Separated sets after partitioning one set

Figure 1: Separation of two sets in two dimensions.

After this, the set $\Omega_1 \backslash \tilde{\Omega}_1$ can be repeatedly partitioned in a similar way. This is generally done for $O(\log|\Omega_1|)$ times in the FMM. Thus, the numerical rank of K in (2.1) is related to $O(\log|\Omega_1|)$. Later for convenience, we say that Ω_1 and Ω_2 are weakly or logarithmically separated, as consistent with the weak admissibility in [19].

Definition 2.3. (*Logarithmically-separated sets*) A set of points Ω_1 is logarithmically α-separated from another set Ω_2 with respect to a constant $\alpha > 1$ if Ω_1 can be partitioned into $O(\log|\Omega_1|)$ subsets that are all α-separated from Ω_2, except for possibly a constant number of subsets containing $O(1)$ points.

Thus, in Figure 1(a), Ω_1 is logarithmically separated from Ω_2. Similarly, it can be verified that in each example in Figure 2, Ω_1 is logarithmically separated from Ω_2.

We then give estimates for the actual numerical ranks of some interaction matrices.

Lemma 2.1. *Suppose ϕ has a finite-term degenerate expansion with respect to a fixed relative tolerance ε and (2.2)–(2.3) hold.*

- *If two sets Ω_1 and Ω_2 are α-separated with $\alpha > 1$, then the discretized matrix K in (2.1) has ε-rank (at most) $r_0 = \left\lceil \log_\alpha \frac{\mu\sqrt{n}}{\varepsilon} \right\rceil$ or*

$$r_0 = O(\log n) + O(|\log\varepsilon|), \tag{2.4}$$

where $n = \min\{|\Omega_1|, |\Omega_2|\}$.

(a) One example (b) Another example

Figure 2: Examples of logarithmically-separated sets in two dimensions, where the points in Ω_1 correspond to a narrow band.

- *If Ω_1 can be partitioned into l subsets each α-separated from Ω_2, then K has ε-rank (at most) $r_1 = lr_0$ or*
$$r_1 = O(l\log n) + O(l|\log\varepsilon|). \tag{2.5}$$
For the two cases, K also has $(\sqrt{n}\varepsilon)$-ranks $O(|\log\varepsilon|)$ and $O(l|\log\varepsilon|)$, respectively.

Proof. If Ω_1 and Ω_2 are α-separated, according to (2.2)–(2.3),

$$K = UV^T + E, \quad \text{with}$$
$$U = (f_j(y_i))_{y_i\in\Omega_1, j=1:r_0}, \quad V = (g_j(z_i))_{z_i\in\Omega_2, j=1:r_0}, \quad |E_{ij}| \le \frac{\mu}{\alpha^{r_0}}|K_{ij}|.$$

This means
$$\|E\|_2 \le \|E\|_F \le \frac{\mu}{\alpha^{r_0}}\|K\|_F \le \frac{\mu\sqrt{n}}{\alpha^{r_0}}\|K\|_2.$$
Setting $\frac{\mu\sqrt{n}}{\alpha^{r_0}} = \varepsilon$ yields $\sigma_{r_0+1}(K) \le \|E\|_2 \le \varepsilon\|K\|_2$. This gives r_0 in (2.4) since μ is a constant. If we relax the tolerance to be $\sqrt{n}\varepsilon$, then the numerical rank becomes $O(|\log\varepsilon|)$.

Then suppose Ω_1 can be partitioned into l subsets Ω_{1k}, $k = 1,\ldots,l$, each α-separated from Ω_2. Accordingly, K can be partitioned into block rows

$$K_k = \left(\phi(|y_i - z_j|)\right)_{y_i\in\Omega_{1k}, z_j\in\Omega_2}, \quad k = 1,\ldots,l. \tag{2.6}$$

Similarly, we have

$$K_k = U_k {V_k}^T + E_k, \quad \text{with} \quad \|E_k\|_F \le \frac{\mu}{\alpha^{r_0}}\|K_k\|_F. \tag{2.7}$$

Thus,

$$K = UV^T + E, \quad \text{with}$$
$$U = \mathrm{diag}(U_1,\ldots,U_l), \quad V = \begin{pmatrix} V_1 & \cdots & V_l \end{pmatrix}, \quad E = \begin{pmatrix} E_1 \\ \vdots \\ E_l \end{pmatrix},$$

where without loss of generality, we suppose the order of the submatrices follows the order of the set partition. Then

$$\|E\|_2 \le \|E\|_F = \sqrt{\sum_{k=1}^{l} \|E_k\|_F^2} \le \frac{\mu}{\alpha^{r_0}} \sqrt{\sum_{k=1}^{l} \|K_k\|_F^2} = \frac{\mu}{\alpha^{r_0}} \|K\|_F \le \frac{\mu\sqrt{n}}{\alpha^{r_0}} \|K\|_2. \tag{2.8}$$

Setting $\frac{\mu\sqrt{n}}{\alpha^{r_0}} = \varepsilon$ still yields r_0 as in (2.4), and since U has column size lr_0, we have $\sigma_{lr_0+1}(K) \le \|E\|_2 \le \varepsilon\|K\|_2$. K then has ε-rank at most $r_1 = lr_0$. □

Remark 2.4. In previous studies, r_0 is typically decided based on the entrywise approximation error of K so that r_0 in (2.4) is $O(|\log\varepsilon|)$. Here, we are interested in an estimate of the actual numerical rank of K, which yields the additional $\log n$ term in (2.4). Although this may look pessimistic, the numerical ranks are usually very small in practice. In addition, the extra $\log n$ term does not substantially impact the global performance of relevant algorithms due to a rank pattern study in [36].

Remark 2.5. In the $O(\cdot)$ notation in (2.4) and (2.5), the hidden constants only depend on fixed α and μ. This will be the case for all our rank estimates. In our results, any dependence on matrix sizes will be explicitly indicated.

The second part of the proposition essentially shows an *accumulative effect* of numerical ranks. In particular, numerical ranks of K related to logarithmically-separated sets look like the following.

Corollary 2.1. *Suppose ϕ has a finite-term degenerate expansion with respect to a fixed relative tolerance ε and (2.2)–(2.3) hold.*

- *If Ω_1 and Ω_2 are logarithmically α-separated, then K has ε-rank (at most) $r_1 = O(r_0 \log m)$ with r_0 in (2.4) or*
$$\tilde{r}_1 = O((\log m)(\log n)) + O((\log m)|\log\varepsilon|).$$
where $m = |\Omega_1|$, $n = \min\{|\Omega_1|, |\Omega_2|\}$.
- *If Ω_1 can be partitioned into s subsets each logarithmically α-separated from Ω_2, then K has ε-rank (at most) $\tilde{r}_2 = O(sr_0 \log m)$.*

Proof. If Ω_1 and Ω_2 are logarithmically α-separated, suppose Ω_1 can be partitioned into $\tilde{l} = O(\log m)$ subsets Ω_{1k}, $k = 1, \ldots, \tilde{l}$, which are all α-separated from Ω_2, except for possibly a constant number of subsets containing $O(1)$ points. For any such subset Ω_{1k} not α-separated from Ω_2, as in the proof of Lemma 2.1, we can set $U_k = I$, $V_k = K_k$, $E_k = 0$ so that (2.7) still holds. Thus, the proof of Lemma 2.1 for (2.8) still holds, so that $\sigma_{\tilde{l}r_0+1}(K) \le \|E\|_2 \le \varepsilon\|K\|_2$ with r_0 as in (2.4).

If Ω_1 can be partitioned into s subsets Ω_{1k}, $k = 1, \ldots, s$, each logarithmically α-separated from Ω_2, we can see that (2.8) still holds. Thus, we can pick $\tilde{r}_2 = O(sr_0 \log m)$ with r_0 as in (2.4) so that $\sigma_{\tilde{r}_2+1}(K) \le \|E\|_2 \le \varepsilon\|K\|_2$. □

2.2 Structure-preserving rank-revealing factorization and representative subset selection

For a matrix such as K in (2.1) with a small numerical rank $\hat{r}_0$, a rank-revealing factorization may be used to compute a low-rank approximation to it. Since we are interested in exploring additional structures within the low-rank approximation, we suppose K_{11} is an $\hat{r}_0 \times \hat{r}_0$ invertible submatrix of K that has the maximum volume (determinant in modulus) [14,31] among all $\hat{r}_0 \times \hat{r}_0$ submatrices. According to [14], we can get an approximation

$$K \equiv \Pi_1 \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \Pi_2^T \approx UBV^T, \quad \text{with} \tag{2.9}$$

$$U = \Pi_1 \begin{pmatrix} K_{11} \\ K_{21} \end{pmatrix}, \quad B = K_{11}^{-1}, \quad V^T = \begin{pmatrix} K_{11} & K_{12} \end{pmatrix} \Pi_2^T \equiv K|_{\mathbf{I}}, \tag{2.10}$$

where U and V^T correspond to selected columns and rows of K, respectively, and Π_1 and Π_2 are permutation matrices. (2.9) can be written in another form:

$$K \approx UK|_{\mathbf{I}}, \quad \text{with } U = \Pi \begin{pmatrix} I \\ E \end{pmatrix}, \quad E = K_{21}K_{11}^{-1}. \tag{2.11}$$

This clearly shows the fact that $K|_{\mathbf{I}}$ corresponds to selected rows of K. A numerically stable way to find (2.9) or (2.11) is the strong rank-revealing QR (SRRQR) factorization [17], which results in E with entries bounded by a small constant. The factorization (2.11) is also called an interpolative decomposition [20] or *structure-preserving rank-revealing* (SPRR) factorization [45].

For convenience, we call $K|_{\mathbf{I}}$ in (2.10) *representative rows* from K, following the terminology in [18]. For K from (2.1), $\mathbf{I}$ corresponds to selected points in Ω_1. Therefore, (2.11) can be understood as the selection of *representative points* (similar to terms in [3,31]). In another word, $\mathbf{I}$ is a *representative subset* (also called skeleton in [9,21]) from Ω_1. If Ω_1 and Ω_2 are well separated, we can then get the approximation (2.11) with small $\hat{r}_0$.

If Ω_1 is logarithmically separated from Ω_2, suppose Ω_1 can be partitioned into $l = O(\log|\Omega_1|)$ subsets Ω_{1k}, each separated from Ω_2. Then we can approximate K as follows:

$$K \equiv \begin{pmatrix} K_1 \\ \vdots \\ K_l \end{pmatrix} \approx UBV^T, \quad \text{with} \tag{2.12}$$

$$U = \mathrm{diag}(U_1,\ldots,U_1), \quad B = \mathrm{diag}(B_1,\ldots,B_l), \quad V = \begin{pmatrix} V_1 & \cdots & V_l \end{pmatrix},$$

where U_i, B_i, V_i have forms like in (2.10). In particular, V^T still corresponds to selected rows of K. Thus, (2.12) can still be understood as the selection of a subset of representative points from Ω_1.

Corresponding to Lemma 2.1, we can estimate the accuracy of the approximations (2.9) and (2.12) as follows.

Proposition 2.1. Suppose ϕ has a finite-term degenerate expansion with respect to a fixed relative tolerance and (2.2)–(2.3) hold.

