Electronic Transactions on Numerical Analysis.
Volume 54, pp. 581–609, 2021.
Copyright © 2021, Kent State University.
ISSN 1068–9613.
DOI: 10.1553/etna_vol54s581

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A STABLE MATRIX VERSION OF THE FAST MULTIPOLE METHOD: STABILIZATION STRATEGIES AND EXAMPLES[*]

DIFENG CAI[†] AND JIANLIN XIA[‡]

Abstract. The fast multipole method (FMM) is an efficient method for evaluating matrix-vector products related to certain discretized kernel functions. The method involves an underlying FMM matrix given by a sequence of smaller matrices (called generators for convenience). Although there has been extensive work in designing and applying FMM techniques, the stability of the FMM and the stable FMM matrix factorization have rarely been studied. In this work, we propose techniques that lead to stable operations with FMM matrices. One key objective is to give stabilization strategies that can be used to provide low-rank approximations and translation relations in the FMM satisfying some stability requirements. The standard Taylor expansions used in FMMs yield basis generators susceptible to instability. Here, we introduce some scaling factors to control the relevant norms of the generators and give a rigorous analysis of the bounds of the entrywise magnitudes. The second objective is to use the one-dimensional case as an example to provide an intuitive construction of FMM matrices satisfying some stability conditions and then convert an FMM matrix into a hierarchically semiseparable (HSS) form that admits stable ULV-type factorizations. This bridges the gap between the FMM and stable FMM matrix factorizations. The HSS construction is done analytically and does not require expensive algebraic compression. Relevant stability studies are given, which show that the resulting matrix forms are suitable for stable operations. Note that the essential stabilization ideas are also applicable to higher dimensions. Extensive numerical tests are given to illustrate the reliability and accuracy.

Key words. numerical stability, fast multipole method, FMM matrix, scaling factor, low-rank approximation, HSS matrix

AMS subject classifications. 65F30, 65F35, 15A23, 15A60

1. Introduction.

Let $\kappa(x, y)$ be a kernel function in a form such as $1/(x-y)$, $1/(x-y)^2$, $\log(x-y)$, or $\log|x-y|$, where $x, y \in \mathbb{C}$, $x \neq y$. Given a set of points

$$\mathbf{s} \equiv \{x_1, \dots, x_n\}, \qquad x_i \in \mathbb{C}, \tag{1.1}$$

let A be an $n \times n$ discretized matrix with entries

$$A_{ij} = \kappa(x_i, x_j), \qquad i \neq j. \tag{1.2}$$

The diagonal entries A_{ii} are defined separately and do not concern us so far. It is well known that the fast multipole method (FMM) [14, 27] can be used to evaluate the product of A with a vector to a given accuracy in linear complexity. As shown in [30], the FMM essentially yields a hierarchical structured approximation of A to a given accuracy. Such a structured approximation is also an example of an $\mathcal{H}^2$-matrix [17, 19]. For convenience, we refer to this approximation derived with the FMM procedure as an *FMM matrix*.

The construction of an FMM matrix often involves appropriate degenerate approximations or truncated expansions of $\kappa(x, y)$. Commonly used expansions are Taylor expansions, multipole expansions, and spherical harmonic expansions. Such expansions provide convenient ways to obtain low-rank approximations of the off-diagonal blocks $(\kappa(x_i, y_j)))_{x_i\in\mathbf{x}_1, y_j\in\mathbf{x}_2}$ of A that correspond to well-separated subsets $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{s}$. This will be made more precise later.

[*]Received January 31, 2021. Accepted September 1, 2021. Published online on October 4, 2021. Recommended by Raf Vandebril. The research of Jianlin Xia was supported in part by an NSF grant DMS-1819166.

[†]Department of Mathematics, Emory University, Atlanta, GA 30322, USA (difeng.cai@emory.edu).

[‡]Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA (xiaj@purdue.edu).

Practical implementations of the FMM have usually been very successful in achieving both high efficiency and nice accuracy. On the other hand, it has also been noticed that numerical stability issues may arise under certain circumstances [8, 9, 10, 15, 26]. Here in particular, we are interested in the stability of the FMM based on Taylor expansions of $\kappa(x, y)$. Taylor expansions can produce low-rank basis matrices with very large entries, although the original matrix entries $\kappa(x_i, y_j)$ may only have modest magnitudes. Examples of such terms are factorials and powers. The artificially large terms may lead to stability issues in the relevant matrix operations, as pointed out in [15]. They can cause a loss of accuracy (due to the magnification of numerical errors) or even overflow. Note that these stability risks can arise even if $\kappa(x, y)$ is non-oscillatory as considered here. This happens especially when the data points are not nicely distributed, like in the case in [26], where accuracy is crucial when dealing with data points that are clustered eigenvalues. Thus, it is important to study the relevant numerical stability. A heuristic strategy to improve the stability is briefly mentioned in [15], but it lacks a rigorous justification or a guarantee of performance. Note that, despite the stability risks, the FMM has worked well for many different problems, likely due to the use of certain basis or translation operators that have some structure or sparsity.

Here, for the FMM based on Taylor expansions, our first objective is to provide a stabilization strategy by analytically obtaining low-rank basis matrices and translation matrices that satisfy some stability requirements. More specifically, we design a scaling strategy where some appropriate scaling factors are chosen to modify the individual terms in the Taylor expansions. Then, for well-separated subsets $\mathbf{x}_1, \mathbf{x}_2 \subset \mathbf{s}$, the block $(\kappa(x_i, y_j)))_{x_i \in \mathbf{x}_1, y_j \in \mathbf{x}_2}$ can be approximated by a low-rank form $\hat{U}\hat{B}\hat{V}^T$, where the entries of $\hat{U}$ and $\hat{V}$ have magnitudes bounded by 1, and, moreover, the entries of $\hat{B}$ have magnitudes bounded by a small multiple of $|\kappa(x, y)|$ evaluated at appropriate centers of $\mathbf{x}_1$ and $\mathbf{x}_2$. See Theorem 2.5 for details. The low-rank approximations in the FMM also involve the key concept of a translation matrix. We give one specific form of the translation matrix and further show that, after scaling with our scaling factors, the entries of the translation matrix also have entrywise magnitudes bounded by 1. See Theorem 2.7. Based on these bounds, the stability of matrix operations with the resulting structured forms can be naturally shown. We illustrate a basic idea of the backward stability analysis in Theorem 2.9.

Our second objective is to extend the stabilization to another structured matrix form so as to bridge the gap between the FMM and stable direct factorizations. We use the one-dimensional (1D) case as an example to provide an intuitive way to express the FMM matrix in an explicit form based on the stabilization strategies. Then the 1D FMM matrix is converted into a hierarchical semiseparable (HSS) form [4, 5, 36] that is frequently used to design structured direct solvers. The 1D case is very useful for computations of PDE solutions, Toeplitz solutions, polynomial computations, and eigenvalue solutions. See, e.g., [4, 6, 11, 23, 24, 29, 31]. The original FMM [14] explains the method in terms of potential evaluations. Here, we illustrate a stable matrix version that can be conveniently understood based on appropriate basis matrices called *contributions*, organized at different hierarchical levels by a nested basis relation. This matrix form is convenient for non-experts to grasp the FMM. An FMM matrix $\hat{A} \approx A$ is given in terms of a sequence of smaller matrices (which we call FMM generators), such as $\hat{U}, \hat{B}, \hat{V}$ as above and some translation matrices. $\hat{A}$ enables fast matrix-vector multiplications, but the stable factorization has been unknown. By converting $\hat{A}$ into an HSS form, we can take advantage of many fast and stable HSS algorithms, especially the so-called HSS ULV factorization [5] with proven nice backward stability [32, 33]. The FMM to HSS conversion is done analytically and avoids explicit algebraic compression like expensive truncated SVDs or randomized sampling used in [21, 22, 28, 36, 37]. The resulting HSS form is represented by a sequence of so-called HSS generators and can be factorized in $\mathcal{O}(n)$ complexity. All the FMM and HSS generators satisfy some norm bounds (see Corollary 3.4) that can be used to show the stability of the FMM and HSS algorithms. The techniques can also be generalized to the 2D case.

Overall, this work provides useful stability safeguards for matrix operations using FMM matrices. We show how and why the stabilization works and point out some essential ideas for the relevant backward stability analysis. We further give an example to illustrate an intuitive matrix version of the stable FMM. An analytical construction of HSS matrices from FMM matrices is also given so as to facilitate stable direct ULV factorizations. Our stabilization strategies are derived in terms of 2D point sets. We would like to emphasize that *the stabilization strategies and the stability studies are not restricted to 2D cases and are also applicable to higher dimensions*. Also, the use of the 1D example to illustrate the FMM matrix form is merely for convenience. The stabilization can also be applied to several kernel functions with related Taylor series expansions.

The structure of the paper is as follows. Section 2 is devoted to the ideas of stabilizing the FMM via stable analytical low-rank approximations and translation operations. In Section 3, the ideas are then used for the construction of the FMM matrix, which is further converted into an HSS matrix. Some discussions and extensive numerical tests are given in Section 4 to illustrate the stability and accuracy.

2. Stabilization of the FMM: stable low-rank approximation and translation operation.

In this section, we show how to obtain low-rank kernel matrix approximations that are suitable for stable operations. We further provide a stable translation relation to derive nested basis matrices. The techniques give essential components for stabilizing the FMM.

2.1. Kernel expansions and low-rank kernel matrix approximations.

Suppose a kernel function $\kappa(x, y)$ has a degenerate approximation of the following form for some points x, y:

$$\kappa(x,y) \approx \sum_{k=0}^{r-1}\sum_{l=0}^{k} \alpha_{k,l}\phi_l(x)\psi_{k-l}(y). \tag{2.1}$$

We suppose that the points are taken from 2D point sets, and they are treated as complex numbers. This assumption can be modified to accommodate higher dimensions. It is well-known that, if r is small compared to the numbers of x, y points, (2.1) yields a low-rank approximation of the kernel matrix (defined by the evaluation of $\kappa(x, y)$ at those x, y points). Here for simplicity, we mainly illustrate our techniques in terms of the following kernel:

$$\kappa(x,y) = \frac{1}{x-y}, \qquad x \neq y. \tag{2.2}$$

Note that *the use of this kernel is only for convenience since the ideas can be immediately extended to several other kernels* with similar degenerate approximations (see Section 2.4 below). For such kernels, Taylor expansions can be used to obtain (2.1).

We provide some details of the expansion following a strategy in [30] so as to facilitate our later derivations. For a set of points $\mathbf{x} \subset \mathbb{C}$, a point $z \in \mathbb{C}$ is said to be a center for $\mathbf{x}$ with a corresponding radius if $|x - z| \leq \delta$ for any $x \in \mathbf{x}$. Such a definition for z and δ is used in [30] and some other FMM work. It is clear that z and δ may not be unique. In case a unique z and δ is required, we may use a disk enclosing the points with the smallest radius. Since the uniqueness is not a concern here, we simply follow the tradition in [30]. The following definition from [30] is a generalization of the classical definition of well-separated sets.

DEFINITION 2.1 ([30]). *Suppose* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are two sets of points in* $\mathbb{C}$ *respectively corresponding to centers* $z_1 \in \mathbb{C}$ *and* $z_2 \in \mathbb{C}$ *and radii* $\delta_1 > 0$ *and* $\delta_2 > 0$. $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are said to be (well)* separated (with separation ratio τ) *if the following admissibility condition holds:*

$$\delta_1 + \delta_2 \le \tau |z_1 - z_2|, \qquad \tau \in (0,1). \tag{2.3}$$

For two well-separated sets $\mathbf{x}_1, \mathbf{x}_2 \subset \mathbf{s}$ as in Definition 2.1 with $\mathbf{s}$ in (1.1), the condition (2.3) implies that, for any $x \in \mathbf{x}_1$ and $y \in \mathbf{x}_2$,

$$\left| \frac{(x - z_1) - (y - z_2)}{z_2 - z_1} \right| \le \frac{\delta_1 + \delta_2}{|z_2 - z_1|} \le \tau. \tag{2.4}$$

Applying a Taylor expansion leads to

$$\begin{aligned} \kappa(x,y) &= -\frac{1}{(z_2 - z_1)[1 - \frac{(x-z_1)-(y-z_2)}{z_2 - z_1}]} \\ &= -\frac{1}{z_2 - z_1} \sum_{k=0}^{r-1} \left[\frac{(x - z_1) - (y - z_2)}{z_2 - z_1} \right]^k + \epsilon_r \\ &= -\sum_{k=0}^{r-1} \frac{k!}{(z_2 - z_1)^{k+1}} \sum_{j=0}^{k} (-1)^{k-j} \frac{(x - z_1)^j}{j!} \frac{(y - z_2)^{k-j}}{(k-j)!} + \epsilon_r \\ &= \sum_{k=0}^{r-1} \alpha_k \sum_{j=0}^{k} (-1)^{k-j} f_j(x - z_1) f_{k-j}(y - z_2) + \epsilon_r, \end{aligned} \tag{2.5}$$

where

$$f_j(x) = \frac{x^j}{j!}, \qquad \alpha_k = -\frac{k!}{(z_2 - z_1)^{k+1}}, \qquad |\epsilon_r| \le \frac{\tau^r}{|z_2 - z_1|(1 - \tau)}. \tag{2.6}$$

Note that, by (2.4),

$$\begin{aligned} |\kappa(x,y)| &\ge \frac{1}{|(x - z_1) - (y - z_2)| + |z_1 - z_2|} \\ &\ge \frac{1}{\tau |z_1 - z_2| + |z_1 - z_2|} = \frac{1}{1+\tau} |\kappa(z_1, z_2)|. \end{aligned} \tag{2.7}$$

Hence, the truncation error ϵ_r can be estimated by

$$|\epsilon_r| \le \frac{\tau^r}{1 - \tau} |\kappa(z_1, z_2)| \le \tau^r \frac{1 + \tau}{1 - \tau} |\kappa(x, y)|,$$

which indicates that the relative error of approximation in (2.5) is bounded by $\tau^r \frac{1+\tau}{1-\tau}$. This is consistent with a conclusion in [30].

According to (2.5) and (2.6), we can then write

$$\kappa(x,y) = u^T \bar{B} v + \epsilon_r, \tag{2.8}$$

where

$$u = \begin{bmatrix} f_0(x-z_1) & f_1(x-z_1) & \cdots & f_{r-1}(x-z_1) \end{bmatrix}^T,$$
$$v = \begin{bmatrix} f_0(y-z_2) & f_1(y-z_2) & \cdots & f_{r-1}(y-z_2) \end{bmatrix}^T,$$

$$\bar{B} = \begin{bmatrix} \alpha_0 & \alpha_1 & \cdots & \alpha_{r-1} \\ \alpha_1 & \iddots & \iddots & \\ \vdots & \iddots & & \\ \alpha_{r-1} & & & 0 \end{bmatrix} \operatorname{diag}\left((-1)^0, (-1)^1, \dots, (-1)^{r-1}\right). \tag{2.9}$$

Here, $\operatorname{diag}(\dots)$ is used to represent a diagonal matrix (or a block diagonal matrix later).

