SIAM J. Matrix Anal. Appl.
Vol. 41, No. 3, pp. 1059–1085

ANALYTICAL LOW-RANK COMPRESSION VIA PROXY POINT SELECTION*

XIN YE[†], JIANLIN XIA[†], AND LEXING YING[‡]

Abstract. It has been known in potential theory that, for some kernel matrices corresponding to well-separated point sets, fast analytical low-rank approximation can be achieved via the use of proxy points. This proxy point method gives a surprisingly convenient way of explicitly writing out approximate basis matrices for a kernel matrix. However, this elegant strategy is rarely known or used in the numerical linear algebra community. It still needs clear algebraic understanding of the theoretical background. Moreover, rigorous quantifications of the approximation errors and reliable criteria for the selection of the proxy points are still missing. In this work, we use contour integration to clearly justify the idea in terms of a class of important kernels. We further provide comprehensive accuracy analysis for the analytical compression and show how to choose nearly optimal proxy points. The analytical compression is then combined with fast rank-revealing factorizations to get compact low-rank approximations and also to select certain representative points. We provide the error bounds for the resulting overall low-rank approximation. This work thus gives a fast and reliable strategy for compressing those kernel matrices. Furthermore, it provides an intuitive way of understanding the proxy point method and bridges the gap between this useful analytical strategy and practical low-rank approximations. Some numerical examples help to further illustrate the ideas.

Key words. kernel matrix, proxy point method, low-rank approximation, approximation error analysis, hybrid compression, strong rank-revealing factorization

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

DOI. 10.1137/19M1247838

1. Introduction.

In this paper, we focus on the low-rank approximation of some kernel matrices: those generated by a smooth kernel function $\kappa(x,y)$ evaluated at two well-separated sets of points $X=\{x_j\}_{j=1}^m$ and $Y=\{y_j\}_{j=1}^n$. We suppose $\kappa(x,y)$ is analytic and a degenerate approximation as follows exists:

$$\kappa(x,y)\approx\sum_{j=1}^{r}\alpha_j\psi_j(x)\varphi_j(y), \tag{1.1}$$

where ψ_j's and φ_j's are appropriate basis functions and α_j's are coefficients independent of x and y. X and Y are well separated in the sense that the distance between them is comparable to their diameters so that r in (1.1) is small. In this case, the corresponding discretized kernel matrix as follows is numerically low rank:

$$K^{(X,Y)}\equiv(\kappa(x,y)_{x\in X,y\in Y}). \tag{1.2}$$

This type of problem frequently arises in a wide range of computations such as numerical solutions of PDEs and integral equations, Gaussian processes, regression with massive data, machine learning, and N-body problems. The low-rank approximation to $K^{(X,Y)}$ enables fast matrix-vector multiplications in methods such as the

*Received by the editors March 4, 2019; accepted for publication (in revised form) by N. Gillis May 26, 2020; published electronically July 20, 2020.
https://doi.org/10.1137/19M1247838

Funding: The work of the second author was partially supported by National Science Foundation grant DMS-1819166.

[†]Department of Mathematics, Purdue University, West Lafayette, IN 47907 (ye83@purdue.edu, xiaj@purdue.edu).

[‡]Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (lexing@stanford.edu).

fast multipole method (FMM) [15]. It can also be used to quickly compute matrix factorization and inversion based on rank structures such as $\mathcal{H}$ [19], $\mathcal{H}^2$ [2, 20], and HSS [5, 48] forms. In fact, relevant low-rank approximations play a key role in rank-structured methods. The success of the so-called fast rank-structured direct solvers relies heavily on the quality and efficiency of low-rank approximations.

According to the Eckhart–Young theorem [9], the best 2-norm low-rank approximation is given by the truncated SVD, which is usually expensive to compute directly. More practical *algebraic compression* methods include rank-revealing factorizations (especially strong rank-revealing QR (SRRQR) [18] and strong rank-revealing LU factorizations [37]), mosaic-skeleton approximations [44], interpolative decomposition [7], CUR decompositions [29], etc. Some of these algebraic methods have a useful feature of *structure preservation* for $K^{(X,Y)}$: relevant resulting basis matrices can be submatrices of the original matrix and are still discretizations of $\kappa(x, y)$ at some subsets. This is a very useful feature that can greatly accelerate some hierarchical rank-structured direct solvers [49, 27, 47]. However, these algebraic compression methods have $\mathcal{O}(rmn)$ complexity and are very costly for large-scale applications. The efficiency may be improved by randomized SVDs [21, 16, 31], which still cost $\mathcal{O}(rmn)$ flops.

Unlike fully algebraic compression, there are also various *analytical compression* methods that take advantage of degenerate approximations like in (1.1) to compute low-rank approximations. The degenerate approximations may be obtained by Taylor expansions, multipole expansions [15], spherical harmonic basis functions [42], Fourier transforms with Poisson's formula [1, 30], Laplace transforms with the Cauchy integral formula [28], Chebyshev interpolations [10], etc. Various other polynomial basis functions may also be used [38].

These analytical approaches can quickly yield low-rank approximations to $K^{(X,Y)}$ by explicitly producing approximate basis matrices. On the other hand, the resulting low-rank approximations are usually not structure preserving in the sense that the basis matrices are not directly related to $K^{(X,Y)}$. This is because the basis functions $\{\psi_j\}$ and $\{\varphi_j\}$ are generally different from $\kappa(x, y)$.

As a particular analytical compression method, the *proxy point method* has attracted a lot of interest in recent years. It is tailored for kernel matrices and is very attractive for different geometries of points [10, 32, 50, 52, 53]. While the methods vary from one to another, they all share the same basic idea and can be summarized in the surprisingly simple Algorithm 1.1, where the details are omitted and will be discussed in later sections. Note that an explicit degenerate form (1.1) is not needed and the algorithm directly produces the matrix $K^{(X,Z)} \equiv (\kappa(x, y)_{x\in X, y\in Z})$ as an approximate column basis matrix in step 2. This feature enables the extension of the ideas of the classical FMM [15] to more general situations, and examples include the recursive skeletonization [22, 32, 36] and kernel independent FMM [33, 52, 53]. The convenient extraction of an approximate column basis matrix is similar to some methods used for data analysis such as the Nyström method and the pseudo-input approximation [8, 13, 26, 40, 46]. (More discussions on this will be given in section 5.)

Notice that $|Z|$ is generally much smaller than $|Y|$ so that $K^{(X,Z)}$ has a much smaller column size than $K^{(X,Y)}$. It is then practical to apply reliable rank-revealing factorizations to $K^{(X,Z)}$ to extract a compact approximate column basis matrix for $K^{(X,Y)}$. This is a *hybrid (analytical/algebraic) compression* scheme, and the proxy point method helps to significantly reduce the compression cost.

The significance of the proxy point method can also be seen from another viewpoint: the selection of *representative points*. When an SRRQR factorization

Algorithm 1.1. *Basic proxy point method for low-rank approximation.*

```
Input: $\kappa(x,y)$, $X$, $Y$
Output: Low-rank approximation $K^{(X,Y)} \approx AB$        ▷ Details in sections 2 and 3
1: Pick a proxy surface $\Gamma$ and a set of proxy points $Z \subset \Gamma$
2: $A \leftarrow K^{(X,Z)}$
3: $B \leftarrow \Phi^{(Z,Y)}$ for a matrix $\Phi^{(Z,Y)}$ such that $K^{(X,Y)} \approx K^{(X,Z)}\Phi^{(Z,Y)}$
```

or interpolative decomposition is applied to $K^{(X,Y)}$, an approximate row basis matrix can be constructed from selected rows of $K^{(X,Y)}$. Suppose those rows correspond to the points $\hat{X} \subset X$. Then $\hat{X}$ can be considered as a subset of representative points. The analytical selection of $\hat{X}$ is not a trivial task. However, with the use of the proxy points Z, we can essentially quickly find $\hat{X}$ based on $K^{(X,Z)}$. (See section 4 for more details.) That is, the set of proxy points Z can serve as a set of auxiliary points based on which the representative points can be quickly identified. In other words, when considering the interaction $K^{(X,Y)}$ between X and Y, we can use the interaction $K^{(X,Z)}$ between X and the proxy points Z to extract the contribution $\hat{X}$ from X.

Thus, the proxy point method is a very convenient and useful tool for researchers working on kernel matrices. However, this elegant method is much less known in the numerical linear algebra community. Indeed, even the compression of some special Cauchy matrices (corresponding to a simple kernel) takes quite a bit of effort in matrix computations [34, 39, 49]. In a recent literature survey [24] that lists many low-rank approximation methods (including a method for kernel matrices), the proxy point method is not mentioned at all. One reason that the proxy point method is not widely known by researchers in matrix computation is the lack of intuitive algebraic understanding of the background.

Moreover, in contrast with the success of the proxy point method in various practical applications, its theoretical justifications are still lacking in the literature. Potential theory [25, Chapter 6] can be used to explain the choice of proxy surface Γ in step 1 of Algorithm 1.1 when dealing with some PDE kernels (when $\kappa(x,y)$ is the fundamental solution of a PDE). However, there is no clear justification of the accuracy of the resulting low-rank approximation, specifically, a clear explanation of such a simple procedure in terms of both the approximation error and the proxy point selection desired, especially from the linear algebra point of view.

Thus, we intend to seek a convenient way to *understand the proxy point method and its accuracy* based on some kernels. The following types of errors will be considered (the notation will be made more precise later):

- The error ε for the approximation of kernel functions $\kappa(x,y)$ with the aid of proxy points.
- The error $\mathcal{E}$ for the low-rank approximation of kernel matrices $K^{(X,Y)}$ via the proxy point method.
- The error $\mathcal{R}$ for practical hybrid low-rank approximations of $K^{(X,Y)}$ based on the proxy point method.

Our main objectives are as follows:

1. Provide an intuitive explanation of the proxy point method using contour integration so as to make this elegant method more accessible to the numerical linear algebra community.
2. Give systematic analysis of the approximation errors of the proxy point method as well as the hybrid compression. We show how the kernel function

approximation error ε and the low-rank compression error $\mathcal{E}$ decay exponentially with respect to the number of proxy points. We also show how our bounds for the error $\mathcal{E}$ are nearly independent of the geometries and sizes of X and Y and why a bound for the error $\mathcal{R}$ may be independent of one set (say, Y).

