A STABLE MATRIX VERSION OF THE 2D FAST MULTIPOLE METHOD*

XIAOFENG OU†, MICHELLE MICHELLE†, AND JIANLIN XIA†

Abstract. The fast multipole method (FMM) is a powerful method for accelerating some kernel matrix-vector multiplications. In this paper, we show an intuitive matrix version of the FMM in two dimensions via degenerate Taylor series expansions and, furthermore, give a simple stabilization strategy to balance relevant low-rank factors so that the factors and some translation operators satisfy certain norm bounds. Based on these, we provide the long-overdue backward stability analysis for the FMM. The matrix version FMM translates the original FMM terminology into simple matrix language with the aim of being more accessible to non-experts. It further makes it convenient to perform the backward stability analysis. The stabilization strategy leads to entrywise backward errors that depend only logarithmically on the matrix size, which shows the superior stability benefit of the FMM on top of its efficiency advantage as compared with usual dense matrix-vector multiplications.

Key words. fast multipole method, kernel matrix, backward stability, low-rank approximation, degenerate expansion, translation relation

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

1. Introduction. Given a kernel function $\kappa(x, y)$ and two set of points $\mathbf{X} = \{x_i\}_{i=1}^M$ and $\mathbf{Y} = \{y_i\}_{i=1}^N$ in the complex plane $\mathbb{C}$, consider the matrix-vector multiplication

$$\phi = Kq, \quad K := [\kappa(x_i, y_j)]_{x_i \in \mathbf{X}, y_j \in \mathbf{Y}}, \tag{1.1}$$

where the matrix K is the *kernel matrix* or *interaction matrix*. Such kernel matrix-vector products appear frequently in scientific computing, engineering, and data analysis. For some kernels, the fast multipole method (FMM) provides an efficient way to evaluate (1.1) to any given accuracy with $O(M + N)$ complexity [14, 27]. The FMM has been widely used in accelerating many different types of numerical computations, such as N-body simulations, integral and differential equation solutions, Gaussian processes, machine learning, as well as classical linear algebra problems like some matrix transforms and eigenvalue solutions. See [8, 22, 25, 26, 27, 29, 30, 36] and the references therein for some examples.

The FMM essentially constructs a rank-structured *FMM matrix* approximation to the kernel matrix K (see, e.g., [28]), which is also known as the hierarchically structured $\mathcal{H}^2$-matrix [18, 19]. The FMM matrix is constructed via some *degenerate* or *separable expansions* of the $\kappa(x, y)$ to obtain low-rank approximations of certain off-diagonal blocks of K. Some commonly used separable expansions include multipole expansions, Taylor expansions, spherical harmonic expansions, Chebyshev interpolations, and some kernel independent strategies. See, e.g., [10, 11, 14, 15, 23, 27, 28, 32, 37].

There exist many implementations of the FMM algorithm which can usually achieve satisfactory accuracy. Nevertheless, it has been previously noticed that stability risks may arise in some separable expansions for constructing the FMM approximations [6, 7, 10, 15]. Some possible causes include fast-growing coefficients in

*The research of Michelle Michelle was supported in part by NSERC Postdoctoral Fellowship. The research of Jianlin Xia was supported in part by NSF grants DMS-1819166 and DMS-2111007. Xiaofeng Ou's work was part of a PhD thesis at Purdue University.

†Department of Mathematics, Purdue University, West Lafayette, IN 47907 (ou17@purdue.edu, mmichell@purdue.edu, xiaj@purdue.edu).

the expansions, products of enormous and minuscule numbers, and extremely large entries in the low-rank approximations. These issues may cause the algorithm to lose accuracy or even break down in certain circumstances. Some heuristic scaling methods to deal with such issues have been discussed in [7, 15] without justification. In [16], for the three-dimensional Laplace equation, stability issues can be resolved via the use of multipole expansions along with plane wave or exponential expansions. More recently, the authors in [6], a rigorous algebraic scaling strategy is proposed to stabilize low-rank approximations obtained from Taylor expansions for some kernels. The scaling strategy is based on Stirling's formula to design some scaling factors and it is challenging to implement the scaled low-rank approximations efficiently. The justification of the effectiveness is also nontrivial. The work in [6] additionally includes a matrix version of the FMM that is believed to be stable, but only in terms of one dimension and without the full stability analysis.

In fact, a general analysis of the backward stability of the FMM has long been overdue, likely due to its complicated nature. An earlier attempt has been made in [33] for a highly simplified one-dimensional (1D) structure, the hierarchically semiseparable (HSS) form [35]. It is also limited to the case where relevant off-diagonal basis matrices are orthogonal, so it is not applicable to the usual FMM.

In response to these limitations, the present paper intends to make the following main contributions. Firstly, we show how to derive stable degenerate expansions directly from Taylor expansions without the need of extra tools like Stirling's formula. The method we use may be viewed as an implicit balancing strategy that can effectively bound appropriate norms ($\|\cdot\|_{\max}$, $\|\cdot\|_1$, or $\|\cdot\|_{1,1}$) of the resulting low-rank factors and translation operators in the FMM. The entries of the factors can be computed in a stable and efficient manner by means of recurrence relations, which is another benefit over the stabilization strategy in [6]. For convenience, the stable degenerate expansions are discussed in terms of a class of 2D generalized Cauchy kernels $\kappa(x,y) = \frac{1}{(x-y)^{1+d}}$ and the 2D logarithmic kernel $\kappa(x,y) = \log\frac{1}{|x-y|}$ as examples. Our analysis and method are also applicable to other non-oscillating kernels where degenerate expansions can be derived via Taylor expansions.

Next, to facilitate our backward stability analysis, we provide a quick intuitive exposition of a matrix version of the 2D FMM, where we translate some FMM terminology into simple matrix language for convenient understanding. The FMM, though very successful, has not been widely accessible to some communities, probably due to the need to understand techniques like multiple expansions, local expansions, and shifting of centers. Here, we make the discussions mainly in terms of matrix factors that are convenient to digest by non-experts. In particular, the use of *basis contributions* avoids the need to distinguish multipole and local expansions.

Thirdly, we rigorously study the stability of the 2D FMM algorithm and provide the long-overdue backward stability analysis. It is worth pointing out FMM algorithms (including our matrix version) usually produce basis matrices that do not have orthonormal columns. This makes it crucial to inspect the backward stability. Here, with our stable low-rank approximations, we can show that the backward error only depends logarithmically on the size of the kernel matrix K instead of linearly as in the case for standard dense matrix-vector multiplications. This fact highlights the stability benefit of hierarchical structured matrix methods over classical matrix methods.

The organization of this paper is as follows. In section 2, we discuss stable degenerate expansions for some kernels, relevant norm bounds for low-rank factors, and

recurrence formulas for quickly and stably computing the low-rank factors. In section 3, stable translation relations of the FMM are shown. In section 4, we interpret standard FMM terminology in simple matrix language and summarize the framework of the matrix version FMM. A rigorous backward stability analysis of the FMM is then given in section 5. To validate our theoretical findings, some numerical experiments are presented in section 6. A collection of notation is given below to facilitate the presentation.

- For an $m \times n$ matrix $A := [a_{ij}]_{1\le i\le m, 1\le j\le n}$, denote

$$\|A\|_{\max} := \max_{1\le i\le m,1\le j\le n} |a_{ij}|, \quad \|A\|_{1,1} := \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|,$$

$$\|A\|_1 := \max_{1\le j\le n} \left(\sum_{i=1}^{m} |a_{ij}|\right), \quad \|A\|_\infty := \max_{1\le i\le m} \left(\sum_{j=1}^{n} |a_{ij}|\right).$$

- For an $m \times n$ matrix A, denote $|A| := [|a_{ij}|]_{1\le i\le m,1\le j\le n}$. If both A and B are $m \times n$, $|A| \le |B|$ is understood as $|a_{ij}| \le |b_{ij}|$ for all $1 \le i \le m$ and $1 \le j \le n$.
- $\mathrm{diag}(\cdot)$ denotes a (block) diagonal matrix.
- $\Re(z)$ and $\Im(z)$ denote the real and imaginary parts of a complex number z, respectively.
- $\binom{\alpha}{n} := \frac{\alpha(\alpha-1)\cdots(\alpha-n+1)}{n!}$ is the generalized binomial coefficient with $\alpha \in \mathbb{C}$.
- $\prod_{i=k}^{j} A_i := A_k A_{k-1} \cdots A_j$ if $j \le k$, where A_i, $i = j, \ldots, k$, are matrices of appropriate sizes. Meanwhile, if $k < j$, then $\prod_{i=k}^{j} A_i := I$, where I is the identity matrix.
- Unless otherwise stated, define the multi-index notation

$$\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_n), \quad \alpha_i \in \{0, 1\}, \quad 1 \le i \le n, \quad n \in \mathbb{N}, \tag{1.2}$$

  and the sum of its components $|\boldsymbol{\alpha}| := \alpha_1 + \cdots + \alpha_n$.
- For $k \ge j$, define

$$\left(\prod_{i=k}^{j} A_i\right)^{\boldsymbol{\alpha}} := A_k^{\alpha_1} \ldots A_j^{\alpha_{k-j+1}}, \tag{1.3}$$

$$\Delta^{\boldsymbol{\alpha}} \left(\prod_{i=k}^{j} A_i\right) := \left(\Delta^{\alpha_1} A_k\right) \ldots \left(\Delta^{\alpha_{k-j+1}} A_j\right), \tag{1.4}$$

  where $\Delta^0 A_i := A_i$ and $\Delta^1 A_i := \Delta A_i$ for $j \le i \le k$.

2. Stable degenerate kernel expansions and low-rank approximations. In this section, we show how to derive stable degenerate expansions that satisfies some norm bounds so as to ensure the stability of the FMM. To facilitate the presentation, we use two important kernels as examples, the generalized Cauchy kernel $\kappa(x, y) = \frac{1}{(x-y)^{1+d}}$, $d \in \mathbb{C}$ and the logarithmic kernel $\kappa(x, y) = \log \frac{1}{|x-y|}$. Then generalizations to other kernels will be discussed.

Consider the evaluation of $\kappa(x, y)$ at two *well-separated* sets of points $\mathbf{x} \subset \mathbf{X}$ and $\mathbf{y} \subset \mathbf{Y}$. Here, we follow the way used in [6, 28] to define well-separated sets $\mathbf{x}$ and $\mathbf{y}$. That is, there is a constant τ called the separation ratio such that

$$\frac{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}{|o_{\mathbf{x}} - o_{\mathbf{y}}|} \le \tau, \quad \tau \in (0, 1), \tag{2.1}$$

where $\delta_{\mathbf{x}}$ and $o_{\mathbf{x}}$ are respectively the radius and the center of a circle that encloses the set $\mathbf{x}$ and $\delta_{\mathbf{y}}$ and $o_{\mathbf{y}}$ can be understood in the same way. This circle may not be unique, but this does not affect the discussions.

We are interested in establishing a degenerate expansion of the form

$$\kappa(x,y) = \sum_{i,j=0}^{r-1} b_{i,j} u_i(x) v_j(y) + e_r, \quad x \in \mathbf{x},\ y \in \mathbf{y}, \tag{2.2}$$

where r is a small positive integer and e_r is the remainder. Suppose $m = |\mathbf{x}|$, $n = |\mathbf{y}|$. Then the corresponding $m \times n$ kernel matrix has a low-rank approximation UBV^{T} as in

$$K_{\mathbf{x},\mathbf{y}} := [\kappa(x_i, y_j)]_{x_i \in \mathbf{x}, y_j \in \mathbf{y}} = UBV^{\mathsf{T}} + E_r, \tag{2.3}$$

where U, B, V are $m \times r$, $r \times r$, and $n \times r$ matrices, respectively, and E_r is the error matrix. The expansion and the remainder will be studied in detail below.

2.1. Generalized Cauchy kernel. We first look at the stable degenerate expansion for the generalized Cauchy kernel $\kappa(x,y) = \frac{1}{(x-y)^{1+d}}$, where $d \in \mathbb{Z}$ (which may be relaxed to $d \in \mathbb{C}$ for some cases as in subsection 2.3). The results generalize those in [6] for the standard Cauchy kernel and further provide an improved stabilization strategy. The stabilization makes sure that the magnitudes of U, B, V in (2.3) satisfy certain bounds.

Our study of the degenerate expansion involves the remainder in a form of Newton's generalized binomial expansion with complex exponents (see, e.g., [13]). The following lemma gives a specific bound for the remainder. To the best of our knowledge, we are not aware of such a bound. The discussion utilizes the generalized binomial coefficient $\binom{\alpha}{n}$ with $\alpha \in \mathbb{C}$.

LEMMA 2.1. *Let* $F(s) = \frac{1}{(1-s)^{1+d}}$ *with* $s, d \in \mathbb{C}$, $|s| < 1$, *and*

$$\mathcal{R}_r(s) = F(s) - \sum_{p=0}^{r-1} \binom{p+d}{p} s^p = \sum_{p=r}^{\infty} \binom{p+d}{p} s^p.$$

Then for any given $\nu > 0$ *with* $\mu := (1+\nu)|s| < 1$,

$$|\mathcal{R}_r(s)| \le \frac{\left|\binom{N_0+d}{N_0}\right|}{(1+\nu)^{N_0}} \left(\frac{\mu^r}{1-\mu}\right) \quad \textit{for all} \quad r \ge N_0 := \left\lceil \frac{|d|}{\nu} \right\rceil.$$

Proof. Let $a_p(s) = \binom{p+d}{p} s^p$. If $p \ge N_0 = \lceil |d|/\nu \rceil$, then

$$\left| \frac{a_{p+1}(s)}{a_p(s)} \right| = \left| \left(1 + \frac{d}{p+1}\right) s \right| \le (1+\nu)|s| = \mu.$$

This implies that $|a_p(s)| \le \left| a_{N_0}(s) \mu^{p-N_0} \right|$. As a result, for $r \ge N_0$,

$$|\mathcal{R}_r(s)| \le \sum_{p=r}^{\infty} |a_p(s)| \le \sum_{p=r}^{\infty} \left| a_{N_0}(s) \mu^{p-N_0} \right| \le \frac{\left|\binom{N_0+d}{N_0}\right|}{(1+\nu)^{N_0}} \frac{\mu^r}{1-\mu}.$$

∎

We now present our first set of main results.

THEOREM 2.2. *Let* $\kappa(x,y) = \frac{1}{(x-y)^{1+d}}, d \in \mathbb{Z}$. *Suppose* $\mathbf{x} = \{x_i\}_{i=1}^n$ *and* $\mathbf{y} = \{y_i\}_{i=1}^n$ *are well separated with separation ratio* τ. *Then the low-rank approximation to the kernel matrix* $K_{\mathbf{x},\mathbf{y}}$ *in* (2.3) *has the following form:*

$$U := \left[\left(\frac{x_i - o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^j\right]_{1\le i\le m, 0\le j\le r-1}, \quad V := \left[\left(\frac{y_i - o_{\mathbf{y}}}{\delta_{\mathbf{y}}}\right)^j\right]_{1\le i\le n, 0\le j\le r-1}, \tag{2.4}$$

$$B := [b_{i,j}]_{0\le i,j\le r-1} \quad \textit{with} \tag{2.5}$$

$$b_{i,j} := \begin{cases} \frac{(-1)^i}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}} \binom{i+j+d}{i+j}\binom{i+j}{i} \left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}}-o_{\mathbf{y}}}\right)^i \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}}-o_{\mathbf{y}}}\right)^j, & 0 \le i+j \le r-1, \\ 0, & i+j > r-1. \end{cases}$$

Moreover, for any given $\nu > 0$ *such that*

$$\mu := (1+\nu) \max_{x\in\mathbf{x}, y\in\mathbf{y}} \left|\frac{(x-o_{\mathbf{x}}) - (y-o_{\mathbf{y}})}{o_{\mathbf{y}} - o_{\mathbf{x}}}\right| < 1, \tag{2.6}$$

we have

$$|E_r| \le \frac{\left|\binom{N_0+d}{N_0}\right|}{(1+\nu)^{N_0}} \left(\frac{\mu^r}{(1-\mu)(1-\tau)^{|1+d|}}\right) |K| \quad \textit{for} \quad r \ge N_0 := \left\lceil \frac{|d|}{\nu} \right\rceil. \tag{2.7}$$

Proof. Let $x \in \mathbf{x}$, $y \in \mathbf{y}$, and

$$s = \frac{(x-o_{\mathbf{x}}) - (y-o_{\mathbf{y}})}{o_{\mathbf{y}} - o_{\mathbf{x}}} \quad \text{or} \quad x - y = (o_{\mathbf{x}} - o_{\mathbf{y}})(1-s). \tag{2.8}$$

Then

$$(x-y)^{1+d} = (o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}(1-s)^{1+d}, \quad d \in \mathbb{Z}. \tag{2.9}$$

Note that (2.9) does not generally hold if $d \notin \mathbb{Z}$. Since $\mathbf{x}$ and $\mathbf{y}$ are well separated with separation ratio τ, we have $|s| \le \frac{\delta_{\mathbf{x}}+\delta_{\mathbf{y}}}{|o_{\mathbf{x}}-o_{\mathbf{y}}|} \le \tau < 1$. Define $\varepsilon_r(s) := \sum_{j=r}^{\infty} \binom{j+d}{j} s^j$. By Taylor and binomial expansions,

$$\begin{aligned}
&\frac{1}{(x-y)^{1+d}} = \frac{1}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}(1-s)^{1+d}} \\
&= \frac{1}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}} \sum_{k=0}^{r-1} \binom{k+d}{k} s^k + \frac{\varepsilon_r(s)}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}} \\
&= \sum_{k=0}^{r-1} \binom{k+d}{k} \sum_{i=0}^{k} \binom{k}{i} (-1)^i \frac{(x-o_{\mathbf{x}})^i (y-o_{\mathbf{y}})^{k-i}}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d+k}} + \frac{\varepsilon_r(s)}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}} \\
&= \sum_{k=0}^{r-1} \sum_{i=0}^{k} \binom{k+d}{k}\binom{k}{i} (-1)^i \frac{(x-o_{\mathbf{x}})^i (y-o_{\mathbf{y}})^{k-i}}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d+k}} + \frac{\varepsilon_r(s)}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}} \\
&= \sum_{k=0}^{r-1} \sum_{i=0}^{k} b_{i,k-i} \left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^i \left(\frac{y-o_{\mathbf{y}}}{\delta_{\mathbf{y}}}\right)^{k-i} + \frac{\varepsilon_r(s)}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{1+d}}.
\end{aligned}$$

This leads to (2.3).