Then, we consider the low-rank approximation of the discretized matrix defined by the evaluation of $\kappa(x,y)$ on $\mathbf{x}_1, \mathbf{x}_2$:

$$K = (\kappa(x_i, x_j))_{x_i \in \mathbf{x}_1, x_j \in \mathbf{x}_2}, \tag{2.10}$$

which has the (i,j)th entry $\kappa(x_i, y_j)$. K is of size $m \times p$ with $m = |\mathbf{x}_1|$, $p = |\mathbf{x}_2|$, and is sometimes referred to as the *interaction* (matrix) between $\mathbf{x}_1$ and $\mathbf{x}_2$. Based on (2.8), K has a low-rank approximation

$$K = \bar{U}\bar{B}\bar{V}^T + K \odot E \approx \bar{U}\bar{B}\bar{V}^T, \tag{2.11}$$

where $\odot$ denotes the entrywise (Hadamard) product and

$$\bar{U} = (f_{j-1}(x_i - z_1))_{m \times r}, \qquad \bar{V} = (f_{j-1}(y_i - z_2))_{p \times r}, \tag{2.12}$$

$$|E_{ij}| \le \tau^r \frac{1+\tau}{1-\tau}. \tag{2.13}$$

Here, the notation $(A_{ij})_{m \times n}$ means an $m \times n$ matrix having the (i,j)th entry A_{ij}. We see that $\bar{U}$ and $\bar{V}$ are fully determined by the sets $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. The matrix $\bar{B}$ is an $r \times r$ matrix that depends only on $z_2 - z_1$.

2.2. The stable low-rank approximation with scaling factors and an analysis of entrywise magnitudes. According to (2.9) and (2.12), the matrices $\bar{U}, \bar{B}, \bar{V}$ in the low-rank approximation (2.11) may have large entrywise magnitudes. This is because of the powers and factorials in (2.6). As mentioned in the introduction, directly using the forms of $\bar{U}, \bar{B}, \bar{V}$ may cause stability issues in the low-rank approximation (2.11) and later operations. The stability issue gets more severe when r or the size of K increases. To ensure numerical stability, we introduce a scaling strategy so as to bound the entries of the factors in the low-rank approximation. We further rigorously justify the effectiveness of the scaling.

One set of scaling parameters is used for each set of points $\mathbf{x_i} \subset \mathbf{s}$ for $\mathbf{s}$ in (1.1). Suppose that $\mathbf{x_i}$ has the center $z_\mathbf{i}$ and radius $\delta_\mathbf{i}$. (Here, we use subscripts in bold fonts to denote indices of point sets.) For a set $\mathbf{x_i}$, define the *scaling factors*

$$\eta_{\mathbf{i},j} = \begin{cases} 1, & j = 0, \\ \left(\frac{j}{e}(2\pi r)^{\frac{1}{2r}} \frac{1}{\delta_\mathbf{i}}\right)^j, & j = 1, 2, \dots, r-1. \end{cases} \tag{2.14}$$

We would like to point out that we first proposed these scaling factors $\eta_{\mathbf{i},j}$ in our earlier preprint [3]. Later, the paper [2] briefly mentioned $\eta_{\mathbf{i},j}$ by citing [3]. Such a form is motivated by Stirling's formula:

$$\lim_{r\to\infty} \frac{r!}{\sqrt{2\pi r}\,(r/e)^r} = 1, \qquad \text{or} \qquad r! \sim \sqrt{2\pi r}\left(\frac{r}{e}\right)^r \text{ for large } r.$$

We use $\eta_{\mathbf{i},j}$ to modify the approximation of K in (2.10) with two separated sets $\mathbf{x}_1$ and $\mathbf{x}_2$. For $x \in \mathbf{x}_1$ and $y \in \mathbf{x}_2$, the expansion in (2.5) can be rewritten as

(2.15)
$$\kappa(x,y) = \sum_{k=0}^{r-1} \alpha_k \sum_{j=0}^{k} (-1)^{k-j} (\eta_{1,j})^{-1} (\eta_{2,k-j})^{-1} \big(\eta_{1,j} f_j(x-z_1)\big)\big(\eta_{2,k-j} f_{k-j}(y-z_2)\big) + \epsilon_r.$$

Compared with (2.11), the approximation of K now becomes

$$K = \hat{U}\hat{B}\hat{V}^T + K \odot E \approx \hat{U}\hat{B}\hat{V}^T, \tag{2.16}$$

where

$$\begin{aligned} \hat{U} &= \left(\eta_{1,j-1} f_j(x_i - z_1)\right)_{m\times r} \equiv \bar{U} S_1, \\ \hat{V} &= \left(\eta_{2,j-1} f_j(y_i - z_2)\right)_{p\times r} \equiv \bar{V} S_2, \\ \hat{B} &= S_1^{-1} \bar{B} S_2^{-1}, \end{aligned} \tag{2.17}$$

and, for $\mathbf{i} = 1, 2$,

$$S_{\mathbf{i}} = \operatorname{diag}\left(\eta_{\mathbf{i},0}, \eta_{\mathbf{i},1}, \ldots, \eta_{\mathbf{i},r-1}\right). \tag{2.18}$$

REMARK 2.2. Here, $\hat{U}$ is a basis matrix that only depends on $\mathbf{x}_1$. In fact, if K is replaced by the interaction between $\mathbf{x}_1$ and any other separated set, $\hat{U}$ remains the same. Thus, $\hat{U}$ can be viewed as the *contribution* of $\mathbf{x}_1$ (to the FMM). $\hat{V}$ can be viewed similarly. An intuitive way of understanding the matrix form of the FMM is to treat the basis matrices as such contributions to the FMM.

To investigate how the new approximation (2.16) enhances the stability, we give bounds for the entries of the matrices $\hat{U}, \hat{V}, \hat{B}$. The following lemmas will be used.

LEMMA 2.3. *For any integer $r > 0$ and any number $\tau \in (0, \frac{4}{5})$,*

$$g_j \equiv \frac{1}{j!}\left(\frac{j}{e}(2\pi r)^{\frac{1}{2r}}\right)^j \le 1, \qquad h_j \equiv \frac{\tau^j}{g_j} < 3\tau, \qquad j = 1, 2, \ldots, r. \tag{2.19}$$

Proof. Let $s = \frac{1}{e}(2\pi r)^{\frac{1}{2r}}$. Then $\frac{1}{e} < s < 1$ and $g_j = \frac{j^j}{j!} s^j$. Since

$$\frac{g_{j+1}}{g_j} = s\left(1 + \frac{1}{j}\right)^j,$$

as j increases, g_j either increases monotonically, decreases monotonically, or first decreases and then increases, depending on r. Thus,

$$g_j \le \max\{g_1, g_r\} = \max\left\{s, \frac{(r/e)^r\sqrt{2\pi r}}{r!}\right\} \le 1.$$

To show the second inequality in (2.19), notice that for any $j > 1$,

$$\frac{h_{j+1}}{h_j} = \tau\frac{g_j}{g_{j+1}} = \tau s^{-1}\left(1+\frac{1}{j}\right)^{-j} < \frac{4}{5}e\left(1+\frac{1}{j}\right)^{-j} < 1.$$

Then for $j > 1$, h_j decreases as j increases. Thus,

$$\max_{j=1,\ldots,r} h_j \le \max\{h_1, h_2\} = \max\left\{e\tau(2\pi r)^{-\frac{1}{2r}}, \frac{1}{2}(e\tau)^2(2\pi r)^{-\frac{1}{r}}\right\} < 3\tau. \qquad \square$$

LEMMA 2.4. *Let k be any positive integer and $\tau > 0$. Then*

$$\max_{t\in(0,\tau)} t^j(\tau-t)^{k-j} = \tau^k \left(\frac{j}{k}\right)^j \left(\frac{k-j}{k}\right)^{k-j}, \qquad j = 1, 2, \ldots, k-1. \tag{2.20}$$

Proof. Let $\varphi(t) = t^j(\tau-t)^{k-j}$, $t \in (0,\tau)$. Since

$$\frac{\mathrm{d}}{\mathrm{d}t}(\log\varphi(t)) = \frac{j}{t} - \frac{k-j}{\tau-t},$$

we can see that $\log\varphi(t)$ has only one critical point $t_0 = j\tau/k$ in $(0,\tau)$ for $j < k$. It can be verified that $\varphi(t_0)$ is the maximum in $(0,\tau)$. Since $\varphi(t_0) = \tau^k\left(\frac{j}{k}\right)^j\left(\frac{k-j}{k}\right)^{k-j}$, we get (2.20). ∎

Based on the lemmas, we can estimate the magnitudes of the entries of the matrices $\hat{U}, \hat{V}, \hat{B}$ in (2.17).

THEOREM 2.5. *Let K be given by (2.10), and let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two separated sets with separation ratio $\tau \in (0, \frac{4}{5})$ and with centers z_1 and z_2, respectively. Then for the approximation in (2.16)–(2.17), the (i,j)th entries of the matrices $\hat{U}, \hat{V}, \hat{B}$ satisfy*

$$|\hat{U}_{ij}| \le 1, \qquad |\hat{V}_{ij}| \le 1, \qquad |\hat{B}_{ij}| \le \max\{1, 3\tau\}|\kappa(z_1,z_2)|.$$

Proof. According to (2.17), $\hat{U}_{ij} = \eta_{1,j-1}f_{j-1}(x_i - z_1)$, where $\eta_{1,j-1}$ is defined in (2.14). Clearly, $|\hat{U}_{ij}| = 1$, for $j = 1$. For $j = 2, \ldots, r$,

$$\begin{aligned}|\hat{U}_{ij}| &= |\eta_{1,j-1}f_{j-1}(x_i-z_1)| = \left(\frac{j-1}{e}(2\pi r)^{\frac{1}{2r}}\frac{1}{\delta_1}\right)^{j-1}\frac{|x_i-z_1|^{j-1}}{(j-1)!}\\ &= \frac{1}{(j-1)!}\left(\frac{j-1}{e}(2\pi r)^{\frac{1}{2r}}\right)^{j-1}\left(\frac{|x_i-z_1|}{\delta_1}\right)^{j-1} = g_{j-1}\cdot\left(\frac{|x_i-z_1|}{\delta_1}\right)^{j-1},\end{aligned} \tag{2.21}$$

where g_{j-1} is defined following (2.19). By Lemma 2.3, $g_{j-1} \le 1$. This together with $|x_i - z_1| \le \delta_1$ leads to $|\hat{U}_{ij}| \le 1$. Similarly, $|\hat{V}_{ij}| \le 1$. We then estimate $|\hat{B}_{ij}|$. According to (2.9) and (2.17),

$$|\hat{B}_{ij}| = |\alpha_k|\eta_{1,i-1}^{-1}\eta_{2,j-1}^{-1}, \qquad i + j \le r + 1,$$

where $k = i + j - 2$ and α_k is given in (2.6). For $k = 0$ or $i = j = 1$, we simply have $|\hat{B}_{11}| = \frac{1}{|z_1-z_2|}$. For $k \ge 1$, we look at different cases for i, j. For $i = 1$ and $j > 1$, we have $\eta_{1,i-1} = 1$ and

$$\begin{aligned}|\hat{B}_{1j}| = |\alpha_k\eta_{2,j-1}^{-1}| &= \frac{(j-1)!}{|z_2-z_1|^j}\left(\frac{(j-1)}{e}(2\pi r)^{\frac{1}{2r}}\frac{1}{\delta_2}\right)^{-j+1}\\ &= \frac{1}{|z_1-z_2|}(j-1)!\left(\frac{(j-1)}{e}(2\pi r)^{\frac{1}{2r}}\right)^{-j+1}\left(\frac{\delta_2}{|z_1-z_2|}\right)^{j-1}\\ &= \frac{1}{|z_1-z_2|}\frac{1}{g_{j-1}}\left(\frac{\delta_2}{|z_1-z_2|}\right)^{j-1}.\end{aligned}$$

According to (2.3),

$$\frac{\delta_2}{|z_1-z_2|} \le \frac{\tau\delta_2}{\delta_1+\delta_2}. \tag{2.22}$$

Then

$$|\hat{B}_{1j}| \le \frac{1}{|z_1-z_2|}\frac{\tau^{j-1}}{g_{j-1}}\left(\frac{\delta_2}{\delta_1+\delta_2}\right)^{j-1} \le \frac{3\tau}{|z_1-z_2|} = 3\tau|\kappa(z_1,z_2)|,$$

where Lemma 2.3 is used. For $j=1$, the derivation is similar to the case when $i=1$. For $i,j>1$, we have $2<k<r$ and

$$\begin{aligned}
|\hat{B}_{ij}| &= |\alpha_k \eta_{1,i-1}^{-1}\eta_{2,j-1}^{-1}| \\
&= \frac{k!}{|z_1-z_2|^{k+1}}\left(\frac{i-1}{e}(2\pi r)^{\frac{1}{2r}}\frac{1}{\delta_1}\right)^{-i+1}\left(\frac{j-1}{e}(2\pi r)^{\frac{1}{2r}}\frac{1}{\delta_2}\right)^{-j+1} \\
&= \frac{1}{|z_1-z_2|}k!\left(\frac{1}{e}(2\pi r)^{\frac{1}{2r}}\right)^{-k}(i-1)^{-i+1}(j-1)^{-j+1} \\
&\quad \times \left(\frac{\delta_1}{|z_1-z_2|}\right)^{i-1}\left(\frac{\delta_2}{|z_1-z_2|}\right)^{j-1} \\
&= \frac{1}{|z_1-z_2|}\frac{k^k}{g_k}(i-1)^{-i+1}(j-1)^{-j+1}\left(\frac{\delta_1}{|z_1-z_2|}\right)^{i-1}\left(\frac{\delta_2}{|z_1-z_2|}\right)^{j-1} \\
&\le \frac{1}{|z_1-z_2|}\frac{k^k}{g_k}(i-1)^{-i+1}(j-1)^{-j+1}\left(\frac{\tau\delta_1}{\delta_1+\delta_2}\right)^{i-1}\left(\frac{\tau\delta_2}{\delta_1+\delta_2}\right)^{j-1},
\end{aligned}$$

where $\frac{\delta_1}{|z_1-z_2|} \le \frac{\tau\delta_1}{\delta_1+\delta_2}$ and (2.22) are used. By setting $t=\frac{\tau\delta_1}{\delta_1+\delta_2}<\tau$ in Lemma 2.4, we further get

$$\begin{aligned}
|\hat{B}_{ij}| &\le \frac{1}{|z_1-z_2|}\frac{k^k}{g_k}(i-1)^{-i+1}(j-1)^{-j+1}\tau^k\left(\frac{i-1}{k}\right)^{i-1}\left(\frac{j-1}{k}\right)^{j-1} \\
&= \frac{1}{|z_1-z_2|}\frac{\tau^k}{g_k} \le \frac{3\tau}{|z_1-z_2|} = 3\tau|\kappa(z_1,z_2)|,
\end{aligned}$$

where Lemma 2.3 is used. This completes the proof. □

Hence, the entries of the basis matrices $\hat{U}$ and $\hat{V}$ in (2.17) have magnitudes bounded by 1. $\hat{B}$ is just a small matrix of size $r\times r$ and its entries have magnitudes bounded by a small multiple of $|\kappa(z_1,z_2)|$ which depends on the two centers only. These bounds ensure the stability of matrix operations with the low-rank approximation $\hat{U}\hat{B}\hat{V}^T$. See Section 2.5 later.

REMARK 2.6. It is clear that our scaling strategy can control the entrywise magnitudes of not only $\hat{U},\hat{V}$, but also $\hat{B}$. This is a significant advantage over simple methods such as a straightforward scaling/normalization of the columns of $\bar{U},\bar{V}$. The latter can make the entries of the resulting $\hat{B}$ matrix poorly scaled or even cause numerical overflow. Even if the simple scaling factors from the latter strategy can be represented as separate diagonal matrices, forming these scaling factors and the entries of $\bar{B}$ in a separate manner can pose numerical issues (since they may still be very large for some cases). The entries of $\bar{B}$ may still be much larger than the original entries in K and thus not be well controlled. With our strategy, the entrywise magnitudes of $\hat{B}$ are under control, and we can integrate our scaling factors into the computation of the entries of $\hat{B}$ if needed.

2.3. A stable translation relation and an analysis of entrywise magnitudes. A key idea for the FMM to reach linear complexity is to exploit a *translation relation* between the so-called local expansions associated with one point set and its subsets [14]. This is essentially using nested basis matrices in the off-diagonal approximations. Here, we give an explicit matrix relation that ensures stable operations. To facilitate later discussions, we assume that a set $\mathbf{x_i} \subset \mathbf{s}$ has center $z_\mathbf{i}$ and radius $\delta_\mathbf{i}$, as mentioned at the beginning of Section 2.2 and that a subset $\mathbf{x_c} \subset \mathbf{x_i}$ has center $z_\mathbf{c}$ and radius $\delta_\mathbf{c}$.