3. Use the error analysis to choose a nearly optimal set of proxy points in the low-rank kernel matrix compression. Our error bounds give a clear guideline to control the errors and to choose the locations of the proxy points so as to find nearly minimum errors. We also give a practical method to quickly estimate the optimal locations.

We conduct such studies based on kernels of the form

$$\kappa(x,y)=\frac{1}{(x-y)^d},\quad x,y\in\mathbb{C},\quad x\neq y, \tag{1.3}$$

where d is a positive integer. Such kernels and their variants are very useful in PDE and integral equation solutions, structured ODE solutions [4], Cauchy matrix computations [39], Toeplitz matrix direct solutions [6, 34, 49], structured divide-and-conquer Hermitian eigenvalue solutions [17, 45], etc. Our derivations and analysis may also be useful for studying other kernels and higher dimensions. This will be considered in future work. (Note that the issue of what kernels the proxy point method can apply to is not the focus here.)

We would like to point out that several of our results like the error analyses in sections 3 and 4 can be easily extended to more general kernels and/or with other approximation methods, as long as a relative approximation error for the kernel function approximation is available. Thus, our studies are useful for more general situations.

Our theoretical studies are also accompanied by various intuitive numerical tests which show that the error bounds nicely capture the error behaviors and also predict the location of the minimum errors.

In the remaining discussions, section 2 is devoted to an intuitive derivation of the proxy point method via contour integration and the analysis of the accuracy (ε) for the approximation of the kernel functions. The analytical low-rank compression accuracy ($\mathcal{E}$) and the nearly optimal proxy point selection are given in section 3. The study is further extended to the analysis of the hybrid low-rank approximation accuracy ($\mathcal{R}$) with representative point selection in section 4. In section 5, the connection between the proxy point method and the Nyström method is discussed. Some notation we use frequently in the paper is listed below.

- The sets under consideration are $X=\{x_j\}_{j=1}^m$ and $Y=\{y_j\}_{j=1}^n$. $Z=\{z_j\}_{j=1}^N$ is the set of proxy points.
- $\mathcal{C}(a;\gamma)$, $\mathcal{D}(a;\gamma)$, and $\bar{\mathcal{D}}(a;\gamma)$ denote respectively the circle, open disk, and closed disk with center $a\in\mathbb{C}$ and radius $\gamma>0$.
- $\mathcal{A}(a;\gamma_1,\gamma_2)=\{z:\gamma_1<|z-a|<\gamma_2\}$ with $0<\gamma_1<\gamma_2$ is an open annulus region.
- $K^{(X,Y)}$ is the $m\times n$ kernel matrix $(\kappa(x_i,y_j)_{x_i\in X,y_j\in Y})$ with $\kappa(x,y)$ in (1.3). Notation such as $K^{(X,Z)}$ and $K^{(\hat{X},Z)}$ will also be used and can be understood similarly.

2. The proxy point method for kernel function approximation and its accuracy.

In this section, we show one intuitive derivation of the proxy point method for the analytical approximation of the kernel functions, followed by detailed approximation error analysis.

Note that the kernel function (1.3) is translation invariant, i.e., $\kappa(x-z, y-z) = \kappa(x,y)$ for any $x \neq y$ and $z \in \mathbb{C}$. Thus, the points X can be moved to be clustered around the origin. Without loss of generality, we always assume $X \subset \mathcal{D}(0;\gamma_1)$ and $Y \subset \mathcal{A}(0;\gamma_2,\gamma_3)$, where the radii satisfy $0 < \gamma_1 < \gamma_2 < \gamma_3$. See Figure 1. This condition is used to characterize the separation of the sets X and Y so as to theoretically guarantee the numerical low-rankness, as often used in applications of the FMM and rank-structured matrix methods. In these methods, the points are hierarchically partitioned into subsets, and the interaction between one subset and those points that are a certain distance away is considered to be numerically low rank. See [15] for some illustrative figures. More discussions on this will be given in section 5.

2.1. Derivation of the proxy point method via contour integration. Consider any two points $x \in X$ and $y \in Y$. Draw a Jordan curve (a simple closed curve) Γ that encloses x while excluding y, and let $\rho > 0$ be large enough so that the circle $\mathcal{C}(0;\rho)$ encloses both Γ and y. See Figure 2(a).

Define the domain Ω_ρ to be the open region inside $\mathcal{C}(0;\rho)$ and outside Γ. Its boundary is $\partial\Omega_\rho := \mathcal{C}(0;\rho) \cup (-\Gamma)$, where $-\Gamma$ denotes the curve Γ in its negative direction. Now consider the function $f(z) := \kappa(x,z)$ on the closed domain $\bar{\Omega}_\rho := \Omega_\rho \cup \partial\Omega_\rho$. The only singularity of $f(z)$ is at $z = x \notin \bar{\Omega}_\rho$. Thus, $f(z)$ is analytic (or holomorphic) on $\bar{\Omega}_\rho$. By the Cauchy integral formula [41],

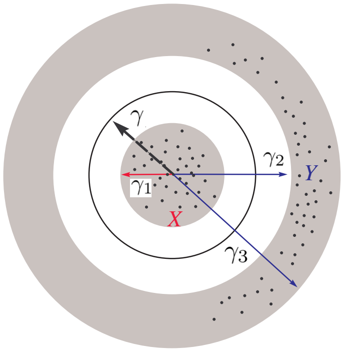
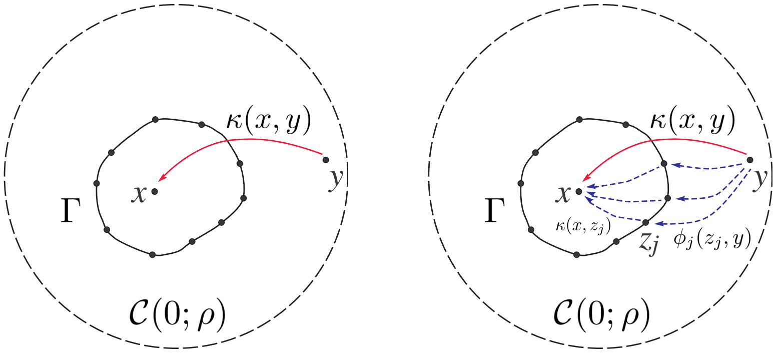


FIG. 1. *Illustration of γ, γ_1, γ_2, γ_3, X, and Y.*



(a) Γ and $\mathcal{C}(0;\rho)$ used in contour integration (b) Approximation of $\kappa(x,y)$

FIG. 2. *Approximating the interaction $\kappa(x,y)$ by $\tilde{\kappa}(x,y)$ in* (2.3) *using proxy points.*

$$\kappa(x,y)=f(y)=\frac{1}{2\pi\mathbf{i}}\int_{\partial\Omega_\rho}\frac{f(z)}{z-y}\mathrm{d}z=\frac{1}{2\pi\mathbf{i}}\int_{\mathcal{C}(0;\rho)}\frac{\kappa(x,z)}{z-y}\mathrm{d}z-\frac{1}{2\pi\mathbf{i}}\int_{\Gamma}\frac{\kappa(x,z)}{z-y}\mathrm{d}z, \tag{2.1}$$

where $\mathbf{i}=\sqrt{-1}$. Note that

$$\left|\int_{\mathcal{C}(0;\rho)}\frac{\kappa(x,z)}{z-y}\mathrm{d}z\right|\le 2\pi\rho\cdot\max_{z\in\mathcal{C}(0;\rho)}\left|\frac{1}{(x-z)^d(z-y)}\right|\le\frac{2\pi\rho}{(\rho-|x|)^d(\rho-|y|)},$$

where the right-hand side goes to zero when $\rho\to\infty$. Thus,

$$\lim_{\rho\to\infty}\int_{\mathcal{C}(0;\rho)}\frac{\kappa(x,z)}{z-y}\mathrm{d}z=0.$$

Take the limit on (2.1) for $\rho\to\infty$, and the first term on the right-hand side vanishes. We get

$$\kappa(x,y)=\frac{1}{2\pi\mathbf{i}}\int_{\Gamma}\frac{\kappa(x,z)}{y-z}\mathrm{d}z. \tag{2.2}$$

Note that this result is different from the Cauchy integral formula in that the point y under consideration is outside the contour Γ in the integral.

To numerically approximate the contour integral (2.2), pick an N-point quadrature rule with quadrature points $\{z_j\}_{j=1}^N\subset\Gamma$ and the corresponding quadrature weights $\{\omega_j\}_{j=1}^N$. Denote by $\tilde{\kappa}(x,y)$ the approximation induced by such a quadrature integration:

$$\tilde{\kappa}(x,y)=\frac{1}{2\pi\mathbf{i}}\sum_{j=1}^N\omega_j\frac{\kappa(x,z_j)}{y-z_j}\equiv\sum_{j=1}^N\kappa(x,z_j)\phi_j(z_j,y)\quad\text{with}\quad\phi_j(z,y)=\frac{\omega_j}{2\pi\mathbf{i}(y-z)}. \tag{2.3}$$

Clearly, $\tilde{\kappa}(x,y)$ in (2.3) is a degenerate approximation to $\kappa(x,y)$ like (1.1). Moreover, it has one additional property of *structure preservation*: the function $\varphi_j(x)$ in this case is $\kappa(x,z_j)$, which is exactly the original kernel $\kappa(x,y)$ with z_j in the role of y. This gives a simple and intuitive explanation of the use of proxy points: the interaction between x and y can essentially be approximated by the interaction between x and some proxy points Z (and later we will further see that Z can be independent of the number of x and y points). These two interactions are made equivalent (in terms of computing potential) through the use of the function ϕ_j. In other words, equivalent charges can be placed on the proxy surface. A pictorial illustration is shown in Figure 2(b).