From (2.8), we have

$$
\begin{aligned}
&|o_{\mathbf{x}} - o_{\mathbf{y}}| = (1-s)^{-1}|x-y| \le (1-\tau)^{-1}|x-y|, \\
&|o_{\mathbf{x}} - o_{\mathbf{y}}| \ge (1+\tau)^{-1}|x-y| \ge (1-\tau)|x-y|.
\end{aligned} \tag{2.10}
$$

Thus,

$$
|o_{\mathbf{x}} - o_{\mathbf{y}}|^{1+d} \ge (1-\tau)^{|1+d|}|x-y|^{1+d}, \quad d \in \mathbb{Z}. \tag{2.11}
$$

Pick any $\nu > 0$ such that (2.6) holds. Then, (2.11) and Lemma 2.1 imply, for $r \ge N_0 := \left\lceil \frac{|d|}{\nu} \right\rceil$,

$$
\left| \frac{\varepsilon_r(s)}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}} \right| \le \frac{\left|\binom{N_0+d}{N_0}\right|}{(1+\nu)^{N_0}} \left(\frac{\mu^r}{1-\mu}\right) \frac{1}{(1-\tau)^{|1+d|}|x-y|^{1+d}}, \tag{2.12}
$$

which gives us (2.7). ∎

Note that the usual way of applying Taylor expansions may produce a low-rank approximation $\widehat{U}\widehat{B}\widehat{V}^{\mathsf{T}}$, where

$$
\widehat{U} := \left[\frac{(x_i - o_{\mathbf{x}})^j}{j!} \right]_{1 \le i \le n, 0 \le j \le r-1}, \quad \widehat{V} := \left[\frac{(y_i - o_{\mathbf{y}})^j}{j!} \right]_{1 \le i \le m, 0 \le j \le r-1}, \tag{2.13}
$$

$$
\widehat{B} = [\widehat{b}_{i,j}]_{0 \le i,j \le r-1} := \begin{cases} \frac{(-1)^i}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d+i+j}} \frac{(i+j+d)!}{d!}, & 0 \le i+j \le r-1, \\ 0, & i+j > r-1. \end{cases} \tag{2.14}
$$

If we define

$$
\Lambda := \operatorname{diag}\left(1, \frac{1}{\delta_{\mathbf{x}}}, \ldots, \frac{(r-1)!}{\delta_{\mathbf{x}}^{r-1}}\right) \quad \text{and} \quad \Omega := \operatorname{diag}\left(1, \frac{1}{\delta_{\mathbf{y}}}, \ldots, \frac{(r-1)!}{\delta_{\mathbf{y}}^{r-1}}\right),
$$

then U, B, V in Theorem 2.2 may be written as

$$
U = \widehat{U}\Lambda, \quad B = \Lambda^{-1}\widehat{B}\Omega^{-1}, \quad \text{and} \quad V = \widehat{V}\Omega. \tag{2.15}
$$

Hence, we can think of our low-rank approximation UBV^{T} in Theorem 2.2 as an *implicitly balanced* version of $\widehat{U}\widehat{B}\widehat{V}^{\mathsf{T}}$ given in (2.13)–(2.14). Such implicit balancing is simply built into the expansion without relying on extra tools like Stirling's formula used in [6]. While both low-rank expansions share the same truncation error, we can clearly see that the entrywise magnitudes in (2.13)–(2.14) may become very large for some data sets. In fact, when the data points are highly stretched so that the distances between some points and their center are large, then the entries in $\widehat{U}, \widehat{V}$ may become large. If the points are highly clustered so that the distance between the centers is small, then the entries in $\widehat{B}$ may be large. These lead to potential numerical instability. On the other hand, our strategy in Theorem 2.2 tries to balance magnitudes of the entries in U, B, V to ensure stability. The effectiveness is shown as follows.

THEOREM 2.3. *The entries of U, V, B in Theorem 2.2 satisfy*

$$
\|U\|_{\max} \le 1, \quad \|V\|_{\max} \le 1, \quad \textit{and} \quad \|B\|_{1,1} \le \frac{K_{\min}}{(1-\tau)^{2+2|d|}}, \tag{2.16}
$$

where $K_{\min} := \min_{x_i \in \mathbf{x}, y_j \in \mathbf{y}} |\kappa(x_i, y_j)|$.

Proof. Since $|x - o_{\mathbf{x}}| \le \delta_{\mathbf{x}}$ and $|y - o_{\mathbf{y}}| \le \delta_{\mathbf{y}}$, we immediately have $\|U\|_{\max} \le 1$ and $\|V\|_{\max} \le 1$. Note

$$\left|\binom{k+d}{k}\right| = \left|\frac{(k+d)(k-1+d)\cdots(1+d)}{k!}\right| \le \binom{k+|d|}{k}.$$

Thus,

$$\begin{aligned}
\|B\|_{1,1} &= \sum_{k=0}^{r-1}\sum_{i=0}^{k} |b_{i,k-i}| \\
&\le |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1}\sum_{i=0}^{k} \binom{k+|d|}{k}\binom{k}{i} \left|\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right|^{i} \left|\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right|^{k-i} \\
&= |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1} \binom{k+|d|}{k} \sum_{i=0}^{k}\binom{k}{i} \left|\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right|^{i} \left|\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right|^{k-i} \\
&\le |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1} \binom{k+|d|}{k} \sum_{i=0}^{k}\binom{k}{i} \left|\frac{\tau\delta_{\mathbf{x}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}\right|^{i} \left|\frac{\tau\delta_{\mathbf{y}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}\right|^{k-i} \\
&= |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1} \binom{k+|d|}{k} \tau^k \left(\sum_{j=0}^{k}\binom{k}{i} \left(\frac{\delta_{\mathbf{x}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}\right)^{i} \left(\frac{\delta_{\mathbf{y}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}\right)^{k-i}\right) \\
&= |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1} \binom{k+|d|}{k} \tau^k \left(\frac{\delta_{\mathbf{x}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}} + \frac{\delta_{\mathbf{y}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}}\right)^{k} \\
&= |o_{\mathbf{x}} - o_{\mathbf{y}}|^{-(1+d)} \sum_{k=0}^{r-1} \binom{k+|d|}{k} \tau^k \le \frac{|\kappa(x,y)|}{(1-\tau)^{|1+d|}} \sum_{k=0}^{\infty} \binom{k+|d|}{k} \tau^k \\
&= \frac{|\kappa(x,y)|}{(1-\tau)^{|1+d|}(1-\tau)^{1+|d|}} \le \frac{|\kappa(x,y)|}{(1-\tau)^{2+2|d|}},
\end{aligned}$$

where (2.1) and (2.11) are used. Since the result above holds for all $x \in \mathbf{x}$, $y \in \mathbf{y}$, we can replace $|\kappa(x,y)|$ with $K_{\min}$ to obtain the desired bound. ∎

2.2. Logarithmic kernel. We now consider the logarithmic kernel evaluated at well-separated sets $\mathbf{x}$ and $\mathbf{y}$ and use Taylor expansions to establish (2.2).

THEOREM 2.4. *Let* $\kappa(x,y) = \log\frac{1}{|x-y|}$. *Suppose* $\mathbf{x} = \{x_i\}_{i=1}^m$ *and* $\mathbf{y} = \{y_j\}_{j=1}^n$ *are well-separated with separation ratio* τ. *Then*

$$K_{\mathbf{x},\mathbf{y}} := [(\kappa(x_i, y_j)]_{x_i\in\mathbf{x}, y_j\in\mathbf{y}} = \Re(UBV^{\mathsf{T}}) + E_r, \tag{2.17}$$

where U *and* V *take the forms in* (2.4) *and* $B := [b_{i,j}]_{0\le i,j\le r-1}$ *with*

$$b_{i,j} := \begin{cases} \log\frac{1}{|o_{\mathbf{x}} - o_{\mathbf{y}}|}, & i = j = 0, \\ \frac{(-1)^i}{i+j}\binom{i+j}{i}\left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^{i}\left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^{j}, & 0 < i + j \le r-1, \\ 0, & i + j > r-1. \end{cases} \tag{2.18}$$

Moreover, $\|E_r\|_{\max} \le \frac{\tau^r}{r(1-\tau)}$.

Proof. Let $x \in \mathbf{x}$, $y \in \mathbf{y}$ and let s be as in (2.8). Define $\varepsilon_r(s) := \sum_{k=r}^{\infty} k^{-1} s^k$. Since $\mathbf{x}$ and $\mathbf{y}$ are well separated with separation ratio τ, we have $|s| \leq \tau < 1$. By Taylor and binomial expansions,

$$\begin{aligned}\log \frac{1}{1-s} &= \sum_{k=1}^{r-1} \frac{s^k}{k} + \varepsilon_r(s) = \sum_{k=1}^{r-1} \sum_{i=0}^{k} \frac{(-1)^i}{k} \binom{k}{i} \frac{(x-o_{\mathbf{x}})^i (y-o_{\mathbf{y}})^{k-i}}{(o_{\mathbf{x}}-o_{\mathbf{y}})^k} + \varepsilon_r(s) \\ &= \sum_{k=1}^{r-1} \sum_{i=0}^{k} b_{i,k-i} \left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^i \left(\frac{y-o_{\mathbf{y}}}{\delta_{\mathbf{y}}}\right)^{k-i} + \varepsilon_r(s),\end{aligned}$$

where $|\varepsilon_r(s)| \leq \sum_{i=r}^{\infty} \frac{\tau^i}{i} \leq \frac{\tau^r}{r(1-\tau)}$. Then,

$$\begin{aligned}\log \frac{1}{|x-y|} &= \log \frac{1}{|o_{\mathbf{x}}-o_{\mathbf{y}}|} + \log \frac{1}{|1-s|} \\ &= \log \frac{1}{|o_{\mathbf{x}}-o_{\mathbf{y}}|} + \Re \left(\log \frac{1}{1-s}\right) = \Re \left(b_{0,0} + \log \frac{1}{1-s}\right) \\ &= \Re \left(b_{0,0} + \sum_{k=1}^{r-1} \sum_{i=0}^{k} b_{i,k-i} \left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^i \left(\frac{y-o_{\mathbf{y}}}{\delta_{\mathbf{y}}}\right)^{k-i} + \varepsilon_r(s)\right) \\ &= \Re \left(\sum_{k=0}^{r-1} \sum_{i=0}^{k} b_{i,k-i} \left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^i \left(\frac{y-o_{\mathbf{y}}}{\delta_{\mathbf{y}}}\right)^{k-i}\right) + \Re\left(\varepsilon_r(s)\right),\end{aligned}$$

which gives (2.17) and (2.18). ∎

Similarly, Theorem 2.4 may be viewed as an implicitly balanced version of a low-rank approximation $\widehat{U}\widehat{B}\widehat{V}^{\mathsf{T}}$ obtained from the usual application of Taylor expansions, where $\widehat{U}, \widehat{V}$ are given in (2.13) and $\widehat{B}$ looks like

$$\widehat{B} = [\widehat{b}_{i,j}]_{0 \leq i,j \leq r-1} := \begin{cases} \log \frac{1}{|o_{\mathbf{x}}-o_{\mathbf{y}}|}, & i = j = 0, \\ (-1)^i \frac{(i+j-1)!}{(o_{\mathbf{x}}-o_{\mathbf{y}})^{i+j}}, & 0 < i+j \leq r-1, \\ 0, & i+j > r-1. \end{cases} \tag{2.19}$$

$\widehat{U}\widehat{B}\widehat{V}^{\mathsf{T}}$ may also suffer from large entrywise magnitudes in the factors. The balancing strategy, on the other hand, can effectively control the magnitudes, as can be seen below.

THEOREM 2.5. *The entries of U, V, B in Theorem 2.4 satisfy*

$$\|U\|_{\max} \leq 1, \quad \|V\|_{\max} \leq 1, \quad \textit{and} \quad \|B\|_{1,1} \leq K_{\min} + 2 \log \frac{1}{1-\tau}, \tag{2.20}$$

where $K_{\min} := \min_{x_i \in \mathbf{x}, y_j \in \mathbf{y}} |\kappa(x_i, y_j)|$.

Proof. U and V have the same forms as in Theorem 2.2 and the reason for $\|U\|_{\max} \leq 1$, $\|V\|_{\max} \leq 1$ is the same as in Theorem 2.3. Now,

$$\begin{aligned}\|B\|_{1,1} &= |b_{0,0}| + \sum_{k=1}^{r-1} \sum_{i=0}^{k} |b_{i,k-i}| = |b_{0,0}| + \sum_{k=1}^{r-1} \frac{1}{k} \sum_{i=0}^{k} \binom{k}{i} \left|\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}}-o_{\mathbf{y}}}\right|^i \left|\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}}-o_{\mathbf{y}}}\right|^{k-i} \\ &\leq |b_{0,0}| + \sum_{k=1}^{r-1} \frac{1}{k} \sum_{i=0}^{k} \binom{k}{i} \left|\frac{\tau\delta_{\mathbf{x}}}{\delta_{\mathbf{x}}+\delta_{\mathbf{y}}}\right|^i \left|\frac{\tau\delta_{\mathbf{y}}}{\delta_{\mathbf{x}}+\delta_{\mathbf{y}}}\right|^{k-i}\end{aligned}$$

$$
\begin{aligned}
&= |b_{0,0}| + \sum_{k=1}^{r-1} \frac{\tau^k}{k} \left(\sum_{i=0}^{k} \binom{k}{i} \left(\frac{\delta_{\mathbf{x}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}} \right)^i \left(\frac{\delta_{\mathbf{y}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}} \right)^{k-i} \right) \\
&= |b_{0,0}| + \sum_{k=1}^{r-1} \frac{\tau^k}{k} \left(\frac{\delta_{\mathbf{x}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}} + \frac{\delta_{\mathbf{y}}}{\delta_{\mathbf{x}} + \delta_{\mathbf{y}}} \right)^k \leq |b_{0,0}| + \sum_{k=1}^{\infty} \frac{\tau^k}{k} \\
&= \left| \log \frac{1}{|o_{\mathbf{x}} - o_{\mathbf{y}}|} \right| + \log \frac{1}{1-\tau} \leq \left| \log \frac{1}{|x-y|} \right| + 2 \log \frac{1}{1-\tau},
\end{aligned}
$$

where we used (2.1) and (2.10). Since the result above holds for all $x \in \mathbf{x}$, $y \in \mathbf{y}$, we immediately obtain the desired bound. ∎

2.3. Extensions, generalizations, and other low-rank approximation methods. The discussions in the previous two subsections are in terms of the generalized Cauchy kernel and the logarithmic kernel. In fact, various generalizations may be made, including extensions of these two types of kernels and generalizations to other kernels.

An extension can be made to the generalized Cauchy kernel in Theorem 2.2 with $d \in \mathbb{Z}$ relaxed to $d \in \mathbb{C}$ for some data points. In the theorem, the assumption $d \in \mathbb{Z}$ is only needed for the identity (2.9) and the inequality (2.11). If $\mathbf{x}$ and $\mathbf{y}$ are subsets of a straight line in $\mathbb{C}$, we can allow $d \in \mathbb{C}$. This will be particularly useful in practice for cases like real point sets $\mathbf{x}$ and $\mathbf{y}$.

COROLLARY 2.6. *If $\mathbf{x}$ and $\mathbf{y}$ are subsets of a straight line in $\mathbb{C}$, then Theorem 2.2 still holds with the assumption $d \in \mathbb{Z}$ relaxed to $d \in \mathbb{C}$, except with an extra factor $e^{2\pi|\Im(1+d)|}$ included on the right-hand side of (2.7).*

Proof. If $\mathbf{x}$ and $\mathbf{y}$ are subsets of a straight line, then the centers $o_{\mathbf{x}}$ and $o_{\mathbf{y}}$ can be set to be also on that line. Hence, s in (2.8) is actually a real number. Note $1 - s > 0$ since $|s| \leq \tau < 1$. Thus, the identity (2.9) still holds with $d \in \mathbb{C}$. Use the inequality

$$
|z|^{\Re(\eta)} e^{-\pi|\Im(\eta)|} \leq |z^\eta| \leq |z|^{\Re(\eta)} e^{\pi|\Im(\eta)|}, \quad z, \eta \in \mathbb{C},
$$

and recall (2.10) to get

$$
\begin{aligned}
\left| (o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d} \right| &\geq |o_{\mathbf{x}} - o_{\mathbf{y}}|^{\Re(1+d)} e^{-\pi|\Im(1+d)|} \\
&\geq (1-\tau)^{|\Re(1+d)|} |x-y|^{\Re(1+d)} e^{-\pi|\Im(1+d)|} \\
&\geq (1-\tau)^{|\Re(1+d)|} \left| (x-y)^{1+d} \right| e^{-2\pi|\Im(1+d)|}.
\end{aligned}
$$

Accordingly, we can update (2.12) by including an extra factor $e^{2\pi|\Im(1+d)|}$ into its right-hand side to get a new bound for $|\frac{\varepsilon_r(s)}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}}|$. The extra factor $e^{2\pi|\Im(1+d)|}$ then appears on the right-hand side of (2.7). ∎

Next, the logarithmic kernel case in Theorem 2.4 is to essentially dealing with the more general kernel $\kappa(x, y) = \log \frac{1}{x-y}$ by taking the real part.