As mentioned in Remark 2.2, we can regard a basis matrix $\hat{U}_\mathbf{i}$ as the contribution of $\mathbf{x_i}$ and a basis matrix $\hat{U}_\mathbf{c}$ as the contribution of $\mathbf{x_c}$. In the FMM, the translation relation is used to connect $\hat{U}_\mathbf{c}$ to the contribution of $\mathbf{x_c}$ to $\hat{U}_\mathbf{i}$. Specifically, the translation relation in our context can be derived for f_j in (2.6) as follows:

$$(2.23)\quad \begin{aligned} f_j(x-z_\mathbf{i}) &= \frac{(x-z_\mathbf{i})^j}{j!} = \frac{((x-z_\mathbf{c})+(z_\mathbf{c}-z_\mathbf{i}))^j}{j!} \\ &= \sum_{i=0}^{j} \frac{(x-z_\mathbf{c})^i}{i!}\frac{(z_\mathbf{c}-z_\mathbf{i})^{j-i}}{(j-i)!} = \sum_{l=0}^{j} \left(\eta_{\mathbf{c},i} f_i(x-z_\mathbf{c})\right)\left(\eta_{\mathbf{c},i}^{-1} f_{j-i}(z_\mathbf{c}-z_\mathbf{i})\right), \end{aligned}$$

where we have included the scaling factor $\eta_{\mathbf{c},i}$ for stability purposes. Therefore, a row in $\hat{U}_\mathbf{i}$ can be written as

$$(2.24)\quad \begin{aligned} &\begin{bmatrix} \eta_{\mathbf{i},0} f_0(x-z_\mathbf{i}) & \cdots & \eta_{\mathbf{i},r-1} f_{r-1}(x-z_\mathbf{i}) \end{bmatrix} \\ &\quad = \begin{bmatrix} \eta_{\mathbf{c},0} f_0(x-z_\mathbf{c}) & \cdots & \eta_{\mathbf{c},r-1} f_{r-1}(x-z_\mathbf{c}) \end{bmatrix} T_{\mathbf{c},\mathbf{i}}, \end{aligned}$$

where $\begin{bmatrix} \eta_{\mathbf{c},0} f_0(x-z_\mathbf{c}) & \cdots & \eta_{\mathbf{c},r-1} f_{r-1}(x-z_\mathbf{c}) \end{bmatrix}$ is a row of $\hat{U}_\mathbf{c}$ and $T_{\mathbf{c},\mathbf{i}}$ is the *translation matrix*

$$(2.25)\quad T_{\mathbf{c},\mathbf{i}} = S_\mathbf{c}^{-1} \begin{bmatrix} f_0(z_\mathbf{c}-z_\mathbf{i}) & \cdots & f_{r-1}(z_\mathbf{c}-z_\mathbf{i}) \\ & \ddots & \vdots \\ & & f_0(z_\mathbf{c}-z_\mathbf{i}) \end{bmatrix} S_\mathbf{i},$$

with $S_\mathbf{i}$ in (2.18) and $S_\mathbf{c}$ defined in the same way. With the translation matrix $T_{\mathbf{c},\mathbf{i}}$, the contribution of x to $\hat{U}_\mathbf{c}$ is related to the contribution of x to $\hat{U}_\mathbf{i}$ as in (2.24).

We then study the entrywise magnitudes of $T_{\mathbf{c},\mathbf{i}}$. To accommodate the general situation that $\mathbf{x_c}$ may be any subset of $\mathbf{x_i}$ resulting from the partitioning of $\mathbf{x_i}$, we suppose that

$$(2.26)\quad \delta_\mathbf{i} - \delta_\mathbf{c} \geq |z_\mathbf{c} - z_\mathbf{i}|,$$

so that the disk defined by $|x - z_\mathbf{c}| \leq \delta_\mathbf{c}$ (that encloses $\mathbf{x_c}$) is fully located inside the disk $|x - z_\mathbf{i}| \leq \delta_\mathbf{i}$ (that encloses $\mathbf{x_i}$).

THEOREM 2.7. *Suppose that* (2.26) *holds. Then the* (i,j)*th entry* $(T_{\mathbf{c},\mathbf{i}})_{i,j}$ *of* $T_{\mathbf{c},\mathbf{i}}$ *defined in* (2.25) *satisfies* $|(T_{\mathbf{c},\mathbf{i}})_{i,j}| \leq 1$.

Proof. $T_{\mathbf{c},\mathbf{i}}$ is an upper triangular matrix, and the (i,j)th entry for $1 \leq i \leq j \leq r$ is

$$(T_{\mathbf{c},\mathbf{i}})_{i,j} = \eta_{\mathbf{i},j-1}\eta_{\mathbf{c},i-1}^{-1} f_{j-i}(z_\mathbf{c}-z_\mathbf{i}).$$

If $j = 1$, then $(T_{\mathbf{c},\mathbf{i}})_{i,j} = 1$. Now suppose that $j > 1$. We look at different cases of i.

1. When $i = 1$, just like in the derivation in (2.21), we have

$$\begin{aligned} |(T_{\mathbf{c},\mathbf{i}})_{i,j}| &= |\eta_{\mathbf{i},j-1} f_{j-1}(z_\mathbf{c}-z_\mathbf{i})| = \left(\frac{j-1}{e}(2\pi r)^{\frac{1}{2r}}\frac{1}{\delta_\mathbf{i}}\right)^{j-1}\frac{|z_\mathbf{c}-z_\mathbf{i}|^{j-1}}{(j-1)!} \\ &= g_{j-1}\cdot\left(\frac{|z_\mathbf{c}-z_\mathbf{i}|}{\delta_\mathbf{i}}\right)^{j-1} \leq 1, \end{aligned}$$

   where Lemma 2.3 and (2.26) are used.

2. When $1 < i < j$,

$$
\begin{aligned}
|(T_{\mathbf{c},\mathbf{i}})_{i,j}| &= (j-1)^{j-1}(i-1)^{-i+1}\left(\frac{1}{e}(2\pi r)^{\frac{1}{2r}}\right)^{j-i}\frac{\delta_{\mathbf{c}}^{i-1}}{\delta_{\mathbf{i}}^{j-1}}\frac{|z_{\mathbf{c}}-z_{\mathbf{i}}|^{j-i}}{(j-i)!}\\
&\le (j-1)^{j-1}(i-1)^{-i+1}\frac{1}{(j-i)!}\left(\frac{1}{e}(2\pi r)^{\frac{1}{2r}}\right)^{j-i}\left(\frac{\delta_{\mathbf{c}}}{\delta_{\mathbf{i}}}\right)^{i-1}\left(1-\frac{\delta_{\mathbf{c}}}{\delta_{\mathbf{i}}}\right)^{j-i}\\
&\qquad \text{(by (2.26))}\\
&\le (j-1)^{j-1}(i-1)^{-i+1}\frac{1}{(j-i)!}\left(\frac{1}{e}(2\pi r)^{\frac{1}{2r}}\right)^{j-i}\left(\frac{i-1}{j-1}\right)^{i-1}\left(\frac{j-i}{j-1}\right)^{j-i}\\
&\qquad \text{(by Lemma 2.4)}\\
&= \frac{1}{(j-i)!}\left(\frac{j-i}{e}(2\pi r)^{\frac{1}{2r}}\right)^{j-i} = g_{j-i} \le 1,
\end{aligned}
$$

where the last inequality is due to Lemma 2.3.

3. When $i = j$,

$$
|(T_{\mathbf{c},\mathbf{i}})_{i,j}| = |\eta_{\mathbf{i},i-1}\eta_{\mathbf{c},i-1}^{-1}| = \left(\frac{\delta_{\mathbf{c}}}{\delta_{\mathbf{i}}}\right)^{i-1} \le 1. \qquad \square
$$

In Section 3.1, we show how the translation matrix $T_{\mathbf{c},\mathbf{i}}$ is used to build a nested basis form for $\hat{U}_{\mathbf{i}}$.

REMARK 2.8. It is worth pointing out that there are also other analytical methods that can produce translation matrices that satisfy similar entrywise bounds. For example, the method in [13] uses an integral form and quadrature approximation to obtain translation operators in diagonal forms with entrywise magnitudes bounded by 1. On the other hand, the resulting basis matrices depend on the quadrature weights, and the bounds for their entries are not studied in [13]. Here, our idea is to integrate the scaling into a simple Taylor expansions so as to control the entrywise magnitudes of all the relevant matrices.

2.4. Generalizations. It can be shown that our results can be generalized to various relevant kernels like $1/(x-y)^k$ with an integer $k > 0$, $\log(x-y)$, $\log|x-y|$, and other kernels with expansions similar to (2.5). In fact, by using the same set of scaling factors as in Section 2.2, we can get bounds similar to those in Theorem 2.5. That is, the entrywise bound for the $\hat{U}, \hat{V}$ basis matrices remain to be 1. The relative entrywise bound for the $\hat{B}$ generators only changes slightly. In our numerical tests in Section 4, tests for different kernels will be given. For some kernels that do not have similar Taylor expansions, the stabilization is beyond the scope of this work.

2.5. Stability. Our stabilization strategies ensure the stability of operations involving the resulting structured forms in the FMM. For example, the stability of applying $\hat{U}\hat{B}\hat{V}^T$ to vectors can be shown as follows.

THEOREM 2.9. *For the $m \times p$ interaction matrix K in (2.10), suppose that the same conditions as in Theorem 2.5 hold, and let $\hat{K} = \hat{U}\hat{B}\hat{V}^T$ be the approximation of K as in (2.16)–(2.17). Then the matrix-vector multiplication $\hat{b} = \hat{U}\hat{B}\hat{V}^T w \approx Kw$ for a vector w satisfies*

$$
\begin{gathered}
\mathrm{fl}(\hat{b}) = (\hat{U}\hat{B}\hat{V}^T + \Delta\hat{K})w, \qquad \textit{with}\\
\|\Delta\hat{K}\|_F \le \max\{1, 3\tau\}(1+\tau)r^2\sqrt{mp}\,\gamma_{p+2r}\|K\|_F + \mathcal{O}(\epsilon_{\mathrm{mach}}^2),
\end{gathered}
$$

where $\mathrm{fl}(\cdot)$ *denotes the numerical result in floating point arithmetic,* ϵ_{mach} *denotes the machine epsilon, and* $\gamma_k = \dfrac{k\epsilon_{\mathrm{mach}}}{1-k\epsilon_{\mathrm{mach}}}$.

Proof. It is commonly known that (see, e.g., [20]) for a matrix C with column size p, the matrix-vector multiplication Cw satisfies the following backward error bound:

$$\mathrm{fl}(Cw) = (C + \Delta C)w, \qquad |\Delta C| \le \gamma_p |C|.$$

Thus, when the matrix-vector multiplication $\hat{K}w = \hat{U}\hat{B}\hat{V}^T w$ is considered, we have

$$\begin{aligned} b_1 &= \mathrm{fl}(\hat{V}^T w) = (\hat{V}^T + \Delta\hat{V}^T)w, & |\Delta\hat{V}^T| &\le \gamma_p |\hat{V}^T|, \\ b_2 &= \mathrm{fl}(\hat{B}b_1) = (\hat{B} + \Delta\hat{B})b_1, & |\Delta\hat{B}| &\le \gamma_r |\hat{B}|, \\ \hat{b} &= \mathrm{fl}(\hat{U}b_2) = (\hat{U} + \Delta\hat{U})b_2, & |\Delta\hat{U}| &\le \gamma_r |\hat{U}|. \end{aligned}$$

Note that $\hat{K}$ is of size $m \times p$ and $\hat{B}$ of size $r \times r$. Combining these results, we get

$$\mathrm{fl}(\hat{U}\hat{B}\hat{V}^T w) = (\hat{U} + \Delta\hat{U})(\hat{B} + \Delta\hat{B})(\hat{V}^T + \Delta\hat{V}^T)w \equiv (\hat{U}\hat{B}\hat{V}^T + \Delta\hat{K})w,$$

where

$$\begin{aligned} \|\Delta\hat{K}\|_F &\le \|\hat{U}\hat{B}(\Delta\hat{V}^T)\|_F + \|\hat{U}(\Delta\hat{B})\hat{V}^T\|_F + \|(\Delta\hat{U})\hat{B}\hat{V}^T\|_F + \mathcal{O}(\epsilon_{\mathrm{mach}}^2) \\ &\le (\gamma_p + 2\gamma_r)\|\hat{U}\|_F \|\hat{B}\|_F \|\hat{V}\|_F + \mathcal{O}(\epsilon_{\mathrm{mach}}^2). \end{aligned}$$

Here, we use the Frobenius norm in the backward error instead of the max-norm since the former is sub-multiplicative but the latter is not. According to Theorem 2.5, we have

$$\begin{aligned} &\|\hat{U}\|_F \le \sqrt{mr}\|\hat{U}\|_{\max} \le \sqrt{mr}, \qquad \|\hat{V}\|_F \le \sqrt{rq}\|\hat{V}\|_{\max} \le \sqrt{rq}, \\ &\|\hat{B}\|_F \le r\|\hat{B}\|_{\max} \le r\max\{1, 3\tau\}|\kappa(z_1, z_2)|, \end{aligned}$$

where z_1 and z_2 are the centers of $\mathbf{x}_1$ and $\mathbf{x}_2$ in (2.10), respectively. Due to the separation condition in Definition 2.1, we have (2.7) for any $x \in \mathbf{x}_1$ and $y \in \mathbf{x}_2$. Thus,

$$\|\hat{B}\|_F \le r\max\{1, 3\tau\}(1+\tau)|\kappa(x,y)| \le r\max\{1, 3\tau\}(1+\tau)\|K\|_F.$$

Accordingly,

$$\begin{aligned} \|\Delta\hat{K}\|_F &\le (\gamma_p + 2\gamma_r)\sqrt{mr}\sqrt{rp}(r\max\{1,3\tau\}(1+\tau)\|K\|_F) + \mathcal{O}(\epsilon_{\mathrm{mach}}^2) \\ &= \max\{1,3\tau\}(1+\tau)r^2\sqrt{mp}\frac{(p+2r)\epsilon_{\mathrm{mach}} - 3rp\epsilon_{\mathrm{mach}}^2}{1-(p+r)\epsilon_{\mathrm{mach}} + rp\epsilon_{\mathrm{mach}}^2}\|K\|_F + \mathcal{O}(\epsilon_{\mathrm{mach}}^2) \\ &\le \max\{1,3\tau\}(1+\tau)r^2\sqrt{mp}\gamma_{p+2r}\|K\|_F + \mathcal{O}(\epsilon_{\mathrm{mach}}^2). \qquad \square \end{aligned}$$

This theorem shows the backward stability of using the low-rank approximation $\hat{K}$ to compute the matrix-vector product $\hat{K}w$ that approximates Kw. For this reason, it makes somewhat more sense to use K in the backward error bound.

Note that if no scaling is used like in the usual FMM, then $\|\hat{U}\|_{\max}$, $\|\hat{V}\|_{\max}$, and/or $\|\hat{B}\|_{\max}$ may potentially get very large, leading to significantly larger backward error bounds. The impact can be observed in the numerical results later.

3. Extension of the stabilization result from FMM to HSS matrices.

We now provide an example of an intuitive analytical construction of an FMM matrix satisfying some stability requirements and, moreover, extend the stabilization result from the FMM matrix to an HSS form. This further connects the FMM with stable and fast ULV-factorizations for HSS matrices.