- If Ω_1 and Ω_2 are α-separated, then for sufficiently large $m \equiv \max\{|\Omega_1|,|\Omega_2|\}$, K_{11} in (2.9) can be chosen to have size
$$\hat{r}_0 = O(\log m) + O(|\log\varepsilon|), \tag{2.13}$$
so that the approximation (2.9) has error bound $\varepsilon\|K\|_2$.
- If Ω_1 can be partitioned into l subsets, each α-separated from Ω_2, and K has the approximation in (2.12) corresponding to (2.6), then for sufficiently large $m \equiv \max\{|\Omega_1|, |\Omega_2|\}$, B in (2.12) can be chosen to have size
$$\hat{r}_1 = O(l\log m) + O(l|\log\varepsilon|), \tag{2.14}$$
so that the approximation (2.12) has error bound $\varepsilon\|K\|_2$.

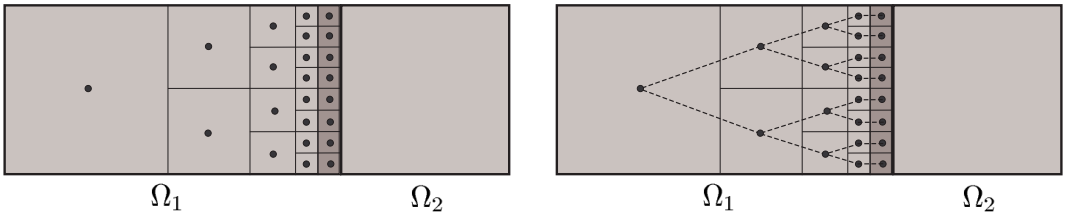
Proof. Suppose Ω_1 and Ω_2 are α-separated. For K_{11} in (2.10) with the maximum volume among all $\hat{r}_0 \times \hat{r}_0$ submatrices of K, according to [14], $\|K_{22} - K_{21}K_{11}^{-1}K_{12}\|_{\max} \le (\hat{r}_0+1)\sigma_{\hat{r}_0+1}(K)$. Then
$$\|K - UBV^T\|_{\max} \le (\hat{r}_0+1)\sigma_{\hat{r}_0+1}(K).$$
Let $n = \min\{|\Omega_1|,|\Omega_2|\}$. From the proof of Lemma 2.1, we have $\sigma_{\hat{r}_0+1}(K) \le \frac{\mu\sqrt{n}}{\alpha^{\hat{r}_0}}\|K\|_2 \le \frac{\mu\sqrt{m}}{\alpha^{\hat{r}_0}}\|K\|_2$. This means,
$$\|K - UBV^T\|_2 \le m\|K - UBV^T\|_{\max} \le \frac{\mu(r_0+1)m^{3/2}}{\alpha^{\hat{r}_0}}\|K\|_2.$$
Thus, by choosing, say, $\hat{r}_0 = \lceil 2\log_\alpha \frac{\mu m}{\varepsilon} \rceil$, we have
$$\|K - UBV^T\|_2 \le \varepsilon\left(\left\lceil 2\log_\alpha \frac{\mu m}{\varepsilon}\right\rceil + 1\right)/\left(\frac{\mu m^{1/2}}{\varepsilon}\right) \le \varepsilon\|K\|_2,$$
for sufficiently large m.

If Ω_1 can be partitioned into l subsets, each α-separated from Ω_2, and K has the approximation in (2.12) corresponding to (2.6), then choose B_k to be the inverse of the $\hat{r}_0 \times \hat{r}_0$ submatrix of K_k with the largest volume. Accordingly,
$$\|K_k - U_k B_k V_k^T\|_{\max} \le \frac{\mu(\hat{r}_0+1)\sqrt{m}}{\alpha^{\hat{r}_0}}\|K_k\|_2.$$
Thus,
$$\begin{aligned}\|K - UBV^T\|_2 &\le m\|K - UBV^T\|_{\max} = m\max_k \|K_k - U_k B_k V_k^T\|_{\max} \\ &\le \frac{\mu(\hat{r}_0+1)m^{3/2}}{\alpha^{\hat{r}_0}}\|K_k\|_2 \le \frac{\mu(\hat{r}_0+1)m^{3/2}}{\alpha^{\hat{r}_0}}\|K\|_2.\end{aligned}$$
Again, by choosing $\hat{r}_0 = \lceil 2\log_\alpha \frac{\mu m}{\varepsilon} \rceil$ as in (2.13) or $\hat{r}_1$ as in (2.14), we get the approximation of K in (2.12) with $\|K - UBV^T\|_2 \le \varepsilon\|K\|_2$. □

This proposition indicates that, to take advantage of the numerical low-rankness of K, we may use an SPRR factorization to select the number of representative points equal to the ε-ranks given in the proposition.

Remark 2.6. Later for convenience, we will not write ε in the rank estimates since it is fixed. Thus, (2.13) becomes $\hat{r}_0 = O(\log m)$ and (2.14) becomes $\hat{r}_1 = O(\log^2 m)$. For example, we may simply say a matrix has ε-rank $O(\log m)$.

When Ω_1 and Ω_2 are not separated and, say, are located within two adjacent boxes like in Figure 3(a), then we can partition Ω_1 into subsets $\Omega_{1,i}$ at multiple levels, as done in the FMM. Suppose the subsets are located within boxes at $l_{\max}$ levels, with the boundary level or level $l_{\max}$ right adjacent to Ω_2. For convenience, suppose each box at level $l_{\max}$ has a constant size h. Level $l_{\max}-1$ also consists of boxes of size h. Also, suppose the partition is fine enough so that the number of points within each box at these two levels is a constant. The box sizes increase for boxes away from the boundary. That is, a box at level l has size twice of that at level $l+1$. (Here for simplicity, we assume all the points are uniformly distributed. Otherwise, adaptive partitioning of the sets would be needed.) Then each box at levels $l_{\max}-1, l_{\max}-2, \ldots$ is well separated from Ω_2, so that representative points can be selected from it, as also done in [9]. (In practice, all the points within a box at levels $l_{\max}$ and $l_{\max}-1$ can be considered as representative points.) The process yields a sequence of representative points.



(a) Representative points at different levels (b) Representative subset tree

Figure 3: Representative points in Ω_1 in the study of the interaction between Ω_1 and Ω_2 in the FMM and the corresponding representative subset tree for organizing the points. The locations of the representative points (marked as black dots) are for illustration purpose only.

Following the selection of representative points like in Figure 3, the collection $\mathbf{I}$ of all the representative points within Ω_1 is also said to be a *representative subset* of Ω_1 (with respect to Ω_2). The process for selecting $\mathbf{I}$ involves the partitioning of Ω_1 into appropriate subsets Ω_{1i} as in the FMM (see Figure 3(a)) followed by the SPRR factorization of the interaction between Ω_{1i} and Ω_2. For convenience, this process of selecting the representative subset $\mathbf{I}$ is denoted by

$$\mathbf{I} = \mathrm{RS}(\Omega_1 | \Omega_2). \tag{2.15}$$

Also, we call $\hat{\mathbf{I}} \equiv \Omega_1 \setminus \mathbf{I}$ the *residual subset* of $\mathbf{I}$ in Ω_1.

2.3 Proper ordering of representative subset

In Figure 3(a), the points in the representative subset $\mathbf{I}$ are located at $l_{\max} = O(\log|\mathbf{I}|)$ hierarchical levels. We can organize these points with the aid of a binary tree, called a *representative subset tree*, where each node corresponds to the box enclosing subset Ω_{1i} and the representative points $\mathbf{I}_i = \mathrm{RS}(\Omega_{1i}|\Omega_2)$. A larger box at level $l-1$ and two smaller adjacent boxes at level l define a parent-children relationship. The boxes at level $l_{\max}$ are associated with the leaves. See Figure 3(b). For convenience, we use $\mathbf{I}^{(l)}$ to denote all the representative points at level l of the tree, called the l-th *slice* of $\mathbf{I}$, so that

$$\mathbf{I} = \bigcup_{l=1}^{l_{\max}} \mathbf{I}^{(l)}, \quad \text{with} \quad \mathbf{I}^{(l)} = \bigcup_{i \text{ at level } l} \mathbf{I}_i. \tag{2.16}$$

In our design of MHS structures later, it is important to order the points within $\mathbf{I}$ in an appropriate way so as to obtain desired rank structures. Here, we order them following the postorder of the representative subset tree. This ensures that the representative points within each slice $\mathbf{I}^{(l)}$ are ordered consecutively in a uniform way.

Definition 2.4. (*Proper order*) If the points in $\mathbf{I}$ are ordered following the postorder of the corresponding representative subset tree, we say that $\mathbf{I}$ is *properly ordered*.

3 MHS structures

We now lay the foundation for MHS structures and show the design of MHS representations.

3.1 HSS structures and motivation for MHS structures

The HSS structure is an efficient tool to study the mutual interactions for points inside 1D domains [8]. One way to define an HSS form is as follows [42]. In an HSS form, an $N \times N$ matrix H is partitioned into a block 2×2 form, and the partition is then recursively done on the two diagonal blocks. This can be organized through a binary tree T called HSS tree. The resulting off-diagonal blocks at all hierarchical levels are represented or approximated by low-rank forms.

In particular, assume i is a node of T with two children c_1 and c_2. In an HSS representation, i is associated with some matrices $D_i, U_i, V_i, R_i, W_i, B_i$ (called HSS generators). These generators are hierarchically defined as

$$D_i \equiv H|_{\mathcal{I}_i \times \mathcal{I}_i} = \begin{pmatrix} D_{c_1} & U_{c_1} B_{c_1} V_{c_2}^T \\ U_{c_2} B_{c_2} V_{c_1}^T & D_{c_2} \end{pmatrix}, U_i = \begin{pmatrix} U_{c_1} R_{c_1} \\ U_{c_2} R_{c_2} \end{pmatrix}, V_i = \begin{pmatrix} V_{c_1} W_{c_1} \\ V_{c_2} W_{c_2} \end{pmatrix}, \tag{3.1}$$

whether $\mathcal{I}_i$ is the index set for D_i in H and satisfies the hierarchical relation $\mathcal{I}_i = \mathcal{I}_{c_1} \cup \mathcal{I}_{c_2}$. For the root node k, $\mathcal{I}_k = \{1:N\}$. It can be seen that U and V are also basis matrices of the

blocks $H|_{\mathcal{I}_i\times(\mathcal{I}_k\setminus\mathcal{I}_i)}$ and $H|_{(\mathcal{I}_k\setminus\mathcal{I}_i)\times\mathcal{I}_i}$, called *HSS blocks*. The maximum rank or numerical rank of all the HSS blocks is called the *HSS rank* of H.

HSS matrices can be constructed based on direct compression, randomized sampling, or analytical strategies. See [7, 22, 24, 35, 45] for some examples.

The HSS structure has some significant benefits, include its simplicity, well-established fast and stable operations, and the convenient error and stability analysis. However, the structure focuses on 1D problems. For higher dimensions, it becomes less effective due to the high HSS ranks. On the other hand, many subblocks of the discretized matrix may still have small numerical ranks following the idea of the FMM, as indicated in the previous section.

Here, we seek to design multi-dimensional structures still based on HSS forms, so as to keep the structure simple and to take advantage of existing HSS algorithms and analysis. This involves the study of the interior structures within the HSS generators, so as to establish a new structure consisting of multiple layers of hierarchical forms. To illustrate this, we consider the discretization of a kernel ϕ over a 2D set Ω with the assumption in Remark 2.1, and the discretized matrix is

$$A=(\phi(|y_i-y_j|))_{y_i,y_j\in\Omega}. \tag{3.2}$$

(The diagonal entries A_{ii} may be specified otherwise.) Let $N=|\Omega|$. The matrix A is $N\times N$ and symmetric.