Other than these kernels, the techniques used in the derivations in the previous two subsections may be applied to other kernels of the form $\kappa(x - y)$, where Taylor expansions are used to obtain degenerate expansions. We can similarly apply Taylor expansions based on the separation condition (2.1), followed by the binomial expansion applied to the powers of the summation form of s in (2.8). The U, V basis matrices take the same forms as in (2.4) in general and the specific form of the B matrix

depends on the actual kernel. The strategy of stabilization by implicit balancing applies just like before.

For a general kernel $\kappa(x, y)$ with $\kappa : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ that is not necessarily translation invariant, if it is sufficiently smooth, we may use the bivariate Taylor expansion (if we choose to stick to Taylor expansions to obtain our low-rank approximation). Let $\boldsymbol{\alpha} := (\alpha_1, \alpha_2)$ and $\boldsymbol{\beta} := (\beta_1, \beta_2)$, where $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{N} \cup \{0\}$. Define $D_1^{\boldsymbol{\alpha}}$ to be the $\boldsymbol{\alpha}$-th partial derivative of the first component of $\kappa(x, y)$ (i.e., $D_1^{\boldsymbol{\alpha}} \kappa(x, y) = \partial_1^{\alpha_1} \partial_1^{\alpha_2} \kappa(x, y)$), and $D_2^{\boldsymbol{\beta}}$ to be the $\boldsymbol{\beta}$-th partial derivative of the second component of $\kappa(x, y)$ (i.e., $D_2^{\boldsymbol{\alpha}} \kappa(x, y) = \partial_2^{\alpha_1} \partial_2^{\alpha_2} \kappa(x, y)$). Expanding the second component about the point $y_0 \in \mathbb{R}^2$, and then for each function of x, further expanding the first component about the point $x_0 \in \mathbb{R}^2$, we have

$$\kappa(x, y) \approx \sum_{|\boldsymbol{\beta}|=0}^{r-1} \frac{(y - y_0)^{\boldsymbol{\beta}}}{\boldsymbol{\beta}!} D_2^{\boldsymbol{\beta}} \kappa(x, y_0) \approx \sum_{|\boldsymbol{\alpha}|=0}^{r-1} \sum_{|\boldsymbol{\beta}|=0}^{r-1} \frac{(x - x_0)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \frac{(y - y_0)^{\boldsymbol{\beta}}}{\boldsymbol{\beta}!} D_1^{\boldsymbol{\alpha}} D_2^{\boldsymbol{\beta}} \kappa(x_0, y_0),$$

where the following standard notation for bivariate Taylor expansions is used: $\boldsymbol{\alpha}! := \alpha_1! \alpha_2!$ and $(x - x_0)^{\boldsymbol{\alpha}} := (x_1 - x_{0,1})^{\alpha_1} (x_2 - x_{0,2})^{\alpha_2}$ with $x := (x_1, x_2)$ and $x_0 := (x_{0,1}, x_{0,2})$. Thus if $\mathbf{x}, \mathbf{y} \subset \mathbb{R}^2$, then by (2.3), we can let

$$\begin{aligned} U &:= \left[\frac{(x - o_{\mathbf{x}})^{\boldsymbol{\alpha}}}{\delta_{\mathbf{x}}^{\boldsymbol{\alpha}}} \right]_{x \in \mathbf{x}, 0 \le |\boldsymbol{\alpha}| \le r-1}, \quad V := \left[\frac{(y - o_{\mathbf{y}})^{\boldsymbol{\beta}}}{\delta_{\mathbf{y}}^{\boldsymbol{\beta}}} \right]_{y \in \mathbf{y}, 0 \le |\boldsymbol{\beta}| \le r-1}, \\ B &:= \left[\frac{\delta_{\mathbf{x}}^{\boldsymbol{\alpha}} \delta_{\mathbf{y}}^{\boldsymbol{\beta}}}{\boldsymbol{\alpha}! \boldsymbol{\beta}!} D_1^{\boldsymbol{\alpha}} D_2^{\boldsymbol{\beta}} \kappa(o_{\mathbf{x}}, o_{\mathbf{y}}) \right]_{0 \le |\boldsymbol{\alpha}| \le r-1, 0 \le |\boldsymbol{\beta}| \le r-1}, \end{aligned} \tag{2.21}$$

where we set the expansion points x_0 and y_0 to be the centers $o_{\mathbf{x}}$ and $o_{\mathbf{y}}$ of the sets $\mathbf{x}$ and $\mathbf{y}$, respectively, and $\delta_{\mathbf{x}}, \delta_{\mathbf{y}} \in \mathbb{R}^2$ are respectively the radii of the sets $\mathbf{x}$ and $\mathbf{y}$. Note that the U, V basis matrices take a form similar to (2.4), and as before the matrix B depends on the kernel. Clearly, $\|U\|_{\max} = \|V\|_{\max} = 1$. To determine the matrix norm of B, we would need to know more information about the underlying kernel itself.

Using Taylor expansions is just one way to find a low-rank approximation of a kernel. Interpolation is another common and useful way to achieve this [11, 37]. From the approximation theory point of view, there are many kinds of interpolation methods [5, 9, 12]. Hence, it is not straightforward to make a general statement on whether or not similar stability issues arise from these interpolation methods. In the literature, it is known that the barycentric interpolation [4, 31, 32] and Chebyshev interpolation [11] are numerically stable from the approximation point of view. However, in the context of the backward stability considered in this paper, we need to analyze the corresponding U, B, V matrices and their norms to determine if similar stability issues arise, which may also depend on the kernel. For example, if we consider Chebyshev interpolations, the matrices U, B, and V can be obtained from the relation in [11, (6)]. A direct inspection suggests that each entry of U and V is can be crudely bounded above by 2. The entries of matrix B solely depend on the kernel evaluations at the Chebyshev nodes. Thus, we require more information on the kernel to determine if stability risks may arise. Having said that, the subject of how to systematically resolve potential stability issues (if any) for individual low-rank approximation techniques serves as an interesting future research direction.

When the kernel under consideration is smooth and behaves well, and the derivatives are known and easily computable, using Taylor expansions is advantageous for

the following reasons. First, the procedure is simple, and it is convenient to know the entries of the U, B, V generators and to control the accuracy. It is also convenient to obtain the translation matrices T. Next, the U, V basis matrices and the translation matrices T have a flavor of kernel independence in the sense that they stay the same for all kernels whose low-rank approximations are obtained by Taylor expansions. Thirdly, the simple forms of the generators ensure the design of simple scaling factors to control their norms and the scaling factors only depend on rank-related factorial terms and the radius of the set as seen in (2.15) and (2.21). Lastly, these generators can be computed efficiently via recurrence relations like in the next subsection.

On the other hand, there exist kernels for which Taylor expansions may not be suitable. Asymptotically smooth kernels are examples of this situation. We refer to [1, Definition 2] or [3, Section 4] for the formal definition of an asymptotically smooth function. From the definition, we only know that the magnitudes of the partial derivatives decay as the distance between two points grows. Still, such kernels may not behave well in certain regions. For example, they may exhibit large variations or oscillations. As pointed out in [3, Section 4], some kernels may even have unbounded derivatives as the two points get closer. Such behaviors often cannot be effectively approximated by Taylor expansions, as the error analysis suggests that one would have to take many terms of such expansions. Interpolations have been used to deal with asymptotically smooth kernels [11]. Even though Taylor expansions may not be suitable, the idea to scale U, B, V matrices such that their norms are bounded to ensure stability may still be relevant regardless of the choice of low-rank approximations.

For oscillating kernels such as the Hankel function, a similar idea along with the use of Graf's formula can be applied. This is discussed in a different paper [24].

2.4. Recurrence relations for computing low-rank factors. To form the required low-rank factors U, B, V, convenient recurrence relations may be used. The construction of the U, V in (2.4) is straightforward. For example, for U in (2.4), we may simply let

$$U_{i,0} = 1, \quad U_{i,j} = \frac{x_i - o_{\mathbf{x}}}{\delta_{\mathbf{x}}} U_{i,j-1}, \quad 1 \le j \le r-1.$$

To obtain the recurrence relation for the entries of the B generators, we can take the generalized Cauchy kernel case in subsection 2.1 and the logarithmic kernel case in subsection 2.2 as examples. For the former case, B has the form in (2.5). Note that for $d \in \mathbb{Z}$ and $k, i \in \mathbb{N} \cup \{0\}$,

$$\binom{k+d}{k} = \frac{k+d}{k}\binom{k-1+d}{k-1}, \quad \binom{k}{i} = \binom{k-1}{i-1} + \binom{k-1}{i}. \tag{2.22}$$

Letting $k = i + j$, taking the product of the above two identities, and plugging the result into (2.5), we get

$$\begin{aligned} b_{i,j} &= \frac{(-1)^i}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}} \binom{i+j+d}{i+j} \binom{i+j}{i} \left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^i \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^j \\ &= \frac{(-1)^i}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}} \frac{i+j+d}{i+j} \binom{i+j-1+d}{i+j-1} \left[\binom{i+j-1}{i-1} + \binom{i+j-1}{i}\right] \\ &\quad \cdot \left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^i \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^j \\ &= \frac{i+j+d}{i+j} \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i,j-1} - \frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i-1,j}\right). \end{aligned}$$

Thus, we have the following recurrence relation to compute the entries of B:

$$b_{i,-1} = b_{-1,j} = 0, \quad b_{0,0} = \frac{(-1)^p}{(o_{\mathbf{x}} - o_{\mathbf{y}})^{1+d}},$$
$$b_{i,j} = \frac{i+j+d}{i+j}\left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i,j-1} - \frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i-1,j}\right), \quad 1 \le i+j \le r-1.$$

For the logarithmic kernel case in subsection 2.2, B has the form in (2.18). Multiply $b_{i,j}$ in (2.18) by $(i+j)/(i+j-1)$ and use the second identity in (2.22) with $k = i+j$ to get

$$\begin{aligned}\frac{i+j}{i+j-1} b_{i,j} &= \frac{(-1)^i}{i+j-1}\binom{i+j}{i}\left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^i \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^j \\ &= \frac{(-1)^i}{i+j-1}\left[\binom{i+j-1}{i-1} + \binom{i+j-1}{i}\right]\left(\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^i \left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}\right)^j \\ &= -\frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i-1,j} + \frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i,j-1}.\end{aligned}$$

Thus, we have the following recurrence relation to compute the entries of B:

$$b_{i,-1} = b_{-1,j} = 0, \quad b_{0,0} = \log\frac{1}{|o_{\mathbf{x}} - o_{\mathbf{y}}|}, \quad b_{1,0} = \frac{-\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}}, \quad b_{0,1} = \frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}},$$
$$b_{i,j} = \frac{i+j-1}{i+j}\left(\frac{\delta_{\mathbf{y}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i,j-1} - \frac{\delta_{\mathbf{x}}}{o_{\mathbf{x}} - o_{\mathbf{y}}} b_{i-1,j}\right), \quad 2 \le i+j \le r-1.$$

Lastly, to obtain the recurrence relation for entries of T in (3.3), we use the second identity in (2.22) with $k = j$ to similarly get

$$t_{0,0} = 1, \quad t_{-1,k-1} = t_{k,k-1} = 0,$$
$$t_{i,j} = \frac{\delta_{\mathbf{x}'}}{\delta_{\mathbf{x}}} t_{i-1,j-1} + \frac{o_{\mathbf{x}'} - o_{\mathbf{x}}}{\delta_{\mathbf{x}}} t_{i,j-1}.$$

3. Stable translation relation. A critical idea for the FMM to reach linear complexity is the use of translation relations in the U and V basis matrices in low-rank approximations like (2.3). The relation allows the use of nested basis forms. For notational clarity, we rewrite (2.3) as

$$K_{\mathbf{x},\mathbf{y}} \approx U_{\mathbf{x}} B_{\mathbf{x},\mathbf{y}} V_{\mathbf{y}}^{\mathsf{T}},$$

where the subscripts are used to emphasize the dependency on the underlying point sets. The matrix $U_{\mathbf{x}}$ is the (approximate) column basis matrix of $K_{\mathbf{x},\mathbf{y}}$ and only depends on $\mathbf{x}$. Thus, it can be understood as the *basis contribution* of $\mathbf{x}$ to $K_{\mathbf{x},\mathbf{y}}$. $V_{\mathbf{y}}$ can be similarly understood. Translation relations connect the basis contribution of a set with those of its subsets.

We now show that, with the basis contributions in the form of (2.4) produced by our stabilization strategy, we can also derive a stable translation relation between $U_{\mathbf{x}}$ and $U_{\mathbf{x}'}$ for $\mathbf{x}' \subset \mathbf{x}$. The translation relation for $V_{\mathbf{y}}$ can be derived analogously. Suppose $\mathbf{x}'$ has center $o_{\mathbf{x}'}$ and radius $\delta_{\mathbf{x}'}$ and $\mathbf{x}$ has center $o_{\mathbf{x}}$ and radius $\delta_{\mathbf{x}}$ so that the corresponding disks $\mathcal{D}_{\mathbf{x}'}$ and $\mathcal{D}_{\mathbf{x}}$ satisfy $\mathcal{D}_{\mathbf{x}'} \subset \mathcal{D}_{\mathbf{x}}$, where

$$\mathcal{D}_{\mathbf{x}'} := \{z \in \mathbb{C} : |z - o_{\mathbf{x}'}| \le \delta_{\mathbf{x}'}\}, \quad \mathcal{D}_{\mathbf{x}} := \{z \in \mathbb{C} : |z - o_{\mathbf{x}}| \le \delta_{\mathbf{x}}\}. \tag{3.1}$$

See Figure 3.1 for an illustration.

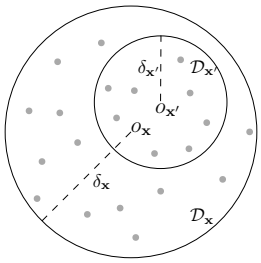


FIG. 3.1. *An illustration of the disks $\mathcal{D}_{\mathbf{x}'}$ and $\mathcal{D}_{\mathbf{x}}$ in Theorem 3.1.*

THEOREM 3.1. *Suppose $\mathbf{x}' \subset \mathbf{x}$ and $\mathcal{D}_{\mathbf{x}'} \subset \mathcal{D}_{\mathbf{x}}$ for $\mathcal{D}_{\mathbf{x}'}$ and $\mathcal{D}_{\mathbf{x}}$ in (3.1). Then the following translation relation holds:*

$$U_{\mathbf{x}',\mathbf{x}} = U_{\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}}, \tag{3.2}$$

where

$$U_{\mathbf{x}',\mathbf{x}} := \left[\left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^j\right]_{x\in\mathbf{x}',0\le j\le r-1}, \quad U_{\mathbf{x}'} := \left[\left(\frac{x-o_{\mathbf{x}'}}{\delta_{\mathbf{x}'}}\right)^j\right]_{x\in\mathbf{x}',0\le j\le r-1},$$

$$T_{\mathbf{x}',\mathbf{x}} := [t_{i,j}]_{0\le i,j\le r-1} \quad \textit{with} \tag{3.3}$$

$$t_{i,j} = \begin{cases} \binom{j}{i}\left(\frac{\delta_{\mathbf{x}'}}{\delta_{\mathbf{x}}}\right)^i\left(\frac{o_{\mathbf{x}'}-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^{j-i}, & 0\le i\le j\le r-1,\\ 0, & 0<j<i\le r-1, \end{cases}$$

Moreover, if $\mathbf{x}'' \subset \mathbf{x}'$ and $\mathbf{x}''$ has center $o_{\mathbf{x}''}$ and radius $\delta_{\mathbf{x}''}$ with the corresponding disk $\mathcal{D}_{\mathbf{x}''} = \{z\in\mathbb{C} : |z-o_{\mathbf{x}''}| \le \delta_{\mathbf{x}''}\} \subset \mathcal{D}_{\mathbf{x}'}$, then

$$T_{\mathbf{x}'',\mathbf{x}} = T_{\mathbf{x}'',\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}}, \tag{3.4}$$

where $T_{\mathbf{x}'',\mathbf{x}}$ and $T_{\mathbf{x}'',\mathbf{x}'}$ are defined like in (3.3), just with the centers and the radii replaced by those associated with corresponding sets.

Proof. Let $x \in \mathbf{x}'$. For $0 \le j \le r-1$, apply binomial expansions to get

$$\left(\frac{x-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^j = \left(\frac{(x-o_{\mathbf{x}'})+(o_{\mathbf{x}'}-o_{\mathbf{x}})}{\delta_{\mathbf{x}}}\right)^j$$
$$= \sum_{i=0}^{j}\binom{j}{i}\left(\frac{x-o_{\mathbf{x}'}}{\delta_{\mathbf{x}'}}\right)^i\left(\frac{\delta_{\mathbf{x}'}}{\delta_{\mathbf{x}}}\right)^i\left(\frac{o_{\mathbf{x}'}-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^{j-i} = \sum_{i=0}^{j}\left(\frac{x-o_{\mathbf{x}'}}{\delta_{\mathbf{x}'}}\right)^i t_{i,j},$$

which yields (3.2).