3.1. An example of an FMM matrix representation. We first integrate the stabilization strategy in the previous section into the FMM framework for constructing an FMM matrix example. For convenience, we consider the 1D case and suppose that the set of points $\mathbf{s}$ in (1.1) is located in an interval $\mathcal{I} \subset \mathbb{R}$. Note that 1D cases are very useful in many different situations [4, 6, 11, 23, 24, 29, 31]. The 1D point set is also just used to simplify the presentation. The strategy below can be easily adapted to more general 1D curves. The essential ideas can also be extended to 2D sets. We consider A in (1.2) being the discretization of κ in (2.2) on $\mathbf{s}$. Given an accuracy ε, we follow the general framework in [30] and use the 1D FMM scheme to produce an FMM matrix $\hat{A}$ such that

$$A = \hat{A} + A \odot E, \qquad \text{with} \quad |E_{ij}| \le \varepsilon. \tag{3.1}$$

According to (2.13), r can be chosen to make $\tau^r \dfrac{1+\tau}{1-\tau} \le \varepsilon$.

3.1.1. Set partitioning and far-field interaction. To conveniently organize the FMM matrix representation, we use a postordered binary tree $\mathcal{T}$ with nodes $\mathbf{i} = 1, 2, \ldots, \mathrm{root}(\mathcal{T})$, where $\mathrm{root}(\mathcal{T})$ denotes the root node. See Figure 3.1. Suppose that $\mathcal{T}$ has L levels such that $n/2^{L-1} = \mathcal{O}(r)$ and $\mathrm{root}(\mathcal{T})$ is at level 0. Partition the set $\mathbf{s}$ hierarchically following $\mathcal{T}$. That is, suppose that each node $\mathbf{i}$ is associated with a subset $\mathbf{x_i} \subset \mathbf{s}$ so that $\mathbf{x}_{\mathrm{root}(\mathcal{T})} = \mathbf{s}$ and $\mathbf{x_i} = \mathbf{x}_{\mathbf{c}_1} \cup \mathbf{x}_{\mathbf{c}_2}$, $\mathbf{x}_{\mathbf{c}_1} \cap \mathbf{x}_{\mathbf{c}_2} = \emptyset$ for any nonleaf node $\mathbf{i}$ with children $\mathbf{c}_1$ and $\mathbf{c}_2$. Based on the subinterval where $\mathbf{x_i}$ is located, we can conveniently determine a center $z_\mathbf{i}$ and a radius $\delta_\mathbf{i}$ of $\mathbf{x_i}$. For each leaf $\mathbf{i}$, the cardinality of $\mathbf{x_i}$ satisfies $m_\mathbf{i} \equiv |\mathbf{x_i}| = \mathcal{O}(r)$.

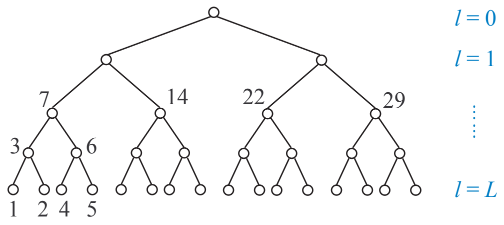


FIG. 3.1. *Example of a postordered tree $\mathcal{T}$ used for the FMM.*

Later for convenience, when $\mathbf{x_i}$ is used, we may simply refer to the node $\mathbf{i}$ of $\mathcal{T}$. For example, given two nodes $\mathbf{i}$ and $\mathbf{j}$ of $\mathcal{T}$ corresponding to two separated sets $\mathbf{x_i}$ and $\mathbf{x_j}$ (as defined in Definition 2.1), respectively, we just say $\mathbf{i}$ and $\mathbf{j}$ are separated.

Suppose that $\mathbf{x_i}$ corresponds to the index set $\mathcal{I}_\mathbf{i}$ so that the submatrix of A corresponding to the row index set $\mathcal{I}_\mathbf{i}$ and column index set $\mathcal{I}_\mathbf{j}$ is $A|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}} \equiv (\kappa(x_i, x_j))_{x_i \in \mathbf{x_i}, x_j \in \mathbf{x_j}}$, which is the interaction between $\mathbf{i}$ and $\mathbf{j}$. When $\mathbf{i}$ and $\mathbf{j}$ are separated, $A|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}}$ can be approximated by a low-rank form like in (2.16) and is said to be a *far-field* interaction. For notational convenience, we rewrite (2.16) as

$$A|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}} = \hat{U}_\mathbf{i} \hat{B}_{\mathbf{i},\mathbf{j}} \hat{V}_\mathbf{j}^T + A|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}} \odot E_{\mathbf{i},\mathbf{j}} \approx \hat{U}_\mathbf{i} \hat{B}_{\mathbf{i},\mathbf{j}} \hat{V}_\mathbf{j}^T, \tag{3.2}$$

where appropriate sets used for the definition of the matrices in (2.17) are replaced by $\mathbf{x_i}$ and $\mathbf{x_j}$. Correspondingly, the centers $z_\mathbf{i}, z_\mathbf{j}$, the radii $\delta_\mathbf{i}, \delta_\mathbf{j}$, and the scaling factors $\eta_{\mathbf{i},j}, \eta_{\mathbf{j},j}$ as in (2.14) are used for the definition of $\hat{U}_\mathbf{i}, \hat{B}_{\mathbf{i},\mathbf{j}}, \hat{V}_\mathbf{j}^T$ in (3.2).

As mentioned in Remark 2.2, we call $\hat{U}_\mathbf{i}$ the *contribution* (matrix) from node $\mathbf{i}$. Clearly, $\hat{U}_\mathbf{i} = \hat{V}_\mathbf{i}$. However, to accommodate more general matrix forms, we still use $\hat{V}_\mathbf{i}^T$ for the row basis matrix in (3.2).

When $\mathbf{i}$ and $\mathbf{j}$ are not separated, they are said to be *near neighbors*, and $A|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$ is a *near-field* interaction. Near-field interactions may be further partitioned so as to generate far-field interactions at finer levels.

3.1.2. The levelwise low-rank approximation. In the FMM, far-field interactions are organized with the aid of *interaction lists* [14], which encode the interactions to consider at each level of partition. Specifically for our case, the *interaction list* $\mathcal{L}_\mathbf{i}$ for node $\mathbf{i}$ of $\mathcal{T}$ is the set of nodes $\mathbf{j}$ at the same level as $\mathbf{i}$ but well-separated from $\mathbf{i}$, and with its parent a near neighbor of $\mathbf{i}$.

Corresponding to level l of $\mathcal{T}$, let $A^{(l)}$ be the submatrix extracted from A by retaining only the blocks $A|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$ for all nodes $\mathbf{i}$ at level l and $\mathbf{j}\in\mathcal{L}_\mathbf{i}$ and zeroing out other blocks in A. For example, for $l=2$, the four nodes in Figure 3.1 have interaction lists $\mathcal{L}_7=\{22,29\}$, $\mathcal{L}_{14}=\{29\}$, $\mathcal{L}_{22}=\{7\}$, $\mathcal{L}_{29}=\{7,14\}$. The corresponding far-field interactions are displayed in Figure 3.2(a). Similarly, the far-field interactions for $l=3,4$ are displayed in Figure 3.2(b–c). Correspondingly, the matrix A can be decomposed levelwise into the following sum of matrices corresponding to far-field interactions and near-field interactions:

$$A = A^{(2)} + \cdots + A^{(L)} + A^{(N)}, \tag{3.3}$$

where $A^{(N)}$ denotes all the near-field interactions at the leaf level L of the partition. $A^{(N)}$ is a block-banded matrix.

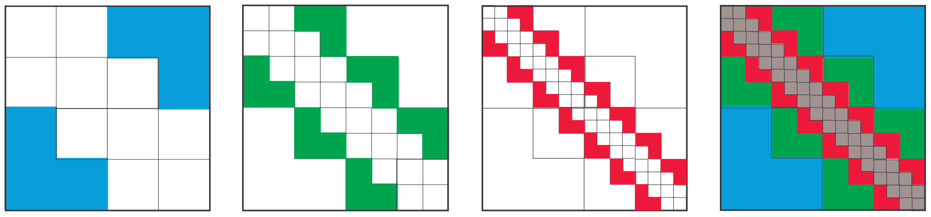

(a) $A^{(2)}$ (b) $A^{(3)}$ (c) $A^{(4)}$ (d) A

FIG. 3.2. *The nonzero patterns of $A^{(l)}$ and how $A^{(l)}$ appears in A, where the grey band in (d) corresponds to $A^{(N)}$.*

For $l\ge 2$, the nonzero block $A|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$ for each node $\mathbf{i}$ at level l and $\mathbf{j}\in\mathcal{L}_\mathbf{i}$ has a low-rank approximation as in (3.2). For convenience, let $\mathbf{i}_1,\ldots,\mathbf{i}_\beta$ be the nodes at level l of $\mathcal{T}$, ordered from left to right. Then, we can write

$$A^{(l)} = \hat{U}^{(l)}\hat{B}^{(l)}(\hat{V}^{(l)})^T + A^{(l)}\odot E^{(l)} \approx \hat{U}^{(l)}\hat{B}^{(l)}(\hat{V}^{(l)})^T, \qquad \text{with} \tag{3.4}$$

$$\hat{U}^{(l)} = \operatorname{diag}(\hat{U}_{\mathbf{i}_1},\ldots,\hat{U}_{\mathbf{i}_\beta}), \qquad \hat{V}^{(l)} = \operatorname{diag}(\hat{V}_{\mathbf{i}_1},\ldots,\hat{V}_{\mathbf{i}_\beta}), \tag{3.5}$$

and $\hat{B}^{(l)}$ and $E^{(l)}$ have the same block nonzero patterns as $A^{(l)}$ with the nonzero blocks $A|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$ of $A^{(l)}$ replaced by $\hat{B}_{\mathbf{i},\mathbf{j}}$ and $E_{\mathbf{i},\mathbf{j}}$, respectively. See Figure 3.3.

From (3.3) and (3.4), we have the following approximation of A:

$$A = \sum_{l=2}^{L}\hat{U}^{(l)}\hat{B}^{(l)}(\hat{V}^{(l)})^T + A^{(N)} + A\odot E \approx \sum_{l=2}^{L}\hat{U}^{(l)}\hat{B}^{(l)}(\hat{V}^{(l)})^T + A^{(N)} \equiv \hat{A}, \tag{3.6}$$

where $E = E^{(2)}+\cdots+E^{(L)}$. Since the nonzero blocks of $E^{(l)}$ for different l do not overlap, E satisfies the bound in (3.1).

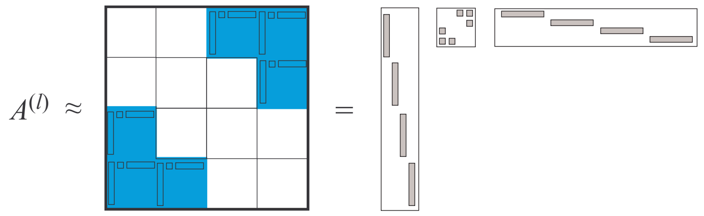


FIG. 3.3. *The nonzero patterns of $\hat{U}^{(l)}$, $\hat{B}^{(l)}$, and $\hat{V}^{(l)}$ in (3.4) for $A^{(l)}$ with $l = 2$ in Figure 3.2(a).*

Thus, $\hat{A}$ is an approximation of A with entrywise relative accuracy ε as in (3.1). It can be easily seen that $\hat{A}$ can be used to compute matrix-vector products in $\mathcal{O}(rnL) = \mathcal{O}(rn\log n)$ flops with $r = \mathcal{O}(|\log\varepsilon|)$. Assuming ε fixed, then this cost becomes $\mathcal{O}(n\log n)$.

3.1.3. Nested basis matrices and the FMM matrix in a telescoping expansion form. The essential strategy to reduce the matrix-vector multiplication cost from $\mathcal{O}(n\log n)$ to $\mathcal{O}(n)$ in the FMM is to use nested basis matrices in the off-diagonal approximations. This utilizes the translation relation in Section 2.3. According to the relation in (2.24), the basis matrices or contributions from a parent node $\mathbf{i}$ of $\mathcal{T}$ and its children $\mathbf{c}_1$ and $\mathbf{c}_2$ are related by

$$\hat{U}_{\mathbf{i}} = \begin{bmatrix} \hat{U}_{\mathbf{c}_1} & \\ & \hat{U}_{\mathbf{c}_2} \end{bmatrix} \begin{bmatrix} \hat{R}_{\mathbf{c}_1} \\ \hat{R}_{\mathbf{c}_2} \end{bmatrix}, \qquad \hat{V}_{\mathbf{i}} = \begin{bmatrix} \hat{V}_{\mathbf{c}_1} & \\ & \hat{V}_{\mathbf{c}_2} \end{bmatrix} \begin{bmatrix} \hat{W}_{\mathbf{c}_1} \\ \hat{W}_{\mathbf{c}_2} \end{bmatrix}, \qquad \text{with} \tag{3.7}$$

$$\hat{R}_{\mathbf{c}_1} = \hat{W}_{\mathbf{c}_1} = T_{\mathbf{c}_1,\mathbf{i}}, \qquad \hat{R}_{\mathbf{c}_2} = \hat{W}_{\mathbf{c}_2} = T_{\mathbf{c}_2,\mathbf{i}}. \tag{3.8}$$

Equation (3.7) shows how the nested basis matrices are obtained.

REMARK 3.1. Note that the translation relation (2.23) is a result of the binomial expansion. Although here $\mathbf{c}_1$ and $\mathbf{c}_2$ are children of $\mathbf{i}$, the translation relation in Section 2.3 is not restricted to the case that $\mathbf{c}$ is a child of $\mathbf{i}$. That is, $T_{\mathbf{c},\mathbf{i}}$ in (2.25) can be used for any descendant $\mathbf{c}$ of $\mathbf{i}$.

The approximation in (3.6) can then be converted into a nested form. That is, let

$$\hat{U}^{(l)} = \hat{U}^{(l+1)}\hat{R}^{(l+1)}, \qquad \hat{V}^{(l)} = \hat{V}^{(l+1)}\hat{W}^{(l+1)}, \quad l = 1, 2, \dots, L-1, \qquad \text{with}$$
$$\hat{R}^{(l+1)} = \operatorname{diag}\left(\begin{bmatrix} \hat{R}_{\mathbf{c}_1} \\ \hat{R}_{\mathbf{c}_2} \end{bmatrix}, \ \mathbf{c}_1, \mathbf{c}_2\text{: children of each node } \mathbf{i} \text{ at level } l \right),$$
$$\hat{W}^{(l+1)} = \operatorname{diag}\left(\begin{bmatrix} \hat{W}_{\mathbf{c}_1} \\ \hat{W}_{\mathbf{c}_2} \end{bmatrix}, \ \mathbf{c}_1, \mathbf{c}_2\text{: children of each node } \mathbf{i} \text{ at level } l \right).$$

We can rewrite the approximation in (3.4) as a recursive relation

$$\hat{U}^{(l)}\hat{B}^{(l)}(\hat{V}^{(l)})^T = \hat{U}^{(L)}\hat{R}^{(L-1)}\cdots\hat{R}^{(l)}\hat{B}^{(l)}(\hat{W}^{(l)})^T\cdots(\hat{W}^{(L-1)})^T(\hat{V}^{(L)})^T, \tag{3.9}$$

where $\hat{U}^{(L)}$ and $\hat{V}^{(L)}$ are defined for the leaf level L as in (3.5).