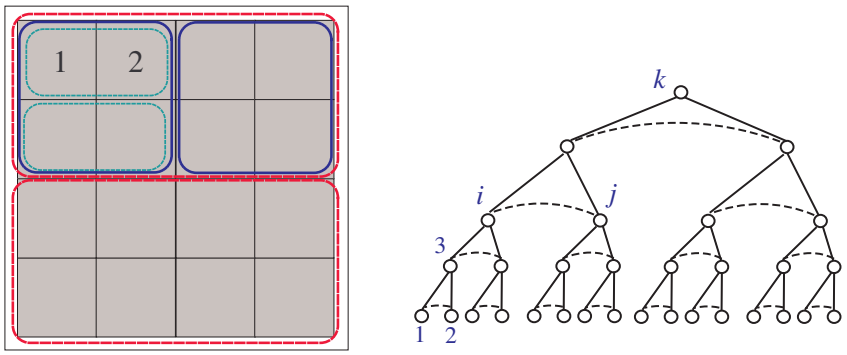
3.2 Outer layer structures

In the design of MHS structures, there are two layers of trees for the 2D case, an outer layer and an inner layer. To explore the outer layer tree structure, we use *nested bisection* to partition the domain/set Ω into a sequence of subdomains. That is, the domain is split into two subdomains, and each subdomain is recursively split. This is similar to the usual nested dissection partitioning [11], but does not involve a separator of points. A postordered binary tree $\mathbf{T}$ called *nested bisection tree* is then set up, where each leaf corresponds to a bottom level subdomain and each nonleaf node corresponds to an upper level domain or the union of the subdomains associated with its children. See Figure 4. Here, we suppose the root corresponds to the entire set Ω and is at level 0 and the leaves are at the largest level. Also for convenience, suppose all the subdomains at the same level of $\mathbf{T}$ include the same number of points.

Remark 3.1. In particular, if Ω is an $M\times M$ uniform mesh, we assume the nested bisection is done with alternating cuts along the horizontal and vertical directions so that each subsect Ω_i for a node i at level $\mathbf{l}$ of the nested bisection tree $\mathbf{T}$ includes $O(M_{\mathbf{l}}^2)$ points with

$$M_{\mathbf{l}}=M/2^{\lfloor \mathbf{l}/2\rfloor}. \tag{3.3}$$

This is just for the convenience of studying the matrix structures below.

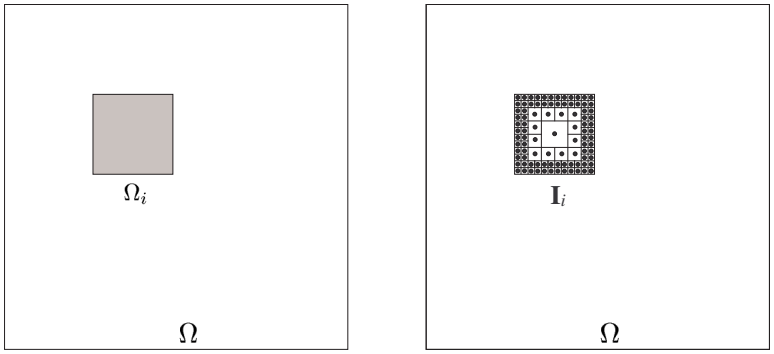


(a) Nested bisection of a domain (b) Nested bisection tree

Figure 4: Nested bisection of a domain and the corresponding nested bisection tree **T**.

After the application of nested bisection to the points in Ω, we reorder the matrix A in (3.2) following the ordering of the leaves of **T**. Later, we suppose A is already reordered. Then construct an HSS approximation to A by compressing the HSS blocks $A|_{\Omega_i \times (\Omega \backslash \Omega_i)}$ and $A|_{(\Omega \backslash \Omega_i) \times \Omega_i}$ for each subset $\Omega_i \subset \Omega$. This is done via the study of the interactions between Ω_i and its exterior or complement $\Omega \backslash \Omega_i$. (Here, we abuse notation and use $A|_{\Omega_i \times (\Omega \backslash \Omega_i)}$ to mean the submatrix corresponding to interaction between Ω_i and $\Omega \backslash \Omega_i$, although strictly speaking, Ω_i and $\Omega \backslash \Omega_i$ are not index subsets for A.)

For example, consider Ω_i in Figure 5(a) and its interaction with $\Omega \backslash \Omega_i$. Similarly to Figure 3, Ω_i is partitioned into multiple levels of boxes of different sizes. The sizes of the boxes double when their locations are farther away from the boundary by one level. These boxes not inside the boundary level are well separated from $\Omega \backslash \Omega_i$. Suppose there are m boxes along the boundary level, then the total number of boxes inside Ω_i that are well separated from $\Omega \backslash \Omega_i$ is $O(m)$. We can then select representative points from each box. The collection of these points is a representative subset $\mathbf{I}_i$ in Ω_i (Figure 5(b)).



(a) A subdomain Ω_i within Ω (b) Representative subset $\mathbf{I}_i$

Figure 5: Study of the interaction between Ω_i and $\Omega \backslash \Omega_i$ for a subdomain $\Omega_i \subset \Omega$.

For convenience, we introduce the following notation.

- For a subset $\Omega_i \subset \Omega$, we use $\hat{\Omega}_i \equiv \Omega \backslash \Omega_i$ to denote the complement of Ω_i in Ω.
- Like in (2.15), $\mathbf{I}_i \subset \Omega_i$ denotes the representative subset within a set Ω_i with respect to $\hat{\Omega}_i$:
$$\mathbf{I}_i = \mathrm{RS}(\Omega_i | \hat{\Omega}_i). \tag{3.4}$$
- $\hat{\mathbf{I}}_i = \Omega_i \backslash \mathbf{I}_i$ denotes the residual subset of $\mathbf{I}_i$ in Ω_i.
- Π_i denotes an appropriate permutation matrix like in (2.9).

In the following, we give detailed studies of the interactions among different subdomains of Ω so as to explore the rank structures in the off-diagonal blocks of A in (3.2). The basic procedure is similar to the HSS construction in [42], but with the off-diagonal compression replaced by representative subset selection. After the nested bisection ordering, we can write

$$A = \begin{pmatrix} A|_{\Omega_1 \times \Omega_1} & A|_{\Omega_1 \times \hat{\Omega}_1} \\ A_{\hat{\Omega}_1 \times \Omega_1} & A|_{\hat{\Omega}_1 \times \hat{\Omega}_1} \end{pmatrix} \begin{matrix} \Omega_1 \\ \hat{\Omega}_1 \end{matrix}.$$

The submatrix $A|_{\Omega_1 \times \hat{\Omega}_1}$ corresponds to the interaction between Ω_1 and $\hat{\Omega}_1$. According to the representative subset section, we can get a low-rank approximation like in (2.11):

$$A|_{\Omega_1 \times \hat{\Omega}_1} \approx U_1 A|_{\mathbf{I}_1 \times \hat{\Omega}_1}, \quad \text{with} \quad U_1 = \Pi_1 \begin{pmatrix} I \\ E_1 \end{pmatrix}. \tag{3.5}$$

The basis matrix U_1 is thus obtained.

Suppose $\Omega_2 \subset \Omega$ is the sibling set of Ω_1 in nested bisection. That is, Ω_1 and Ω_2 correspond to a pair of sibling nodes in the tree in Figure 4. Just like above, we can obtain a representative subset $\mathbf{I}_2 \subset \Omega_2$ (Figure 6(a)), so that

$$A|_{\Omega_2 \times \hat{\Omega}_2} = \begin{pmatrix} A|_{\Omega_2 \times \Omega_1} & A|_{\Omega_2 \times \hat{\Omega}_3} \end{pmatrix} \approx U_2 A|_{\mathbf{I}_2 \times \hat{\Omega}_2}, \quad \text{with} \quad U_2 = \Pi_2 \begin{pmatrix} I \\ E_2 \end{pmatrix}, \tag{3.6}$$

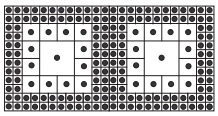
where $\hat{\Omega}_3$ is the complement of the parent set $\Omega_3 = \Omega_1 \cup \Omega_2$.

We can then write A as

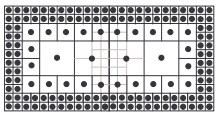
$$A = \begin{pmatrix} A|_{\Omega_1 \times \Omega_1} & A|_{\Omega_1 \times \Omega_2} & A|_{\Omega_1 \times \hat{\Omega}_3} \\ A|_{\Omega_2 \times \Omega_1} & A|_{\Omega_2 \times \Omega_2} & A|_{\Omega_2 \times \hat{\Omega}_3} \\ A|_{\hat{\Omega}_3 \times \Omega_1} & A|_{\hat{\Omega}_3 \times \Omega_2} & A|_{\hat{\Omega}_3 \times \hat{\Omega}_3} \end{pmatrix} \begin{matrix} \Omega_1 \\ \Omega_2 \\ \hat{\Omega}_3 \end{matrix},$$

As in symmetric HSS constructions, it is natural to let

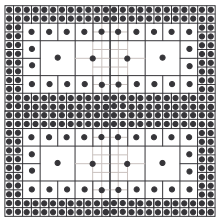
$$D_1 = A|_{\Omega_1 \times \Omega_1}, \quad D_2 = A|_{\Omega_2 \times \Omega_2}, \quad B_1 = A|_{\mathbf{I}_1 \times \mathbf{I}_2}, \tag{3.7}$$



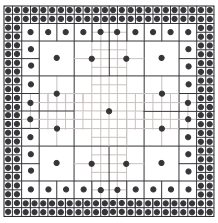
(a) Representative subsets in Ω_1 and Ω_2



(b) Representative subset in $\Omega_1 \cup \Omega_2$



(c) The upper level



(d) Representative subset at the upper level

Figure 6: Study of the interactions between some sets and their complements in Ω.

so that

$$D_3 \equiv \begin{pmatrix} A|_{\Omega_1\times\Omega_1} & A|_{\Omega_1\times\Omega_2} \\ A|_{\Omega_2\times\Omega_1} & A|_{\Omega_2\times\Omega_2} \end{pmatrix} \approx \begin{pmatrix} D_1 & U_1 B_1 U_2^T \\ U_2 B_2 U_1^T & D_2 \end{pmatrix}. \tag{3.8}$$

The choice of B_1 is due to the selection of the representative subsets so that it is just a submatrix of A [18,24].