Now, for any $x \in \mathcal{D}_{\mathbf{x}''}$, let $u_{x,\mathbf{x}''} = \begin{bmatrix} 1 & \frac{x-o_{\mathbf{x}''}}{\delta_{\mathbf{x}''}} & \cdots & \left(\frac{x-o_{\mathbf{x}''}}{\delta_{\mathbf{x}''}}\right)^{r-1}\end{bmatrix}$ and similarly define $u_{x,\mathbf{x}'}$ and $u_{x,\mathbf{x}}$ using the corresponding centers and radii. Then,

$$u_{x,\mathbf{x}''}T_{\mathbf{x}'',\mathbf{x}} = \left[\sum_{i=0}^{j}\left(\frac{x-o_{\mathbf{x}''}}{\delta_{\mathbf{x}''}}\right)^i\binom{j}{i}\left(\frac{\delta_{\mathbf{x}''}}{\delta_{\mathbf{x}}}\right)^i\left(\frac{o_{\mathbf{x}''}-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^{j-i}\right]_{0\le j\le r-1}$$
$$= \left[\sum_{i=0}^{j}\binom{j}{i}\left(\frac{x-o_{\mathbf{x}''}}{\delta_{\mathbf{x}}}\right)^i\left(\frac{o_{\mathbf{x}''}-o_{\mathbf{x}}}{\delta_{\mathbf{x}}}\right)^{j-i}\right]_{0\le j\le r-1}$$

$$= \left[\left(\frac{x - o_{\mathbf{x}}}{\delta_{\mathbf{x}}} \right)^i \right]_{0 \le j \le r-1} = u_{x,\mathbf{x}}.$$

Similarly, we have

$$u_{x,\mathbf{x}''} T_{\mathbf{x}'',\mathbf{x}'} = u_{x,\mathbf{x}'}, \quad u_{x,\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}} = u_{x,\mathbf{x}}.$$

Thus,

$$u_{x,\mathbf{x}''} T_{\mathbf{x}'',\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}} = u_{x,\mathbf{x}} = u_{x,\mathbf{x}''} T_{\mathbf{x}'',\mathbf{x}},$$

which holds for any $x \in \mathcal{D}_{\mathbf{x}''}$. Therefore, we may take r unique points $x_i \in \mathcal{D}_{\mathbf{x}''}$, $i = 0, 1, \ldots, r-1$ to form an $r \times r$ matrix $W = \left[\left(\frac{x_i - o_{\mathbf{x}''}}{\delta_{\mathbf{x}''}} \right)^j \right]_{0 \le i,j \le r-1}$ with rows $u_{x_i,\mathbf{x}''}$ so that

$$W T_{\mathbf{x}'',\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}} = W T_{\mathbf{x}'',\mathbf{x}}.$$

Note that W is a Vandermonde matrix defined on distinct nodes so it is invertible. Thus, $T_{\mathbf{x}'',\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}} = T_{\mathbf{x}'',\mathbf{x}}$. ∎

$U_{\mathbf{x}',\mathbf{x}}$ is a submatrix of the basis contribution $U_{\mathbf{x}}$ from the set $\mathbf{x}$. Thus, $T_{\mathbf{x}',\mathbf{x}}$ is a translation matrix which connects the basis contribution $U_{\mathbf{x}'}$ to $U_{\mathbf{x}}$ via $U_{\mathbf{x}',\mathbf{x}} = U_{\mathbf{x}'} T_{\mathbf{x}',\mathbf{x}}$.

COROLLARY 3.2. *With the conditions in Theorem 3.1, the entries of* $T_{\mathbf{x}',\mathbf{x}}$ *satisfy* $\|T_{\mathbf{x}',\mathbf{x}}\|_1 = 1$.

Proof. Since $\mathcal{D}_{\mathbf{x}'} \subset \mathcal{D}_{\mathbf{x}}$, we have $|o_{\mathbf{x}'} - o_{\mathbf{x}}| \le \delta_{\mathbf{x}} - \delta_{\mathbf{x}'}$. Thus, for all $0 \le j \le r-1$, (3.3) satisfies

$$\sum_{i=0}^{j} |t_{i,j}| \le \sum_{i=0}^{j} \binom{j}{i} \left(\frac{\delta_{\mathbf{x}'}}{\delta_{\mathbf{x}}} \right)^i \left(\frac{\delta_{\mathbf{x}} - \delta_{\mathbf{x}'}}{\delta_{\mathbf{x}}} \right)^{j-i} = 1.$$

On the other hand, $\|T_{\mathbf{x}',\mathbf{x}}\|_1 \ge 1$ because $t_{0,0} = 1$. Therefore, $\|T_{\mathbf{x}',\mathbf{x}}\|_1 = 1$. ∎

Theorem 3.1 and Corollary 3.2 indicate that, for all kernels with degenerate expansions that lead to U, V basis matrices as in (2.4), our stabilization strategy preserves the translation relation like the original FMM and further guarantees that all the translation matrices like $T_{\mathbf{x}',\mathbf{x}}$ have 1-norm equal to 1.

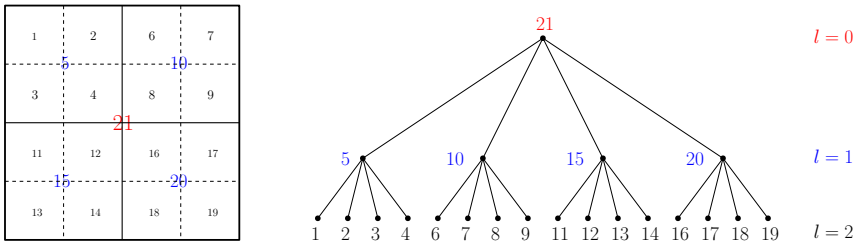
Before we move on to the next section, we make a remark on how the analysis in this section can be applied to the case when the bivariate Taylor expansion is used to obtain a degenerate expansion of a kernel that may not be translation invariant. By (2.21), we have

$$U = \left[\frac{(x - o_{\mathbf{x}})^{\boldsymbol{\alpha}}}{\delta_{\mathbf{x}}^{\boldsymbol{\alpha}}} \right]_{x \in \mathbf{x}, 0 \le |\boldsymbol{\alpha}| \le r} = \left[\left(\frac{x_1 - o_{\mathbf{x},1}}{\delta_{\mathbf{x},1}} \right)^{\alpha_1} \left(\frac{x_2 - o_{\mathbf{x},2}}{\delta_{\mathbf{x},2}} \right)^{\alpha_2} \right]_{x \in \mathbf{x}, 0 \le |\boldsymbol{\alpha}| \le r},$$

where $x := (x_1, x_2)$, $o_{\mathbf{x}} := (o_{\mathbf{x},1}, o_{\mathbf{x},2})$, $\delta_{\mathbf{x}} := (\delta_{\mathbf{x},1}, \delta_{\mathbf{x},2})$, and $\boldsymbol{\alpha} := (\alpha_1, \alpha_2)$. The translation relation for the matrix U in (2.21) can be obtained by applying Theorem 3.1 individually to $\left(\frac{x_1 - o_{\mathbf{x},1}}{\delta_{\mathbf{x},1}} \right)^{\alpha_1}$ and $\left(\frac{x_2 - o_{\mathbf{x},2}}{\delta_{\mathbf{x},2}} \right)^{\alpha_2}$ and appropriately combining the entries by taking into account of the multi-index. The same holds for the matrix V in (2.21).

4. Stable matrix version of the FMM. In this section, we present the 2D FMM in terms of an intuitive stable matrix version based on the results in the previous two sections. We give a concise illustration of the main steps without involving too many details or using expansion forms in the original FMM in [14]. This provides a convenient way for non-experts to quickly digest the method and also serves as a necessary preparation for our backward stability analysis in the next section.

For a given accuracy, the FMM in [14] essentially constructs an *FMM matrix* approximation $\tilde{K}$ to the kernel matrix K in (1.1), where K is the evaluation of a degenerate kernel function $\kappa(x, y)$ at all $x \in \mathbf{X}$ and $y \in \mathbf{Y}$. For convenience, we suppose $\mathbf{X}$ and $\mathbf{Y}$ are located within a square domain $\mathbf{S}$ in $\mathbb{C}$ and also $N = |\mathbf{X}| = |\mathbf{Y}|$. The FMM begins with a hierarchical partitioning of $\mathbf{S}$. Quadrisect $\mathbf{S}$ recursively and adaptively until the number of points in each subdomain is at most a prespecified constant N_0. A two-level partitioning example is given in Figure 4.1. We use a postordered quadtree $\mathcal{T}$ to organize the partition, which is called the *FMM tree* for convenience. Each node is labeled with a single index $\mathbf{i}$ and corresponds to $\mathbf{S}$ when $\mathbf{i}$ is the root or to a subdomain otherwise. The root node is assumed to be at level 0 and the children of a level-l node are at level $l + 1$. The total number of levels of $\mathcal{T}$ is L, which is assumed to be $O(\log\left(\frac{N}{N_0}\right))$. We use $\mathrm{lv}(\mathbf{i})$ to denote the level of node $\mathbf{i}$. We sometimes also use a node $\mathbf{i}$ to mean its corresponding subdomain.



(a) Domain partitioning (b) FMM tree

FIG. 4.1. *Example of a two-level domain partitioning and the corresponding postordered FMM tree* $\mathcal{T}$.

4.1. Interpretation of FMM terminology in matrix language. To facilitate the understanding of the FMM for non-experts, we interpret some essential FMM terminology in terms of the matrix forms.

Use $\mathbf{x_i}$ to denote the subset of $\mathbf{X}$ that are located in the subdomain $\mathbf{i}$, and $\mathbf{y_i}$ to denote the subset of $\mathbf{Y}$ that are located in the subdomain $\mathbf{i}$. Let $o_\mathbf{i}$ and $\delta_\mathbf{i}$ be the center and radius of the subdomain (or in other words, the sub-square) $\mathbf{i}$. Without loss of generality, we let the centers and radii of the subsets $\mathbf{x_i}$ and $\mathbf{y_i}$ to be equal to those of the subdomain; i.e., $o_{\mathbf{x_i}} = o_{\mathbf{y_i}} = o_\mathbf{i}$ and $\delta_{\mathbf{x_i}} = \delta_{\mathbf{y_i}} = \delta_\mathbf{i}$. For convenience, we write $K_{\mathbf{i,j}}$ as the block of K defined by subsets $\mathbf{x_i}$ and $\mathbf{y_j}$:

$$K_{\mathbf{i,j}} = K_{\mathbf{x_i},\mathbf{y_j}} = [\kappa(x, y)]_{x\in\mathbf{x_i},y\in\mathbf{y_j}}.$$

When $\mathbf{i}$ and $\mathbf{j}$ are not well separated, they are said to be *neighbors* and $K_{\mathbf{i,j}}$ is referred as a *near-field interaction* or *near-field block*. $K_{\mathbf{i,j}}$ is a diagonal block if $\mathbf{i} = \mathbf{j}$. Otherwise, it is often an off-diagonal block near the diagonal, but it may also be further away because of the ordering of the subdomains.

When $\mathbf{i}$ and $\mathbf{j}$ are well separated with separation ratio τ, the block $K_{\mathbf{i,j}}$ is referred to as the *far-field interaction* or *far-field block* and has a low-rank approximation as

discussed in section 2:

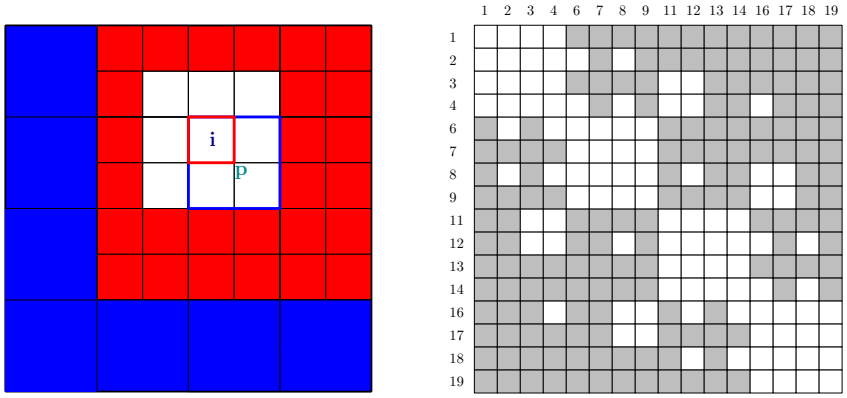
$$K_{\mathbf{i},\mathbf{j}} = U_{\mathbf{i}} B_{\mathbf{i},\mathbf{j}} V_{\mathbf{j}}^{\mathsf{T}} + E_{\mathbf{i},\mathbf{j}} \approx U_{\mathbf{i}} B_{\mathbf{i},\mathbf{j}} V_{\mathbf{j}}^{\mathsf{T}}, \tag{4.1}$$

where we write $U_{\mathbf{i}} = U_{\mathbf{x}_\mathbf{i}}$, $B_{\mathbf{i},\mathbf{j}} = B_{\mathbf{x}_\mathbf{i},\mathbf{y}_\mathbf{j}}$, $V_{\mathbf{j}} = V_{\mathbf{y}_\mathbf{j}}$ for convenience. $U_{\mathbf{i}}$ and $V_{\mathbf{j}}$ are the *basis contributions* from $\mathbf{x}_\mathbf{i}$ and $\mathbf{y}_\mathbf{j}$, respectively. In the original FMM in [14], $U_{\mathbf{i}}$ and $V_{\mathbf{j}}$ are essentially produced by two different types of expansions, the so-called *local expansion* and *multipole expansion*, respectively.

In contrast, here we use centers for both $\mathbf{x}_\mathbf{i}$ and $\mathbf{y}_\mathbf{j}$ like in [28] so that $U_{\mathbf{i}}$ and $V_{\mathbf{j}}$ have similar forms like in (2.4) (just with different points, centers, and radii). Such a setup makes it very convenient to handle $U_{\mathbf{i}}$ and $V_{\mathbf{j}}$ and they are treated in the same way: simply as basis contributions from the respective point sets. There is then *no need to distinguish* local expansions from multipole expansions. Indeed, if $\mathbf{X}$ and $\mathbf{Y}$ are the same set, then we may set $U_{\mathbf{i}}$ and $V_{\mathbf{i}}$ to be the same.

Similarly, $B_{\mathbf{i},\mathbf{j}}$ corresponds to the *multipole-to-local* translation operation in [14]. Now when the translation operation in section 3 is used so as to construct a basis contribution $U_{\mathbf{i}}$ associated with a node $\mathbf{i}$ from those associated with its children $\mathbf{i}_\alpha$, we can use a translation operator $T_{\mathbf{i}_\alpha,\mathbf{i}}$ defined like in (3.3). $T_{\mathbf{i}_\alpha,\mathbf{i}}$ then corresponds to a so-called *multipole-to-multipole* translation in [14]. When row basis contributions like $V_{\mathbf{i}}$ are considered, we can similarly see that $T_{\mathbf{i}_\alpha,\mathbf{i}}$ also corresponds to a *local-to-local* translation in [14].

For near-field interactions, a concept of *interaction list* is used in [14] to explore finer-level rank structures. The interaction list $\mathcal{L}_{\mathbf{i}}$ associated with a node $\mathbf{i}$ of the FMM tree $\mathcal{T}$ is defined to be the collection of nodes $\mathbf{j}$ that satisfies these conditions: (i) $\mathrm{lv}(\mathbf{j}) = \mathrm{lv}(\mathbf{i})$; (ii) $\mathbf{i}$ and $\mathbf{j}$ are well separated with separation ratio τ; (iii) the parents of $\mathbf{i}$ and $\mathbf{j}$ are neighbors. An example is given in Figure 4.2(a) for a node $\mathbf{i}$ with $\mathrm{lv}(\mathbf{i}) = 3$ and its parent $\mathbf{p}$ within a three-level partition of $\mathbf{S}$. It is easy to see for any node $\mathbf{i} \in \mathcal{T}$, its interaction list $\mathcal{L}_{\mathbf{i}}$ has at most 27 nodes. The interaction list decides which far-field subblocks are to be considered within a near-field block. Figure 4.2(b) gives an example of the far-field interactions at level 2.



(a) Partitioning of the domain (b) Far-field blocks at level 2

FIG. 4.2. *Interaction list and far-field blocks. In (a), the red squares correspond to nodes in the interaction list of* $\mathbf{i}$ *and the red squares correspond to nodes in the interaction list of* $\mathbf{p}$*. (b) shows the far-field blocks at level* 2 *where each gray block is a far-field block that has a low-rank approximation like in* (4.1) *and the numbers beside the matrix correspond to the leaf level nodes in Figure* 4.1*(b).*

Following these discussions, we summarize key FMM terminology and the corresponding matrix forms in Table 4.1.

TABLE 4.1
FMM terminology and the corresponding matrix forms.