Inserting (3.9) into (3.6), we obtain the following *telescoping expansion* of $\hat{A}$:

$$\begin{aligned} \hat{A} = \hat{U}^{(L)}\Big(\hat{R}^{(L-1)}\Big(\cdots(\hat{R}^{(2)}\hat{B}^{(2)}(\hat{W}^{(2)})^T + \hat{B}^{(3)}) \\ \cdots\Big)(\hat{W}^{(L-1)})^T + \hat{B}^{(L)}\Big)(\hat{V}^{(L)})^T + A^{(N)}, \end{aligned} \tag{3.10}$$

which is the hierarchical matrix form produced by the FMM or the *FMM matrix*. For convenience, we call the matrices $\hat{U}_{\mathbf{i}}, \hat{V}_{\mathbf{i}}, \hat{R}_{\mathbf{i}}, \hat{W}_{\mathbf{i}}, \hat{B}_{\mathbf{i}}$ *FMM generators*. We also suppose that each node $\mathbf{i}$ of the FMM tree $\mathcal{T}$ is associated with FMM generators $\hat{U}_\mathbf{i}, \hat{V}_\mathbf{i}, \hat{R}_\mathbf{i}, \hat{W}_\mathbf{i}, \hat{B}_\mathbf{i}$. Due to the nested bases, the $\hat{U}_\mathbf{i}, \hat{V}_\mathbf{i}$ generators associated with a nonleaf node $\mathbf{i}$ are not explicitly stored. The total storage for the FMM matrix $\hat{A}$ is then just $\mathcal{O}(rn)$. The cost for multiply the FMM matrix and a vector now becomes $\mathcal{O}(rn)$.

3.2. The general idea of transforming FMM into HSS matrices. We now consider the conversion of the FMM matrix $\hat{A}$ in (3.10) into an HSS form. Note that in [2, 36, 37], the construction of HSS matrices is based on algebraic strategies. It is also possible to use analytical compression as done in [23, 39, 40, 41] for HSS constructions, but it is unclear whether the resulting HSS forms satisfy the stability requirements or not. Here, we use an analytical way to convert the FMM matrix into an HSS form. The resulting HSS form has a generator representation with the generators satisfying proper norm bounds.

An HSS matrix can be organized with the aid of a binary tree called HSS tree [36]. Here, we can use the same binary tree $\mathcal{T}$ as in Figure 3.1. An HSS form for $\hat{A}$ can be defined with the aid of a set of *HSS generators* $D_\mathbf{i}, U_\mathbf{i}, V_\mathbf{i}, R_\mathbf{i}, W_\mathbf{i}, B_\mathbf{i}$:

$$\hat{A} = D_{\mathrm{root}(\mathcal{T})}, \qquad D_\mathbf{i} = \begin{bmatrix} D_{\mathbf{c}_1} & U_{\mathbf{c}_1} B_{\mathbf{c}_1} V_{\mathbf{c}_2}^T \\ U_{\mathbf{c}_2} B_{\mathbf{c}_2} V_{\mathbf{c}_1}^T & D_{\mathbf{c}_2} \end{bmatrix}, \tag{3.11}$$

$$U_\mathbf{i} = \begin{bmatrix} U_{\mathbf{c}_1} & \\ & U_{\mathbf{c}_2} \end{bmatrix} \begin{bmatrix} R_{\mathbf{c}_1} \\ R_{\mathbf{c}_2} \end{bmatrix}, \qquad V_\mathbf{i} = \begin{bmatrix} V_{\mathbf{c}_1} & \\ & V_{\mathbf{c}_2} \end{bmatrix} \begin{bmatrix} W_{\mathbf{c}_1} \\ W_{\mathbf{c}_2} \end{bmatrix}, \tag{3.12}$$

where $\mathbf{c}_1, \mathbf{c}_2$ are the left and right children of a nonleaf node $\mathbf{i}$, respectively.

We use $\{1:n\}$ to denote the set $\{1, 2, \ldots, n\}$. Also, let $\mathcal{I}_\mathbf{i}$ be the index set associated with $D_\mathbf{i}$ such that $D_\mathbf{i} = \hat{A}|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{i}}$. Then we see from (3.11)–(3.12) that the columns of $U_\mathbf{i}$ span the column space of the block $A|_{\mathcal{I}_\mathbf{i} \times (\{1:n\} \backslash \mathcal{I}_\mathbf{i})}$. Similarly, the columns of $V_\mathbf{i}$ span the column space of the block $(A|_{(\{1:n\} \backslash \mathcal{I}_\mathbf{i}) \times \mathcal{I}_\mathbf{i}})^T$. Equation (3.12) indicates that the $U_\mathbf{i}, V_\mathbf{i}$ basis matrices have nested forms.

The HSS form also has a telescoping expansion [22]:

$$\hat{A} = U^{(L)} \Big(R^{(L-1)} \big(\cdots (R^{(2)} B^{(1)} (W^{(2)})^T + B^{(2)}) \cdots \big) (W^{(L-1)})^T + B^{(L-1)} \Big) \cdot (V^{(L)})^T + D^{(L)}, \tag{3.13}$$

where

$$\begin{aligned} D^{(L)} &= \mathrm{diag}(D_\mathbf{i},\ \mathbf{i}\text{: each node at level } L), \\ U^{(L)} &= \mathrm{diag}(U_\mathbf{i},\ \mathbf{i}\text{: each node at level } L), \\ V^{(L)} &= \mathrm{diag}(V_\mathbf{i},\ \mathbf{i}\text{: each node at level } L), \\ R^{(l)} &= \mathrm{diag}\left(\begin{bmatrix} R_{\mathbf{c}_1} \\ R_{\mathbf{c}_2} \end{bmatrix},\ \mathbf{c}_1, \mathbf{c}_2\text{: children of each node } \mathbf{i} \text{ at level } l < L \right), \\ W^{(l)} &= \mathrm{diag}\left(\begin{bmatrix} W_{\mathbf{c}_1} \\ W_{\mathbf{c}_2} \end{bmatrix},\ \mathbf{c}_1, \mathbf{c}_2\text{: children of each node } \mathbf{i} \text{ at level } l < L \right), \\ B^{(l)} &= \mathrm{diag}\left(\begin{bmatrix} 0 & B_{\mathbf{c}_1} \\ B_{\mathbf{c}_2} & 0 \end{bmatrix},\ \mathbf{c}_1, \mathbf{c}_2\text{: children of each node } \mathbf{i} \text{ at level } l < L \right). \end{aligned}$$

The telescoping expansion in (3.13) is similar to the expansion in (3.10) for the FMM. These two telescoping expansions have the following differences:

- In (3.10), the last term $A^{(N)}$ for the near-field interactions has a block-banded form, while in (3.13) only the diagonal blocks are considered as near-field interactions so that the last term $D^{(L)}$ has a block-diagonal form.

- Accordingly, the $\hat{U}^{(L)}$, $\hat{V}^{(L)}$ basis matrices in (3.10) are different from $U^{(L)}$, $V^{(L)}$ in (3.13), respectively, since they are basis matrices for different off-diagonal blocks. The matrices $\hat{R}^{(l)}$, $\hat{W}^{(l)}$ in (3.10) are also different from $R^{(l)}$, $W^{(l)}$ in (3.13), respectively.
- In (3.10), $\hat{B}^{(l)}$ has a block nonzero pattern similar to $A^{(l)}$, illustrated in Figure 3.2, while in (3.13), $B^{(l)}$ has a block-diagonal form.

We will resolve these differences by showing how to construct an HSS form from the FMM form. It should be noted that the HSS form that is generated is for the FMM matrix $\hat{A}$ in (3.10). That is, we are design an HSS approximation to A.

The basic idea of constructing the HSS form of $\hat{A}$ is to find HSS representations for the *far-field matrix* $\hat{A}^{(F)} \equiv \hat{A} - A^{(N)}$ and the near-field matrix $A^{(N)}$ separately and then to merge the two sets of HSS generators. In Figure 3.2(d), $A^{(N)}$ corresponds to the grey banded matrix along the diagonal and $\hat{A}^{(F)}$ corresponds to the remaining part of the matrix. To distinguish the generators for different matrices, we use the following notation.

- $\hat{U}, \hat{V}$, etc.: FMM generators of $\hat{A}^{(F)}$ from the FMM procedure in Section 3.1.
- U, V, etc.: HSS generators for the HSS form of $\hat{A}$.
- $\tilde{U}, \tilde{V}$, etc.: HSS generators for the HSS form of $\hat{A}^{(F)}$.
- $\breve{U}, \breve{V}$, etc.: HSS generators for the HSS form of $\hat{A}^{(N)}$.

The HSS representation for the near-field part $A^{(N)}$ can be explicitly written out based on its block-banded form. The main task is then to find the HSS representation of the far-field part $\hat{A}^{(F)}$. We do this in two steps:

1. First, we write each off-diagonal block in a low-rank form

$$\hat{A}^{(F)}|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}} = \tilde{U}_\mathbf{i} \tilde{B}_\mathbf{i} \tilde{V}_\mathbf{j}^T, \tag{3.14}$$

   where $\mathbf{i}$ and $\mathbf{j}$ are sibling nodes in $\mathcal{T}$ (denoted as $\mathbf{j} = \mathrm{sib}(\mathbf{i})$) with the corresponding index sets $\mathcal{I}_\mathbf{i}$ and $\mathcal{I}_\mathbf{j}$ in A, respectively. As in Section 3.1, we suppose each node $\mathbf{i}$ is associated with a set of points $\mathbf{x}_\mathbf{i} \in \mathbf{s}$.
2. Then we write the $\tilde{U}, \tilde{V}$ basis matrices in nested forms. That is, we obtain the $\tilde{R}, \tilde{W}$ generators in (3.12).

The two steps above will be elaborated in Sections 3.3 and 3.4, respectively. The HSS representations for $\hat{A}^{(F)}$ and $\hat{A}^{(N)}$ will be merged to form an HSS representation for $\hat{A}$ in Section 3.5.

3.3. Low-rank forms of the off-diagonal blocks of $\hat{A}^{(F)}$.

For sibling nodes $\mathbf{i}, \mathbf{j}$ of $\mathcal{T}$, we find the HSS generators $\tilde{U}_\mathbf{i}, \tilde{B}_\mathbf{i}, \tilde{V}_\mathbf{j}$ so as to write $\hat{A}^{(F)}|_{\mathcal{I}_\mathbf{i} \times \mathcal{I}_\mathbf{j}}$ in the form of (3.14).

The FMM procedure yields a partition that accounts for all far-field interactions between subsets of $\mathbf{x}_\mathbf{i}$ and $\mathbf{s} \backslash \mathbf{x}_\mathbf{i}$. Accordingly, $\mathcal{I}_\mathbf{i}$ is partitioned into subsets following the partitioning of $\mathbf{x}_\mathbf{i}$. Later for convenience, we consider the partition of the index set $\mathcal{I}_\mathbf{i}$ instead of $\mathbf{x}_\mathbf{i}$. Note that subsets resulting from the partitioning of $\mathcal{I}_\mathbf{i}$ correspond to the descendants of the node $\mathbf{i}$ in $\mathcal{T}$. Figure 3.4 illustrates the partitioning of $\mathcal{I}_\mathbf{i}$, and the subsets correspond to the nodes marked in Figure 3.5. These nodes form a set which we call the *partition list* associated with $\mathbf{i}$.

DEFINITION 3.2. *Suppose that $\mathcal{T}$ is a postordered full binary tree. Let $\mathbf{c}_1$ and $\mathbf{c}_\beta$ be the leftmost and rightmost leaf descendants of a node* $\mathbf{i}$*, respectively. Let $\mathcal{P}_1$ be the set of all the nodes in the path from* $\mathrm{par}(\mathbf{c}_1)$ *(the parent of $\mathbf{c}_1$) to the left child of* $\mathbf{i}$ *and $\mathcal{P}_2$ be the set of all the nodes in the path from* $\mathrm{par}(\mathbf{c}_\beta)$ *to the right child of* $\mathbf{i}$*. Then the* partition list *associated with* $\mathbf{i}$ *of $\mathcal{T}$ is*

$$\Omega_\mathbf{i} = \{\mathbf{c}_1\} \cup \{\textit{the right child of each } \mathbf{j} \in \mathcal{P}_1\} \cup \{\textit{the left child of each } \mathbf{j} \in \mathcal{P}_2\} \cup \{\mathbf{c}_\beta\}.$$

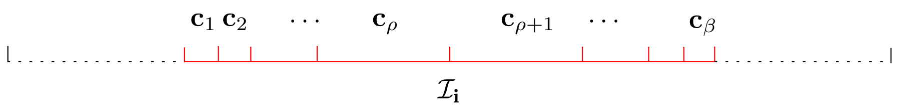

$\mathbf{c}_1$ $\mathbf{c}_2$ $\cdots$ $\mathbf{c}_\rho$ $\mathbf{c}_{\rho+1}$ $\cdots$ $\mathbf{c}_\beta$
$\mathcal{I}_\mathbf{i}$

FIG. 3.4. *Partitioning of the index set $\mathcal{I}_\mathbf{i}$ associated with node* $\mathbf{i}$.

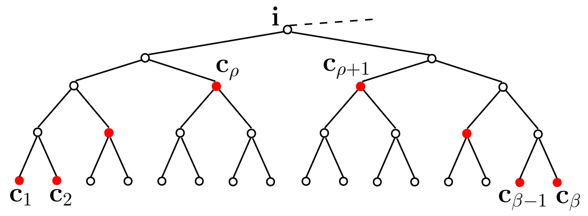

$\mathbf{i}$
$\mathbf{c}_\rho$
$\mathbf{c}_{\rho+1}$
$\mathbf{c}_1$ $\mathbf{c}_2$
$\mathbf{c}_{\beta-1}$ $\mathbf{c}_\beta$

FIG. 3.5. *Nodes in the partition list $\Omega_\mathbf{i}$ (marked as red solid nodes) corresponding to the partition of $\mathcal{I}_\mathbf{i}$ in Figure 3.4.*

Thus, $\Omega_\mathbf{i}$ consists of nodes $\mathbf{c_1}$ and $\mathbf{c}_\beta$ corresponding to the boundaries of $\mathcal{I}_\mathbf{i}$ and nodes at levels as high as possible for the interior subsets of $\mathcal{I}_\mathbf{i}$. When we study the interaction between $\mathbf{i}$ and other nodes, $\Omega_\mathbf{i}$ is used to provide a way to systematically organize the partition of $\mathcal{I}_\mathbf{i}$. The resulting partition similar as in Figure 3.4 is also used in [5].

We then find $\tilde{U}_\mathbf{i}$, $\tilde{V}_\mathbf{j}$, and $\tilde{B}_\mathbf{i}$ in (3.14). The FMM procedure yields a partition of $\mathcal{I}_\mathbf{i} \cup \mathcal{I}_\mathbf{j}$, leading to a blockwise agglomeration [18] of $\hat{A}^{(F)}|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$. For convenience, suppose that $\Omega_\mathbf{i}$ has the following form as marked in Figures 3.4–3.5:

$$\Omega_\mathbf{i} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_\rho, \mathbf{c}_{\rho+1}, \ldots, \mathbf{c}_\beta\}, \tag{3.15}$$

where $\mathbf{c}_\rho$ and $\mathbf{c}_{\rho+1}$ are the left and right children of $\mathbf{i}$, respectively. Similarly, suppose that $\Omega_\mathbf{j}$ has the following form:

$$\Omega_\mathbf{j} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_\xi, \mathbf{d}_{\xi+1}, \ldots, \mathbf{d}_\theta\}, \tag{3.16}$$

where $\mathbf{d}_\xi$ and $\mathbf{d}_{\xi+1}$ are the left and right children of $\mathbf{j}$, respectively. As shown in Section 3.1.1, for each pair of separated sets $\mathbf{c}_i$ and $\mathbf{d}_j$, we can find a low-rank form

$$\hat{A}^{(F)}|_{\mathcal{I}_{\mathbf{c}_i}\times\mathcal{I}_{\mathbf{d}_j}} = \hat{U}_{\mathbf{c}_i}\hat{B}_{\mathbf{c}_i,\mathbf{d}_j}\hat{V}_{\mathbf{d}_j}^T. \tag{3.17}$$

Note that $\hat{A}^{(F)}|_{\mathcal{I}_{\mathbf{c}_i}\times\mathcal{I}_{\mathbf{d}_j}} = 0$ if $\mathbf{c}_i$ and $\mathbf{d}_j$ are near neighbors. In such a case, we can set $\hat{B}_{\mathbf{c}_i,\mathbf{d}_j} = 0$ so that (3.17) still holds. Then we can assemble all the blocks $\hat{A}^{(F)}|_{\mathcal{I}_{\mathbf{c}_i}\times\mathcal{I}_{\mathbf{d}_j}}$ for $i = 1, \ldots, \beta$, $j = 1, \ldots, \theta$, into $\tilde{U}_\mathbf{i}\tilde{B}_\mathbf{i}\tilde{V}_\mathbf{j}^T$ in (3.14), where

$$\tilde{U}_\mathbf{i} = \operatorname{diag}(\hat{U}_{\mathbf{c}_1}, \ldots, \hat{U}_{\mathbf{c}_\beta}), \quad \tilde{V}_\mathbf{j} = \operatorname{diag}(\hat{V}_{\mathbf{d}_1}, \ldots, \hat{V}_{\mathbf{d}_\theta}), \tag{3.18}$$

$$\tilde{B}_\mathbf{i} = \begin{bmatrix} \hat{B}_{\mathbf{c}_1,\mathbf{d}_1} & \cdots & \hat{B}_{\mathbf{c}_1,\mathbf{d}_\theta} \\ \vdots & \cdots & \vdots \\ \hat{B}_{\mathbf{c}_\beta,\mathbf{d}_1} & \cdots & \hat{B}_{\mathbf{c}_\beta,\mathbf{d}_\theta} \end{bmatrix}. \tag{3.19}$$

An illustration of (3.14) with (3.18)–(3.19) is provided in Figure 3.6.