2.2. Approximation error analysis. Although the approximation (2.3) holds for any proxy surface Γ satisfying the given conditions and for any quadrature rule, we still need to make specific choices in order to obtain a more practical error bound. First, we assume the proxy surface to be a circle, $\Gamma=\mathcal{C}(0;\gamma)$, which is one of the most popular choices in related work and is also consistent with our assumptions at the beginning of section 2. For now, the proxy surface Γ is only assumed to be between X and Y, i.e., $\gamma_1<\gamma<\gamma_2$ as in Figure 1, and we will come back to discuss more on this later. Second, the quadrature rule is chosen to be the composite trapezoidal rule with

$$z_j = \gamma \exp\left(\frac{2j\pi\mathbf{i}}{N}\right), \quad \omega_j = \frac{2\pi\mathbf{i}}{N} z_j, \quad j = 1, 2, \ldots, N. \tag{2.4}$$

This choice can be justified by noting that the trapezoidal rule converges exponentially fast if applied to a periodic integrand [43]. Our results later also align with this. Moreover, if no specific direction is more important that others, the trapezoidal rule performs uniformly well on all directions of the complex plane $\mathbb{C}$. Some related discussions of this issue can be found in [23, 51].

As a result of the above assumptions, the function $\phi_j(z,y)$ in (2.3) becomes the following form:

$$\phi(z,y) = \frac{1}{N}\frac{z}{y-z}, \quad y \neq z,$$

where we dropped the subscript j since j does not explicitly appear on the right-hand side. Also, we define

$$g(z) = \frac{1}{z-1}, \quad z \neq 1.$$

The following lemma will be used in the analysis of the approximation error for $\kappa(x,y)$.

Lemma 2.1. *Let $\{z_j\}_{j=1}^N$ be the points defined in* (2.4). *Then the following result holds for all $z \in \mathbb{C}\backslash\{z_j\}_{j=1}^N$:*

$$\sum_{j=1}^{N} \frac{z_j}{z - z_j} = Ng\left(\left(\frac{z}{\gamma}\right)^N\right). \tag{2.5}$$

Proof. For any integer p, we have

$$\sum_{j=1}^{N} z_j^p = \begin{cases} N\gamma^p & \text{if } p \text{ is a multiple of } N, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

If $|z| < \gamma$, then $|z/z_j| < 1$ for $j = 1, 2, \ldots, N$ and

$$\begin{aligned}
\sum_{j=1}^{N} \frac{z_j}{z - z_j} &= -\sum_{j=1}^{N} \frac{1}{1 - z/z_j} = -\sum_{j=1}^{N}\sum_{k=0}^{\infty}\left(\frac{z}{z_j}\right)^k = -\sum_{k=0}^{\infty}\left(z^k \sum_{j=1}^{N} z_j^{-k}\right) \\
&= -\sum_{l=0}^{\infty} z^{lN} N\gamma^{-lN} \quad \text{(with (2.6), only } k = lN \text{ terms left)} \\
&= -\frac{N}{1 - z^N/\gamma^N} = Ng\left(\left(\frac{z}{\gamma}\right)^N\right).
\end{aligned}$$

If $|z| > \gamma$, then $|z_j/z| < 1$ for $j = 1, 2, \ldots, N$ and

$$\begin{aligned}
\sum_{j=1}^{N} \frac{z_j}{z - z_j} &= \sum_{j=1}^{N}\left(\frac{z}{z - z_j} - 1\right) = -N + \sum_{j=1}^{N} \frac{z}{z - z_j} = -N + \sum_{j=1}^{N} \frac{1}{1 - z_j/z} \\
&= -N + \sum_{j=1}^{N}\sum_{k=0}^{\infty}\left(\frac{z_j}{z}\right)^k = -N + \sum_{k=0}^{\infty}\left(z^{-k}\sum_{j=1}^{N} z_j^k\right)
\end{aligned}$$

$$
\begin{aligned}
&= -N + \sum_{l=0}^{\infty} z^{-lN} N \gamma^{lN} \quad \text{(with (2.6), only } k = lN \text{ terms left)} \\
&= -N + \frac{N}{1-\gamma^N/z^N} = \frac{N}{z^N/\gamma^N - 1} = Ng\left(\left(\frac{z}{\gamma}\right)^N\right).
\end{aligned}
$$

Finally, since both sides of (2.5) are analytic functions on $\mathbb{C}\backslash\{z_j\}_{j=1}^N$ and they agree on z with $|z| \neq \gamma$, by continuity, they must also agree on z when $|z| = \gamma$, $z \notin \{z_j\}_{j=1}^N$. This completes the proof. ◻

In the following theorem, we derive an analytical expression for the accuracy of approximating $\kappa(x,y)$ by $\tilde{\kappa}(x,y)$. Without loss of generality, assume $x \neq 0$.

THEOREM 2.2. *Suppose $\kappa(x,y)$ in* (1.3) *is approximated by $\tilde{\kappa}(x,y)$ in* (2.3) *which is obtained from the composite trapezoidal rule with* (2.4). *Assume $x \neq 0$. Then*

$$
\tilde{\kappa}(x,y) = \kappa(x,y)\left(1 + \varepsilon(x,y)\right), \tag{2.7}
$$

where $\varepsilon(x,y)$ is the relative approximation error

$$
\varepsilon(x,y) := \frac{\tilde{\kappa}(x,y) - \kappa(x,y)}{\kappa(x,y)} = g\left(\left(\frac{y}{\gamma}\right)^N\right) + \sum_{j=0}^{d-1} \frac{(y-x)^j}{j!} \frac{\mathrm{d}^j}{\mathrm{d}x^j} g\left(\left(\frac{\gamma}{x}\right)^N\right). \tag{2.8}
$$

Proof. We prove this theorem by induction on d. For $d = 1$, substituting (2.4) into (2.3) yields

$$
\begin{aligned}
\tilde{\kappa}(x,y) &= \frac{1}{N}\sum_{j=1}^{N} \frac{z_j}{(x-z_j)(y-z_j)} = \frac{1}{N(x-y)} \sum_{j=1}^{N} \frac{(x-z_j)-(y-z_j)}{(x-z_j)(y-z_j)} z_j \\
&= \frac{1}{N(x-y)} \left(\sum_{j=1}^{N} \frac{z_j}{y-z_j} - \sum_{j=1}^{N} \frac{z_j}{x-z_j} \right) \\
&= \frac{1}{N(x-y)} \left(Ng\left(\left(\frac{y}{\gamma}\right)^N\right) - Ng\left(\left(\frac{x}{\gamma}\right)^N\right) \right) \quad \text{(Lemma 2.1)} \\
&= \frac{1}{x-y}\left[1 + g\left(\left(\frac{y}{\gamma}\right)^N\right) + g\left(\left(\frac{\gamma}{x}\right)^N\right)\right].
\end{aligned}
$$

Thus, (2.7) holds for $d = 1$.

Now suppose (2.7) holds for $d = k$ with k a positive integer. Equating (2.3) and (2.7) (with $d = k$) and plugging in $\kappa(x,y)$ we get

$$
\sum_{j=1}^{N} \frac{\phi_j(z_j,y)}{(x-z_j)^k} = \frac{1}{(x-y)^k}\left[1 + g\left(\left(\frac{y}{\gamma}\right)^N\right) + \sum_{j=0}^{k-1} \frac{(y-x)^j}{j!} \frac{\mathrm{d}^j}{\mathrm{d}x^j} g\left(\left(\frac{\gamma}{x}\right)^N\right)\right].
$$

The derivatives of the left- and right-hand sides with respect to x are, respectively, $-k\sum_{j=1}^{N} \frac{\phi_j(z_j,y)}{(x-z_j)^{k+1}}$ and

$$
\begin{aligned}
&\frac{-k}{(x-y)^{k+1}}\left[1+g\left(\left(\frac{y}{\gamma}\right)^N\right)+\sum_{j=0}^{k-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right]\\
&\quad+\frac{1}{(x-y)^k}\left[\sum_{j=0}^{k-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^{j+1}}{\mathrm{d}x^{j+1}}g\left(\left(\frac{\gamma}{x}\right)^N\right)-\sum_{j=1}^{k-1}\frac{(y-x)^{j-1}}{(j-1)!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right]\\
&\quad=\frac{-k}{(x-y)^{k+1}}\left[1+g\left(\left(\frac{y}{\gamma}\right)^N\right)+\sum_{j=0}^{k-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right]\\
&\qquad+\frac{1}{(x-y)^k}\frac{(y-x)^{k-1}}{(k-1)!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}g\left(\left(\frac{\gamma}{x}\right)^N\right)\quad\text{(all terms cancel except for } j=k-1\text{)}\\
&\quad=\frac{-k}{(x-y)^{k+1}}\left[1+g\left(\left(\frac{y}{\gamma}\right)^N\right)+\sum_{j=0}^{k}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right].
\end{aligned}
$$

Thus,

$$
\sum_{j=1}^{N}\frac{\phi(z_j,y)}{(x-z_j)^{k+1}}=\frac{1}{(x-y)^{k+1}}\left[1+g\left(\left(\frac{y}{\gamma}\right)^N\right)+\sum_{j=0}^{k}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right].
$$

That is, (2.7) holds for $d=k+1$. By induction, (2.7)–(2.8) are true for any positive integer d. □

With the analytical expression (2.8) we can give a rigorous upper bound for the approximation error.

THEOREM 2.3. *Suppose $0<|x|<\gamma_1<\gamma<|y|$. With all the assumptions in Theorem 2.2, there exists a positive integer N_1 such that for any $N>N_1$, the approximation error (2.8) is bounded by*

$$
|\varepsilon(x,y)|\le g\left(\left|\frac{y}{\gamma}\right|^N\right)+c\,g\left(\left|\frac{\gamma}{x}\right|^N\right), \tag{2.9}
$$

where $c=1$ if $d=1$, and otherwise

$$
c=2+2\sum_{j=1}^{d-1}\frac{[(|y/x|+1)N]^j(2d)^{j-1}}{j!}. \tag{2.10}
$$

Proof. For any positive integer N,

$$
\left|g\left(\left(\frac{y}{\gamma}\right)^N\right)\right|=\frac{1}{|(y/\gamma)^N-1|}\le\frac{1}{|y/\gamma|^N-1}=g\left(\left|\frac{y}{\gamma}\right|^N\right).
$$

Thus, we only need to prove the following bound:

$$
\left|\sum_{j=0}^{d-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right|\le c\,g\left(\left|\frac{\gamma}{x}\right|^N\right). \tag{2.11}
$$

When $d=1$, it's easy to verify that the above inequality holds for $c=1$ and any positive integer N. We now consider the case when $d\ge 2$.