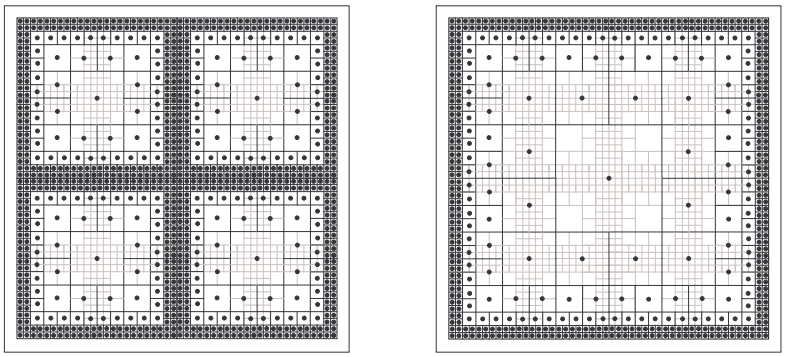
In HSS construction, the next step is to conduct compression associated with the parent node 3 so as to find a nested basis matrix $U_3 = \begin{pmatrix} U_1 & \\ & U_2 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$ for

$$A|_{\Omega_3\times\hat{\Omega}_3} \equiv \begin{pmatrix} A|_{\Omega_1\times\hat{\Omega}_3} \\ A|_{\Omega_2\times\hat{\Omega}_3} \end{pmatrix}.$$

U_3 results from the interaction between Ω_3 and $\hat{\Omega}_3$. To find $\begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$, we need to study the interaction between $\mathbf{I}_1 \cup \mathbf{I}_2$ and $\hat{\Omega}_3$. That is, we select a representative subset $\mathbf{I}_3$ from $\mathbf{I}_1 \cup \mathbf{I}_2$. In Figure 6(a), we can see that some representative points in $\mathbf{I}_1$ and $\mathbf{I}_2$ becomes interior points located within some boxes well separated from $\hat{\Omega}_3$. We just need to further select representative points from these points and keep the other representative points in $\mathbf{I}_1 \cup \mathbf{I}_2$. See Figure 6(b). This representative subset selection produces

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \Pi_3 \begin{pmatrix} I \\ E_3 \end{pmatrix}.$$

When we move to upper levels, similar procedures apply. See Figures 6(c–d) and 7(a–b). This is repeated for all the nodes (except root($\mathbf{T}$)) of the nested bisection tree $\mathbf{T}$ (Figure 4), so as to produce an HSS approximation to A.



(a) Representative subsets within some subdomains at a certain level

(b) Representative subset at an upper level

Figure 7: Representative subsets at upper levels.

Clearly, the number of points within each representative subset $\mathbf{I}_i$ is directly related to the numerical rank of the HSS block $A|_{\Omega_i\times\hat{\Omega}_i}$. Thus, we have the following result which is consistent with the FMM.

Lemma 3.1. *Suppose Ω is an $M\times M$ uniform mesh with the nested bisection tree $\mathbf{T}$ generated as in Remark 3.1. Then for sufficiently large M, the HSS block $A|_{\Omega_i\times\hat{\Omega}_i}$ associated with node i at level $\mathbf{l}$ of $\mathbf{T}$ has ε-rank*

$$\tilde{r}_{\mathbf{l}} = O(M_{\mathbf{l}}\log M), \tag{3.9}$$

where $M_{\mathbf{l}}$ is given in (3.3).

Proof. During the process of finding $\mathbf{I}_i$ as in (3.4), Ω_i can be partitioned into $O(M_{\mathbf{l}})$ subdomains, each α-separated from $\hat{\Omega}_i$. See Figure 5(b) for an illustration. According to Proposition 2.1, with the numerical rank $\tilde{r}_{\mathbf{l}}$ in (3.9), the resulting approximation from the representative subset selection can reach a relative 2-norm approximation accuracy ε for $A|_{\Omega_i\times\hat{\Omega}_i}$. □

3.3 Inner layer structures

In Lemma 3.1, the HSS rank of A is as large as $O(M\log M) = O(\sqrt{N}\log N)$. Then an HSS approximation to A is generally not very effective. For example, it costs around $O(N^{3/2})$ flops to factorize it [36]. To improve the efficiency, we study the inner-layer structures or the structures within the HSS generators from the previous subsection.

We set a switching level $\mathbf{l}_s$ for the nodes of the nested bisection tree $\mathbf{T}$, so that if a node is at a level above $\mathbf{l}_s$, we exploit the inner structures of the HSS generators. Thus, the generators below $\mathbf{l}_s$ are treated as in the regular HSS case. This avoids operating on blocks that are too small, and also ensures that the outer HSS generator sizes are large enough for the asymptotic inner-layer rank estimates to hold. We can establish a two-layer tree $\mathcal{T}$ from $\mathbf{T}$, which has outer-layer nodes from levels 0 to $\mathbf{l}_s$ of $\mathbf{T}$. A node i at

level $\mathbf{l}_s$ of $\mathbf{T}$ is treated as a leaf of $\mathcal{T}$ and D_i is treated as an HSS form generator. The off-diagonal numerical ranks of D_i satisfy Lemma 3.1 and the HSS tree of D_i is simply the subtree of $\mathbf{T}$ associated with i, which is an inner-layer tree. The nonleaf nodes of $\mathcal{T}$ are also associated with inner-layer trees, as shown next.

3.3.1 Structures within the B generators

First, we show the structures within the B_i generators. According to the previous discussions, B_i has the form (see, e.g., (3.7))

$$B_i = A|_{\mathbf{I}_i \times \mathbf{I}_j}, \tag{3.10}$$

where $j = \text{sib}(i)$. Note that B_i may be a rectangular matrix, while the work in [9] sets it to be square and further needs it to be invertible. The work in [9] also uses boundary points as representative points for selecting B_i from A. Here, the representative subsets $\mathbf{I}_i$ and $\mathbf{I}_j$ are more general. See Figure 8(a) for an illustration of $\mathbf{I}_i$ and $\mathbf{I}_j$. To facilitate the study of the interior structures within B_i, we split $\mathbf{I}_i$ and $\mathbf{I}_j$ as:

$$\mathbf{I}_i = \bar{\mathbf{I}}_i \cup (\mathbf{I}_i \backslash \bar{\mathbf{I}}_i), \quad \mathbf{I}_j = \bar{\mathbf{I}}_j \cup (\mathbf{I}_j \backslash \bar{\mathbf{I}}_j), \tag{3.11}$$

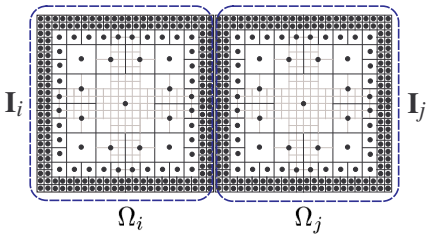
where $\bar{\mathbf{I}}_i$ corresponds to representative points located within those subdomains of Ω_i that have larger sizes farther away from the interface between Ω_i and Ω_j, similarly to those subdomains generated in Figure 3. $\bar{\mathbf{I}}_j$ can be similarly understood. Figure 8(b) illustrates $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$. In (3.11), for convenience, we also suppose $\bar{\mathbf{I}}_i$ is ordered before $\mathbf{I}_i \backslash \bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ is ordered before $\mathbf{I}_j \backslash \bar{\mathbf{I}}_j$. Then B_i in (3.10) can be written as

$$B_i = \begin{pmatrix} A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j} & A|_{\bar{\mathbf{I}}_i \times (\mathbf{I}_j \backslash \bar{\mathbf{I}}_j)} \\ A|_{(\mathbf{I}_i \backslash \bar{\mathbf{I}}_i) \times \bar{\mathbf{I}}_j} & A|_{(\mathbf{I}_i \backslash \bar{\mathbf{I}}_i) \times (\mathbf{I}_j \backslash \bar{\mathbf{I}}_j)} \end{pmatrix}. \tag{3.12}$$

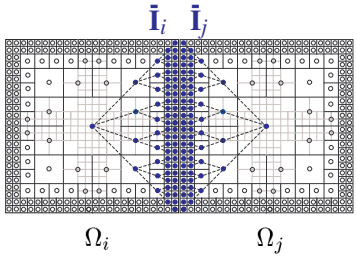
We would like to show that B_i can be approximated by an HSS form when $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ are properly ordered. Similarly to Figure 3, the points within $\bar{\mathbf{I}}_i$ can be organized with the aid of a representative subset tree. A postorder of the nodes in this tree are then applied so as to obtain a proper order of $\bar{\mathbf{I}}_i$ as in Definition 2.4. $\bar{\mathbf{I}}_j$ is similarly ordered. See Figure 8(b). We would like to show $A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$ can be approximated by an HSS form.

Remark 3.2. An HSS approximation to $A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$ needs a consistent strategy to order and partition $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ so as to generate the HSS partition. For example, assume Ω is an $M \times M$ uniform mesh with the nested bisection tree $\mathbf{T}$ generated as in Remark 3.1. Then we suppose $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ are always partitioned simultaneously by straight cuts perpendicular to the interface between Ω_i and Ω_j. See Figure 8(c). In the proper ordering of $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$, we also make sure that any off-diagonal block $A|_{\bar{\mathbf{I}}_{i,1} \times \bar{\mathbf{I}}_{j,2}}$ of $A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$ always corresponds to $\bar{\mathbf{I}}_{i,1}$ and $\bar{\mathbf{I}}_{j,2}$ lying on the opposite sides of one of those straight cuts.

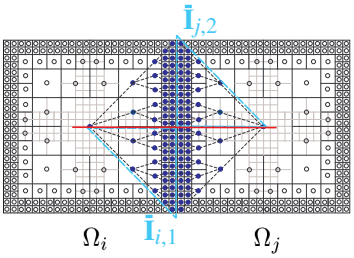
Based on these, we have the following result.



(a) Representative subsets $\mathbf{I}_i$ and $\mathbf{I}_j$



(b) $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ (marked as solid dots)



(c) Partitioning of $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$

Figure 8: Study of the interactions between $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ for siblings i and j of $\mathbf{T}$.

Theorem 3.1. *Suppose ϕ has a finite-term degenerate expansion with respect to a fixed relative tolerance and (2.2)–(2.3) hold. Assume Ω is an $M \times M$ uniform mesh with the nested bisection tree $\mathbf{T}$ generated as in Remark 3.1. Let Ω_i and Ω_j be a pair of sibling subsets at level $\mathbf{l}$ of $\mathbf{T}$ and $\mathbf{I}_i$ and $\mathbf{I}_j$ be representative subsets selected from Ω_i and Ω_j, respectively, like in (3.4). Assume $\mathbf{I}_i$ and $\mathbf{I}_j$ are partitioned and ordered as in (3.11) and $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ are further properly ordered as in Remark 3.2. Then B_i in (3.10) can be approximated by an HSS form with HSS rank $O(\log^3 M_{\mathbf{l}})$ with $M_{\mathbf{l}}$ in (3.3), so that each approximated HSS block of B_i has relative approximation accuracy ε.*

Proof. With (3.11), B_i looks like (3.12). We first show $A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$ can be approximated by an HSS form with HSS rank $O(\log^3 M_{\mathbf{l}})$ so that each approximated HSS block has relative approximation accuracy ε. Following the partitioning in Remark 3.2, suppose the one level of partitioning of $\bar{\mathbf{I}}_i$ and $\bar{\mathbf{I}}_j$ looks like

$$\bar{\mathbf{I}}_i = \bar{\mathbf{I}}_{i,1} \cup \bar{\mathbf{I}}_{i,2}, \quad \bar{\mathbf{I}}_j = \bar{\mathbf{I}}_{j,1} \cup \bar{\mathbf{I}}_{j,2}. \tag{3.13}$$

$A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$ can be then written as

$$A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j} = \begin{pmatrix} A|_{\bar{\mathbf{I}}_{i,1} \times \bar{\mathbf{I}}_{j,1}} & A|_{\bar{\mathbf{I}}_{i,1} \times \bar{\mathbf{I}}_{j,2}} \\ A|_{\bar{\mathbf{I}}_{i,2} \times \bar{\mathbf{I}}_{j,1}} & A|_{\bar{\mathbf{I}}_{i,2} \times \bar{\mathbf{I}}_{j,2}} \end{pmatrix}.$$

It is sufficient to show that the ε-rank of $A|_{\bar{\mathbf{I}}_{i,1} \times \bar{\mathbf{I}}_{j,2}}$ is $O(\log^3 M_{\mathbf{l}})$ and it is similar to study other HSS blocks of $A|_{\bar{\mathbf{I}}_i \times \bar{\mathbf{I}}_j}$.