3.4. Nested $\tilde{U}$, $\tilde{V}$ basis matrices. We then derive the nested forms of the basis matrices. Suppose that $\mathbf{i}$ and $\mathbf{j}$ are a pair of sibling nodes with parent $\mathbf{p} = \operatorname{par}(\mathbf{i})$. Suppose that the

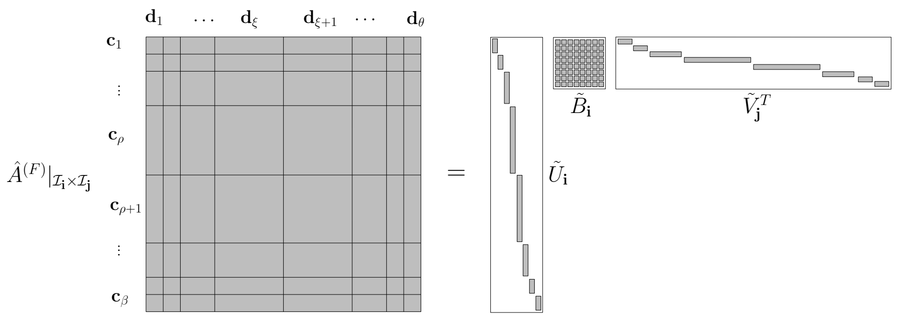


FIG. 3.6. *Illustration of* (3.14) *with* (3.18)–(3.19) *for the low-rank form of* $\hat{A}^{(F)}|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}$, *where* $\mathbf{j} = \mathrm{sib}(\mathbf{i})$.

partition lists $\Omega_\mathbf{i}$ and $\Omega_\mathbf{j}$ associated with $\mathbf{i}$ and $\mathbf{j}$ are given in (3.15) and (3.16), respectively, which are used for the partitioning of the corresponding index sets $\mathcal{I}_\mathbf{i}$ and $\mathcal{I}_\mathbf{j}$. Let the index set associated with $\mathbf{p}$ in A be $\mathcal{I}_\mathbf{p} = \mathcal{I}_\mathbf{i} \cup \mathcal{I}_\mathbf{j}$. Then the partition list $\Omega_\mathbf{p}$ associated with $\mathbf{p}$ can be obtained by merging and modifying $\Omega_\mathbf{i}$ and $\Omega_\mathbf{j}$. This is illustrated in Figure 3.7. We can then let

$$\Omega_\mathbf{p} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_\rho, \mathbf{e}_1, \mathbf{e}_2, \mathbf{d}_{\xi+1}, \ldots, \mathbf{d}_\theta\},$$

where $\mathbf{e}_1 = \mathrm{par}(\mathbf{c}_{\rho+1})$ and $\mathbf{e}_2 = \mathrm{par}(\mathbf{d}_\xi)$. Note that the nodes $\mathbf{c}_{\rho+1}, \ldots, \mathbf{c}_\beta$ are descendants of $\mathbf{e}_1$ and $\mathbf{d}_1, \ldots, \mathbf{d}_\xi$ are descendants of $\mathbf{e}_2$.

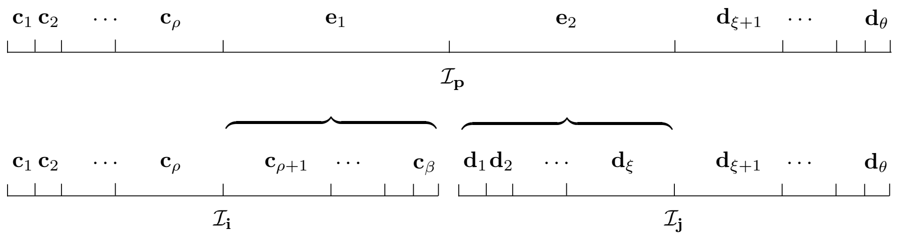


FIG. 3.7. *Merging the partitions of* $\mathcal{I}_\mathbf{i}$ *and* $\mathcal{I}_\mathbf{j}$ *to form the partition of* $\mathcal{I}_\mathbf{p}$.

As in (3.18), we have

$$\tilde{U}_\mathbf{p} = \mathrm{diag}(\hat{U}_{\mathbf{c}_1}, \ldots, \hat{U}_{\mathbf{c}_\rho}, \hat{U}_{\mathbf{e}_1}, \hat{U}_{\mathbf{e}_2}, \hat{U}_{\mathbf{d}_{\xi+1}}, \ldots, \hat{U}_{\mathbf{d}_\theta}).$$

From the translation relations in (2.24) and (3.8) and noticing Remark 3.1, $\hat{U}_{\mathbf{e}_1}$ and $\hat{U}_{\mathbf{e}_2}$ satisfy

$$\hat{U}_{\mathbf{e}_1} = \mathrm{diag}(\hat{U}_{\mathbf{c}_{\rho+1}} T_{\mathbf{c}_{\rho+1},\mathbf{e}_1}, \ldots, \hat{U}_{\mathbf{c}_\beta} T_{\mathbf{c}_\beta,\mathbf{e}_1}), \quad \hat{U}_{\mathbf{e}_2} = \mathrm{diag}(\hat{U}_{\mathbf{d}_1} T_{\mathbf{d}_1,\mathbf{e}_2}, \ldots, \hat{U}_{\mathbf{d}_\xi} T_{\mathbf{d}_\xi,\mathbf{e}_2}),$$

where the translation matrices $T_{\mathbf{c},\mathbf{e}_1}, T_{\mathbf{d},\mathbf{e}_2}$ are defined similar as in (2.25). Then

$$\begin{aligned}\tilde{U}_\mathbf{p} &= \mathrm{diag}(\hat{U}_{\mathbf{c}_1}, \ldots, \hat{U}_{\mathbf{c}_\rho}, \hat{U}_{\mathbf{c}_{\rho+1}} T_{\mathbf{c}_{\rho+1},\mathbf{e}_1}, \ldots, \hat{U}_{\mathbf{c}_\beta} T_{\mathbf{c}_\beta,\mathbf{e}_1}, \\ &\qquad\qquad \hat{U}_{\mathbf{d}_1} T_{\mathbf{d}_1,\mathbf{e}_2}, \ldots, \hat{U}_{\mathbf{d}_\xi} T_{\mathbf{d}_\xi,\mathbf{e}_2}, \hat{U}_{\mathbf{d}_{\xi+1}}, \ldots, \hat{U}_{\mathbf{d}_\theta}) \\ &= \mathrm{diag}(\tilde{U}_\mathbf{i}\tilde{R}_\mathbf{i}, \tilde{U}_\mathbf{j}\tilde{R}_\mathbf{j}),\end{aligned}$$

where

$$\tilde{R}_\mathbf{i} = \begin{bmatrix} \mathrm{diag}\left(I, \begin{bmatrix} T_{\mathbf{c}_{\rho+1},\mathbf{e}_1} \\ \vdots \\ T_{\mathbf{c}_\beta,\mathbf{e}_1} \end{bmatrix}\right) & 0 \end{bmatrix}, \quad \tilde{R}_\mathbf{j} = \begin{bmatrix} 0 & \mathrm{diag}\left(\begin{bmatrix} T_{\mathbf{d}_1,\mathbf{e}_2} \\ \vdots \\ T_{\mathbf{d}_\xi,\mathbf{e}_2} \end{bmatrix}, I\right) \end{bmatrix}.$$

Here, the zero blocks are chosen to make $\tilde{R}_{\mathbf{i}}$ and $\tilde{R}_{\mathbf{j}}$ have the same column size as $\tilde{U}_{\mathbf{p}}$. Thus we get the nested basis relationship

$$\tilde{U}_{\mathbf{p}} = \begin{bmatrix} \tilde{U}_{\mathbf{i}} & \\ & \tilde{U}_{\mathbf{j}} \end{bmatrix} \begin{bmatrix} \tilde{R}_{\mathbf{i}} \\ \tilde{R}_{\mathbf{j}} \end{bmatrix}.$$

This yields the nested relation for the $\tilde{U}$ basis matrices. We can similarly derive a nested basis relationship for $\tilde{V}_{\mathbf{i}}$. Since the translation matrices only depend on the relevant centers of subsets, $\tilde{R}_{\mathbf{i}}$ and $\tilde{W}_{\mathbf{i}}$ are only determined by the partition of $\mathcal{I}_{\mathbf{i}}$ and are independent of the actual points in $\mathcal{I}_{\mathbf{i}}$. It follows that the HSS generator satisfy

$$\tilde{W}_{\mathbf{i}} = \tilde{R}_{\mathbf{i}}. \tag{3.20}$$

At this point, we obtain all the $\tilde{U}, \tilde{V}, \tilde{R}, \tilde{W}, \tilde{B}$ generators for $\hat{A}^{(F)}$. The $\tilde{D}$ generators of $\hat{A}^{(F)}$ are zero blocks. Clearly, the generators have block structures that can be explored to save storage and computational costs.

3.5. HSS representation for $\hat{A}$. We now construct an HSS representation for $A^{(N)}$ so as to get an HSS form for $A = \hat{A}^{(F)} + A^{(N)}$. $A^{(N)}$ is a block-banded matrix. Suppose that $A^{(F)}$ and $A^{(N)}$ are partitioned conformably. Then the HSS form of $A^{(N)}$ can be explicitly written as [34]:

$$\begin{aligned}
&\check{U}_{\mathbf{i}} = I, \quad \check{V}_{\mathbf{i}} = I, \quad \text{for a leaf } \mathbf{i},\\
&\check{R}_{\mathbf{i}} = \begin{cases} [I \quad 0] & \text{if } \mathbf{i} \text{ is a leaf and } \mathbf{i} < \mathrm{sib}(\mathbf{i}),\\ [0 \quad I] & \text{if } \mathbf{i} \text{ is a leaf and } \mathbf{i} > \mathrm{sib}(\mathbf{i}),\\ \mathrm{diag}\,(I, 0)\,, & \text{if } \mathbf{i} \text{ is a nonleaf node and } \mathbf{i} < \mathrm{sib}(\mathbf{i}),\\ \mathrm{diag}\,(0, I)\,, & \text{if } \mathbf{i} \text{ is a nonleaf node and } \mathbf{i} > \mathrm{sib}(\mathbf{i}),\end{cases}\\
&\check{W}_{\mathbf{i}}\text{:} \quad \text{in the same form as } \check{R}_{\mathbf{i}},\\
&\check{B}_{\mathbf{i}} = \begin{cases} A|_{\mathcal{I}_{\mathbf{i}} \times \mathcal{I}_{\mathrm{sib}(\mathbf{i})}}, & \text{if } \mathbf{i} \text{ is a leaf and } \mathbf{i} < \mathrm{sib}(\mathbf{i}),\\ A|_{\mathcal{I}_{\mathrm{sib}(\mathbf{i})} \times \mathcal{I}_{\mathbf{i}}}, & \text{if } \mathbf{i} \text{ is a leaf and } \mathbf{i} > \mathrm{sib}(\mathbf{i}),\\ \begin{bmatrix} & 0 \\ A|_{\mathcal{I}_{\mathbf{i}} \times \mathcal{I}_{\mathrm{sib}(\mathbf{i})}} & \end{bmatrix}, & \text{if } \mathbf{i} \text{ is a nonleaf node and } \mathbf{i} < \mathrm{sib}(\mathbf{i}),\\ \begin{bmatrix} & A|_{\mathcal{I}_{\mathrm{sib}(\mathbf{i})} \times \mathcal{I}_{\mathbf{i}}} \\ 0 & \end{bmatrix}, & \text{if } \mathbf{i} \text{ is a nonleaf node and } i > \mathrm{sib}(\mathbf{i}).\end{cases}
\end{aligned} \tag{3.21}$$

With the HSS generators for $A^{(F)}$ and $A^{(N)}$ at hand, it is easy to verify (see, e.g., [34]) that the HSS generators for $\hat{A}$ are given by:

$$\begin{aligned}
D_{\mathbf{i}} &= \tilde{D}_{\mathbf{i}} + \check{D}_{\mathbf{i}}, & B_{\mathbf{i}} &= \mathrm{diag}(\tilde{B}_{\mathbf{i}}, \check{B}_{\mathbf{i}}),\\
U_{\mathbf{i}} &= \begin{bmatrix} \tilde{U}_{\mathbf{i}} & \check{U}_{\mathbf{i}} \end{bmatrix}, & V_{\mathbf{i}} &= \begin{bmatrix} \tilde{V}_{\mathbf{i}} & \check{V}_{\mathbf{i}} \end{bmatrix},\\
R_{\mathbf{i}} &= \mathrm{diag}(\tilde{R}_{\mathbf{i}}, \check{R}_{\mathbf{i}}), & W_{\mathbf{i}} &= \mathrm{diag}(\tilde{W}_{\mathbf{i}}, \check{W}_{\mathbf{i}}).
\end{aligned} \tag{3.22}$$

Due to the summation, the sizes of some generators such as $B_{\mathbf{i}}$ may be larger than necessary. If a more compact HSS form is desired, then a recompression step may be applied as done in some other HSS methods [7, 12, 35].

It can be shown that the cost to construct the HSS matrix is also $\mathcal{O}(rn)$. The ULV factorization of the resulting HSS form costs $\mathcal{O}(r^2 n)$.

REMARK 3.3. Here in the 1D case, both the FMM and the HSS forms use binary trees. For 2D problems, quad-trees are typically used for the FMM. If there is a need to convert a

2D FMM matrix to an HSS form, then we may rederive the FMM matrix form based on the repeated bisection of the domain so as to generate a binary tree structure. Then the conversion to an HSS form can follow a procedure similar to the 1D case by agglomerating low-rank subblocks to form an approximation to an off-diagonal block. However, there will be a lot more of such subblocks (as many as $O(\sqrt{n})$) than in the 1D case (at most $O(\log n)$). The maximum off-diagonal rank in the HSS form will be as large as $O(\sqrt{n})$. This makes the resulting HSS form less attractive than for the 1D case. In three dimensions, the off-diagonal rank will be even higher.