FMM terminology	Nodes	Corresponding matrix forms
Far-field interaction	$\mathbf{i} \leftrightarrow \mathbf{j}$	$K_{\mathbf{i},\mathbf{j}}$: well-separated $\mathbf{i}$, $\mathbf{j}$
Near-field interaction	$\mathbf{i} \leftrightarrow \mathbf{j}$	$K_{\mathbf{i},\mathbf{j}}$: $\mathbf{i}$, $\mathbf{j}$ not well separated
Interaction list	$\mathbf{i}$	All $\mathbf{j}$ at level $l = \mathrm{lv}(\mathbf{i})$ so that $B_{\mathbf{i},\mathbf{j}}$ is a nonzero block of $B^{(l)}$ (later in (4.2))
Multipole expansion	$\mathbf{j}$	$V_{\mathbf{j}}$ in (4.1)
Local expansion	$\mathbf{i}$	$U_{\mathbf{i}}$ in (4.1)
Multipole-to-local expansion	$\mathbf{j} \to \mathbf{i}$	$B_{\mathbf{i},\mathbf{j}}$ in (4.1)
Multipole-to-multipole expansion	$\mathbf{c} \to \mathbf{i}$	$T_{\mathbf{x}_\mathbf{c},\mathbf{x}_\mathbf{i}}$ like in (3.3)
Local-to-local expansion	$\mathbf{c} \to \mathbf{i}$	$T_{\mathbf{x}_\mathbf{c},\mathbf{x}_\mathbf{i}}$ like in (3.3)

4.2. Levelwise low-rank approximation. We can now look at the matrix structure of K. At level $l = 1$, all the nodes of the FMM tree $\mathcal{T}$ are neighbors. At level $l \geq 2$, we can define a level-l far-field matrix $K^{(l)}$ by retaining all the far-field blocks $K_{\mathbf{i},\mathbf{j}}$ with $\mathrm{lv}(\mathbf{i}) = \mathrm{lv}(\mathbf{j}) = l$ and $\mathbf{j} \in \mathcal{L}_\mathbf{i}$ and setting other blocks of K to zero. An example of $K^{(2)}$ for the two-level case is given in Figure 4.2(b). Each $K^{(l)}$ takes the form of

$$K^{(l)} = U^{(l)} B^{(l)} (V^{(l)})^\mathsf{T} + E^{(l)} \quad \text{with} \tag{4.2}$$
$$U^{(l)} = \mathrm{diag}(U_{\mathbf{i}_1}, \ldots, U_{\mathbf{i}_k}) \quad \text{and} \quad V^{(l)} = \mathrm{diag}(V_{\mathbf{i}_1}, \ldots, V_{\mathbf{i}_k}),$$

where $\{\mathbf{i}_j\}_{j=1}^k$ is the set of nodes at level l and $B^{(l)}$ and $E^{(l)}$ have the same block structure as $K^{(l)}$, just with $K_{\mathbf{i},\mathbf{j}}$ replaced by $B_{\mathbf{i},\mathbf{j}}$ and $E_{\mathbf{i},\mathbf{j}}$, respectively. We also define the near-field matrix $K^{(0)}$ by retaining only the near-field blocks $K_{\mathbf{i},\mathbf{j}}$ with $\mathbf{i}$ and $\mathbf{j}$ leaf-level neighbors (including the case $\mathbf{i} = \mathbf{j}$) and setting other blocks of K to zero. Then the kernel matrix K can be decomposed as

$$K = K^{(0)} + \sum_{l=2}^{L} K^{(l)} = K^{(0)} + \sum_{l=2}^{L} U^{(l)} B^{(l)} (V^{(l)})^\mathsf{T} + E, \tag{4.3}$$

where the error term $E := \sum_{l=2}^{L} E^{(l)}$ and the nonzero pattern of $E^{(l)}$ does not overlap for different l. A matrix approximation to K is then obtained as

$$\tilde{K} := K^{(0)} + \sum_{l=2}^{L} U^{(l)} B^{(l)} (V^{(l)})^\mathsf{T}. \tag{4.4}$$

$\tilde{K}$ can be multiplied with a vector in $O(LN) = O(N \log(N))$ complexity when N_0 is a constant.

4.3. Nested basis, FMM matrix, and FMM algorithm. To further accelerate the matrix-vector multiplication cost with $\tilde{K}$ to $O(N)$, the FMM exploits the nested relations in $U^{(l)}$ and $V^{(l)}$. This may be based on the results in section 3. Suppose a non-leaf node $\mathbf{i} \in \mathcal{T}$ has four children $\{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ in the hierarchical partition of $\mathbf{S}$ and this node $\mathbf{i}$ contains the following subsets of $\mathbf{X}$ and $\mathbf{Y}$:

$$\mathbf{x}_\mathbf{i} \cup \mathbf{y}_\mathbf{i} = \mathbf{x}_{\mathbf{c}_1} \cup \mathbf{x}_{\mathbf{c}_2} \cup \mathbf{x}_{\mathbf{c}_3} \cup \mathbf{x}_{\mathbf{c}_4} \cup \mathbf{y}_{\mathbf{c}_1} \cup \mathbf{y}_{\mathbf{c}_2} \cup \mathbf{y}_{\mathbf{c}_3} \cup \mathbf{y}_{\mathbf{c}_4}.$$

For $1 \le j \le 4$, let $T_{\mathbf{c}_j,\mathbf{i}}$ be a translation matrix from node $\mathbf{c}_j$ to node $\mathbf{i}$ so that

$$U_{\mathbf{i}} = \operatorname{diag}(U_{\mathbf{c}_1}, \ldots, U_{\mathbf{c}_4}) R_{\mathbf{i}} \quad \text{with} \quad R_{\mathbf{i}} := (T_{\mathbf{c}_1,\mathbf{i}}^{\mathsf{T}}, \ldots, T_{\mathbf{c}_4,\mathbf{i}}^{\mathsf{T}})^{\mathsf{T}}, \tag{4.5}$$

where $T_{\mathbf{c}_j,\mathbf{i}}$ is defined like in (3.3) with the relevant centers and radii of $\mathbf{x}'$ and $\mathbf{x}$ replaced by those corresponding to the node $\mathbf{c}_j$ and the node $\mathbf{i}$, respectively. As mentioned earlier, the centers and radii of the subsets of $\mathbf{X}$ and $\mathbf{Y}$ are chosen to be equal to those of the subdomains they belong to. Thus, the translation matrix $R_{\mathbf{i}}$ can be reused to get $V_{\mathbf{i}} = \operatorname{diag}(V_{\mathbf{c}_1}, \ldots, V_{\mathbf{c}_4}) R_{\mathbf{i}}$.

Thus, we obtain the following *nested basis* relation

$$\begin{aligned} &U^{(l)} = U^{(l+1)} R^{(l)}, \quad V^{(l)} = V^{(l+1)} R^{(l)} \quad \text{with} \\ &R^{(l)} = \operatorname{diag}(R_{\mathbf{i}_1}, \ldots, R_{\mathbf{i}_k}), \quad \{\mathbf{i}_j\}_{j=1}^{k}\text{: nodes at level } l, \end{aligned} \tag{4.6}$$

From this, we get a *telescoping expansion* of $\tilde{K}$ which is the final *FMM matrix*:

$$\begin{aligned} \tilde{K} &= K^{(0)} + \sum_{l=2}^{L} U^{(L)} \Big(\prod_{\tilde{l}=L-1}^{l} R^{(\tilde{l})} \Big) B^{(l)} \Big(\prod_{\tilde{l}=L-1}^{l} R^{(\tilde{l})} \Big)^{\mathsf{T}} (V^{(L)})^{\mathsf{T}} \\ &= K^{(0)} + U^{(L)} \Big(R^{(L-1)} \Big(\cdots \big(R^{(2)} B^{(2)} (R^{(2)})^{\mathsf{T}} + B^{(3)} \big) \cdots \Big) (R^{(L-1)})^{\mathsf{T}} + B^{(L)} \Big) (V^{(L)})^{\mathsf{T}}, \end{aligned} \tag{4.7}$$

which is given in terms of the U, V, R, B matrices called *FMM generators*.

In the actual representation of $\tilde{K}$ in (4.7), it just needs to assemble the generators $U_{\mathbf{i}}$, $V_{\mathbf{i}}$ for leaf nodes $\mathbf{i}$, all the translation matrices $T_{\mathbf{i}}$, and all the $B_{\mathbf{i},\mathbf{j}}$ generators. Upper-level basis matrices are not explicitly stored. Then the matrix-vector product with $\tilde{K}$ can be computed through bottom-up and top-down traversals along the FMM tree $\mathcal{T}$. See Algorithm 4.1. The FMM matrix-vector multiplication now just needs $O(N)$ complexity. Specifically, if we choose $N_0 = O(r)$, where r is the expansion order in the degenerate expansion (2.2) and the far-field low-rank approximation (4.1), then the FMM complexity is $O(r^2 N)$. r is a constant for a given accuracy.

Algorithm 4.1 FMM matrix-vector multiplication $\phi = \tilde{K} q$

$v^{(L)} \leftarrow (V^{(L)})^{\mathsf{T}} q$
$t^{(L)} \leftarrow B^{(L)} v^{(L)}$
for level $l = L-1, \ldots, 2$ **do** ▷ *Bottom-up traversal*
  $v^{(l)} \leftarrow \big(R^{(l)}\big)^{\mathsf{T}} v^{(l+1)}$
  $t^{(l)} \leftarrow B^{(l)} v^{(l)}$
end for
$u^{(2)} \leftarrow t^{(2)}$
for level $l = 3, \ldots, L$ **do** ▷ *Top-down traversal*
  $u^{(l)} \leftarrow R^{(l-1)} u^{(l-1)} + t^{(l)}$
end for
$\phi \leftarrow U^{(L)} u^{(L)} + K^{(0)} q$ ▷ *Evaluation*

Remark 4.1. In Algorithm 4.1, we present the algorithm in terms of levelwise operations and include superscripts in the intermediate vectors $v^{(l)}$ and $t^{(l)}$ so as to facilitate our later stability analysis. In actual implementations, just two intermediate

vectors v and t are needed. In all the levelwise products like $(R^{(l)})^{\mathsf{T}} v^{(l+1)}$, local products associated with each node at level l are evaluated so as to perform relevant block diagonal multiplications. Essentially, it is not hard to write the FMM algorithm in terms of traversals of the FMM tree nodes where local products are communicated between parents and children and between siblings.

5. Backward stability of the FMM. Given the matrix form FMM that is based on the kernel expansions in Section 2 and the translation relation in Section 3, we now study the backward stability of the FMM.

Assumption 5.1. To facilitate the proof of the backward stability of the FMM, we make the following assumptions.

1. $|\mathbf{X}| = |\mathbf{Y}| = N$ for $\mathbf{X}, \mathbf{Y}$ in (1.1).
2. $\mathcal{T}$ is a full quadtree with L levels so that there are 4^l nodes at level l for $0 \le l \le L$.
3. For each leaf node $\mathbf{i} \in \mathcal{T}$, $U_{\mathbf{i}}$ and $V_{\mathbf{i}}$ have column sizes r and row sizes $|\mathbf{x}_{\mathbf{i}}| = |\mathbf{y}_{\mathbf{i}}| = N_0 = O(r)$. Accordingly, $L(\approx \frac{1}{2}\log_2\left(\frac{N}{N_0}\right)) \le \log_2(N)$.
4. For each node $\mathbf{i}$ at level $l > 1$, $R_{\mathbf{i}}$ in (4.5) has column size r. This is based on the column size of the translation matrix like in (3.3).
5. The low-rank factors for each far-field block in (4.1) with $\mathrm{lv}(\mathbf{i}) = l > 1$ and $\mathbf{j} \in \mathcal{L}_{\mathbf{i}}$ satisfy

$$\|U_{\mathbf{i}}\|_{\max} \le c_U, \quad \|V_{\mathbf{j}}\|_{\max} \le c_V, \quad \text{and} \quad \|B_{\mathbf{i},\mathbf{j}}\|_{1,1} \le c_B K_{\min},$$

   where $K_{\min} := \min_{x\in\mathbf{X}, y\in\mathbf{Y}} |\kappa(x,y)|$ and the positive constants c_U, c_V, c_B are independent of nodes $\mathbf{i}$, $\mathbf{j}$ and level l.
6. For a node $\mathbf{i}$ at level l_1 and a descendant $\mathbf{j}$ of $\mathbf{i}$ at level $l_2 > l_1$, $T_{\mathbf{j},\mathbf{i}}$ as the translation matrix like in (3.3) satisfies

$$T_{\mathbf{j},\mathbf{i}} = T_{\mathbf{j},\mathbf{d}_{l_2}} T_{\mathbf{d}_{l_2},\mathbf{d}_{l_2-1}} \cdots T_{\mathbf{d}_{l_1+1},\mathbf{i}} = \prod_{l=l_2}^{l_1} T_{\mathbf{d}_{l+1},\mathbf{d}_l}, \quad \|T_{\mathbf{j},\mathbf{i}}\|_1 \le 1, \tag{5.1}$$

   where the nodes of $\mathcal{T}$ in the path between $\mathbf{j}$ and $\mathbf{i}$ are $\mathbf{d}_{l_2+1} := \mathbf{j}, \mathbf{d}_{l_2}, \ldots, \mathbf{d}_{l_1+1}$, $\mathbf{d}_{l_1} := \mathbf{i}$, $T_{\mathbf{d}_{l+1},\mathbf{d}_l}$ is a translation matrix from $\mathbf{d}_{l+1}$ to $\mathbf{d}_l$. For example, the product form of the translation matrices in Theorem 3.1 satisfies (3.4), and the norm in Corollary 3.2 satisfies (5.1). $T_{\mathbf{j},\mathbf{i}}$ is a diagonal block of $\prod_{l=l_2+1}^{l_1} R^{(l)}$, where $R^{(l)}$ is defined in (4.6).
7. The error matrix E in (4.3) satisfies

$$|E| \le \varepsilon |K|, \quad \varepsilon > 0. \tag{5.2}$$

   (After the main Theorem 5.8, we will comment on how this condition is related to the two classes of kernels we used before.)
8. The inequality $3\gamma_r \log_2(N) \le 1/2$ holds.

The low-rank factors in item 5 above do not necessarily need to be obtained from Taylor expansions. The inequality in item 8 holds for most practical situations. It serves only as a technical condition in the proof.

The following standard result is for the backward stability of dense matrix-vector multiplications. Throughout this section, we use notation like ΔA in (5.4) below to mean backward error terms. Also, $\mathrm{fl}(\cdot)$ stands for the floating point result of an operator and ϵ_{mach} stands for the machine precision.

LEMMA 5.2. *[20, (3.11) in Section 3.5] Let $A \in \mathbb{C}^{r\times r}$, $q \in \mathbb{C}^r$, and*

$$\gamma_r := \frac{r\epsilon_{\text{mach}}}{1 - r\epsilon_{\text{mach}}}. \tag{5.3}$$

Then the numerical matrix-vector product satisfies

$$\text{fl}(Aq) = (A + \Delta A)q, \tag{5.4}$$

where ΔA satisfies

$$|\Delta A| \le \gamma_r |A|, \quad \|\Delta A\|_1 \le \gamma_r \|A\|_1, \quad \|\Delta A\|_\infty \le \gamma_r \|A\|_\infty.$$

Our stability analysis will involve multiple types of norms and the following norm inequalities will be frequently used later.

LEMMA 5.3. *Let $X \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{n\times r}$, and $Y \in \mathbb{C}^{r\times s}$. Then,*

$$\|XAY\|_{\max} \le \|X\|_\infty \|A\|_{\max} \|Y\|_1 \quad \textit{and} \quad \|XAY\|_{\max} \le \|X\|_{\max} \|A\|_{1,1} \|Y\|_{\max}.$$

Proof. Since $|(XA)_{ij}| = |\sum_k X_{ik}A_{kj}| \le \|X\|_\infty \|A\|_{\max}$, we have

$$\begin{aligned} \|XAY\|_{\max} &\le \|X\|_\infty \|AY\|_{\max} = \|X\|_\infty \|Y^{\mathsf{T}} A^{\mathsf{T}}\|_{\max} \\ &\le \|X\|_\infty \|Y^{\mathsf{T}}\|_\infty \|A^{\mathsf{T}}\|_{\max} = \|X\|_\infty \|A\|_{\max} \|Y\|_1, \\ |(XAY)_{ij}| &= \Big| \sum_{p,k} X_{ip} A_{pk} Y_{kj} \Big| \le \|X\|_{\max} \|A\|_{1,1} \|Y\|_{\max}. \end{aligned}$$

∎

The next proposition results from the matrix form of the FMM and can be verified by induction.

PROPOSITION 5.4. *For $1 < l_1 \le l_2 \le L-1$, the matrix $R^{(l_2,l_1)} := \prod_{l=l_2}^{l_1} R^{(l)}$ with $R^{(l)}$ defined in (4.6) satisfies the following properties.*

- *$R^{(l_2,l_1)}$ is a $4^{l_1} \times 4^{l_1}$ block diagonal matrix with each diagonal block $T_{\mathbf{i}}$ of size $(4^{l_2+1-l_1} r) \times r$ corresponding to a node $\mathbf{i}$ at level l_1.*
- *If $\mathbf{i}$ is at level l_1, $\mathbf{j}$ is at level l_2+1 and is a descendant of $\mathbf{i}$, and the nodes in the path of $\mathcal{T}$ between $\mathbf{j}$ and $\mathbf{i}$ are $\mathbf{j} = \mathbf{d}_{l_2+1}, \mathbf{d}_{l_2}, \ldots, \mathbf{d}_{l_1+1}, \mathbf{i} = \mathbf{d}_{l_1}$, then the $r \times r$ submatrix of $T_{\mathbf{i}}$ corresponding to $\mathbf{j}$ is $T_{\mathbf{j},\mathbf{i}}$ in (5.1).*

Now, we study the backward stability of the FMM. We shall show that its backward error depends logarithmically on N. The stability analysis is done for the key stages of the FMM in Algorithm 4.1. To prove the main result Theorem 5.8, we need Lemma 5.5 to account for the error accumulated in the bottom-up traversal stage, Lemma 5.6 to account for the error accumulated in the top-down traversal stage, and Lemma 5.7 to account for the error accumulated in the evaluation stage. To obtain the error bounds, we repeatedly make use of the following simple identity:

$$\prod_{i=k}^{j} (A_i + \Delta A_i) = \prod_{i=k}^{j} A_i + \sum_{|\boldsymbol{\alpha}|=1}^{k-j+1} \Delta^{\boldsymbol{\alpha}} \left(\prod_{i=k}^{j} A_i \right), \quad k \ge j, \tag{5.5}$$

where $\boldsymbol{\alpha}$ is defined in (1.2) and $\Delta^{\boldsymbol{\alpha}}(\cdot)$ is defined in (1.4). Additionally, we exploit the block structures at each level and non-overlapping nonzero patterns across different levels. The norm bounds of the FMM generators are used to achieve the overall backward error bound. With Assumption 5.1, the techniques used in the proof of Theorem 5.8 does not depend on any specific kernels.