It can be verified that, for any positive integer i,

$$\frac{\mathrm{d}}{\mathrm{d}x} g^i\left(\left(\frac{\gamma}{x}\right)^N\right) = \frac{iN}{x}\left[g^i\left(\left(\frac{\gamma}{x}\right)^N\right) + g^{i+1}\left(\left(\frac{\gamma}{x}\right)^N\right)\right], \tag{2.12}$$

where g^i denotes function g raised to power i. Hence, the derivatives appearing in (2.11) all have the following form:

$$\frac{\mathrm{d}^j}{\mathrm{d}x^j} g\left(\left(\frac{\gamma}{x}\right)^N\right) = \frac{1}{x^j}\sum_{i=1}^{j+1} \alpha_i^{(j)} g^i\left(\left(\frac{\gamma}{x}\right)^N\right), \tag{2.13}$$

where $\alpha_i^{(j)}$ $(1 \le i \le j+1,\ 0 \le j \le d-1)$ are constants.

We claim that, when $N > d$ and for any $0 \le j \le d-1$, there exist constants $\beta^{(j)}$ dependent on d so that

$$|\alpha_i^{(j)}| \le \beta^{(j)} N^j, \quad 1 \le i \le j+1.$$

This claim can be proved by induction on j. It is obviously true when $j = 0$, and $\beta^{(0)} = 1$ in this case. When $j = 1$, (2.12) means that the claim is true with $\alpha_1^{(1)} = \alpha_2^{(1)} = N$ and $\beta^{(1)} = 1$. Suppose the claim holds for $j = k$ with $1 \le k \le d-2$ (where we also assume $d > 2$, since otherwise the claim is already proved). Then

$$\begin{aligned}
\frac{\mathrm{d}^{k+1}}{\mathrm{d}x^{k+1}} g\left(\left(\frac{\gamma}{x}\right)^N\right) &= \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{x^k}\sum_{i=1}^{k+1}\alpha_i^{(k)} g^i\left(\left(\frac{\gamma}{x}\right)^N\right)\right) \\
&= -\frac{k}{x^{k+1}}\sum_{i=1}^{k+1}\alpha_i^{(k)} g^i\left(\left(\frac{\gamma}{x}\right)^N\right) + \frac{1}{x^k}\sum_{i=1}^{k+1}\alpha_i^{(k)}\frac{iN}{x}\left[g^i\left(\left(\frac{\gamma}{x}\right)^N\right) + g^{i+1}\left(\left(\frac{\gamma}{x}\right)^N\right)\right] \\
&\qquad \text{(by (2.12))} \\
&= \frac{1}{x^{k+1}}\left[(N-k)\alpha_1^{(k)} g\left(\left(\frac{\gamma}{x}\right)^N\right) + \sum_{i=2}^{k+1}\left((iN-k)\alpha_i^{(k)} + N(i-1)\alpha_{j-1}^{(k)}\right) g^i\left(\left(\frac{\gamma}{x}\right)^N\right)\right. \\
&\qquad \left. + N(k+1)\alpha_{k+1}^{(k)} g^{k+2}\left(\left(\frac{\gamma}{x}\right)^N\right)\right].
\end{aligned}$$

Thus, the coefficients satisfy the following recurrence relation:

$$\alpha_i^{(k+1)} = \begin{cases} (N-k)\alpha_1^{(k)}, & i = 1, \\ (iN-k)\alpha_i^{(k)} + N(i-1)\alpha_{i-1}^{(k)}, & 2 \le i \le k+1, \\ N(k+1)\alpha_{k+1}^{(k)}, & i = k+2. \end{cases}$$

Therefore, when $N > d$, we can pick (conservatively)

$$\beta^{(k+1)} = 2d\beta^{(k)}, \tag{2.14}$$

so that $|\alpha_i^{(k+1)}| \le \beta^{(k+1)} N^{k+1}$. That is, the claim holds for $j = k+1$ and this finishes the induction.

Now, we go back to prove (2.11). By (2.13),

(2.15)

$$\begin{aligned}&\left|\sum_{j=0}^{d-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right| = \left|\sum_{j=0}^{d-1}\left[\frac{(y-x)^j}{j!}\frac{1}{x^j}\sum_{i=1}^{j+1}\alpha_i^{(j)}g^i\left(\left(\frac{\gamma}{x}\right)^N\right)\right]\right|\\ &\quad\le \sum_{j=0}^{d-1}\left[\frac{(|y/x|+1)^j}{j!}\sum_{i=1}^{j+1}|\alpha_i^{(j)}|\,g^i\left(\left|\frac{\gamma}{x}\right|^N\right)\right] \le \sum_{j=0}^{d-1}\left[\frac{(|y/x|+1)^j}{j!}\beta^{(j)}N^j\sum_{i=1}^{j+1}g^i\left(\left|\frac{\gamma}{x}\right|^N\right)\right].\end{aligned}$$

Set

$$N_1 = \max\{d, \lceil \log 3/\log|\gamma_1/x|\rceil\}. \tag{2.16}$$

Then for $N > N_1$, $|\gamma/x|^N > |\gamma_1/x|^N > 3$ and $g\left(|\gamma/x|^N\right) < 1/2$. Thus, for $1 \le j \le d-1$,

$$\sum_{i=1}^{j+1} g^i\left(\left|\frac{\gamma}{x}\right|^N\right) \le 2g\left(\left|\frac{\gamma}{x}\right|^N\right).$$

Continuing on (2.15), for $N > N_1$, we get

(2.17)

$$\left|\sum_{j=0}^{d-1}\frac{(y-x)^j}{j!}\frac{\mathrm{d}^j}{\mathrm{d}x^j}g\left(\left(\frac{\gamma}{x}\right)^N\right)\right| \le cg\left(\left|\frac{\gamma}{x}\right|^N\right), \quad \text{with} \quad c = 2\sum_{j=0}^{d-1}\frac{(|y/x|+1)^j}{j!}\beta^{(j)}N^j.$$

Note that with the way $\beta^{(j)}$ is picked as in (2.14), $\beta^{(j)}$ satisfies

$$\beta^{(j)} = (2d)^{j-1}\beta^{(1)} = (2d)^{j-1}, \; j = 1, 2, \ldots, d-1.$$

Then c in (2.17) becomes (2.10). Thus, (2.11) holds with c in (2.10). □

The upper bound (2.9) in Theorem 2.3 has two implications.

- Since $g(|y/\gamma|^N)$ and $g(|\gamma/x|^N)$ decay almost exponentially with N and c is just a polynomial in N, d, and $|y/x|$ with degrees up to $d-1$, the bound in (2.9) decays roughly exponentially as N increases.
- The bound can help us identify a nearly optimal radius γ of the proxy surface Γ so as to minimize the error. This is given in the following theorem.

THEOREM 2.4. *Suppose* $0 < |x| < \gamma_1 < |y|$ *and* $\kappa(x, y)$ *in* (1.3) *is approximated by* $\tilde{\kappa}(x, y)$ *in* (2.3) *with* (2.4)*. If the upper bound in* (2.9) *is viewed as a real function in* γ *on the interval* $(|x|, |y|)$*, then there exists a positive integer* N_2 *independent of* γ*, such that for* $N > N_2$*,*

1. *this upper bound has a unique minimizer* $\gamma^* \in (|x|, |y|)$*;*
2. *the minimum of this upper bound decays asymptotically as* $\mathcal{O}\left(|y/x|^{-N/2}\right)$*.*

Proof. To find the minimizer, we just need to consider the real function

$$h(t) = \frac{1}{b/t - 1} + \frac{c}{t/a - 1}, \quad t \in (a, b),$$

where $a = |x|^N$, $b = |y|^N$, and c is either equal to 1 (for $d = 1$) or defined in (2.10) (for $d \ge 2$). The derivative of the function is

$$h'(t) = \frac{p(t)}{(t-a)^2(t-b)^2} \quad \text{with} \quad p(t) = (b - ac)t^2 + 2ab(c-1)t + ab(a - bc).$$

Consider $p(t)$, which is a quadratic polynomial in t with the following properties:

- The coefficient of the second order term is
$$b - ac = |x|^N \left(|y/x|^N - c\right).$$
Since c is either equal to 1 (for $d = 1$) or a polynomial in N, d, and $|y/x|$ with degrees up to $d-1$ (for $d \geq 2$), there exists N_2 larger than N_1 in Theorem 2.3 such that $|y/x|^N > c$ for any $N > N_2$. Thus, $b - ac > 0$ for $N > N_2$.
- The discriminant is $4abc(a-b)^2 > 0$.
- When evaluated at $t = a$ and $t = b$, the function $p(t)$ gives respectively
$$p(a) = -ac(a-b)^2 < 0, \quad p(b) = b(a-b)^2 > 0.$$

All the properties above combined indicate that $p(t)$ has one root $t_0 \in (a, b)$ and $h'(t) < 0$ on (a, t_0) and $h'(t) > 0$ on (t_0, b). Thus, t_0 is the only zero of $p(t)$ in $[a, b]$ and $\gamma^* = \sqrt[N]{t_0}$ is the unique minimizer of the upper bound in (2.9). The requirements for picking N_2 are $N_2 > N_1$ and $|y/x|^{N_2} > c$. Hence, N_2 is independent of γ.

To prove the second part of the theorem, we explicitly compute the root t_0 of $p(t) = 0$ in (a, b) and substitute it into $h(t)$ to get
$$h(t_0) = \frac{2\sqrt{cb/a} + (c+1)}{b/a - 1} = \frac{2\sqrt{c}|y/x|^{N/2} + (c+1)}{|y/x|^N - 1} \sim \mathcal{O}\left(\left|\frac{y}{x}\right|^{-N/2}\right).$$
The details involve tedious algebra and are omitted here. □

In the proof, we can actually find the minimizer but are not explicitly writing it out. The reason is that the minimizer depends on x and y and it makes more sense to write a minimizer later when we consider the low-rank approximation of the kernel matrix. See the next section.