According to the proof of Lemma 3.1, $|\bar{\mathbf{I}}_i| = O(M_1)$. The points within $\bar{\mathbf{I}}_{i,1}$ are located at $O(\log M_1)$ slices or levels of the representative subset tree associated with $\bar{\mathbf{I}}_i$. Each such slice is logarithmically separated from $\mathbf{I}_{j,2}$. Setting $l = O(\log^2 M_1)$ in Proposition 2.1 yields that $A|_{\bar{\mathbf{I}}_{i,1}\times\bar{\mathbf{I}}_{j,2}}$ has ε-rank $O(\log^3 M_1)$.

We then look at the second block column $A|_{\mathbf{I}_i\times(\mathbf{I}_j\setminus\bar{\mathbf{I}}_j)}$ in (3.12). Similarly, $\mathbf{I}_j\setminus\bar{\mathbf{I}}_j$ includes $O(\log M_1)$ levels or slices, each logarithmically separated from $\mathbf{I}_i$. Thus, Proposition 2.1 means $A|_{\mathbf{I}_i\times(\mathbf{I}_j\setminus\bar{\mathbf{I}}_j)}$ has ε-rank $O(\log^3 M_1)$.

Overall, we can see that B_i in (3.12) can be approximated by an HSS form with HSS rank $O(\log^3 M_1)$ so that each approximated HSS block has relative approximation accuracy ε. □

Thus, the generator B_i has an inner HSS structure. Remark 3.2 essentially also provides a way to generate the HSS tree for B_i. This HSS tree then serves as an inner-layer tree associated with node i of $\mathcal{T}$. Note that essentially we only need to approximate the $(1,1)$ block in (3.12) by an HSS form and then the $(2,2)$ block by a low-rank form. Approximating the entire B matrix by an HSS form is more general and is easier to present with fewer details. In practical implementations, some B generators may involve black-box permutations and it may not be necessary to distinguish such different types of $(1,1)$ and $(2,2)$ blocks.

3.3.2 Structures within the R generators

Next, we show the structures within R_i and R_j with $j = \mathrm{sib}(i)$. Let $p = \mathrm{par}(i)$. The parent domain is $\Omega_p = \Omega_i \cup \Omega_j$. Due to the lower level compression, the compression of $A|_{\Omega_p\times\hat{\Omega}_p}$ reduces to the compression of $A|_{(\mathbf{I}_i\cup\mathbf{I}_j)\times\hat{\Omega}_p}$. The SPRR factorization leads to

$$A|_{(\mathbf{I}_i\cup\mathbf{I}_j)\times\hat{\Omega}_p} \approx \begin{pmatrix} R_i \\ R_j \end{pmatrix} A|_{\mathbf{I}_p\times\hat{\Omega}_p}, \quad \text{with} \quad \begin{pmatrix} R_i \\ R_j \end{pmatrix} = \Pi_p \begin{pmatrix} I \\ E_p \end{pmatrix} \begin{matrix} \mathbf{I}_p \\ \hat{\mathbf{I}}_p \end{matrix}, \tag{3.14}$$

where the identity matrix corresponds to the representative subset $\mathbf{I}_p$ from $\mathbf{I}_i \cup \mathbf{I}_j$, and E_p corresponds to the residual subset $\hat{\mathbf{I}}_p = (\mathbf{I}_i \cup \mathbf{I}_j)\setminus\mathbf{I}_p$.

Like in Figure 6, some boxes near the boundary of Ω_i and Ω_j become well separated from $\hat{\Omega}_p$. For convenience, we use $\tilde{\mathbf{I}}_p$ to denote the subset of representative points in $\mathbf{I}_i \cup \mathbf{I}_j$ that are inside those boxes, and call $\tilde{\mathbf{I}}_p$ the *compressible subset*. See Figure 9(a). Also, denote the representative subset of $\tilde{\mathbf{I}}_p$ with respect to $\hat{\Omega}_p$ by

$$\check{\mathbf{I}}_p = \mathrm{RS}(\tilde{\mathbf{I}}_p | \hat{\Omega}_p), \tag{3.15}$$

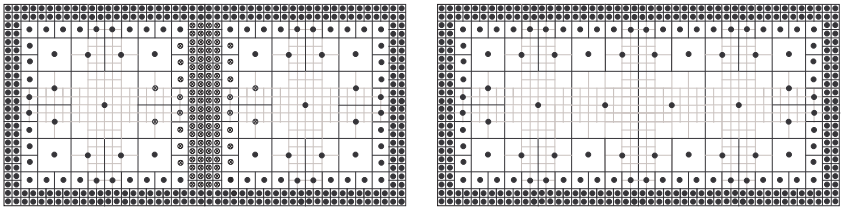
so that

$$\mathbf{I}_p = ((\mathbf{I}_i \cup \mathbf{I}_j)\setminus\tilde{\mathbf{I}}_p) \cup \check{\mathbf{I}}_p. \tag{3.16}$$

Then we have

$$\hat{\mathbf{I}}_p = (\mathbf{I}_i \cup \mathbf{I}_j)\setminus\mathbf{I}_p = \tilde{\mathbf{I}}_p \setminus \check{\mathbf{I}}_p.$$

That is, to obtain $\mathbf{I}_p$ from $\mathbf{I}_i \cup \mathbf{I}_j$, we replace $\tilde{\mathbf{I}}_p$ by its representative set $\check{\mathbf{I}}_p$. The points that we drop from $\tilde{\mathbf{I}}_p$ form the residual subset $\hat{\mathbf{I}}_p$.



(a) Compressible subset $\tilde{\mathbf{I}}_p$ (marked as $\otimes$) in $\mathbf{I}_i \cup \mathbf{I}_j$

(b) $\mathbf{I}_p$ (with $\tilde{\mathbf{I}}_p$ replaced by $\check{\mathbf{I}}_p$)

Figure 9: Forming the representative subset $\mathbf{I_p}$ from $\mathbf{I_i} \cup \mathbf{I_j}$ by replacing the compressible subset $\tilde{\mathbf{I}}_\mathbf{p}$ by its representative subset $\check{\mathbf{I}}_\mathbf{p}$ with respect to $\hat{\Omega}_p$.

We then study the structure of E_p in (3.14). Suppose $m = |\mathbf{I}_p|$. The SPRR factorization finds an $m \times m$ invertible submatrix of $A|_{(\mathbf{I}_i \cup \mathbf{I}_j) \times \hat{\Omega}_p}$, denoted $A|_{\mathbf{I}_p \times \mathbf{J}_p}$ ($\mathbf{J}_p \subset \hat{\Omega}_p$), whose determinant is sufficiently large. That is, the selection of the representative subset $\mathbf{I}_p$ returns a numerical column basis matrix for $A|_{(\mathbf{I}_i \cup \mathbf{I}_j) \times \hat{\Omega}_p}$:

$$\Pi_p \begin{pmatrix} I \\ E_p \end{pmatrix} A|_{\mathbf{I}_p \times \mathbf{J}_p} = \Pi_p \begin{pmatrix} A|_{\mathbf{I}_p \times \mathbf{J}_p} \\ E_p A|_{\mathbf{I}_p \times \mathbf{J}_p} \end{pmatrix}, \tag{3.17}$$

which is also a submatrix of $A|_{(\mathbf{I}_i \cup \mathbf{I}_j) \times \hat{\Omega}_p}$. As in (2.11),

$$E_p = A|_{\hat{\mathbf{I}}_p \times \mathbf{J}_p} (A|_{\mathbf{I}_p \times \mathbf{J}_p})^{-1}.$$

We show $A|_{\hat{\mathbf{I}}_p \times \mathbf{J}_p}$ is numerically low rank so that E_p can be approximated by a low-rank form.

Theorem 3.2. *Suppose the conditions in Theorem 3.1 holds. Let $m = |\mathbf{I}_p|$ and $A|_{\mathbf{I}_p \times \mathbf{J}_p}$ be the $m \times m$ invertible submatrix of $A|_{(\mathbf{I}_i \cup \mathbf{I}_j) \times \hat{\Omega}_p}$ with the largest determinant among all of its $m \times m$ submatrices. Then $A|_{\hat{\mathbf{I}}_p \times \mathbf{J}_p}$ has ε-rank $O(\log^3 m)$.*

Proof. We show this with the aid of the compressible subset $\tilde{\mathbf{I}}_p$ as illustrated in Figure 9(a). Notice $\tilde{\mathbf{I}}_p \subset \Omega_p$ and $\mathbf{J}_p \subset \hat{\Omega}_p$. Just like in Figure 3, using a representative subset tree, we can organize the points in $\tilde{\mathbf{I}}_p$ into $O(\log|\tilde{\mathbf{I}}_p|)$ slices, each logarithmically separated from $\hat{\Omega}_p$ and also $\mathbf{J}_p$. Setting $l = O(\log^2 m)$ in Proposition 2.1 yields that $A|_{\tilde{\mathbf{I}}_p \times \mathbf{J}_p}$ has ε-rank $O(\log^3 m)$. Accordingly, $A|_{\hat{\mathbf{I}}_p \times \mathbf{J}_p}$ has ε-rank $O(\log^3 m)$ since it is a submatrix of $A|_{\tilde{\mathbf{I}}_p \times \mathbf{J}_p}$. □

Theorems 3.1 and 3.2 indicate that, if Ω is an $M \times M$ uniform mesh, then A in (3.2) can be approximated by an HSS form with structured generators. The results can be

extended to more general point sets with similar strategies. We have only shown the analysis for uniform meshes since it is easier to rigorously characterize the conditions and conclusions.

In addition, although the ε-rank $O(\log^3 m)$ may not look so small, its impact on the overall cost and storage is not significant due to a rank pattern study in [36] since m is level dependent. It is also possible to further reduce the bound to $O(\log^2 m)$ if we take advantage of certain common basis matrices. This is not critical and we use a conservative bound $O(\log^3 m)$ to avoid some technical details.

3.4 MHS representation

To systematically take advantage of the inner-layer structures within the HSS generators, we define the MHS representation as follows. Here for generality purpose, the representation is defined for a general nonsymmetric form, although most our other discussions are for symmetric forms just for convenience.

Definition 3.1. A multi-layer *hierarchically semiseparable (MHS) matrix* is an HSS matrix whose generators are further HSS, MHS, or low-rank matrices. In particular, a two-layer MHS matrix $\mathcal{A}$ with a corresponding *MHS tree* $\mathcal{T}$ is recursively defined as follows. $\mathcal{T}$ includes two layers of postordered binary trees. The outer-layer tree has nodes $i=1,2,\ldots,k$, where k is the root. Each node i is associated with HSS generators $D_i, U_i, V_i, R_i, W_i, B_i$. Furthermore, the generators are structured as described below.