3.6. Norm bounds for the generators and additional stability discussions. Now we would like to briefly illustrate that the structured representations given in the previous sections satisfy some stability requirements needed for several computations such as matrix-vector multiplications (with the FMM or HSS form) and ULV factorizations (with the HSS form). The backward stability of several commonly used HSS algorithms has been studied in [5, 32, 33], where the stability analysis essentially relies on the following conditions:

- The U, V generators have bounded norms.
- The B generators have norms bounded by a small constant times the norm of A.

Thus, our purpose is to show that the FMM and HSS generators that we obtain using our stabilization strategy satisfy such norm requirements. Based on the analysis in Section 2, we have the following bounds for the norms of the FMM and HSS generators.

COROLLARY 3.4. *Suppose that* (2.26) *holds for any descendant* $\mathbf{c}$ *of a nonleaf node* $\mathbf{i}$ *in* $\mathcal{T}$. *Then for the approximation* $\hat{A}$ *to* A *in* (1.2) *with* (2.2) *and* $\tau \in (0, \frac{4}{5})$, *the FMM generators* $\hat{U}, \hat{V}, \hat{B}$ *in* (2.17) *and* $\hat{R}, \hat{W}$ *in* (3.8) *satisfy*

$$\begin{aligned}
&\|\hat{U}\|_{\max} \le 1, \qquad \|\hat{V}\|_{\max} \le 1, \qquad \|\hat{R}\|_{\max} \le 1, \qquad \|\hat{W}\|_{\max} \le 1,\\
&\|\hat{B}\|_{\max} \le \max\{1, 3\tau\}(1+\tau)\|A\|_{\max}.
\end{aligned}$$

The HSS generators U, V, R, W, B *in* (3.22) *satisfy*

$$\begin{aligned}
&\|U\|_{\max} \le 1, \qquad \|V\|_{\max} \le 1, \qquad \|R\|_{\max} \le 1, \qquad \|W\|_{\max} \le 1,\\
&\|B\|_{\max} \le \max\{1, 3\tau\}(1+\tau)\|A\|_{\max}.
\end{aligned}$$

Proof. The max-norm results for the generators $\hat{U}, \hat{V}, \hat{R}, \hat{W}$ are immediate from Theorems 2.5 and 2.7. When $\hat{A}|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}} = \hat{U}_\mathbf{i}\hat{B}_{\mathbf{i},\mathbf{j}}\hat{V}_\mathbf{j}^T$, as in (3.2), for two separated point sets $\mathbf{x}_\mathbf{i}$ and $\mathbf{x}_\mathbf{j}$, we can use Theorem 2.5 and the derivation in the proof of Theorem 2.9 to get

$$\|\hat{B}_{\mathbf{i},\mathbf{j}}\|_{\max} \le \max\{1, 3\tau\}(1+\tau)\|A|_{\mathcal{I}_\mathbf{i}\times\mathcal{I}_\mathbf{j}}\|_{\max} \le \max\{1, 3\tau\}(1+\tau)\|A\|_{\max}. \tag{3.23}$$

Next, it is clear from (3.21) that the HSS generators $\check{U}, \check{V}, \check{R}, \check{W}$ for $\hat{A}^{(N)}$ have entrywise magnitudes bounded by 1. Thus, it can be seen from (3.22) that the HSS generators U, V, R, W for $\hat{A}$ have entrywise magnitudes bounded by 1. The HSS generators $\tilde{B}$ as in (3.19) also satisfy the bound in (3.23). Then,

$$\begin{aligned}\|B\|_{\max} &\le \max\{\|\tilde{B}\|_{\max}, \|\check{B}\|_{\max}\} \le \max\{\max\{1, 3\tau\}(1+\tau)\|A\|_{\max}, \|A\|_{\max}\} \\ &\le \max\{1, 3\tau\}(1+\tau)\|A\|_{\max}.\end{aligned}$$

We thus get the bound for $\|B\|_{\max}$. ∎

Based on these norm bounds and the stability study in Section 2.5, the stability of the overall FMM algorithm and the HSS matrix-vector multiplication can be naturally shown. The stability analysis is similar to that in [32]. In fact, such stability can be conveniently understood based on the telescoping expansions in (3.10) and (3.13). The stability of ULV factorizations and solutions for the HSS form of $\hat{A}$ can be shown similarly to the work in [32, 33]. The actual derivations involve lengthy technical details and thus the readers are referred to [32, 33].

4. Numerical tests.

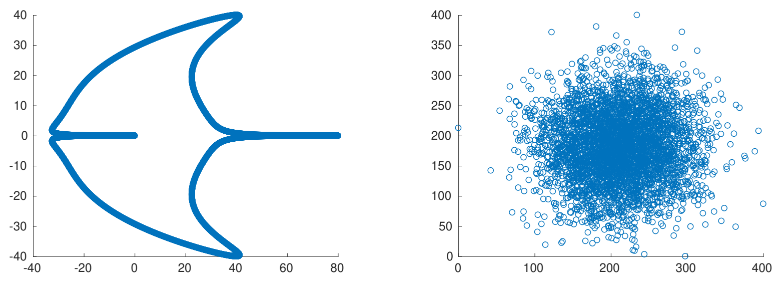
Here, we use some numerical examples to demonstrate the performance of our techniques and support the analysis. We show how our stable FMM/HSS constructions with the scaling strategy control the norms of the generators and the approximation accuracy. We also test the accuracy of a direct solution. Different types of kernels as follows are tested:

$$\kappa_1(x,y) = \begin{cases} \frac{1}{x-y}, & \text{if } x \neq y, \\ 1, & \text{otherwise,} \end{cases} \tag{4.1}$$

$$\kappa_2(x,y) = \begin{cases} \frac{1}{(x-y)^2}, & \text{if } x \neq y, \\ 1, & \text{otherwise,} \end{cases} \qquad \kappa_3(x,y) = \begin{cases} \log|x-y|, & \text{if } x \neq y, \\ 1, & \text{otherwise.} \end{cases} \tag{4.2}$$

To account for factors like the scale and the distribution of point sets, the kernels are evaluated at various 1D and 2D point sets.

- Set s_1: A set of uniform grid points in $[0, 1]$.
- Set s_2: A set of randomly generated points in $[0, 1]$.
- Set s_3: A set of points on the boundary curve of a stingray shape defined by the coordinates
$$\left(40\sin\frac{(2i-1)\pi}{n} + 40\cos^4\frac{2(2i-1)\pi}{n},\ 40\cos^5\frac{(2i-1)\pi}{n}\right), \qquad i = 1, 2, \ldots, n.$$
See Figure 4.1(a) for an illustration.
- Set s_4: A set of uniform grid points in $[0, 400] \times [0, 400]$.
- Set s_5: A set of randomly generated points in $[0, 400] \times [0, 400]$. See Figure 4.1(b) for an example.



(a) s_3 (b) s_5

FIG. 4.1. *Illustration of the points in examples of* s_3 *and* s_5.

To generate a binary tree $\mathcal{T}$ for the FMM/HSS matrix construction, we hierarchically bisect each set. Separated subsets are adaptively identified in the partitioning process.

4.1. Entrywise magnitudes of the generators. We illustrate the benefit of the proposed stable FMM/HSS matrix construction by investigating the entrywise magnitudes of the generators with and without applying the scaling strategy (denoted New and Unscaled in the tests, respectively). According to (3.22) and Corollary 3.4, we just need to report the entrywise magnitudes for the HSS version since the results are almost the same for the FMM case. To inspect how New differs from Unscaled, we report the entrywise magnitudes of the HSS generators of $\hat{A}^{(F)}$ as follows:

$$\mathcal{B} \equiv \max_{\mathbf{i}\in\mathcal{T}} \|\tilde{B}_{\mathbf{i}}\|_{\max}, \qquad \mathcal{U} \equiv \max_{\mathbf{i}\in\mathcal{T}} \|\tilde{U}_{\mathbf{i}}\|_{\max}, \qquad \mathcal{R} \equiv \max_{\mathbf{i}\in\mathcal{T}} \|\tilde{R}_{\mathbf{i}}\|_{\max}. \tag{4.3}$$

Results for the generators $\tilde{V}$ and $\tilde{W}$ are not shown since they are similar to those for $\tilde{U}$ and $\tilde{R}$, respectively.

We pick the number of points in each point set (or the order of A) as $n = 4096$ and set each leaf level partition to include at most 256 points. The separation ration τ in Definition 2.1 is set to $\frac{1}{2}$ for the sets $\mathrm{s}_1, \mathrm{s}_2$ and $\frac{\sqrt{2}}{2}$ for $\mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5$. The number of expansion terms r increases from 5 to 30 to illustrate how the standard Taylor series expansion leads to large entrywise magnitudes of the generators.

For the kernel $\kappa_1(x, y)$ in (4.1), the results for the maximum entrywise magnitudes (4.3) are given in Tables 4.1 and 4.2. As r increases, the maximum entrywise magnitudes of some generators from Unscaled becomes quite large. For some cases, even a small increase in r leads to a rapid increase in the entrywise magnitudes, and the magnitudes become significantly larger than $\|A\|_{\max}$. Such large magnitudes occur for different generators, depending on the point set. On the other hand, New fully resolves this issue and produces generators with uniformly bounded matrix entries regardless of the scale and the distribution of the point sets. That is, all $\mathcal{U}, \mathcal{R}$ are bounded by 1, which is consistent with Corollary 3.4. The $\mathcal{B}$-values are also bounded by modest constants.

TABLE 4.1
Maximum entrywise magnitudes of the HSS generators of $\hat{A}^{(F)}$ obtained with Unscaled and New for $\kappa_1(x, y)$ discretized on the sets $\mathrm{s}_1, \mathrm{s}_2$.

Set	$\|A\|_{\max}$	r	Unscaled		New	
			$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$	$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
s_1	4.10e3	5	1.04e05	1.00	5.33	1.00
		10	6.77e12	1.00	5.33	1.00
		15	7.03e21	1.00	5.33	1.00
		20	4.24e31	1.00	5.33	1.00
		25	9.34e41	1.00	5.33	1.00
		30	5.75e52	1.00	5.33	1.00
s_2	3.60e7	5	1.06e08	1.00	21.3	1.00
		10	7.09e18	1.00	21.3	1.00
		15	7.52e30	1.00	21.3	1.00
		20	4.64e43	1.00	21.3	1.00
		25	1.05e57	1.00	21.3	1.00
		30	6.58e70	1.00	21.3	1.00

TABLE 4.2
Maximum entrywise magnitudes of the HSS generators of $\hat{A}^{(F)}$ obtained with Unscaled *and* New *for $\kappa_1(x, y)$ discretized on the sets $\mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5$.*

Set	$\|A\|_{\max}$	r	Unscaled			New	
			$\mathcal{B}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
$\mathbf{s}_3$	4.71e13	5	2.35e-02	1.49e04	1.96e02	2.35e-02	1.00
		10	2.35e-02	8.59e06	5.49e02	2.69e-02	1.00
		15	2.35e-02	3.12e08	5.49e02	3.00e-02	1.00
		20	2.35e-02	1.95e09	5.49e02	3.20e-02	1.00
		25	2.35e-02	3.34e09	5.49e02	3.32e-02	1.00
		30	2.35e-02	3.34e09	5.49e02	3.42e-02	1.00
$\mathbf{s}_4$	1.00	5	3.59e-03	9.78e05	1.00	3.59e-03	1.00
		10	3.59e-03	1.06e11	1.00	4.39e-03	1.00
		15	3.59e-03	7.19e14	1.00	4.90e-03	1.00
		20	3.59e-03	8.42e17	1.00	5.21e-03	1.00
		25	3.59e-03	2.70e20	1.00	5.42e-03	1.00
		30	3.59e-03	3.09e22	1.00	5.57e-03	1.00
$\mathbf{s}_5$	2.98e1	5	9.30e-03	5.90e06	1.00	9.30e-03	1.00
		10	9.30e-03	6.02e12	1.00	1.09e-02	1.00
		15	9.30e-03	3.86e17	1.00	1.22e-02	1.00
		20	9.30e-03	4.27e21	1.00	1.30e-02	1.00
		25	9.30e-03	1.29e25	1.00	1.35e-02	1.00
		30	9.30e-03	1.40e28	1.00	1.39e-02	1.00

Similar results can also observed for other kernel functions. We repeat some tests with the kernels $\kappa_2(x, y)$ and $\kappa_3(x, y)$ in (4.2). The results are shown in Table 4.3. Again, while some generators from Unscaled have large magnitudes, the generators from New always have well-controlled entrywise magnitudes.

Other than increasing r, another way to demonstrate the advantage of New over Unscaled is to increase the number of points n in a set while keeping the points still within the given interval. In this way, the points get more clustered, and the entries in (2.9) and (2.12) used in Unscaled get larger. For example, for $\kappa_1(x, y)$ discretized on $\mathbf{s}_2$, we fix $r = 20$ and increase n. The $\mathcal{B}$-magnitudes are displayed in Figure 4.2. It can be observed that $\mathcal{B}$ from Unscaled increases quickly with n, while it remains well bounded for New. We can observe similar comparisons for the other sets and kernels.

REMARK 4.1. In practice, even if r is very small (say, smaller than 10), Unscaled may still provide generators with huge entries that pose stability risks. Also, we have used computational domains with different sizes to show that Unscaled is susceptible to the problem setting but New is much more robust.

4.2. Accuracy and efficiency. The large magnitudes of the entries of the generators can cause a loss of accuracy for structured algorithms using the generators. To demonstrate this, we perform some operations for the generators in (3.22). The recompression step mentioned after (3.22) is first applied with full machine precision as tolerance to avoid introducing extra approximation errors. The resulting generators are used for matrix-vector multiplications and linear system solutions via ULV factorizations and solutions. Without the recompression, the unscaled version can often give reasonable accuracies in matrix-vector multiplications. However, it is quite sensitive to more complicated operations such as recompression. In addition, it can encounter overflow for larger ranks.

TABLE 4.3
Maximum entrywise magnitudes of the HSS generators of $\hat{A}^{(F)}$ obtained with Unscaled *and* New *for the kernels in* (4.2) *discretized on the sets* $\mathrm{s}_2, \mathrm{s}_5$.

Kernel	Set	$\|A\|_{\max}$	r	Unscaled			New	
$\kappa_2(x,y)$	s_2	1.30e15		$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$		$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
			5	1.13e10	1.00		5.63e02	1.00
			10	1.51e21	1.00		7.41e02	1.00
			15	2.41e33	1.00		8.28e02	1.00
			20	1.98e46	1.00		8.80e02	1.00
			25	5.57e59	1.00		9.16e02	1.00
			30	4.21e73	1.00		9.41e02	1.00
	s_5	8.90e02		$\mathcal{B}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
			5	8.65e-05	5.90e06	1.00	2.22e-04	1.00
			10	8.65e-05	6.02e12	1.00	2.93e-04	1.00
			15	8.65e-05	3.86e17	1.00	3.33e-04	1.00
			20	8.65e-05	4.27e21	1.00	3.65e-04	1.00
			25	8.65e-05	1.29e25	1.00	3.87e-04	1.00
			30	8.65e-05	1.40e28	1.00	4.03e-04	1.00
$\kappa_3(x,y)$	s_2	1.74e01		$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$		$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
			5	1.24e06	1.00		3.06	1.00
			10	3.69e16	1.00		3.06	1.00
			15	2.52e28	1.00		3.06	1.00
			20	1.14e41	1.00		3.06	1.00
			25	2.04e54	1.00		3.06	1.00
			30	1.06e68	1.00		3.06	1.00
	s_5	8.90e02		$\mathcal{B}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\max\{\mathcal{U},\mathcal{R}\}$
			5	5.76	5.90e06	1.00	5.76	1.00
			10	5.76	6.02e12	1.00	5.76	1.00
			15	5.76	3.86e17	1.00	5.76	1.00
			20	5.76	4.27e21	1.00	5.76	1.00
			25	5.76	1.29e25	1.00	5.76	1.00
			30	5.76	1.40e28	1.00	5.76	1.00

For each matrix-vector multiplication, we generate a random vector w and multiply the approximate matrix with w to get a vector $\hat{b}$, which approximates the exact vector $b = Aw$. For $\kappa_1(x,y)$ discretized on the point sets as above, the resulting matrix-vector multiplication errors $\frac{\|\hat{b}-b\|_1}{\|b\|_1}$ are given in Table 4.4. In exact arithmetic, when r increases, the approximate matrix gets more accurate, and the error $\frac{\|\hat{b}-b\|_1}{\|b\|_1}$ should decrease. However, with Unscaled, only modest accuracies are achieved. Specifically for the sets $\mathrm{s}_1, \mathrm{s}_2$, the accuracy in Table 4.4 does not improve much for increasing r. For the sets $\mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5$, the accuracy in Table 4.4 initially improves with increasing r but then decreases. On the other hand, such situations do not occur with New. For all the sets, the accuracy increases with r to near the machine precision. For the kernels $\kappa_2(x,y)$ and $\kappa_3(x,y)$, the results are given in Table 4.5.