3. Low-rank approximation accuracy and proxy point selection in the proxy point method for kernel matrices.

With the kernel $\kappa(x, y)$ in (1.3) approximated by $\tilde{\kappa}(x, y)$ in (2.3), a low-rank approximation to $K^{(X,Y)}$ in (1.2) as follows is obtained:

$$K^{(X,Y)} \approx \tilde{K}^{(X,Y)} := (\tilde{\kappa}(x, y)_{x\in X, y\in Y}) = K^{(X,Z)}\Phi^{(Z,Y)}, \tag{3.1}$$

where $\Phi^{(Z,Y)} = (\phi(z, y)_{z\in Z, y\in Y})$. The analysis in subsection 2.2 provides entrywise approximation errors for (3.1) (with implicit dependence on x). Now, we consider normwise approximation errors for $K^{(X,Y)}$ and obtain relative error bounds independent of the specific x and y points. The error analysis will be further used to estimate the optimal choice of the radius γ for the proxy surface in the low-rank approximation. We look at the cases $d = 1$ and $d \geq 2$ separately.

3.1. The case $d = 1$.

In this case, the proof of Theorem 2.2 for $d = 1$ gives an explicit expression for the entrywise approximation error

$$\varepsilon(x, y) = g\left(\left(\frac{\gamma}{x}\right)^N\right) + g\left(\left(\frac{y}{\gamma}\right)^N\right). \tag{3.2}$$

We then have the following result on the low-rank approximation error in Frobenius norm.

Proposition 3.1. *Suppose $d = 1$ and $\kappa(x, y)$ in* (1.3) *is approximated by $\tilde{\kappa}(x, y)$ in* (2.3) *with* (2.4)*. If $0 < |x| < \gamma_1 < \gamma < \gamma_2 < |y|$ for all $x \in X, y \in Y$, then for any $N > 0$,*

$$\frac{\|\tilde{K}^{(X,Y)} - K^{(X,Y)}\|_F}{\|K^{(X,Y)}\|_F} \le g\left(\left(\frac{\gamma}{\gamma_1}\right)^N\right) + g\left(\left(\frac{\gamma_2}{\gamma}\right)^N\right). \tag{3.3}$$

Moreover, if the upper bound on the right-hand side is viewed as a function in γ, it has a unique minimizer $\gamma^* = \sqrt{\gamma_1\gamma_2}$ and the minimum is $2g\left((\gamma_2/\gamma_1)^{N/2}\right)$ which decays asymptotically as $\mathcal{O}\left(|\gamma_2/\gamma_1|^{-N/2}\right)$.

Proof. The approximation error bound (3.3) is a direct application of the entrywise error in (3.2) together with the fact that $g(t)$ monotonically decreases for $t > 1$.

To find the minimizer of the right-hand side of (3.3), we can either follow the proof in Theorem 2.4 or simply use the following explicit expression:

$$\begin{aligned} g\left((\gamma/\gamma_1)^N\right) + g\left((\gamma_2/\gamma)^N\right) &= \frac{1}{(\gamma/\gamma_1)^N - 1} + \frac{1}{(\gamma_2/\gamma)^N - 1} \\ &= -1 + \frac{(\gamma_2/\gamma_1)^N - 1}{(\gamma_2/\gamma_1)^N + 1 - \left((\gamma/\gamma_1)^N + (\gamma_2/\gamma)^N\right)}. \end{aligned}$$

We just need to minimize $(\gamma/\gamma_1)^N + (\gamma_2/\gamma)^N$, which reaches its minimum at $\gamma^* = \sqrt{\gamma_1\gamma_2}$. ◻

Remark 3.2. Although it is not easy to choose γ to minimize the approximation error directly, the minimizer γ^* for the bound in (3.3) can serve as a reasonable estimate of the minimizer for the error. These can be seen from an intuitive numerical example below. In addition, the minimum $2g\left((\gamma_2/\gamma_1)^{N/2}\right)$ of the bound in (3.3) decays nearly exponentially as N increases. Thus, to reach a relative approximation accuracy τ, we can conveniently decide the number of proxy points:

$$N = \mathcal{O}\left(\frac{\log(1/\tau)}{\log(\gamma_2/\gamma_1)}\right).$$

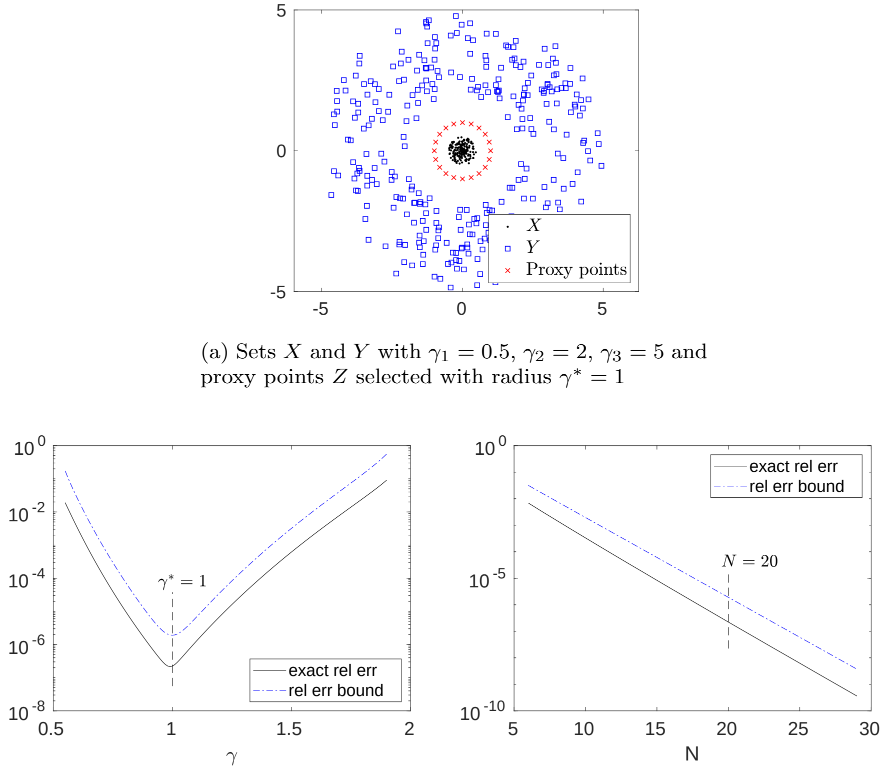
Clearly, N does not depend on the number of points or the geometries of X, Y. It only depends on τ and γ_2/γ_1, which indicates the separation of X and Y. This is consistent with the conclusions in the FMM context [42].

Example 1. We use an example to illustrate the results in Proposition 3.1 for $d = 1$. The points in X and Y are uniformly chosen from their corresponding regions and are plotted in Figure 3(a), where $m = |X| = 200$, $n = |Y| = 300$, $\gamma_1 = 0.5$, $\gamma_2 = 2$, and $\gamma_3 = 5$.

First, we fix the number of proxy points $N = 20$ and let γ vary. We plot the actual error $\mathcal{E}_N(\gamma) := \|\tilde{K}^{(X,Y)} - K^{(X,Y)}\|_F / \|K^{(X,Y)}\|_F$ and the error bound in (3.3). See Figure 3(b). We can see that both plots are V-shaped lines and the error bound is a close estimate of the actual error. Moreover, the bound nicely captures the error behavior, and the actual error reaches its minimum almost at the same location where the error bound is minimized: $\gamma^* = \sqrt{\gamma_1\gamma_2} = 1$. Thus, γ^* is a nice choice to minimize the error. The proxy points Z with radius γ^* are plotted in Figure 3(a).

Then in Figure 3(c), we fix $\gamma = \gamma^*$ and let N vary. Again, the error bound provides a nice estimate for the error. Furthermore, both the error and the bound decay exponentially like $\mathcal{O}\left(|\gamma_2/\gamma_1|^{-N/2}\right) = \mathcal{O}(2^{-N})$.

3.2. The case $d > 2$. In this case, there is no simple explicit formula for $\varepsilon(x, y)$ like in (3.2). The results in Theorems 2.3 and 2.4 cannot be trivially extended to study the normwise error either since no lower bound is imposed on $|x|$ in $|y/x|$. Nevertheless, we can derive a bound as follows.



(a) Sets X and Y with $\gamma_1 = 0.5$, $\gamma_2 = 2$, $\gamma_3 = 5$ and proxy points Z selected with radius $\gamma^* = 1$

(b) Varying proxy surface radius γ

(c) Varying number of proxy points N

FIG. 3. *Example* 1*: For $d = 1$, the selection of the proxy points and the actual relative error $\mathcal{E}_N(\gamma)$ compared with its upper bound in Proposition* 3.1 *for different γ and N.*

PROPOSITION 3.3. *Suppose $d \geq 2$ and $\kappa(x, y)$ in* (1.3) *is approximated by $\tilde{\kappa}(x, y)$ in* (2.3) *with* (2.4)*. If $0 < |x| < \gamma_1 < \gamma < \gamma_2 < |y| < \gamma_3$ for all $x \in X, y \in Y$, then there exists a positive integer N_3 independent of γ such that for $N > N_3$,*

$$\frac{\|\tilde{K}^{(X,Y)} - K^{(X,Y)}\|_F}{\|K^{(X,Y)}\|_F} \leq g\left(\left(\frac{\gamma_2}{\gamma}\right)^N\right) + \hat{c}\, g\left(\left(\frac{\gamma}{\gamma_1}\right)^N\right), \tag{3.4}$$

where

$$\hat{c} = 2 + 2\sum_{j=1}^{d-1} \frac{[(|\gamma_3/\gamma_1| + 1)N]^j (2d)^{j-1}}{j!}. \tag{3.5}$$

Moreover, if the upper bound in (3.4) *is viewed as a real function in γ on the interval (γ_1, γ_2), then*