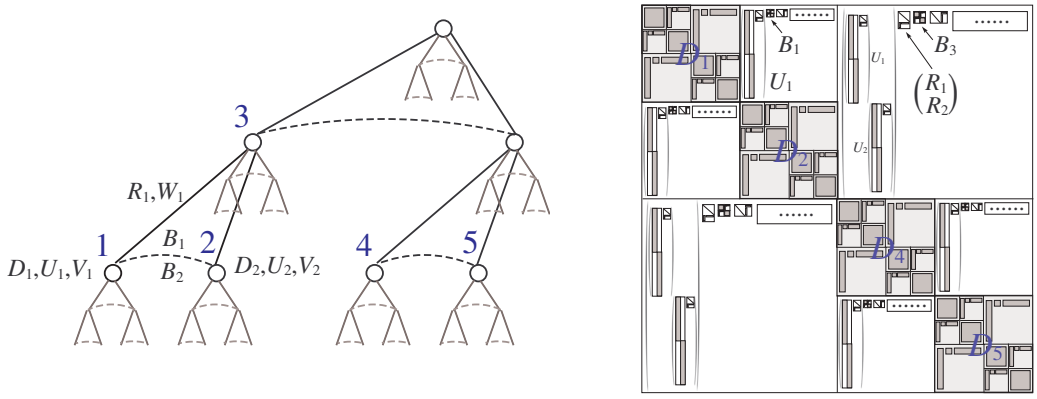
1. All the generators D_i associated with the leaves i of $\mathcal{T}$ are in HSS forms.
2. All the B_i generators associated with the nodes $i \neq k$ of $\mathcal{T}$ are in (rectangular) HSS forms.
3. All the R, W generators used to construct the generators U_i, V_i associated with nonleaf nodes $i \neq k$ of $\mathcal{T}$ as in (3.1) are in the forms of $\begin{pmatrix} R_{c_1} \\ R_{c_2} \end{pmatrix} = \Pi_i \begin{pmatrix} I \\ E_i \end{pmatrix}$ and $\begin{pmatrix} W_{c_1} \\ W_{c_2} \end{pmatrix} = \Theta_i \begin{pmatrix} I \\ F_i \end{pmatrix}$, respectively, where c_1 and c_2 are the children of i, Π_i and Θ_i are permutation matrices, and E_i and F_i are low-rank matrices.

Each node i of $\mathcal{T}$ is associated with an inner-layer HSS tree for the D_i and/or B_i generators. All the inner-layer HSS generators and the low-rank forms of E_i and F_i are called the MHS generators. The outer HSS rank is the maximum of the sizes from the smaller dimension of each B_i generator of the outer HSS form. The *MHS rank* of $\mathcal{A}$ is the maximum of the HSS ranks of all the leaf level D_i generators, the HSS ranks of all the B_i generators, and the ranks of all E_i, F_i.

Thus, a two-layer MHS structure is an outer-layer HSS structure with an extra inner layer of HSS or low-rank structures. This is also called an HSS2D structure in our earlier

report [39] and is similar to (but more general than) the 2D HSS form in [9]. See Figure 10 for an illustration. Here by an MHS structure, we usual mean the two-layer one. Similar considerations can be given for higher dimensions via the use of multiple layers. The basic idea is to "lift" low-rank forms to HSS forms, HSS forms to two-layer MHS forms, two-layer MHS forms to three-layer MHS forms, etc. The process would be much more technical than the two-layer case. The two-layer MHS structure is sufficient for practical computations such as the approximation of dense Schur complements in the direct factorizations of 3D discretized elliptic PDEs.

For notational consistency, suppose the MHS tree has $\mathbf{l}_s$ outer levels, with the root at level 0. When we say a node of $\mathcal{T}$ is at level $\mathbf{l}$, we mean the outer level $\mathbf{l}$.



(a) MHS tree (b) MHS structure

Figure 10: An MHS matrix corresponding to a two-layer MHS tree, where some outer-layer HSS generators are marked and the inner layer trees in (a) are for the structured outer D,B generators.

Remark 3.3. We make some remarks on certain practical issues about the generators in the definition.

1. Permutations may also be involved in the leaf level D_i generators and the B_i generators in order for them to have HSS forms. This is to accommodate possible reordering of the corresponding representative points. The permutations do not interfere with the off-diagonal rank structures, since the off-diagonal basis matrices U,V also involve permutations.

2. The B_i generators may be non-square matrices, which is different from the strategy in [9] where B_i needs to be not only square but also invertible.

3. For a leaf i, there is no restriction on the structure of the U_i,V_i generators, which are formed based on the generators associated with the inner HSS generators of D_i.

4. The outer HSS rank of $\mathcal{A}$ is just a way to measure the HSS rank of $\mathcal{A}$ when it is considered as a usual HSS form.

5. The number of outer levels $\mathbf{l}_s$ can be adjusted so as to roughly optimize the performance for different situations. An example is shown below.

The storage of an MHS form $\mathcal{A}$ can be counted as follows. Let N be its size, r be its MHS rank, and $\tilde{r}$ be its outer HSS rank and also the size of all outer B generators. Choose $\mathbf{l}_s$ so that the D_i generator for any outer-layer leaf i has size $O(\tilde{r})$. With these uniform rank bounds, each HSS form D_i or B_i generator needs storage $O(r\tilde{r})$. The storage is similar for each structured R_i or W_i generator. The outer HSS tree has $O(\frac{N}{\tilde{r}})$ nodes. Thus, the total storage is

$$\sigma = O(r\tilde{r}\cdot\frac{N}{\tilde{r}}) = O(rN).$$

In particular, when $\mathcal{A}$ is used to approximate A in (3.2) with Ω an $M\times M$ uniform mesh, we can choose $\mathbf{l}_s$ so that the D_i generators for the outer HSS form has size

$$N_s \equiv O(\frac{N}{2^{\mathbf{l}_s}}) = O(N^{1/2}). \tag{3.18}$$

In this way, following a rank pattern study in [36], the storage for each D_i generator is $O(N_s\log N_s) = O(N^{1/2}\log N)$. Based on Theorems 3.1 and 3.2, the storage for each B,R,W generator is $O\left(\left(\frac{N}{2^{\mathbf{l}}}\right)^{1/2}\log^4\frac{N}{2^{\mathbf{l}}}\right)$. Thus, the total storage is

$$\begin{aligned}\sigma &= 2^{\mathbf{l}_s}O(N^{1/2}\log N) + \sum_{\mathbf{l}=1}^{\mathbf{l}_s} 2^{\mathbf{l}}O\left(\left(\frac{N}{2^{\mathbf{l}}}\right)^{1/2}\log^4\frac{N}{2^{\mathbf{l}}}\right)\\ &= O(N\log N) + O(N^{1/2}2^{\mathbf{l}_s/2}\log^4 N) = O(N\log N),\end{aligned}$$

where (3.18) is used. In comparison, the (outer) HSS form needs storage

$$\sigma = \sum_{\mathbf{l}=1}^{O(\log N)} 2^{\mathbf{l}}O\left(\frac{N}{2^{\mathbf{l}}}\log^2 N\right) = O(N\log^3 N).$$

In practice, when the HSS and MHS structured forms are used for some matrix operations, the difference in the costs is even more significant than the difference in the storage, which is typically the case for structures with different numbers of hierarchical layers [10, 40, 41].

4 Design of MHS algorithms

Due to the multi-layer structure, it is convenient to reuse some basic ideas and algorithms in HSS methods to design MHS algorithms such as construction, factorization, solution,

and multiplication. Since the focus of this work is the design of the MHS structure, we just briefly mention the design of some algorithms, mainly the construction of an MHS approximation $\mathcal{A}$ to the matrix A in (3.2) and some related practical issues. The actual implementations will be left to other work.

In MHS constructions, we can first construct the outer-layer representation following the derivation procedure in Section 3 and then explore the inner-layer structures. The method can be based on either analytical or algebraic approaches.

An analytical MHS construction can be designed if a kernel expansion in (2.2) is known. We can construct the outer HSS form following the procedure in [7], where relevant off-diagonal basis matrices are analytically constructed based on the kernel expansion. We may also use an idea of analytical compression based on the so-called proxy point method [9, 25, 46, 47].

More specifically, for a leaf i of $\mathcal{T}$, find an initial numerical column basis $\tilde{U}_i$ for $A|_{\Omega_i \times \hat{\Omega}_i}$ based on the kernel expansion. $\tilde{U}_i$ is then converted into the generator U_i in a form like in (2.11) via an SPRR factorization. This enables to identify the representative subset $\mathbf{I}_i$. For a nonleaf node p with children i and j, find a numerical column basis $\tilde{U}_p$ for $\begin{pmatrix} A|_{\mathbf{I}_i \times \hat{\Omega}_p} \\ A|_{\mathbf{I}_j \times \hat{\Omega}_p} \end{pmatrix}$. As mentioned in Section 3.3.2, this only needs to be done on $A|_{\check{\mathbf{I}}_p \times \hat{\Omega}_p}$. The representative subset selection (3.15) yields $\check{\mathbf{I}}_p$ and then $\mathbf{I}_p$ in (3.16). The representative subset selection also gives E_p in (3.14).

Next, we find the inner structures. For the D_i generators associated with a leaf i, the inner-layer structures are directly from the outer HSS construction. For the generators $B_i \equiv A|_{\mathbf{I}_i \times \mathbf{I}_j}$ associated with a node i, the kernel expansion can be used to find an HSS approximation to B_i with the method in [7]. For the R generators in (3.14), a low-rank approximations to E_i may be computed quickly if the sets like $\mathbf{J}_p$ in (3.17) can be quickly identified. Otherwise, we may use direct compression.

In the analytical construction, the main costs include the following.

- The cost to construct the outer HSS form via the algorithm in [7] is $\sum_{\mathbf{l}=1}^{O(\log N)} O(2^{\mathbf{l}}\tilde{r}^2) = O(\tilde{r}N)$, which is $O(N^{1.5}\log N)$ based on Lemma 3.1 if Ω is an $M \times M$ uniform mesh. However, this cost is mainly for evaluating the entries of the generators which are either from A (for B_i in (3.10)) or have some parameterized forms (since the numerical basis matrices $\tilde{U}_i$ are scaled Vandermonde matrices [7]). Thus, this cost can be absorbed into the later costs for constructing inner-layer structures when the structured outer generators are used.

- The cost to compute SPRR factorizations for all the basis matrices $\tilde{U}_i$ is $\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}}\tilde{r}^3) = O(\tilde{r}^3 2^{\mathbf{l}_s})$. If Ω is an $M \times M$ uniform mesh, Lemma 3.1 can be used to get the cost of $O(N^2 \log^3 N)$ due to (3.18). However, this may be further reduced due to the structures in $\tilde{U}_i$. If we take advantage of the scaled Vandermonde form of $\tilde{U}_i$, a low-rank approximation may be quickly computed based on methods in [13, 18, 26, 27, 43].

This can reduce to cost to $\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}}\tilde{r}^2) = O(\tilde{r}^2 2^{\mathbf{l}_s})$, which becomes $O(N^{1.5}\log^2 N)$. Furthermore, Lemma 3.1 actually indicates $\tilde{r}$ is level dependent. Based on (3.9), this cost should actually be

$$\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}}\tilde{r}_{\mathbf{l}}^2) = \sum_{\mathbf{l}=1}^{\mathbf{l}_s} O\left(2^{\mathbf{l}}\left(M/2^{\lfloor \mathbf{l}/2 \rfloor}\right)^2 \log^2 M\right) = O(M^2\log^3 M) = O(N\log^3 N).$$

- The cost to find inner-layer structures for the E_i matrices is $\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}} r^2 \tilde{r})$, or if the scaled Vandermonde structured of $\tilde{U}_i$ is considered, $\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}} r\tilde{r})$. This cost is then just a low-order term as compared with the cost in the previous item.