We then fix $r = 20$ and increase n. Figure 4.3(a) displays the relative errors of the matrix-vector multiplications for one case. Much higher accuracies are achieved for all n with New than with Unscaled.

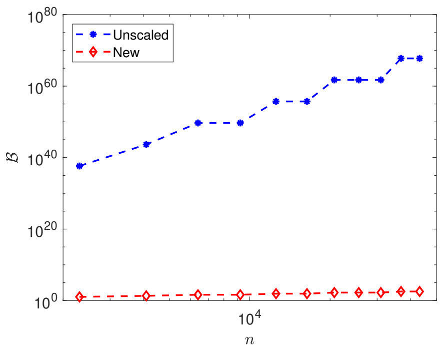


FIG. 4.2. *Maximum entrywise magnitude $\mathcal{B}$ in (4.3) from* Unscaled *and* New *for $\hat{A}^{(F)}$ with $\kappa_1(x,y)$ discretized on $\mathbf{s}_2$ of different sizes n.*

TABLE 4.4
$\frac{\|\hat{b}-b\|_1}{\|b\|_1}$*: accuracy of matrix-vector multiplications based on* Unscaled *and* New *with the kernel $\kappa_1(x,y)$.*

	r	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$
Unscaled	5	7.03e-06	1.34e-06	4.13e-13	1.11e-04	1.20e-04
	10	8.30e-08	5.55e-08	1.50e-10	6.32e-07	6.93e-07
	15	8.35e-08	5.50e-08	1.05e-08	7.18e-09	1.98e-01
	20	8.47e-08	5.49e-08	7.24e-06	2.95e-01	2.62e-01
	25	7.61e-08	5.49e-08	9.78e-02	3.21e-01	2.91e-01
	30	7.65e-08	5.49e-08	2.13e-01	3.25e-01	2.99e-01
New	5	7.03e-06	1.33e-06	6.84e-14	1.11e-04	1.20e-04
	10	1.14e-08	2.67e-09	1.78e-14	6.32e-07	6.93e-07
	15	2.72e-11	5.55e-12	3.26e-14	7.18e-09	7.20e-09
	20	7.84e-14	1.81e-14	3.17e-14	9.93e-11	1.08e-10
	25	1.62e-15	1.83e-15	2.16e-14	1.76e-12	1.92e-12
	30	1.54e-15	1.81e-15	4.60e-14	3.49e-14	4.17e-14

We can similarly compare the accuracy for solving linear system by ULV factorization and ULV solution. We form the right-hand side vector $b = Aw$ with a random vector w and suppose that $\hat{w}$ is the approximate solution. For $\kappa_1(x,y)$ discretized on the five sets as above, Table 4.6 gives the relative residuals $\frac{\|A\hat{w}-b\|_1}{\|b\|_1}$. With Unscaled, only modest accuracies can be achieved for some cases and very inaccurate results are produced for the other cases. With New, the relative residuals reduce with increasing r to near the machine precision.

Similarly, with $r = 20$ and varying n, the accuracy results for one test is given in Figure 4.3(b). While the accuracy with Unscaled remains modest and gets worse with increasing n, the accuracy with New stays high for all the values of n.

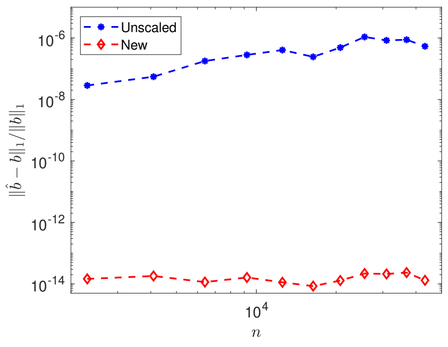
Finally, it is convenient to verify the efficiency of some relevant structured algorithms. Such efficiency studies have been done extensively in the existing literature. Here, we just use Figure 4.4 with $r = 20$ to present the storage needed for the generators for $\hat{A}^{(F)}$, which essentially reflects the cost needed to multiply $\hat{A}^{(F)}$ with a vector. The storage in Figure 4.4 is roughly linear in n.

5. Conclusions.

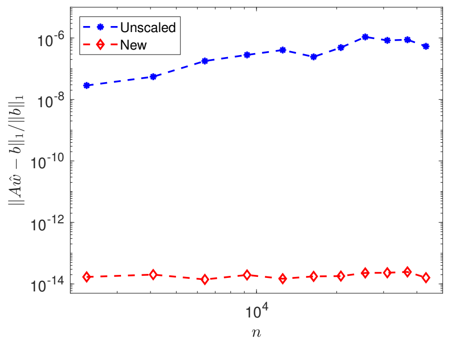
In this paper, stabilization strategies and backward stability studies are given for relevant low-rank approximations and translation relations in an intuitive matrix

TABLE 4.5
$\frac{\|\hat{b}-b\|_1}{\|b\|_1}$: *accuracy of matrix-vector multiplications based on* Unscaled *and* New *with the kernels* $\kappa_2(x,y)$ *and* $\kappa_3(x,y)$.

	r	$\kappa_2(x,y)$		$\kappa_3(x,y)$	
		$\mathbf{s}_2$	$\mathbf{s}_5$	$\mathbf{s}_2$	$\mathbf{s}_5$
Unscaled	5	2.66e-11	4.40e-06	1.11e-04	2.11e-05
	10	1.26e-12	4.76e-08	2.80e-06	6.07e-08
	15	1.26e-12	1.55e-02	2.80e-06	5.43e-01
	20	1.26e-12	1.94e-02	2.80e-06	6.67e-01
	25	1.26e-12	1.97e-02	2.80e-06	7.14e-01
	30	1.26e-12	2.03e-02	2.80e-06	6.71e-01
New	5	2.66e-11	4.40e-06	1.11e-04	2.11e-05
	10	7.88e-14	4.76e-08	9.99e-08	6.07e-08
	15	2.14e-15	7.26e-10	1.71e-10	4.22e-10
	20	1.89e-15	1.45e-11	3.60e-13	4.98e-12
	25	1.89e-15	3.16e-13	3.97e-15	6.22e-14
	30	1.89e-15	9.29e-15	4.00e-15	4.14e-15



(a) $\frac{\|\hat{b}-b\|_1}{\|b\|_1}$ for matrix-vector multiplications.



(b) $\frac{\|A\hat{w}-b\|_1}{\|b\|_1}$ for linear system solutions.

FIG. 4.3. *Accuracies of matrix-vector multiplications and linear system solutions based on* Unscaled *and* New *with the kernel* $\kappa_1(x,y)$ *discretized on* $\mathbf{s}_2$ *for different sizes* n.

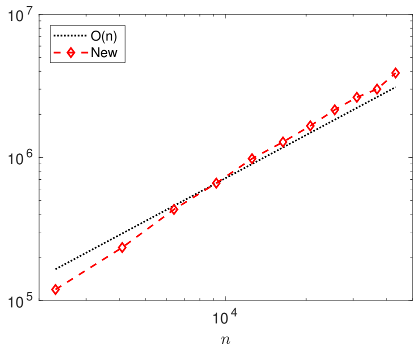


FIG. 4.4. *Storage costs (number of nonzero entries) for the* $\tilde{D}, \tilde{U}, \tilde{V}, \tilde{R}, \tilde{W}, \tilde{B}$ *generators for* $\hat{A}^{(F)}$ *from* New *for* $\kappa_1(x,y)$ *discretized on* $\mathbf{s}_2$ *for different numbers of points* n.

TABLE 4.6
Residuals of ULV solutions after ULV factorizations based on New *with the kernel* $\kappa_1(x, y)$.

	r	$\mathbf{s}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$
Unscaled	5	7.03e-06	1.34e-06	4.75e-10	8.85e-04	2.54e-04
	10	8.30e-08	5.55e-08	3.62e-07	2.47e-06	1.41e-06
	15	8.35e-08	5.50e-08	4.40e-04	1.42e-08	5.34e+02
	20	8.47e-08	5.49e-08	9.20e+01	1.32e+01	4.13e+02
	25	7.61e-08	5.49e-08	3.08e+10	1.36e+01	2.59e+02
	30	7.65e-08	5.49e-08	6.73e+08	2.55e+01	8.92e+03
New	5	7.03e-06	1.33e-06	4.79e-09	8.85e-04	2.54e-04
	10	1.14e-08	2.67e-09	2.10e-12	2.47e-06	1.41e-06
	15	2.72e-11	5.55e-12	1.04e-12	1.42e-08	1.82e-08
	20	7.96e-14	2.01e-14	2.33e-13	2.53e-10	2.22e-10
	25	4.41e-15	6.77e-15	2.93e-13	2.61e-12	2.13e-12
	30	4.90e-15	6.49e-15	4.23e-13	4.83e-14	8.30e-14

version of the FMM. An FMM matrix example is also presented, followed by ideas to convert the FMM matrix into an HSS form that admits stable factorizations. The stable matrix version FMM employs a scaling strategy to revise the low-rank approximations based on Taylor expansions for some kernel functions. Rigorous norm bounds are provided for the FMM and HSS generators. These bounds lead to backward stability of fast matrix-vector multiplications with these matrices. The HSS form can be used for stably solving linear systems via an ULV factorization.

Since the approximation based on Taylor expansions can be substituted by other approximations such as polynomial interpolations [11, 16, 38], numerical integrations [1, 39], and kernel-independent FMM [23, 40, 41], we expect that our ideas can also be generalized to various other types of FMM. Our stabilization strategies are derived based on 2D point sets but can also be extended to higher dimensions. It is convenient to generalize the norm bounds and the stability analysis in Sections 2.5 and 3.6. Although we only give the FMM matrix using one-dimensional sets as an example, the essential ideas can be directly modified for higher dimensions. Some details will appear in [25].

REFERENCES

[1] C. R. ANDERSON, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 923–947.
[2] D. CAI, E. CHOW, L. ERLANDSON, Y. SAAD, AND Y. XI, *SMASH: structured matrix approximation by separation and hierarchy*, Numer. Linear Algebra Appl., 25 (2018), Art. e2204, 25 pages.
[3] D. CAI AND J. XIA, *A stable and efficient matrix version of the fast multipole method*, Preprint, Purdue University, West Lafayette. https://www.math.purdue.edu/~cai92/fmm2hss.pdf
[4] S. CHANDRASEKARAN, P. DEWILDE, M. GU, W. LYONS, AND T. PALS, *A fast solver for HSS representations via sparse matrices*, SIAM J. Matrix Anal. Appl., 29 (2006/07), pp. 67–81.
[5] S. CHANDRASEKARAN, M. GU, AND T. PALS, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 603–622.
[6] S. CHANDRASEKARAN, M. GU, X. SUN, J. XIA, AND J. ZHU, *A superfast algorithm for Toeplitz systems of linear equations*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1247–1266.
[7] E. CORONA, P.-G. MARTINSSON, AND D. ZORIN, *An $o(n)$ direct solver for integral equations on the plane*, Appl. Comput. Harmon. Anal., 38 (2015), pp. 284–317.
[8] E. DARVE, *The fast multipole method: numerical implementation*, J. Comput. Phys., 160 (2000), pp. 195–240.
[9] E. DARVE AND P. HAVÉ, *Efficient fast multipole method for low-frequency scattering*, J. Comput. Phys., 197 (2004), pp. 341–363.

[10] ———, *A fast multipole method for Maxwell equations stable at all frequencies*, Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 362 (2004), pp. 603–628.
[11] A. Dutt, M. Gu, and V. Rokhlin, *Fast algorithms for polynomial interpolation, integration, and differentiation*, SIAM J. Numer. Anal., 33 (1996), pp. 1689–1711.
[12] A. Gillman, *Fast Direct Solvers for Elliptic Partial Differential Equations*, PhD. Thesis, Department of Applied Mathematics, University of Colorado, Boulder, 2011.
[13] Z. Gimbutas, N. F. Marshall, and V. Rokhlin, *A fast simple algorithm for computing the potential of charges on a line*, Appl. Comput. Harmon. Anal., 49 (2020), pp. 815–830.
[14] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
[15] ———, *On the efficient implementation of the fast multipole algorithm*, Technical Report RR-602, Department of Computer Science, Yale University, New Haven, 1988.
[16] W. Hackbusch and S. Börm, *H^2-matrix approximation of integral operators by interpolation*, Appl. Numer. Math., 43 (2002), pp. 129–143.
[17] ———, *Data-sparse approximation by adaptive H^2-matrices*, Computing, 69 (2002), pp. 1–35.
[18] W. Hackbusch, B. N. Khoromskij, and R. Kriemann, *Hierarchical matrices based on a weak admissibility criterion*, Computing, 73 (2004), pp. 207–243.
[19] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter, *On $\mathcal{H}^2$-matrices*, in Lectures on Applied Mathematics, H.-J. Bungartz, R. H. W. Hoppe, and C. Zenger, eds., Springer, Berlin, 2000, pp. 9–29.
[20] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[21] 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.
[22] 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.
[23] P. G. Martinsson and V. Rokhlin, *An accelerated kernel-independent fast multipole method in one dimension*, SIAM J. Sci. Comput., 29 (2007), pp. 1160–1178.
[24] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.
[25] X. Ou and J. Xia, *A stable matrix version of the fast multipole method: the 2d version with stability analysis*, Preprint.
[26] ———, *Superdc: stable superfast divide-and-conquer eigenvalue decomposition*, Preprint on arXiv, 2021. https://arxiv.org/abs/2108.04209
[27] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.
[28] F. H. Rouet, X. S. Li, P. Ghysels, and A. Napov, *A distributed-memory package for dense hierarchically semi-separable matrix computations using randomization*, ACM Trans. Math. Software, 42 (2016), Art. 27, 35 pages.
[29] J. Shen, Y. Wang, and J. Xia, *Fast structured direct spectral methods for differential equations with variable coefficients, I. The one-dimensional case*, SIAM J. Sci. Comput., 38 (2016), pp. A28–A54.
[30] X. Sun and N. P. Pitsianis, *A matrix version of the fast multipole method*, SIAM Rev., 43 (2001), pp. 289–300.
[31] J. Vogel, J. Xia, S. Cauley, and V. Balakrishnan, *Superfast divide-and-conquer method and perturbation analysis for structured eigenvalue solutions*, SIAM J. Sci. Comput., 38 (2016), pp. A1358–A1382.
[32] 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.
[33] 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.
[34] J. Xia, *Fast Direct Solvers for Structured Linear Systems of Equations*, PhD. Thesis, University of California, Berkeley, 2006.
[35] ———, *On the complexity of some hierarchical structured matrix algorithms*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 388–410.
[36] 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.
[37] 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.
[38] N. Yarvin and V. Rokhlin, *A generalized one-dimensional fast multipole method with application to filtering of spherical harmonics*, J. Comput. Phys., 147 (1998), pp. 594–609.
[39] 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.
[40] L. Ying, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457.

[41] 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.