1. *this upper bound has a unique minimizer*

$$\gamma^* = \left(\frac{(\gamma_2^N - \gamma_1^N)\sqrt{(\gamma_1\gamma_2)^N \hat{c}} - (\gamma_1\gamma_2)^N(\hat{c} - 1)}{\gamma_2^N - \gamma_1^N \hat{c}}\right)^{1/N} \in (\gamma_1, \gamma_2); \tag{3.6}$$

2. *the minimum of this upper bound decays asymptotically as $\mathcal{O}\left(|\gamma_2/\gamma_1|^{-N/2}\right)$.*

Proof. Following the proof of Theorem 2.4, we can set N_3 to be the maximum of N_2 in Theorem 2.4 for all $x \in X$. Based on the entrywise error bound in (2.9), we can just show the following inequalities for $N > N_3$ and any $x \in X, y \in Y$:

$$g\left(\left|\frac{y}{\gamma}\right|^N\right) < g\left(\left(\frac{\gamma_2}{\gamma}\right)^N\right), \quad cg\left(\left|\frac{\gamma}{x}\right|^N\right) < \hat{c}\,g\left(\left(\frac{\gamma}{\gamma_1}\right)^N\right).$$

The first inequality is obvious. We then focus on the second one. Just for the purpose of this proof, we write c in (2.10) as $c(|x|,|y|)$ to indicate its dependency on $|x|$ and $|y|$. $c(|x|,|y|)$ can be viewed as a degree-$(d-1)$ polynomial in $1/|x|$ and $|y|$ with all positive coefficients.

Write

$$c(|x|,|y|)\,g\left(\left|\frac{\gamma}{x}\right|^N\right) = \left[c(|x|,|y|)|x|^{d-1}\right]\left[g\left(\left|\frac{\gamma}{x}\right|^N\right)|x|^{1-d}\right].$$

The first term $c(|x|,|y|)|x|^{d-1}$ is a polynomial in $|x|$ with all positive coefficients and increases with $|x|$. The second term is

$$g\left(\left|\frac{\gamma}{x}\right|^N\right)|x|^{1-d} = \frac{|x|^{N-d+1}}{\gamma^N - |x|^N}.$$

With $N > N_3$, it can be shown that this term is also strictly increasing in $|x|$ for $0 < |x| < \gamma_1 < \gamma$.

Thus for any $x \in X, y \in Y$,

$$c(|x|,|y|)\,g\left(\left|\frac{\gamma}{x}\right|^N\right) < c(\gamma_1,|y|)\,g\left(\left|\frac{\gamma}{\gamma_1}\right|^N\right) < c(\gamma_1,\gamma_3)\,g\left(\left|\frac{\gamma}{\gamma_1}\right|^N\right) = \hat{c}\,g\left(\left|\frac{\gamma}{\gamma_1}\right|^N\right),$$

where the constant $\hat{c}$ is defined in (3.5) which is c in (2.10) with $|y/x|$ replaced by γ_3/γ_1.

The minimizer γ^* in (3.6) for the upper bound is the root of a quadratic polynomial in (γ_1,γ_2) and can be obtained following the proof of Theorem 2.4. □

Based on this corollary, we can draw conclusions similar to those in Remark 3.2. In addition, although γ_3 is needed so that Y is on a bounded domain in order to derive the error bound (3.4), we believe such a limitation is not needed in practice. In fact, the analytical compression tends to be more accurate when the points y are farther away from the set X. Also, if γ_3 is too large, then we may slightly shift the x points to make sure $|x|$ is larger than a positive number γ_0 so as to similarly derive an error bound using γ_0 instead of γ_3.

3.3. A practical method to estimate the optimal radius γ. In Propositions 3.1 and 3.3, the upper bounds are used to estimate the optimal choice of γ for the radius of the proxy surface. In practice, it is possible that the upper bound may be conservative, especially when $d > 1$. Thus, we also propose the following method to quickly obtain a numerical estimate of the optimal choice.

In Propositions 3.1 and 3.3, the optimal γ^* is independent of the number of points in X and Y and their distribution. This feature motivates the idea to pick subsets $X_0 \subset \mathcal{D}(0;\gamma_1)$ and $Y_0 \subset \mathcal{A}(0;\gamma_2,\gamma_3)$ and use them to estimate the actual error. That is, we would expect the following two quantities to have similar behaviors when γ varies in (γ_1,γ_2):

$$\mathcal{E}_N^0(\gamma) := \frac{\|K^{(X_0,Y_0)} - \tilde{K}^{(X_0,Y_0)}\|_F}{\|K^{(X_0,Y_0)}\|_F}, \quad \mathcal{E}_N(\gamma) := \frac{\|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F}{\|K^{(X,Y)}\|_F}. \tag{3.7}$$

$\mathcal{E}_N^0(\gamma)$ can be used as an estimator of the actual approximation error $\mathcal{E}_N(\gamma)$. Note that $K^{(X_0,Y_0)}$ and $\tilde{K}^{(X_0,Y_0)}$ are computable through (1.3) and (2.3), respectively, so $\mathcal{E}_N^0(\gamma)$ can be computed explicitly, and the cost is extremely small if $|X_0| \ll |X|$ and $|Y_0| \ll |Y|$.

Note that in rank-structured matrix computations, often an admissible condition or separation parameter is prespecified for the compression of multiple off-diagonal blocks. In the case of kernel matrices, it means that the process of estimating the optimal γ needs to be run only once and can then be used in multiple compression steps.

Example 2. We use an example to demonstrate the numerical selection of the optimal γ. Consider $d = 2, 3$ and the two sets X and Y in Example 1 with the same values $\gamma_1, \gamma_2, \gamma_3$ (see Figure 3(a)). Fix $N = 30$.

For the sets X_0 and Y_0 we choose, we set $l = |X_0| = |Y_0|$ to be 1, 2, or 3. We make sure $x = \gamma_1$ and $y = \gamma_2$ as points of $\mathbb{C}$ are always in X_0 and Y_0, respectively. Thus, $\mathcal{E}_N^0(\gamma)$ is more likely to capture the behavior of $\mathcal{E}_N(\gamma)$. Any additional points in X_0 are uniformly distributed in the circle $\mathcal{C}(0;\gamma_1)$ and any additional points in Y_0 are uniformly distributed in $\mathcal{C}(0;\gamma_2)$.

With $l = 1$, both $\mathcal{E}_N(\gamma)$ and $\mathcal{E}_N^0(\gamma)$ are plotted. See Figures 4(a) and 4(b) for $d = 2$ and 3, respectively. We can see that $\mathcal{E}_N^0(\gamma)$ already gives a good estimate of the behavior of $\mathcal{E}_N(\gamma)$ for both cases. Then in Figures 4(c) and 4(d) we plot $\mathcal{E}_N^0(\gamma)$ for $l = 1, 2, 3$ and zoom in at around the minimum since they almost coincide with each other away from the minimum. The minimums of the three cases are very close to each other.

4. Low-rank approximation accuracy in hybrid compression and representative point selection.

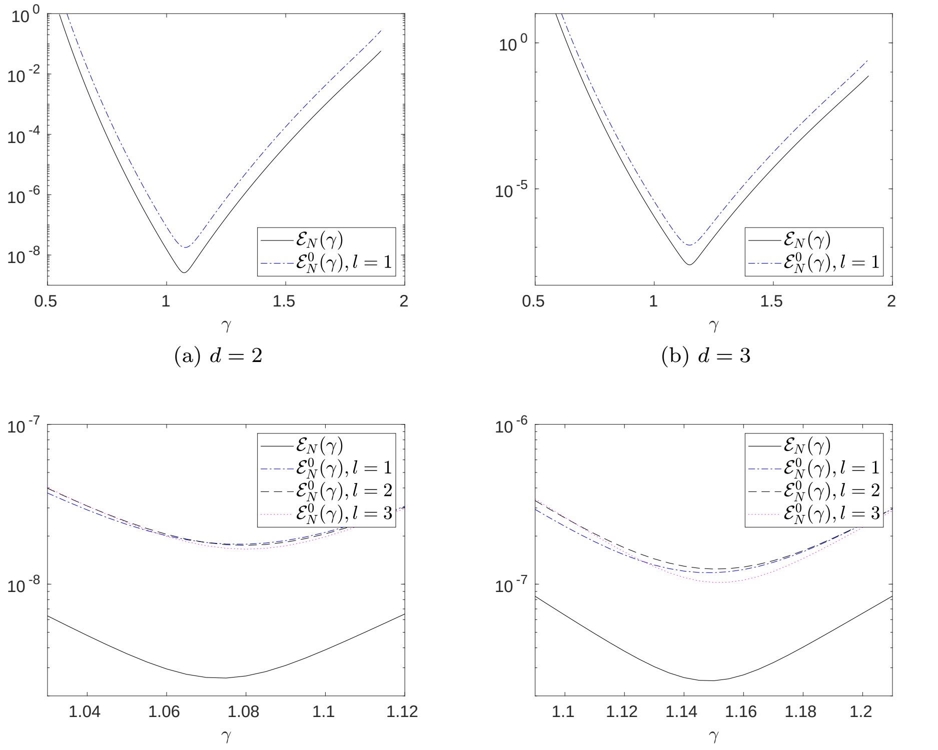
The analytical compression in section 3 can serve as a preliminary low-rank approximation, which is typically followed by an algebraic compression step to get a more compact low-rank approximation. In this section, we analyze the approximation error of such hybrid (analytical/algebraic) compression method applied to $K^{(X,Y)}$.

Suppose $m = |X|$ and $n = |Y|$ are sufficiently large and $N = |Z|$ is fixed. With the preliminary low-rank approximation in (3.1), since $K^{(X,Z)}$ has a much smaller column size than $K^{(X,Y)}$, it becomes practical to apply an SRRQR factorization to $K^{(X,Z)}$ to obtain the following approximation:

$$K^{(X,Z)} \approx U K^{(\hat{X},Z)} \quad \text{with} \quad U = P \begin{pmatrix} I \\ E \end{pmatrix}, \tag{4.1}$$

where P is a permutation matrix so that $K^{(\hat{X},Z)}$ a submatrix of $K^{(X,Z)}$ corresponding to a subset $\hat{X} \subset X$. $\hat{X}$ can be referred to as a set of *representative points* of X. (4.1) is an interpolative decomposition of $K^{(X,Z)}$. It is also called structure-preserving rank-revealing (SPRR) factorization in [49] since $K^{(\hat{X},Z)}$ is a submatrix of $K^{(X,Z)}$.