LEMMA 5.5. *With Assumption 5.1, the bottom-up traversal stage of Algorithm 4.1 satisfies*

$$\text{fl}(v^{(l)}) = \left(\sum_{|\boldsymbol{\alpha}|=0}^{L-l} \Delta^{\boldsymbol{\alpha}}\left(R^{(L-1,l)}\right)\right)^{\mathsf{T}} \left(V^{(L)} + \Delta V^{(L)}\right)^{\mathsf{T}} q, \tag{5.6}$$

$$\text{fl}(t^{(l)}) = \left(B^{(l)} + \Delta B^{(l)}\right)\left(\sum_{|\boldsymbol{\alpha}|=0}^{L-l} \Delta^{\boldsymbol{\alpha}}\left(R^{(L-1,l)}\right)\right)^{\mathsf{T}} \left(V^{(L)} + \Delta V^{(L)}\right)^{\mathsf{T}} q, \tag{5.7}$$

where

$$\left|(\Delta V^{(L)})^{\mathsf{T}}\right| \le \gamma_{N_0}\left|(V^{(L)})^{\mathsf{T}}\right|, \quad |\Delta R^{(l)}| \le \gamma_r |R^{(l)}|, \quad \textit{and} \quad |\Delta B^{(l)}| \le \gamma_r |B^{(l)}|. \tag{5.8}$$

Proof. Within this stage, the following floating point operations are performed:

$$\begin{aligned}
&\text{fl}(v^{(L)}) = (V^{(L)} + \Delta V^{(L)})^{\mathsf{T}} q, && |(\Delta V^{(L)})^{\mathsf{T}}| \le \gamma_{N_0}|(V^{(L)})^{\mathsf{T}}|,\\
&\text{fl}(v^{(l)}) = (R^{(l)} + \Delta R^{(l)})^{\mathsf{T}}\,\text{fl}(v^{(l+1)}), && |\Delta R^{(l)}| \le \gamma_r |R^{(l)}|,\\
&\text{fl}(t^{(l)}) = (B^{(l)} + \Delta B^{(l)})\,\text{fl}(v^{(l)}), && |\Delta B^{(l)}| \le \gamma_r |B^{(l)}|,
\end{aligned}$$

where $\Delta V^{(L)}$, $\Delta R^{(l)}$, and $\Delta B^{(l)}$ have the same block structures as $V^{(L)}$, $R^{(l)}$, and $B^{(l)}$, respectively, and the backward error bounds result from Lemma 5.2. Expanding these recurrence relations and applying the identity in (5.5), we have (5.6) and (5.7). ∎

LEMMA 5.6. *With Assumption 5.1, the top-down traversal stage of Algorithm 4.1 satisfies*

$$\begin{aligned}
\text{fl}(u^{(L)}) = \sum_{k=2}^{L} &(M^{(k)} + \Delta M^{(k)})(I + \Delta Z^{(k)})(B^{(k)} + \Delta B^{(k)})(P^{(k)} + \Delta P^{(k)})^{\mathsf{T}}\\
&(V^{(L)} + \Delta V^{(L)})^{\mathsf{T}} q,
\end{aligned} \tag{5.9}$$

where (5.8) *holds,* $M^{(k)} := \tilde{M}^{(L-1,k)}$, $\tilde{M}^{(j)} := R^{(j)}$, $P^{(k)} := R^{(L-1,k)}$,

$$\begin{aligned}
\Delta M^{(k)} &:= \sum_{|\boldsymbol{\alpha}|=1}^{L-k} \Delta^{\boldsymbol{\alpha}}(\tilde{M}^{(L-1,k)}), \quad \Delta P^{(k)} := \sum_{|\boldsymbol{\alpha}|=1}^{L-k} \Delta^{\boldsymbol{\alpha}}(R^{(L-1,k)}),\\
\Delta \tilde{M}^{(j)} &:= \Delta R^{(j)} + \Delta Z^{(j+1)} R^{(j)} + \Delta Z^{(j+1)} \Delta R^{(j)},\\
\Delta Z^{(2)} &:= 0, \quad \textit{and} \quad |\Delta Z^{(k)}| \le \epsilon_{\text{mach}} |I|.
\end{aligned} \tag{5.10}$$

Furthermore, for a given node $\mathbf{i}$ *at level* L*, the blocks of* $M^{(k)}$ *and* $\Delta M^{(k)}$ *associated with the translation from the node* $\mathbf{i}$ *to its parent/ancestor at level* k*, denoted by* $M_{\mathbf{i}}^{(k)}$ *and* $\Delta M_{\mathbf{i}}^{(k)}$ *respectively, satisfy*

$$\|M_{\mathbf{i}}^{(k)}\|_1 \le 1 \quad \textit{and} \quad \|\Delta M_{\mathbf{i}}^{(k)}\|_1 \le 6\gamma_r \log_2(N). \tag{5.11}$$

For a given node $\mathbf{j}$ *at level* L*, the blocks of* $P^{(k)}$ *and* $\Delta P^{(k)}$ *associated with translation from the node* $\mathbf{j}$ *to its parent/ancestor at level* k*, denoted by* $P_{\mathbf{j}}^{(k)}$ *and* $\Delta P_{\mathbf{j}}^{(k)}$ *respectively, satisfy*

$$\|P_{\mathbf{j}}^{(k)}\|_1 \le 1 \quad \textit{and} \quad \|\Delta P_{\mathbf{j}}^{(k)}\|_1 \le 2\gamma_r \log_2(N). \tag{5.12}$$

Proof. Within this stage, the following floating point operations are performed:

$$\begin{aligned}&\mathrm{fl}(u^{(2)}) = \mathrm{fl}(t^{(2)}),\\ &\mathrm{fl}(u^{(l)}) = (I + \Delta Z^{(l)})\left((R^{(l-1)} + \Delta R^{(l-1)})\,\mathrm{fl}(u^{(l-1)}) + \mathrm{fl}(t^{(l)})\right), \quad l > 2.\end{aligned}$$

Expanding these recurrence relations, we have

$$\mathrm{fl}(u^{(L)}) = \sum_{k=2}^{L}\left(\prod_{j=L}^{k+1}(I + \Delta Z^{(j)})(R^{(j-1)} + \Delta R^{(j-1)})\right)(I + \Delta Z^{(k)})\,\mathrm{fl}(t^{(k)}). \tag{5.13}$$

Since $(I+\Delta Z^{(j)})(R^{(j-1)}+\Delta R^{(j-1)}) = \tilde{M}^{(j-1)}+\Delta\tilde{M}^{(j-1)}$, the identity in (5.5) yields

$$\prod_{j=L}^{k+1}(I + \Delta Z^{(j)})(R^{(j-1)} + \Delta R^{(j-1)}) = \tilde{M}^{(L-1,k)} + \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\Delta^{\boldsymbol{\alpha}}\left(\tilde{M}^{(L-1,k)}\right).$$

Plugging the above relation and (5.7) into (5.13), we obtain (5.9).

We now turn to the bounds in (5.11). Let $\mathbf{i}$ be a node at level L. Then, by item 6 of Assumption 5.1 and the definition of $M_{\mathbf{i}}^{(k)}$, we have $\|M_{\mathbf{i}}^{(k)}\|_1 \le 1$. Define $\tilde{M}_{\mathbf{i}}^{(k)}$ to be the block of $\tilde{M}^{(k)}$ corresponding to a node at level k whose descendant at level L is the node $\mathbf{i}$. The blocks $R_{\mathbf{i}}^{(k-1)}$ and $\Delta R_{\mathbf{i}}^{(k-1)}$ are defined similarly. Then, we have

$$\begin{aligned}\left\|\Delta\tilde{M}_{\mathbf{i}}^{(k-1)}\right\|_1 &= \left\|\Delta R_{\mathbf{i}}^{(k-1)} + \Delta Z_{\mathbf{i}}^{(k)} R_{\mathbf{i}}^{(k-1)} + \Delta Z_{\mathbf{i}}^{(k)}\Delta R_{\mathbf{i}}^{(k-1)}\right\|_1\\ &\le \gamma_r + \epsilon_{\mathrm{mach}} + \epsilon_{\mathrm{mach}}\gamma_r \le 3\gamma_r,\end{aligned} \tag{5.14}$$

where $Z_{\mathbf{i}}^{(k)}$ is an identity matrix whose size matches the block $R_{\mathbf{i}}^{(k-1)}$ such that $|\Delta Z_{\mathbf{i}}^{(k)}| \le \epsilon_{\mathrm{mach}}|Z_{\mathbf{i}}^{(k)}|$. In what follows, $\tilde{M}_{\mathbf{i}}^{(L-1,k)}$ stands for the block of $\tilde{M}^{(L-1,k)}$ corresponding to a node at level k whose descendant at level L is the node $\mathbf{i}$. Now, to prove the second inequality in (5.11), we have

$$\begin{aligned}\left\|\Delta M_{\mathbf{i}}^{(k)}\right\|_1 &\le \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\left\|\Delta^{\boldsymbol{\alpha}}\left(\tilde{M}_{\mathbf{i}}^{(L-1,k)}\right)\right\|_1 = \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\left\|\left(\Delta^{\alpha_1}\tilde{M}_{\mathbf{i}}^{(L-1)}\right)\cdots\left(\Delta^{\alpha_{L-k}}\tilde{M}_{\mathbf{i}}^{(k)}\right)\right\|_1\\ &\le \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\left\|\Delta^{\alpha_1}\tilde{M}_{\mathbf{i}}^{(L-1)}\right\|_1\cdots\left\|\Delta^{\alpha_{L-k}}\tilde{M}_{\mathbf{i}}^{(k)}\right\|_1 \le \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\binom{L-k}{|\boldsymbol{\alpha}|}(3\gamma_r)^{|\boldsymbol{\alpha}|}\\ &\le \sum_{|\boldsymbol{\alpha}|=1}^{L-k}(3\gamma_r\log_2(N))^{|\boldsymbol{\alpha}|} \le 3\gamma_r\log_2(N)\sum_{|\boldsymbol{\alpha}|=0}^{\infty}2^{-|\boldsymbol{\alpha}|} = 6\gamma_r\log_2(N),\end{aligned}$$

where we used (5.14) in the third inequality, the fact that $\binom{L-k}{|\boldsymbol{\alpha}|} \le (\log_2(N))^{|\boldsymbol{\alpha}|}$ in the second last inequality and item 8 of Assumption 5.1 in the last inequality. We proved both bounds in (5.11).

Finally, we prove the bounds in (5.12). The calculation is similar to what we did before. By item 6 of Assumption 5.1 and the definition of $P_{\mathbf{j}}^{(k)}$, we have $\left\|P_{\mathbf{j}}^{(k)}\right\|_1 \le 1$. Now, to prove the second inequality in (5.12), we have

$$\left\|\Delta P_{\mathbf{j}}^{(k)}\right\|_1 \le \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\left\|\Delta^{\boldsymbol{\alpha}}\left(R_{\mathbf{j}}^{(L-1,k)}\right)\right\|_1 = \sum_{|\boldsymbol{\alpha}|=1}^{L-k}\left\|\left(\Delta^{\alpha_1}R_{\mathbf{j}}^{(L-1)}\right)\cdots\left(\Delta^{\alpha_{L-k}}R_{\mathbf{j}}^{(k)}\right)\right\|_1$$

$$\leq \sum_{|\boldsymbol{\alpha}|=1}^{L-k} \left\| \left(\Delta^{\alpha_1} R_{\mathbf{j}}^{(L-1)} \right) \right\|_1 \cdots \left\| \left(\Delta^{\alpha_{L-k}} R_{\mathbf{j}}^{(k)} \right) \right\|_1$$
$$\leq \sum_{|\boldsymbol{\alpha}|=1}^{L-k} \binom{L-k}{|\boldsymbol{\alpha}|} \gamma_r^{|\boldsymbol{\alpha}|} \leq \sum_{|\boldsymbol{\alpha}|=1}^{L-k} (\gamma_r \log_2(N))^{|\boldsymbol{\alpha}|}$$
$$\leq \gamma_r \log_2(N) \sum_{|\boldsymbol{\alpha}|=0}^{\infty} 2^{-|\boldsymbol{\alpha}|} \leq 2\gamma_r \log_2(N),$$

where we used the fact that $\binom{L-k}{|\boldsymbol{\alpha}|} \leq (\log_2(N))^{|\boldsymbol{\alpha}|}$ in the second last inequality and item 8 of Assumption 5.1 in the last inequality. We proved both bounds in (5.12). ∎

LEMMA 5.7. *With Assumption 5.1, the evaluation stage of Algorithm 4.1 satisfies*

$$\mathrm{fl}(\phi) = \mathrm{fl}(\tilde{K}q) = (K + \Delta K)q, \tag{5.15}$$

where $\Delta K := -E + \Delta\tilde{K}$, $|E| \leq \varepsilon |K|$,

$$\Delta\tilde{K} := \sum_{k=2}^{L} \sum_{|\boldsymbol{\beta}|=1}^{7} \left(\Delta^{\beta_1} H\right) \left(\Delta^{\beta_2} U^{(L)}\right) \left(\Delta^{\beta_3} M^{(k)}\right) \left(\Delta^{\beta_4} Z^{(k)}\right) \left(\Delta^{\beta_5} B^{(k)}\right) \left(\Delta^{\beta_6} P^{(k)}\right)^{\mathsf{T}} \left(\Delta^{\beta_7} V^{(L)}\right)^{\mathsf{T}} + \Delta K^{(0)} + \Delta H K^{(0)} + \Delta H \Delta K^{(0)}, \tag{5.16}$$

with $\beta_i \in \{0, 1\}$ *for* $1 \leq i \leq 7$, $M^{(k)}$, $\Delta M^{(k)}$, $P^{(k)}$, $\Delta P^{(k)}$ *are defined as in Lemma 5.6, the inequalities in (5.8), (5.10) hold,*

$$|\Delta H| \leq \epsilon_{mach} |I|, \quad |\Delta U^{(L)}| \leq \gamma_r |U^{(L)}|, \quad |\Delta K^{(0)}| \leq \gamma_w |K|,$$

and w *stands for the maximum number of columns of near-field blocks.*

Proof. Within this stage, the following floating point operations are performed:

$$\mathrm{fl}(\phi) = (I + \Delta H) \left((U^{(L)} + \Delta U^{(L)}) \, \mathrm{fl}(u^{(L)}) + (K^{(0)} + \Delta K^{(0)}) q \right),$$

where $\mathrm{fl}(u^{(L)})$ is given in (5.9). By (5.5) and (4.3), we can rewrite $\mathrm{fl}(\phi)$ as

$$\begin{aligned}
\mathrm{fl}(\phi) &= \sum_{k=2}^{L} U^{(L)} R^{(L-1,k)} B^{(k)} (R^{(L-1,k)})^{\mathsf{T}} (V^{(L)})^{\mathsf{T}} q \\
&\quad + \sum_{k=2}^{L} \sum_{|\boldsymbol{\beta}|=1}^{7} \left(\Delta^{\beta_1} H\right) \left(\Delta^{\beta_2} U^{(L)}\right) \left(\Delta^{\beta_3} M^{(k)}\right) \left(\Delta^{\beta_4} Z^{(k)}\right) \left(\Delta^{\beta_5} B^{(k)}\right) \\
&\qquad \left(\Delta^{\beta_6} P^{(k)}\right)^{\mathsf{T}} \left(\Delta^{\beta_7} V^{(L)}\right)^{\mathsf{T}} q \\
&\quad + (K^{(0)} + \Delta K^{(0)} + \Delta H K^{(0)} + \Delta H \Delta K^{(0)}) q, \\
&= (\tilde{K} + \Delta\tilde{K}) q = (K - E + \Delta\tilde{K}) q,
\end{aligned}$$

from which we obtain (5.15). ∎

THEOREM 5.8. *With Assumption 5.1, the FMM matrix-vector product in Algorithm 4.1 satisfies*

$$\text{fl}(\tilde{K}q) = (K + \Delta K)q \quad \textit{with} \tag{5.17}$$
$$|\Delta K| \leq (768r^2 c_U c_B c_V \max\{\gamma_r, \gamma_{N_0}\}\log_2(N) + 4\max\{\varepsilon, \gamma_w, \epsilon_{mach}\})|K|,$$

where $\tilde{K}$ is defined in (4.4) and ε is the relative approximation accuracy of $\tilde{K}$ in (5.2).

Proof. We have computed how the error propagates in Lemma 5.5, Lemma 5.6, and Lemma 5.7. Our goal now is to bound $|\Delta K|$ in Lemma 5.7. To this end, we want to find an entrywise upper bound for each term in the double summation of (5.16) by considering the following node-wise inequality from a leaf node $\mathbf{j}$ at level L to the leaf node $\mathbf{i}$ at level L (via the bottom-up and top-down traversals). The subscript $\mathbf{i},\mathbf{j}$ are used to emphasize the dependence on these leaf nodes.