- The cost to find inner structures for the B_i generators is $\sum_{\mathbf{l}=1}^{\mathbf{l}_s} O(2^{\mathbf{l}} r\tilde{r})$, which is also a low-order term as compared with the cost in the second item.

Overall, when we fully take advantage of the structures, the complexity of the MHS construction can be reduced to nearly $O(N)$, although the actual algorithm implementation needs to take careful of many technical details.

Algebraic MHS construction strategies can also be designed. Explicit HSS constructions in [42] may be used and are expensive. However, it can serve as a black-box construction method. A faster way is to use a randomized HSS construction in [24, 45] together with the FMM for matrix-vector multiplications.

It is still an open problem to construct MHS approximations to problems with small MHS ranks in nearly $O(N)$ complexity using only algebraic techniques. Note that randomized HSS construction for problems with small HSS ranks and fast matrix-vector multiplications can already reach nearly linear complexity [22–24, 45]. This may provide some hints for linear complexity randomized MHS constructions.

Remark 4.1. There are some practical issues to pay attention to. Since Π_p results from a representative subset selection from $\mathbf{I}_i \cup \mathbf{I}_j$, it only separates $\mathbf{I}_i \cup \mathbf{I}_j$ into sets $\mathbf{I}_p$ and $\hat{\mathbf{I}}_p$, but does not guarantee that $\mathbf{I}_p$ is properly ordered as needed for the HSS approximation of $A|_{(\mathbf{I}_p \cup \hat{\mathbf{I}}_p)\times(\mathbf{I}_p \cup \hat{\mathbf{I}}_p)}$. Thus, we also reorder the set $\mathbf{I}_p$ based on either graph reordering methods or the geometric connectivity in the mesh so as to reveal the HSS structure. For convenience, we assume Π_p also includes such internal reordering. In practical implementations, for simplicity, the proper ordering for $\mathbf{I}_p$ and other sets may be based on the reverse Cuthill-McKee (RCM) method. Note that $\mathbf{I}_i \cup \mathbf{I}_j$ is also reordered accordingly. Since B_i is given by $A|_{\mathbf{I}_i \times \mathbf{I}_j}$ as in (3.10), it is more convenient to approximate $\Pi_p^T \begin{pmatrix} & B_i \\ B_i^T & \end{pmatrix} \Pi_p$ by a square HSS form. This will be useful when we consider the MHS factorization in [40].

Other MHS algorithms can also be designed. For example, A multi-layer MHS factorization procedure can be designed based on repeated elimination of representative points like the methods in [21,38]. The factorization at each layer follows the frameworks in HSS

ULV factorizations and even the multifrontal methods. The entire factorization procedure involves many steps and the details will be given in [40].

It is clear that we can take advantage of existing HSS methods in designing MHS algorithms. Furthermore, the multi-layer tree structure makes it convenient to analyze the resulting MHS algorithms. For example, the backward stability analysis for some HSS methods in [33, 34] relies on the idea that the numerical errors only propagate $O(\log N)$ times along HSS trees. Here with MHS trees, this is still the situation, so we should expect to still see nice stability behaviors for MHS methods. In fact, by replacing operations on some large dense generators by structured ones, the stability is likely to get even better. This will be investigated in detail in future work.

5 Numerical experiments

To verify the existence and effectiveness of MHS structures, we consider some discretized matrices A as in (3.2). Since the main purpose of this work is to confirm the feasibility of MHS approximations, we consider kernel functions ϕ in (3.2) that are known to be suitable for FMM methods. This paves the way for studying more sophisticated functions in the future. Once the MHS structure is verified, it will be feasible to solve relevant linear systems by direct solvers instead of iterative ones.

We report some rank bounds, storage, and representative subset selection to show the feasibility of MHS approximations. The following measurements are used.

- $\tilde{r}$: numerical measurement of outer HSS rank when A is approximated by an HSS form.
- r: numerical measurement of MHS rank.
- σ: storage for the structured matrix approximation in terms of the number of nonzeros in the generators.

The rank measurements $\tilde{r}$ and r are decided based on appropriate generator sizes of the resulting structured forms. For example, if E is approximated by a compressed form GH^T, the column size of G is used as the rank measurement for E. To measure $\tilde{r}$, we find the smaller of the row and column sizes of each outer B generator and then pick the maximum of all these sizes. To measure r, we are more conservative. We find the row and column sizes of all the inner-layer B generators and the rank measurements for all the E generators and then take the maximum.

Example 5.1. First, consider ϕ to be the 2D Laplace free-space Green's function and

$$A_{ij}=\begin{cases} 1, & i=j, \\ \frac{h^2}{2\pi}\log|y_i-y_j|, & i\neq j, \end{cases}$$

where y_j's are points on a uniform grid in the domain $[-1,1]\times[-1,1]$ with M points in each direction.

We inspect whether A can be approximated by compact MHS forms for a given tolerance. More specifically, we look at the inner structures within an outer HSS approximation to A, as discussed in Section 3.3. Accordingly, to show the advantage of MHS structures over HSS structures, we also report the results when A is directly approximated by an HSS form (which is just the outer-layer HSS form of the MHS approximation). Relevant partitions and ordering follow Remarks 3.1 and 3.2. To get an initial numerical column basis $\tilde{U}_i$ for $A|_{\Omega_i \times \hat{\Omega}_i}$, a proxy point method is used with sufficient accuracy. However, in all rank-revealing factorization steps, a relative tolerance $\tau = 10^{-6}$ is used. To simplify implementations, the inner-layer HSS approximations to the D,B generators are obtained with the HSS construction in [42] using the tolerance τ. The matrix size $N = M^2$ ranges from 128^2 to 2048^2. The number of outer HSS levels $\mathbf{l}_s$ varies accordingly.

As shown in Table 1, when the mesh dimension M doubles, the HSS rank $\tilde{r}$ roughly doubles, which is consistent with Lemma 3.1. On the other hand, the MHS rank r remains about the same. For $N = 2048^2$, $\tilde{r}$ is almost 40 times as large as r. This shows the feasibility and effectiveness of inner rank structures within the outer generators which are otherwise considered dense in regular HSS methods.

Table 1: Example 5.1. Rank measurements and storage of MHS approximations as compared with those of HSS approximations.

N		128^2	256^2	512^2	1024^2	2048^2
$\mathbf{l}_s$		6	8	10	12	14
Rank measurement	$\tilde{r}$ (HSS)	247	472	898	1710	3302
	r (MHS)	80	80	80	83	84
Storage σ	HSS	5.60e06	2.74e07	1.30e08	5.76e08	2.53e09
	MHS	6.35e06	2.69e07	1.12e08	4.45e08	1.78e09

The storage for the MHS and HSS approximations is also given Table 1 and is further plotted in Figure 11. It indicates that the MHS forms have nearly $O(N)$ storage. In addition, if the resulting MHS approximations are used for linear system solutions based on the algorithm in [40], we can observe solution accuracies around $O(10^{-6})$. Since our focus is not on the algorithms, we do not report the actual solution performance.

Remark 5.1. The comparison in the storage of MHS and HSS approximations is not as significant as the rank comparison due to multiple reasons. First, the HSS/MHS rank measurements are for the worst case generator sizes. Some outer B generators have sizes much smaller than $\tilde{r}$ so that the inner structures do not significantly reduce the storage. (However, note that the cost of factorizations is often dominated by the cost for few largest matrices, which is why we expect the difference in the factorization cost to be much larger.) Nevertheless, the storage difference is expected to be more significant for larger N. Next, as mentioned in Remark 4.1, in the MHS approximations, we store the HSS approximation of $\Pi_p^T \begin{pmatrix} & B_i \\ B_i^T & \end{pmatrix} \Pi_p$ which has some zeros in the generators. For

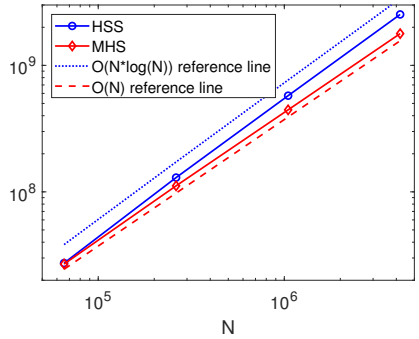


Figure 11: Example 5.1. Storage of MHS approximations as compared with HSS ones.

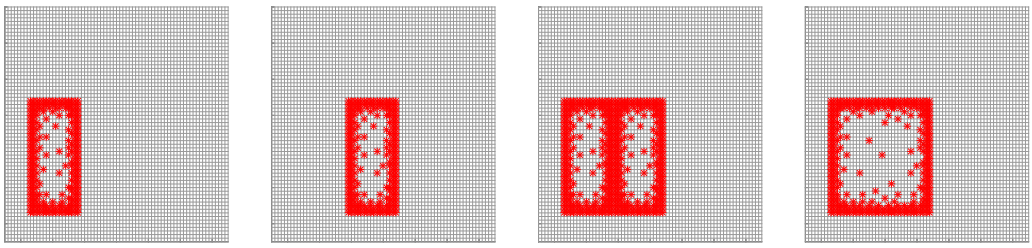
generality, these zeros are treated as dense in our storage count for the MHS form. This overestimates the MHS storage a little bit.

Example 5.2. Then we consider ϕ as the 3D Laplace free-space Green's function, and the matrix A is given by

$$A_{ij} = \begin{cases} 1, & i = j, \\ -\frac{h^2}{4\pi}\frac{1}{|y_i - y_j|}, & i \neq j. \end{cases}$$

The other setups are the same as in the previous example.

In this case, unlike the previous example where representative points cluster near the boundaries of a domain like in [9], the SPRR factorization yields representative points that may also be away from the boundaries. This can be observed from Figure 12, which illustrates representative subsets $\mathbf{I}_i$ and $\mathbf{I}_j$ corresponding to two sibling subdomains Ω_i and Ω_j in one mesh, respectively. The figure also shows the representative subset $\mathbf{I}_p$ selected from $\mathbf{I}_i \cup \mathbf{I}_j$.

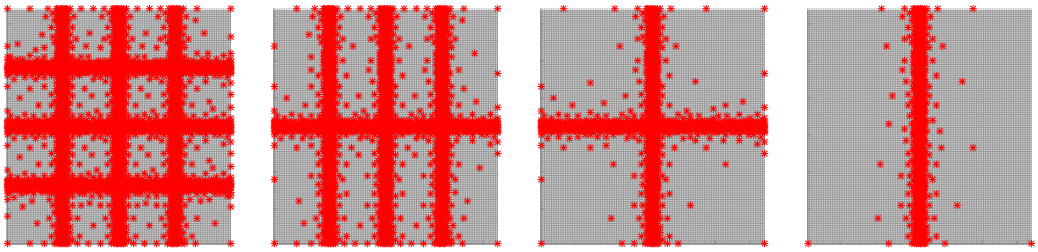


(a) $\mathbf{I}_i$ (b) $\mathbf{I}_j$ (c) $\mathbf{I}_i \cup \mathbf{I}_j$ (d) $\mathbf{I}_p$

Figure 12: Example 5.2. Some examples of representative subsets in a mesh (zoomed in), and the selection of $\mathbf{I}_p$ from lower level representative subsets $\mathbf{I}_i$ and $\mathbf{I}_j$, where $j = \mathrm{sib}(i)$, $p = \mathrm{par}(i)$.