Although U generally does not have orthonormal columns, the SRRQR factorization keeps its norm under control in the sense that entries of E have magnitudes bounded by a constant e (e.g., $e = 2$ or $\sqrt{N}$). See [18] for details.



(a) $d = 2$ (b) $d = 3$

(c) $d = 2$, zoomed in around the critical point (d) $d = 3$, zoomed in around the critical point

FIG. 4. *Example 2: For $d = 2$ and 3, how the estimator $\mathcal{E}_N^0(\gamma)$ with $l = 1, 2, 3$ compares with the actual error $\mathcal{E}_N(\gamma)$.*

We then have

$$K^{(X,Y)} \approx \tilde{K}^{(X,Y)} = K^{(X,Z)}\Phi^{(Z,Y)} \qquad \text{(by (2.3) and (3.1))} \tag{4.2a}$$

$$\approx UK^{(\hat{X},Z)}\Phi^{(Z,Y)} \qquad \text{(by (4.1))} \tag{4.2b}$$

$$= U\tilde{K}^{(\hat{X},Y)} \qquad \text{(by (2.3) and similar to (3.1))} \tag{4.2c}$$

$$\approx UK^{(\hat{X},Y)}, \qquad \text{(by } \tilde{\kappa}(x,y) \approx \kappa(x,y)\text{)}, \tag{4.2d}$$

which is an SPRR factorization of $K^{(X,Y)}$.

Similarly, an SRRQR factorization can further be applied to $K^{(\hat{X},Y)}$ to produce

$$K^{(\hat{X},Y)} \approx K^{(\hat{X},\hat{Y})}V^T \quad \text{with} \quad V = Q\begin{pmatrix} I \\ F \end{pmatrix}, \tag{4.3}$$

where Q is a permutation matrix and $\hat{Y} \subset Y$. The approximation (4.2) together with (4.3) essentially enables us to quickly select representative points from both X and Y. In other words, we have a skeleton factorization of $K^{(X,Y)}$ as

$$K^{(X,Y)} \approx UK^{(\hat{X},\hat{Y})}V^T. \tag{4.4}$$

Note that computing an SPRR or skeleton factorization for $K^{(X,Y)}$ directly (or to find a submatrix $K^{(\hat{X},\hat{Y})}$ with the largest "volume" [14, 44]) is typically prohibitively expensive for large m and n. Here, the proxy point method substantially reduces the cost. In fact, (4.2a) and (4.2c) are done analytically with no computation cost. Only the SRRQR factorizations of skinny matrices ($K^{(X,Z)}$ and/or $K^{(\hat{X},Y)}$) are needed. The total compression cost is $\mathcal{O}(mNr)$ for (4.2) or $\mathcal{O}(mNr + nr^2)$ for (4.4) instead of $\mathcal{O}(mnr)$ in the case of direct compression, where $r = |\hat{X}| \geq |\hat{Y}|$. As we have discussed before, N is only a constant independent of m and n. Thus, this procedure is significantly more efficient than applying SRRQR factorizations directly to the original kernel matrix.

The next theorem concerns the approximation error of the hybrid compression via either (4.2) or (4.4).

THEOREM 4.1. *Suppose* $0 < |x| < \gamma_1 < \gamma < \gamma_2 < |y| < \gamma_3$ *for any* $x \in X, y \in Y$ *and the N proxy points in Z are located on the proxy surface with radius* γ^*. *Let* $r = |\hat{X}|$ *and let the relative tolerance in the kernel approximation be* τ_1 *(i.e.,* $|\varepsilon(x,y)| < \tau_1$ *for* $\varepsilon(x,y)$ *in* (2.7)*) and the relative approximation tolerance (in Frobenius norm) in the SRRQR factorizations* (4.1) *and* (4.3) *be* τ_2. *Assume the entries of E in* (4.1) *and F in* (4.3) *have magnitudes bounded by e. Then the approximation of* $K^{(X,Y)}$ *by* (4.2) *satisfies*

$$\frac{\|K^{(X,Y)} - UK^{(\hat{X},Y)}\|_F}{\|K^{(X,Y)}\|_F} < s_1\tau_1 + s_2\tau_2, \tag{4.5}$$

where

$$s_1 = 1 + \sqrt{r + (m-r)re^2}\sqrt{1 - \frac{(m-r)(\gamma_2 - \gamma_1)^{2d}}{m(\gamma_1 + \gamma_3)^{2d}}}, \quad s_2 = \frac{\gamma^*(\gamma_1 + \gamma_3)^d}{(\gamma_2 - \gamma^*)(\gamma^* - \gamma_1)^d}.$$

The approximation of $K^{(X,Y)}$ *by* (4.4) *satisfies*

$$\frac{\|K^{(X,Y)} - UK^{(\hat{X},\hat{Y})}V^T\|_F}{\|K^{(X,Y)}\|_F} < s_1\tau_1 + \tilde{s}_2\tau_2, \tag{4.6}$$

where $\tilde{s}_2 = s_2 + s_1 - 1$.

Proof. The following inequalities for $x \in X, y \in Y, z \in Z$ will be useful in the proof:

$$|\phi(z,y)| < \frac{\gamma^*}{N(\gamma_2 - \gamma^*)}, \tag{4.7}$$

$$|\kappa(x,z)| < \frac{1}{(\gamma^* - \gamma_1)^d}, \tag{4.8}$$

$$\frac{1}{(\gamma_1 + \gamma_3)^d} < |\kappa(x,y)| < \frac{1}{(\gamma_2 - \gamma_1)^d}. \tag{4.9}$$

Note that

(4.10)

$$\begin{aligned}
&\|K^{(X,Y)} - UK^{(\hat{X},Y)}\|_F \\
&\quad \leq \|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F + \|\tilde{K}^{(X,Y)} - UK^{(\hat{X},Y)}\|_F \\
&\quad \leq \|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F + \|\tilde{K}^{(X,Y)} - U\tilde{K}^{(\hat{X},Y)}\|_F + \|U\tilde{K}^{(\hat{X},Y)} - UK^{(\hat{X},Y)}\|_F \\
&\quad = \|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F + \|K^{(X,Z)}\Phi^{(Z,Y)} - UK^{(\hat{X},Z)}\Phi^{(Z,Y)}\|_F \\
&\qquad + \|U\tilde{K}^{(\hat{X},Y)} - UK^{(\hat{X},Y)}\|_F \qquad\qquad \text{(by (4.2a)–(4.2c))}
\end{aligned}$$

$$\begin{aligned}&\le \|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F + \|K^{(X,Z)} - UK^{(\hat{X},Z)}\|_F \|\Phi^{(Z,Y)}\|_F \\ &\quad + \|U\|_F \|K^{(\hat{X},Y)} - \tilde{K}^{(\hat{X},Y)}\|_F.\end{aligned}$$

Now, we derive upper bounds separately for the three terms in the last step above.

(i) The first term is the approximation error for the original kernel matrix from the proxy point method. Then

$$\|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F \le \tau_1 \|K^{(X,Y)}\|_F. \tag{4.11}$$

(ii) Next, from the SPRR factorization of $K^{(X,Z)}$,

$$\|K^{(X,Z)} - UK^{(\hat{X},Z)}\|_F \|\Phi^{(Z,Y)}\|_F \le \tau_2 \|K^{(X,Z)}\|_F \|\Phi^{(Z,Y)}\|_F.$$

Since $\Phi^{(Z,Y)}$ is $N \times n$, (4.7) means

$$\|\Phi^{(Z,Y)}\|_F < \sqrt{Nn} \frac{\gamma^*}{N(\gamma_2 - \gamma^*)} = \sqrt{\frac{n}{N}} \frac{\gamma^*}{\gamma_2 - \gamma^*}.$$

Similarly, (4.8) and (4.9) mean

$$\frac{\|K^{(X,Z)}\|_F^2}{\|K^{(X,Y)}\|_F^2} < \frac{mN/(\gamma^* - \gamma_1)^{2d}}{mn/(\gamma_1 + \gamma_3)^{2d}} = \frac{N}{n} \frac{(\gamma_1 + \gamma_3)^{2d}}{(\gamma^* - \gamma_1)^{2d}}.$$

Then

$$\begin{aligned}\|K^{(X,Z)} - UK^{(\hat{X},Z)}\|_F \|\Phi^{(Z,Y)}\|_F &< \tau_2 \sqrt{\frac{n}{N}} \frac{\gamma^*}{\gamma_2 - \gamma^*} \|K^{(X,Z)}\|_F \\ &< \tau_2 \frac{\gamma^* (\gamma_1 + \gamma_3)^d}{(\gamma_2 - \gamma^*)(\gamma^* - \gamma_1)^d} \|K^{(X,Y)}\|_F.\end{aligned} \tag{4.12}$$

(iii) Third,

$$\|U\|_F = \left\| P \begin{pmatrix} I \\ E \end{pmatrix} \right\|_F = \left\| \begin{pmatrix} I \\ E \end{pmatrix} \right\|_F \le \sqrt{r + (m-r)re^2},$$

$$\|K^{(\hat{X},Y)} - \tilde{K}^{(\hat{X},Y)}\|_F \le \tau_1 \|K^{(\hat{X},Y)}\|_F.$$

According to (4.9),

$$\begin{aligned}\frac{\|K^{(\hat{X},Y)}\|_F^2}{\|K^{(X,Y)}\|_F^2} &= 1 - \frac{\|K^{(X \backslash \hat{X},Y)}\|_F^2}{\|K^{(X,Y)}\|_F^2} \\ &\le 1 - \frac{(m-r)n/(\gamma_1+\gamma_3)^{2d}}{mn/(\gamma_2 - \gamma_1)^{2d}} = 1 - \frac{(m-r)(\gamma_2-\gamma_1)^{2d}}{m(\gamma_1 + \gamma_3)^{2d}}.\end{aligned}$$