Since $|\Delta U^{(L)}| \leq \gamma_r |U^{(L)}|$, $\left|\left(\Delta V^{(L)}\right)^{\mathsf{T}}\right| \leq \gamma_{N_0}\left|\left(V^{(L)}\right)^{\mathsf{T}}\right|$, and the numbers of columns $U_{\mathbf{i}}^{(L)}$, $V_{\mathbf{j}}^{(L)}$ are both equal to r, and item 5 of Assumption 5.1 holds, we have

$$\left\|\Delta U_{\mathbf{i}}^{(L)}\right\|_\infty \leq \gamma_r r c_U \quad \text{and} \quad \left\|\left(\Delta V_{\mathbf{j}}^{(L)}\right)^{\mathsf{T}}\right\|_1 \leq \gamma_{N_0} r c_V. \tag{5.18}$$

for given nodes $\mathbf{i},\mathbf{j}$ at level L.

In what follows, let $H_{\mathbf{i}}$ be an identity matrix whose size matches the block corresponding to the node $\mathbf{i}$ at level L of the matrix $U^{(L)}$ (i.e., $U_{\mathbf{i}}^{(L)}$), which satisfies $|\Delta H_{\mathbf{i}}| \leq \epsilon_{\text{mach}}|H_{\mathbf{i}}|$. Furthermore, let $Z_{\mathbf{i}}^{(k)}$ be an identity matrix whose size matches the block of the matrix $B^{(k)}$ containing the interaction at level k of parent nodes of $\mathbf{i},\mathbf{j}$ (i.e., $B_{\mathbf{i},\mathbf{j}}^{(k)}$), which satisfies $|\Delta Z_{\mathbf{i}}^{(k)}| \leq \epsilon_{\text{mach}}|Z_{\mathbf{i}}^{(k)}|$.

Now, by a direct calculation,

(5.19)

$$\begin{aligned}
&\left\|(\Delta^{\beta_1}H_{\mathbf{i}})\left(\Delta^{\beta_2}U_{\mathbf{i}}^{(L)}\right)\left(\Delta^{\beta_3}M_{\mathbf{i}}^{(k)}\right)\left(\Delta^{\beta_4}Z_{\mathbf{i}}^{(k)}\right)\left(\Delta^{\beta_5}B_{\mathbf{i},\mathbf{j}}^{(k)}\right)\left(\Delta^{\beta_6}P_{\mathbf{j}}^{(k)}\right)^{\mathsf{T}}\left(\Delta^{\beta_7}V_{\mathbf{j}}^{(L)}\right)^{\mathsf{T}}\right\|_{\max} \\
&\leq \left\|(\Delta^{\beta_1}H_{\mathbf{i}})\left(\Delta^{\beta_2}U_{\mathbf{i}}^{(L)}\right)\right\|_\infty \left\|\left(\Delta^{\beta_3}M_{\mathbf{i}}^{(k)}\right)\left(\Delta^{\beta_4}Z_{\mathbf{i}}^{(k)}\right)\left(\Delta^{\beta_5}B_{\mathbf{i},\mathbf{j}}^{(k)}\right)\left(\Delta^{\beta_6}P_{\mathbf{j}}^{(k)}\right)^{\mathsf{T}}\right\|_{\max} \\
&\qquad \cdot \left\|\left(\Delta^{\beta_7}V_{\mathbf{j}}^{(L)}\right)^{\mathsf{T}}\right\|_1 \\
&\leq \left\|(\Delta^{\beta_1}H_{\mathbf{i}})\left(\Delta^{\beta_2}U_{\mathbf{i}}^{(L)}\right)\right\|_\infty \left\|\left(\Delta^{\beta_3}M_{\mathbf{i}}^{(k)}\right)\left(\Delta^{\beta_4}Z_{\mathbf{i}}^{(k)}\right)\right\|_{\max} \left\|\Delta^{\beta_5}B_{\mathbf{i},\mathbf{j}}^{(k)}\right\|_{1,1} \\
&\qquad \cdot \left\|\left(\Delta^{\beta_6}P_{\mathbf{j}}^{(k)}\right)^{\mathsf{T}}\right\|_{\max} \left\|\left(\Delta^{\beta_7}V_{\mathbf{j}}^{(L)}\right)^{\mathsf{T}}\right\|_1 \\
&\leq \left\|(\Delta^{\beta_1}H_{\mathbf{i}})\right\|_\infty \left\|\left(\Delta^{\beta_2}U_{\mathbf{i}}^{(L)}\right)\right\|_\infty \left\|\Delta^{\beta_3}M_{\mathbf{i}}^{(k)}\right\|_1 \left\|\Delta^{\beta_4}Z_{\mathbf{i}}^{(k)}\right\|_1 \left\|\Delta^{\beta_5}B_{\mathbf{i},\mathbf{j}}^{(k)}\right\|_{1,1} \\
&\qquad \cdot \left\|\left(\Delta^{\beta_6}P_{\mathbf{j}}^{(k)}\right)^{\mathsf{T}}\right\|_{\max} \left\|\left(\Delta^{\beta_7}V_{\mathbf{j}}^{(L)}\right)^{\mathsf{T}}\right\|_1 \\
&\leq \epsilon_{\text{mach}}^{\beta_1}\left(\gamma_r^{\beta_2} r c_U\right)\left(6\gamma_r\log_2(N)\right)^{\beta_3}\epsilon_{\text{mach}}^{\beta_4}\left(\gamma_r^{\beta_5} c_B K_{\min}\right)\left(2\gamma_r\log_2(N)\right)^{\beta_6}\left(\gamma_{N_0}^{\beta_7} r c_V\right) \\
&\leq r^2 c_U c_V c_B K_{\min}\epsilon_{\text{mach}}^{\beta_1+\beta_4}\gamma_r^{\beta_2}\left(6\gamma_r\log_2(N)\right)^{\beta_3+\beta_6}\gamma_r^{\beta_5}\gamma_{N_0}^{\beta_7},
\end{aligned}$$

where we repeatedly applied (5.3) to the first two inequalities, and the bounds in (5.11), (5.12) to the last inequality. We also observe that the inner summation of the first term in (5.16) produces the same block structure as $U^{(k)}B^{(k)}\left(V^{(k)}\right)^{\mathsf{T}}$. This means that for different levels $k = 2, \ldots, L$, the nonzero patterns of this inner summation do not overlap. Therefore,

(5.20)

$$
\begin{aligned}
|\Delta K| &\le |E| + \sum_{k=2}^{L} \Bigg| \sum_{|\boldsymbol{\beta}|=1}^{7} \left(\Delta^{\beta_1} H\right) \left(\Delta^{\beta_2} U^{(L)}\right) \left(\Delta^{\beta_3} M^{(k)}\right) \left(\Delta^{\beta_4} Z^{(k)}\right) \left(\Delta^{\beta_5} B^{(k)}\right) \\
&\qquad \cdot \left(\Delta^{\beta_6} P^{(k)}\right)^{\mathsf{T}} \left(\Delta^{\beta_7} V^{(L)}\right)^{\mathsf{T}} \Bigg| + |\Delta K^{(0)}| + |\Delta H K^{(0)}| + |\Delta H \Delta K^{(0)}| \\
&\le |E| + \max_{k=2,\ldots,L} \Bigg| \sum_{|\boldsymbol{\beta}|=1}^{7} \left(\Delta^{\beta_1} H\right) \left(\Delta^{\beta_2} U^{(L)}\right) \left(\Delta^{\beta_3} M^{(k)}\right) \left(\Delta^{\beta_4} Z^{(k)}\right) \left(\Delta^{\beta_5} B^{(k)}\right) \\
&\qquad \cdot \left(\Delta^{\beta_6} P^{(k)}\right)^{\mathsf{T}} \left(\Delta^{\beta_7} V^{(L)}\right)^{\mathsf{T}} \Bigg| + |\Delta K^{(0)}| + |\Delta H K^{(0)}| + |\Delta H \Delta K^{(0)}| \\
&\le \varepsilon |K| + \sum_{|\boldsymbol{\beta}|=1}^{7} \binom{7}{|\boldsymbol{\beta}|} \left(r^2 c_U c_V c_B \epsilon_{\text{mach}}^{\beta_1+\beta_4} \gamma_r^{\beta_2} \left(6\gamma_r \log_2(N)\right)^{\beta_3+\beta_6} \gamma_r^{\beta_5} \gamma_{N_0}^{\beta_7} \right) |K| \\
&\qquad + (\gamma_w + \epsilon_{\text{mach}} + \epsilon_{\text{mach}}\gamma_w)|K| \\
&\le (2^7)(6) r^2 c_U c_V c_B \max\{\gamma_r, \gamma_{N_0}\} \log_2(N) |K| + (\varepsilon + \gamma_w + \epsilon_{\text{mach}} + \epsilon_{\text{mach}}\gamma_w)|K| \\
&\le (768 r^2 c_U c_V c_B \max\{\gamma_r, \gamma_{N_0}\} \log_2(N) + 4\max\{\varepsilon, \gamma_w, \epsilon_{\text{mach}}\})|K|,
\end{aligned}
$$

where we used the node-wise estimate in (5.19) to arrive at the third inequality. This completes the proof. ∎

The theorem shows that, with proper norm bounds of the generators, the FMM is backward stable and the backward error depends logarithmically on the size of K. This result clearly highlights an advantage of using a hierarchical structure matrix method over the standard matrix-vector multiplication, where the backward error depends linearly on N (see Lemma 5.2). Such a stability advantage has also been observed in some other hierarchical structured matrix algorithms in [2, 33, 34].

The theorem does not rely on a specific kernel and instead only needs certain norm bounds for the FMM generators. Our stabilization strategy essentially ensures such norm bounds. The following corollary states the backward errors of the FMM for the generalized Cauchy kernel and the logarithmic kernel discussed in section 2. For the logarithmic kernel, even though the bounds of $\|B_{\mathbf{i},\mathbf{j}}\|_{1,1}$ and $|E|$ are slightly different from items 5 and 7 of Assumption 5.1, respectively, we can follow the same proof idea.

COROLLARY 5.9. *Suppose all the conditions in Assumption 5.1 hold, except items 5 and 7. Let τ be the separation ratio in (2.1) and ε be a given error tolerance.*

- *If $\kappa(x, y) = (x - y)^{-(1+d)}$, $d \in \mathbb{Z}$ and the expansion order r is chosen such that E in (4.3) satisfies $|E| \le \varepsilon |K|$, then the FMM matrix-vector product in Algorithm 4.1 satisfies (5.17) with*

$$
|\Delta K| \le \left(\frac{768 r^2 \max\{\gamma_r, \gamma_{N_0}\} \log_2(N)}{(1-\tau)^{2+2|d|}} + 4\max\{\varepsilon, \gamma_w, \epsilon_{mach}\} \right) |K|.
$$

- *If $\kappa(x,y) = -\log|x-y|$ and the expansion order r is chosen such that $|E| \le \varepsilon$ in (4.3), then the FMM matrix-vector product in Algorithm 4.1 satisfies (5.17) with*

$$|\Delta K| \le \left(768r^2 \max\{\gamma_r, \gamma_{N_0}\}\log_2(N) + 3\max\{\gamma_w, \epsilon_{mach}\}\right)|K| + \varepsilon \\ - 1536r^2 \max\{\gamma_r, \gamma_{N_0}\}\log(1-\tau).$$

Proof. If $\kappa(x,y) = (x-y)^{-(1+d)}$, $d \in \mathbb{Z}$, by Theorem 2.3 and Theorem 3.1, we have $c_U = c_V = 1$ and $c_B = (1-\tau)^{-(2+2|d|)}$ in Theorem 5.8. These yield the desired result.

If $\kappa(x,y) = -\log|x-y|$, by Theorem 2.5 and Theorem 3.1, we have $c_U = c_V = 1$ and $\|B_{\mathbf{i},\mathbf{j}}\|_{1,1} \le |K| + 2\log\frac{1}{1-\tau}$, where $\mathrm{lv}(\mathbf{i}) = l$ and $\mathbf{j} \in \mathcal{L}_{\mathbf{i}}$ with $l = 2, \ldots, L$. The ideas used in proving Theorem 5.8 are still valid. We just need to appropriately replace $|K|$ in the second inequality of (5.20) with $|K| + 2\log\frac{1}{1-\tau}$ and use the bound $|E| \le \varepsilon$ to get the desired result. ∎

The choices of the expansion order r in the above corollary are guaranteed by Theorem 2.2 and Theorem 2.4. Note that the backward error for the logarithmic kernel in Corollary 5.9 is not exactly a relative error bound, due to the bounds of the error matrix in Theorem 2.4 and $\|B\|_{1,1}$ in Theorem 2.5.

6. Numerical experiments. We now present two sets of numerical experiments to validate our theoretical findings. In the first set of numerical experiments, we verify how the norms of the FMM generators are controlled as discussed in sections 2 and 3. In the second set of numerical experiments, we validate the backward error bound discussed in section 5.

6.1. Norm bounds of FMM generators. We randomly generate two sets $\mathbf{X}$ and $\mathbf{Y}$ such that each set contains 150^2 points. The real and imaginary parts of each $x \in \mathbf{X}$ are independently sampled from the standard normal distribution, shift, and scale them so that $\Re(x), \Im(x) \in [0, 400]$. Each point in $\mathbf{Y}$ is generated the same way. Afterwards, we further scale both sets in various ways to simulate situations that lead to instabilities (see Tables 6.1 to 6.3). The first situation is when there is a small distance between the centers (Table 6.1), while the second situation is when there is a large distance between the points and its center (Table 6.2). Both situations may also simultaneously arise as in Table 6.3. Our goal is to observe how the scaled sets affect the magnitude of the far-field FMM generators and consequently the result of the matrix-vector multiplication.

We set each partition at the leaf level to contain at most $N_0 = 2^5$ points. For each test below, we report the expansion order (numerical rank) r, the accuracy of the FMM matrix-vector multiplication, and the following maximum norms:

$$\mathcal{U} := \max(\max_{\mathbf{i}\in\mathcal{T}} \|U_{\mathbf{i}}\|_{\max}, \max_{\mathbf{i}\in\mathcal{T}} \|V_{\mathbf{i}}\|_{\max}), \quad \mathcal{B} := \max_{\mathbf{i}\in\mathcal{T}} \|B_{\mathbf{i}}\|_{\max}, \quad \mathcal{R} := \max_{\mathbf{i}\in\mathcal{T}} \|T_{\mathbf{i}}\|_{\max}.$$

For a given vector q, recall that ϕ is the direct dense matrix-vector product in (1.1) and $\tilde{\phi}$ is the result from the FMM as described in Algorithm 4.1. To assess the accuracy of the FMM matrix-vector multiplication, we record the relative error $\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$ with q being a vector of randomly generated numbers sampled from the standard normal distribution.

We conduct three representative tests for the generalized Cauchy kernel (see Table 6.1 for $d = 0$ and Table 6.3 for $d = 1$) and the logarithmic kernel (see Table 6.2).

Even though we only present tests for the generalized Cauchy kernel with $d = 0, 1$, similar stability issues are encountered for other choices of d. All the computations are done in Matlab. Each of the tables has two columns that are labeled as 'Balanced' and 'Original'. The results reported in the 'Balanced' column correspond to the case, where the far-field FMM generators take the forms of (2.4) and (2.5) for the generalized Cauchy kernel and the forms of (2.4) and (2.18) for the logarithmic kernel. Meanwhile, the results reported in the 'Original' column correspond to the case, where the far-field FMM generators take the forms of (2.13) and (2.14) for the generalized Cauchy kernel and the forms of (2.13) and (2.19) for the logarithmic kernel.

TABLE 6.1

Comparison of the balanced and the original versions of the FMM for the generalized Cauchy kernel with the sets $10^{-4}\mathbf{X}$ *and* $10^{-4}\mathbf{Y}$ *(i.e., we multiply the randomly generated sets* $\mathbf{X}$ *and* $\mathbf{Y}$ *by a factor of* 10^{-4}*), where* $d = 0$, $\tau = 0.6$, $N_0 = 2^5$, *and* $L = 8$. *See Figure 6.1 for a hierarchical partitioning of a square domain containing* $10^{-4}\mathbf{X}$ *and* $10^{-4}\mathbf{Y}$.

	Balanced				Original			
r	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$
10	1	1	1.1E+3	5.9E-6	1	1	1.1E+36	5.9E-6
20	1	1	1.1E+3	5.6E-9	1	1	1.2E+78	5.6E-9
30	1	1	1.1E+3	1.7E-11	1	1	2.7E+122	1.7E-11
40	1	1	1.1E+3	4.4E-14	1	1	1.9E+168	4.5E-14
50	1	1	1.1E+3	4.6E-15	1	1	1.8E+215	4.6E-15
60	1	1	1.1E+3	4.6E-15	1	1	1.3E+263	4.6E-15
70	1	1	1.1E+3	4.6E-15	1	1	Inf	NaN
80	1	1	1.1E+3	4.6E-15	1	1	Inf	NaN
90	1	1	1.1E+3	4.6E-15	1	1	Inf	NaN
100	1	1	1.1E+3	4.6E-15	1	1	Inf	NaN

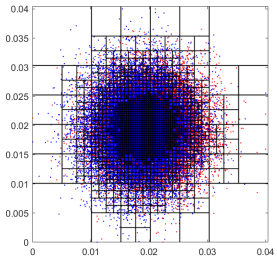


FIG. 6.1. *A hierarchical partitioning of a square domain, which contains points from* $\mathbf{X}$ *(in red) and points from* $\mathbf{Y}$ *(in red) that are scaled differently as described in Table 6.1.*

Recall that the entries (2.14) and (2.19) may become very large if the distance between the centers is small and/or r is large. Hence, it is not surprising that some results of $\mathcal{B}$ and $\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$ in the 'Original' column of Tables 6.1 to 6.3 appear as Inf and NaN in Matlab. This is because Matlab encounters overflow. A similar problem is encountered for the 'Original' column if the distance between some points in a set and the center of this set is very large. From Tables 6.2 and 6.3, we see that the entries of (2.13) and translation matrices reported in the 'Original' column become very large that we run into the same overflow issue.