For one mesh, Figure 13 plots the collections of representative subsets for all the nodes at some outer levels of the MHS tree. They correspond to collections of so-called reduced

matrices in the ULV-type factorization in [38] and can also be considered as a generalization of the skeletons in [21]. This is consistent with Figures 6 and 7 and clearly shows how the residual subsets $\hat{\mathbf{I}}_i$ at each level are eliminated during the formation of the upper level representative subsets. The figure can also be viewed as a sparsification of the mesh. This will be useful for understanding the MHS factorization process in [40].



(a) $\mathbf{l}=3$ (b) $\mathbf{l}=2$ (c) $\mathbf{l}=1$ (d) $\mathbf{l}=0$

Figure 13: Example 5.2. Collections of representative subsets at some outer levels $\mathbf{l}$ of the MHS tree, where the mesh is 128×128.

The rank structures are reported in Table 2. Although the MHS ranks in this case are higher than in the previous example, they are still much smaller than the outer HSS ranks. Note that the ordering of the representative points impacts the measured MHS ranks, as mentioned in Remark 4.1. We expect to be able to further reduce the MHS rank measurements with improved ordering strategies. On the other hand, the rank structure in Theorem 3.2 is less dependent on the ordering. Thus, we also report the maximum of the measured numerical ranks of the E_i matrices for the nodes i of $\mathcal{T}$, denoted r_E. This bound is a more precise measurement of the intrinsic structures within the off-diagonal basis generators. r_E is quite smaller than r in Table 2.

Table 2: Example 5.2. Rank measurements and storage of MHS approximations as compared with those of HSS approximations.

N		128^2	256^2	512^2	1024^2	2048^2
$\mathbf{l}_s$		6	8	10	12	14
Rank measurement	$\tilde{r}$ (HSS)	377	743	1471	2932	5893
	r (MHS)	99	136	201	384	510
	r_E (MHS)	99	114	131	139	152
Storage σ	HSS	8.50e06	4.70e07	2.45e08	1.22e09	5.95e09
	MHS	1.06e07	4.84e07	2.09e08	8.74e08	3.58e09

The storage for the MHS and HSS approximations is also given Table 2 and is further plotted in Figure 14. The difference in the storage is more significant than in the previous example. We expect the difference to be even larger for bigger matrix sizes.

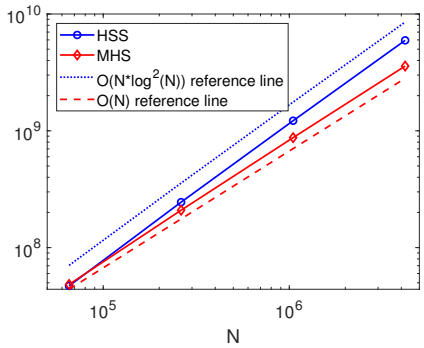


Figure 14: Example 5.2. Storage of MHS approximations as compared with HSS ones.

6 Conclusions

In this work, we have given theoretical foundations for the design of the MHS structure. The MHS structure extends the HSS structure to multiple dimensions by recursively incorporating HSS and low-rank structures into the generators of outer-layer HSS forms. Based on FMM and algebraic methods for selecting representative points, we have shown the existence of MHS structures within the approximation of some multi-dimensional discretized dense matrices. The multi-layer design makes it convenient to explore multi-dimensional FMM structures by taking advantage of existing HSS algorithms and analysis. In particular, it facilitates the design of fast and stable multi-layer hierarchical factorizations.

The work provides a proof-of-concept study for multi-layer hierarchical structures. The MHS structure in two dimensions can be used approximate some 2D discretized integral equations or dense Schur complements in some 3D discretized PDEs, which can lead to direct solvers with nearly linear complexity. Further optimization of the ordering strategies for the representative points is expected to be done. We also expect to inspect more practical discretized problems. Efficient algorithms and implementations will be developed for the purpose of large-scale dense and sparse direct solutions for multi-dimensional problems. Some developments will be included in [40].

Acknowledgements

The research of Jianlin Xia was supported in part by an NSF CAREER Award DMS-1255416 and NSF grant DMS-1819166.

References

[1] A. AMINFAR, S. AMBIKASARAN, E. DARVE, *A fast block low-rank dense solver with applications to finite-element matrices*, J. Comp. Phys., 304 (2016), pp. 170–188.

[2] R. BEATSON AND L. GREENGARD, *A Short Course on Fast Multipole Methods*, In M. Ainsworth, & al (Eds.), Wavelets, multilevel methods, and elliptic PDEs, pp. 1–37, Numerical Mathematics and Scientific Computation, Oxford University Press.
[3] M. BEBENDORF AND R. VENN, *Constructing nested bases approximations from the entries of non-local operators*, Numer. Math., 121 (2012), pp. 609–635.
[4] S. BÖRM, L. GRASEDYCK, AND W. HACKBUSCH, *Introduction to hierarchical matrices with applications*, Eng. Anal. Bound. Elem., 27 (2003), pp. 405–422.
[5] S. BÖRM AND W. HACKBUSCH, *Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices*, Computing, 69 (2002), pp. 1–35.
[6] S. BÖRM AND K. REIMER, *Efficient arithmetic operations for rank-structured matrices based on hierarchical low-rank updates*, Comput. Visual Sci. 16 (2013, pp. 247–258.
[7] D. CAI AND J. XIA, *A stable matrix version of the fast multipole method*, SIAM J. Matrix Anal. Appl., submitted, (2020).
[8] S. CHANDRASEKARAN, P. DEWILDE, M. GU, AND T. PALS, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 603–622.
[9] E. CORONA, P. G. MARTINSSON, AND D. ZORIN, *An $O(N)$ direct solver for integral equations in the plane*, Appl. Comput. Harmon. Anal. 38 (2015), pp. 284–317.
[10] I. S. DUFF AND J. K. REID, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Bans. Math. Software, 9 (1983), pp. 302–325.
[11] J. A. GEORGE, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
[12] A. GILLMAN, P. YOUNG, AND P. G. MARTINSSON, *A direct solver with $O(N)$ complexity for integral equations on one-dimensional domains*, Front. Math. China, 7 (2012), pp. 217–247.
[13] I. GOHBERG, T. KAILATH, AND V. OLSHEVSKY, *Fast Gaussian elimination with partial pivotingnfor matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.
[14] S. A. GOREINOV AND E. E. TYRTYSHNIKOV, *The maximal-volume concept in approximation by low-rank matrices*, in Contemporary Mathematics, vol 280, 2001, pp. 47–52.
[15] L. GRASEDYCK, R. KRIEMANN, AND S. LE BORNE, *Domain decomposition based $\mathcal{H}$-LU preconditioning*, Numer. Math., 112 (2009), pp. 565–600.
[16] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comp. Phys., 73 (1987), pp. 325–348.
[17] M. GU AND S. C. EISENSTAT, *Efficient algorithms for computing a strong-rank revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
[18] M. GU AND J. XIA, *A numerically stable superfast Toeplitz solver*, Berkeley Matrix Computations and Scientific Computing Seminar, http://math.berkeley.edu/~mgu /Seminar/Fall2009/ToeplitzSeminarTalk.pdf, (2009).
[19] W. HACKBUSCH, B. N. KHOROMSKIJ, AND R. KRIEMANN, *Hierarchical matrices based on a weak admissibility criterion*, Computing, 73 (2004), pp 207–243.
[20] N. HALKO, P. G. MARTINSSON, AND J. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, 53 (2011), pp. 217–288.
[21] K. L. HO AND L. YING, *Hierarchical interpolative factorization for elliptic operators: integral equations*, Comm. Pure Appl. Math., 69 (2016), pp. 1314–1353.
[22] L. LIN, J. LU, AND L. YING, *Fast construction of hierarchical matrix representation from matrix-vector multiplication*, J. Comput. Phys., 230 (2011), pp. 4071–4087.
[23] X. LIU, J. XIA, AND M. V. DE HOOP, *Parallel randomized and matrix-free direct solvers for large*

structured dense linear systems, SIAM J. Sci. Comput., 38 (2016), pp. S508–S538.
[24] P. G. MARTINSSON, *A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix*, SIAM. J. Matrix Anal. Appl., 32 (2011), pp. 1251–1274.
[25] P. G. MARTINSSON AND V. ROKHLIN, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.
[26] V. PAN, *Fast Approximate Computations with Cauchy Matrices, Polynomials and Rational Functions*, In: Hirsch E.A., Kuznetsov S.O., Pin JÉ, Vereshchagin N.K. (eds) Computer Science - Theory and Applications. CSR 2014. Lecture Notes in Computer Science, vol 8476. Springer, Cham.
[27] V. PAN, *Transformations of matrix structures work again*, Linear Algebra Appl., 465(2015), pp. 107–138.
[28] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.
[29] P. G. SCHMITZ AND L. YING, *A fast direct solver for elliptic problems on general meshes in 2D*, J. Comput. Phys., 231 (2012), pp. 1314–1338.
[30] X. SUN AND N. P. PITSIANIS, *A matrix version of the fast multipole method*, SIAM Review, 43 (2001), pp. 289–300.
[31] E. E. TYRTYSHNIKOV, *Mosaic-skeleton approximations*, Calcolo, 33 (1996), 47–57.
[32] S. WANG, M. V. DE HOOP, AND J. XIA, *Acoustic inverse scattering via Helmholtz operator factorization and optimization*, J. Comput. Phys., 229 (2010), pp. 8445–8462.
[33] Y. XI AND J. XIA, *On the stability of some hierarchical rank structured matrix algorithms*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1279–1303.
[34] Y. XI, J. XIA, S. CAULEY, AND V. BALAKRISHNAN, *Superfast and stable structured solvers for Toeplitz least squares via randomized sampling*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 44–72.
[35] Y. XI, J. XIA, AND R. CHAN, *A fast randomized eigensolver with structured LDL factorization update*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 974–996.
[36] J. XIA, *On the complexity of some hierarchical structured matrix algorithms*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 388–410.
[37] J. XIA, *Efficient structured multifrontal factorization for general large sparse matrices*, SIAM J. Sci. Comput., 35 (2013), pp. A832–A860.
[38] J. XIA, *Randomized sparse direct solvers*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 197–227.
[39] J. XIA, *O(n) complexity randomized 3D direct solver with HSS2D structure*, Purdue GMIG Report 14-18, April 2014.
[40] J. XIA, *Unifying hierarchically structured sparse and dense matrix factorizations*, under preparation.
[41] J. XIA, S. CHANDRASEKARAN, M. GU, AND X. S. LI, *Superfast multifrontal method for large structured linear systems of equations*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1382–1411.
[42] J. XIA, S. CHANDRASEKARAN, M. GU, AND X. S. LI, *Fast algorithms for hierarchically semiseparable matrices*, Numer. Linear Algebra Appl., 17 (2010), pp. 953–976.
[43] J. XIA AND M. GU, *A multi-structured stable and superfast Toeplitz solver*, preprint, 2009.
[44] J. XIA, Y. XI, S. CAULEY, AND V. BALAKRISHNAN, *Fast sparse selected inversion*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1283–1314.
[45] J. XIA, Y. XI, AND M. GU, *A superfast structured solver for Toeplitz linear systems via randomized sampling*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 837–858.
[46] X. YE, J. XIA, AND L. YING, *Analytical low-rank compression via proxy point selection*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 1059–1085.
[47] L. YING, G. BIROS, AND D. ZORIN, *A kernel-independent adaptive fast multipole algorithm in*

two and three dimensions, J. Comput. Phys., 196 (2004), pp. 591–626.