TABLE 6.2

Comparison of the balanced and the original versions of the FMM for the logarithmic kernel with randomly generated sets $10^2\mathbf{X}$ and $10^2\mathbf{Y}$ (i.e, we multiply the randomly generated sets $\mathbf{X}$ and $\mathbf{Y}$ by a factor of 10^2), where $\tau = 0.6$, $N_0 = 2^5$, and $L = 8$. The hierarchical partitioning of a square domain containing $10^2\mathbf{X}$ and $10^2\mathbf{Y}$ looks the same as Figure 6.1 except the scales of the axes are different.

	Balanced				Original			
r	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$
10	1	1	1.1E+1	2.5E-7	1.1E+29	1.3E+29	1.1E+1	2.5E-7
20	1	1	1.1E+1	1.2E-10	9.2E+55	1.4E+56	1.1E+1	1.2E-10
30	1	1	1.1E+1	6.1E-13	3.6E+80	6.5E+80	1.1E+1	6.1E-13
40	1	1	1.1E+1	1.3E-14	4.3E+103	9.7E+103	1.1E+1	1.3E-14
50	1	1	1.1E+1	1.3E-14	4.08E+125	1.1E+126	1.1E+1	1.3E-14
60	1	1	1.1E+1	1.3E-14	5.0E+146	1.7E+147	1.1E+1	1.3E-14
70	1	1	1.1E+1	1.3E-14	1.1E+167	4.8E+167	1.1E+1	1.3E-14
80	1	1	1.1E+1	1.3E-14	9.1E+182	3.2E+187	1.1E+1	1.3E-14
90	1	1	1.1E+1	1.3E-14	Inf	Inf	1.1E+1	NaN
100	1	1	1.1E+1	1.3E-14	Inf	Inf	1.1E+1	NaN
110	1	1	1.1E+1	1.3E-14	Inf	Inf	1.1E+1	NaN

TABLE 6.3

Comparison of the balanced and the original versions of the FMM for the generalized Cauchy kernel with points in sets $\mathbf{X}$ and $\mathbf{Y}$ scaled differently so that some points are tightly clustered, where $d = 1$, $\tau = 0.6$, $N_0 = 2^5$, and $L = 26$. See Figure 6.2 for a hierarchical partitioning of a square domain containing points from sets $\mathbf{X}$ and $\mathbf{Y}$ that are scaled differently.

	Balanced				Original			
r	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$	$\mathcal{U}$	$\mathcal{R}$	$\mathcal{B}$	$\frac{\|\tilde{\phi}-\phi\|_2}{\|\phi\|_2}$
10	1	1	6.3E+4	7.3E-8	2.7E+28	7.5E+28	8.8E+32	7.3E-8
20	1	1	6.3E+4	1.7E-10	4.7E+54	4.1E+55	5.7E+68	1.7E-10
30	1	1	6.3E+4	5.3E-13	3.8E+78	1.0E+80	6.0E+106	5.3E-13
40	1	1	6.3E+4	2.1E-15	9.6E+100	8.2E+102	1.8E+146	2.1E-15
50	1	1	6.3E+4	1.3E-15	1.9E+122	5.0E+124	6.4E+186	1.3E-15
60	1	1	6.3E+4	1.3E-15	4.8E+142	4.0E+145	1.7E+228	1.3E-15
70	1	1	6.3E+4	1.3E-15	2.3E+162	6.0E+165	2.3E+270	1.3E-15
80	1	1	6.3E+4	1.3E-15	2.5E+181	2.1E+185	Inf	NaN
90	1	1	6.3E+4	1.3E-15	Inf	Inf	Inf	NaN
100	1	1	6.3E+4	1.8E-15	Inf	Inf	Inf	NaN
110	1	1	6.3E+4	1.8E-15	Inf	Inf	Inf	NaN

In particular, Table 6.3 describes a situation where the 'Original' version simultaneously encounters two of the foregoing issues. The corresponding point sets $\mathbf{X}$ and $\mathbf{Y}$ are shown in Figure 6.2 together with the adaptive hierarchical partitioning of the domain.

In comparison, regardless how the sets are scaled, the FMM that uses our stabilization strategy in section 2 (see the 'Balanced' column) automatically handles the scaling effects and overcomes these instability/overflow issues.

We also comment that, before the 'Original' version encounters overflow (when r is not very large) in these tests, it is indeed able to achieve good accuracies. However,

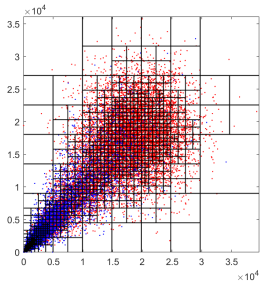


FIG. 6.2. *A hierarchical partitioning of a square domain, which contains points from* $\mathbf{X}$ *(in red) and points from* $\mathbf{Y}$ *(in red) that are scaled differently (with the aid of nonlinear transformations) as described in Table 6.3.*

like shown in [6], the corresponding FMM matrix is still susceptible to stability issues. If some other operations (like reorthogonalization and recompression) are applied to the generators produced by the 'Original' version, then the accuracy of the FMM can be quite poor.

6.2. Backward errors. Our next goal is to observe how the backward errors behave as the level changes. We first describe the setup and provide more explanations after. We randomly generate two new sets $\mathbf{X}$ and $\mathbf{Y}$ such that each set contains 2^{10} points. The real and imaginary parts of each point in $\mathbf{X}$ and $\mathbf{Y}$ are randomly sampled from the uniform distribution defined on the unit interval. Then, we define

$$\mathbf{X}_l := 2^{-l}\mathbf{X} \quad \text{and} \quad \mathbf{Y}_l := 2^{-l}\mathbf{Y} + \left(\frac{2^l-1}{2^l} + \frac{2^l-1}{2^l}\mathrm{i}\right), \quad l = 3, \ldots, 21,$$

where i stands for the imaginary number. Clearly, $\mathbf{X}_l \cup \mathbf{Y}_l$, is contained in $\mathbf{S} := \{z \in \mathbb{C} : \Re(z), \Im(z) \in [0,1]\}$. Next, we define $\mathbf{i}_k := \{z \in \mathbf{S} : \Re(z), \Im(z) \in [0, 2^{-k}]\}$ and $\mathbf{j}_k := \{z \in \mathbf{S} : \Re(z), \Im(z) \in [1-2^{-k}, 1]\}$ for $k \geq 3$. As before, $o_{\mathbf{i}_k}$ and $\delta_{\mathbf{i}_k}$ respectively denote the center and radius of the node $\mathbf{i}_k$. A similar thing holds for $o_{\mathbf{j}_k}$ and $\delta_{\mathbf{j}_k}$. We can directly see that $\mathbf{X}_l$ is contained in the node $\mathbf{i}_l$, while $\mathbf{Y}_l$ is contained in the node $\mathbf{j}_l$.

To compute the backward errors, we follow the formula stated in [21, (3.6)]. That is, given a randomly generated vector q, we compute

(6.1)
$$\mathcal{E}_l := \max_{1\leq i\leq 2^{10}} \frac{|(U_{\mathbf{X}_l,\mathbf{i}_3} B_{\mathbf{i}_3,\mathbf{j}_3} V^{\mathsf{T}}_{\mathbf{Y}_l,\mathbf{j}_3} q - U_{\mathbf{X}_l} T_{\mathbf{i}_l,\mathbf{i}_{l-1}} \ldots T_{\mathbf{i}_4,\mathbf{i}_3} B_{\mathbf{i}_3,\mathbf{j}_3} T^{\mathsf{T}}_{\mathbf{j}_4,\mathbf{j}_3} \ldots T^{\mathsf{T}}_{\mathbf{j}_l,\mathbf{j}_{l-1}} V^{\mathsf{T}}_{\mathbf{Y}_l} q)_i|}{(|U_{\mathbf{X}_l,\mathbf{i}_3} B_{\mathbf{i}_3,\mathbf{j}_3} V^{\mathsf{T}}_{\mathbf{Y}_l,\mathbf{j}_3}||q|)_i},$$

where

$$U_{\mathbf{X}_l,\mathbf{i}_3} := \left[\left(\frac{x - o_{\mathbf{i}_3}}{\delta_{\mathbf{i}_3}}\right)^j\right]_{x\in\mathbf{X}_l, 0\leq j\leq r-1}, \quad U_{\mathbf{X}_l} := \left[\left(\frac{x - o_{\mathbf{i}_l}}{\delta_{\mathbf{i}_l}}\right)^j\right]_{x\in\mathbf{X}_l, 0\leq j\leq r-1},$$

$V_{\mathbf{Y}_l,\mathbf{j}_3}, V_{\mathbf{Y}_l}$ are defined similarly, $B_{\mathbf{i}_3,\mathbf{j}_3}$ takes the form of (2.5) or (2.18) (its entries depend on the radii and centers of the nodes $\mathbf{i}_3$ and $\mathbf{j}_3$), $T_{\mathbf{i}_l,\mathbf{i}_{l-1}}$ takes the form of (3.3) (its entries also depend on the radii and centers of the nodes $\mathbf{i}_l$ and $\mathbf{i}_{l-1}$), and the subscript i denotes the i-th component of a vector. By convention, we set $\mathcal{E}_3$:=0.

This setup aims to simulate the case where we deal with a kernel matrix that is increasing in size by fixing the number of points in the leaf nodes and consequently

letting the number of levels grow. We focus on a single block of the kernel matrix and compute the product of this block with a vector. To eliminate the influence of the approximation error, we treat the low-rank approximation to this block at the top level, $l = 3$, as the exact form. We then compare this product against the product of the low-approximation of this block obtained via the nested relation with the same vector. For simplicity, we assume that all points are contained in the unit square. So, to allow the growing number of levels in the experiment, we make the leaf nodes sets $\mathbf{X}_l$ and $\mathbf{Y}_l$ to be increasingly clustered at the bottom left and top right corners of the domain.

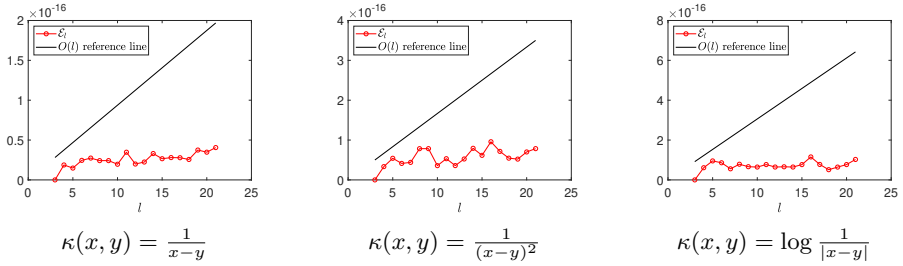


FIG. 6.3. *Illustrations of the behaviors of the backward errors as the number of levels l increases. We fix the rank to be $r = 128$.*

7. Conclusions. This work gives a stabilization strategy to overcome a stability issue in the FMM and then justify the backward stability in terms of an intuitive matrix version FMM. This stabilization strategy amounts to balancing relevant low-rank factors. Even though we focus on two types of important kernels, the techniques are applicable to other kernels whose degenerate expansions are based on Taylor expansions. We have proved that the resulting low-rank factors have bounded norms and entries. Additionally, the entries of these low-rank factors can be computed conveniently by recurrence relations. We have also presented a matrix version of the 2D in an intuitive manner in hopes of making the FMM more accessible to a larger scientific computing community. The long-overdue rigorous analysis on the backward stability of the FMM is then performed and demonstrates the stability advantage of such hierarchical structured algorithms. In our forthcoming work [24], we shall rigorously study a stable low-rank approximation for an oscillating kernel such as the 2D Helmholtz kernel.

Acknowledgement. The authors would like to thank the two anonymous referees for some insightful suggestions.

REFERENCES

[1] M. BEBENDORF, *Approximation of boundary element matrices*, Numer. Math, 86 (2000), pp. 565–589.
[2] T. BELLA, V. OLSHEVSKY, AND M. STEWART, *Nested product decomposition of quasiseparable matrices*, SIAM. J. Matrix Anal. Appl., 34 (2013), pp. 1520–1555.
[3] A. BRANDT, *Multilevel computations of integral transforms and particle interactions with oscillatory kernels*, Comput. Phys. Commun., 65 (1991), pp. 24–38.
[4] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501-517.
[5] M. D. BUHMANN, *Radial basis functions*, 1st ed, Cambridge University Press, 2003.
[6] D. CAI AND J. XIA, *A stable matrix version of the fast multipole method: stabilization strategies and examples*, Electron. Trans. Numer. Anal., 54 (2021), pp. 581–609.

[7] H. Cheng, W. Crutchfield, Z. Gimbutas, L. Greengard, J. Huang, V. Rokhlin, N. Yarvin, J. Zhao, *Remarks on the implementation of the wideband FMM for the Helmholtz equation in two dimensions*, Inverse problems, multi-scale analysis and efffective medium theory, Contemp. Math., 408, Amer. Math. Soc., Providence, RI, (2006), pp. 99–110.

[8] M. H. Cho, J. Huang, D. Chen, and W. Cai, *A heterogeneous FMM for layered media Helmholtz equation I: Two layers in* $\mathbb{R}^2$, J. Comput. Phys., 369 (2018), 237–251.

[9] P. J. Davis, *Interpolation and approximation*, 1st ed., Dover Publications, 1963.

[10] E. Darve, *The fast multipole method: numerical implementation*, J. Comput. Phys., 160 (2000), pp. 195–240.

[11] W. Fong and E. Darve, *The black-box fast multipole method*, J. Comput. Phys., 228 (2009), pp. 8712–8725.

[12] M. Gasca and T. Sauer, *Polynomial interpolation in several variables*, Adv. Comput. Math., 12 (2000), pp. 377-410.

[13] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd edition, Reading, MA, Addison-Wesley, (1994).

[14] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[15] L. Greengard and V. Rokhlin, *On the efficient implementation of the fast multipole algorithm*, Technical Report RR-602, Department of Computer Science, Yale University, New Haven, (1988).

[16] L. Greengard and V. Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer., (1997), pp. 229–269.

[17] M. Gu and S. C. Eisenstat, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276.

[18] W. Hackbusch, *A sparse matrix arithmetic based on* $\mathcal{H}$*-matrices*, Computing, 62 (1999), pp. 89–108.

[19] W. Hackbusch and S. Börm, *Data-sparse approximation by adaptive* $\mathcal{H}^2$*-matrices*, Computing 69, (2002), pp. 1–35.

[20] N. J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[21] N. J. Higham and T. Mary, *Sharper probabilistic backward error analysis for basic linear algebra kernels with random data*, SIAM J. Sci. Comput., 42, (2020), no. 5, pp. A3427-A3446.

[22] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, and S. Jiang, *A fast multipole method for the Rotne-Prager-Yamakawa tensor and its applications*, J. Comput. Phys., 234 (2013), pp. 133–139.

[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] M. Michelle, X. Ou, and J. Xia, *A stable matrix version of the fast multipole method for the 2D Helmholtz kernel*, preprint.

[25] V. Y. Pan, *Transformations of matrix structures work again*, Linear Algebra Appl., 465 (2015), pp. 107–138.

[26] X. Ou and J. Xia, *SuperDC: superfast divide-and-conquer eigenvalue decomposition for rank-structured matrices*, SIAM J. Sci. Comput., 44 (2022), pp. A3041–A3066.

[27] V. Rokhlin, *Rapid solution of integral equations of scattering theory in two dimensions*, J. Comput. Phys., 86 (1990), pp. 414–439.

[28] X. Sun and N. P. Pitsianis, *A matrix version of the fast multipole method*, SIAM Rev., 43 (2001), pp. 289–300.

[29] D. Sushnikova, L. Greengard, M. O'Neil, and M. Rachh, *FMM-LU: a fast direct solver for multiscale boundary integral eqautions in three dimensions*, Multiscale Model. Simul., 21 (2023), no. 4, pp. 1570–1601.

[30] 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.

[31] L. Wang, R. Krasny, and S. Tlupova, *A kernel-independent treecode based on barycentric lagrange interpolation*, Commun. Comput. Phys., 28(4), (2020), pp. 1415-1436.

[32] L. Wilson, N. Vaughn, R. Krasny, *A GPU-accelerated fast multipole method based on barycentric Lagrange interpolation and dual tree traversal*, Comput. Phys. Commun., 265 (2021), 108017.

[33] Y. Xi and J. Xia, *On the stability of some hierarchical rank structured matrix algorithms*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1279–1303.

[34] Y. Xi, J. Xia, S. Cauley, and V. Balakrishnan, *Superfast and stable structured solvers for Toeplitz least squares via randomized sampling*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 44–72.
[35] 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.
[36] A. Yesypenko, C. Chen, P.-G. Martinsson, *SkelFMM: a simplified fast multipole method based on recursive skeletonization*, arXiv:2310.16668v1, pp 1–22.
[37] 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.