Then

$$\begin{aligned}&\|U\|_F \|K^{(\hat{X},Y)} - \tilde{K}^{(\hat{X},Y)}\|_F \\ &\le \tau_1 \sqrt{r + (m-r)re^2} \sqrt{1 - \frac{(m-r)(\gamma_2 - \gamma_1)^{2d}}{m(\gamma_1 + \gamma_3)^{2d}}} \|K^{(X,Y)}\|_F.\end{aligned} \tag{4.13}$$

Combining the results (4.11)–(4.13) from the three steps above yields (4.5). To show (4.6), we use the following inequality:

$$\begin{aligned}&\|K^{(X,Y)} - UK^{(\hat{X},\hat{Y})}V^T\|_F\\&\quad\le \|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F + \|K^{(X,Z)}\Phi^{(Z,Y)} - UK^{(\hat{X},Z)}\Phi^{(Z,Y)}\|_F\\&\qquad + \|U\tilde{K}^{(\hat{X},Y)} - UK^{(\hat{X},Y)}\|_F + \|UK^{(\hat{X},Y)} - UK^{(\hat{X},\hat{Y})}V^T\|_F.\end{aligned}$$

Then the proof can proceed similarly. □

If e in SRRQR factorizations is a constant, with fixed N, the two constants in (4.5) scale roughly as $s_1 = \mathcal{O}(\sqrt{m})$ and $s_2 = \mathcal{O}(1)$. Moreover, once the annulus region $\mathcal{A}(0;\gamma_2,\gamma_3)$ is fixed, the set Y is completely irrelevant to the algorithm for obtaining the approximation (4.2) and the error bound (4.5). The column basis matrix U and the set $\hat{X}$ of representative points can be obtained with only the set X, and the error analysis in (4.5) applies to any set Y in $\mathcal{A}(0;\gamma_2,\gamma_3)$.

Remark 4.2. Note that our error analyses in the previous section and this section are not necessarily restricted to the particular kernel like in (1.3) or the proxy point selection method. In fact, the error bounds can be easily modified for more general kernels and/or with other approximation methods as long as a relative error bound for the kernel function approximation is available. This bound is τ_1 in Theorem 4.1.

We then use a comprehensive example to show the accuracies of the analytical compression and the hybrid compression, as well as the selections of the proxy points and the representative points.

Example 3. We generate a triangular finite element mesh on a rectangle domain $[0,2]\times[0,1]$ based on the package MESHPART [11]. The two sets of points X and Y are the mesh points as shown in Figure 5, where $|X| = 821$, $|Y| = 4125$, $\gamma_1 = 0.3$, and $\gamma_2 = 0.45$. We compute the low-rank approximation in (4.2) and report the relative errors in the analytical compression step and the hybrid low-rank approximation, respectively:

$$\mathcal{E}_N(\gamma) = \frac{\|K^{(X,Y)} - \tilde{K}^{(X,Y)}\|_F}{\|K^{(X,Y)}\|_F},\quad \mathcal{R}_N(\gamma) = \frac{\|K^{(X,Y)} - UK^{(\hat{X},Y)}\|_F}{\|K^{(X,Y)}\|_F}.$$

In the first set of tests, the number of proxy points N is chosen to reach a relative tolerance $\tau_1 = 10\varepsilon_{\text{mach}}$ in the proxy point method, where $\varepsilon_{\text{mach}}$ is the machine precision. (Note that τ_1 is the tolerance for approximating $\kappa(x,y)$, and the actual

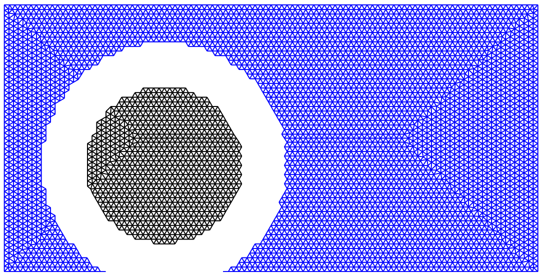
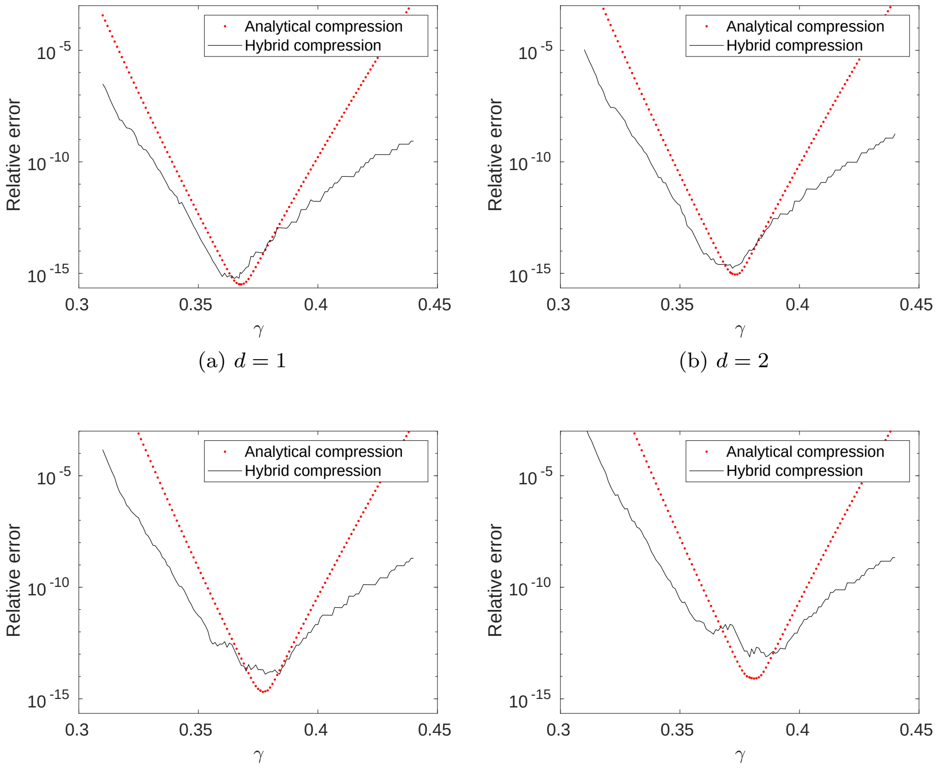


Fig. 5. *Example 3: Sets X and Y in the mesh, where the image is based on the package MESHPART* [11].



(a) $d = 1$ (b) $d = 2$

(c) $d = 3$ (d) $d = 4$

Fig. 6. *Example 3: $\mathcal{E}_N(\gamma)$ in the analytical compression step and $\mathcal{R}_N(\gamma)$ in the hybrid low-rank approximation with varying radius γ.*

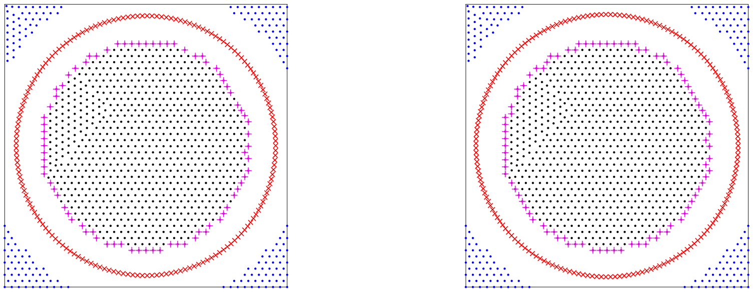
computed Frobenius-norm matrix approximation error $\mathcal{E}_N(\gamma)$ may be slightly larger due to floating point errors.)

We vary the radius γ for the proxy surface between γ_1 and γ_2. For $d = 1, 2, 3, 4$, $\mathcal{E}_N(\gamma)$ and $\mathcal{R}_N(\gamma)$ are shown in Figure 6. In practice, we can use the method in subsection 3.3 to obtain an approximate optimal radius $\tilde{\gamma}^*$. To show that $\tilde{\gamma}^*$ is very close to the actual optimal radius, we can look at Figure 6(a) for $d = 1$. Here, $N = 169$ and $\tilde{\gamma}^* = 0.3675$, which is very close to the actual optimal radius 0.3678. In addition, the error bound in Proposition 3.1 can be used to provide another estimate $\sqrt{\gamma_1\gamma_2} = 0.3674$. Both estimates are very close to the actual minimizer, which indicates the effectiveness of the error analysis and the minimizer estimations. When $\gamma = \tilde{\gamma}^*$, we have $\mathcal{E}_N(\gamma) = 3.2106E - 16$ and $\mathcal{R}_N(\gamma) = 1.1008E - 15$, and the numerical rank resulting from the hybrid compression is 78. The numerical rank produced by SVD under a similar relative error is 68.

Similar results are obtained for $d = 2, 3, 4$. See Figure 6 and Table 1. We notice that $\mathcal{E}_N(\gamma)$ is sometimes larger than $\mathcal{R}_N(\gamma)$, especially when γ is closer to X or Y. This is likely due to the different amount of evaluations of the kernel function in the error computations. The kernel function evaluations may have higher numerical errors when γ gets closer to γ_1 or γ_2. When γ is not too close to γ_1 or γ_2, $\mathcal{R}_N(\gamma)$ is smaller than $\mathcal{E}_N(\gamma)$, which is consistent with the theoretical estimates. Here, no stabilization is integrated into the proxy point method (which may be fixed based on

Table 1
Example 3: Hybrid compression results, where $\tilde{\gamma}^*$ is the approximate optimal radius.

d	N	Optimal γ	$\tilde{\gamma}^*$	Numerical rank	$\mathcal{E}_N(\tilde{\gamma}^*)$	$\mathcal{R}_N(\tilde{\gamma}^*)$
1	169	0.3678	0.3675	78	$3.2106E-16$	$1.1008E-15$
2	179	0.3733	0.3713	88	$1.0431E-15$	$2.1817E-15$
3	187	0.3774	0.3759	93	$2.3565E-15$	$2.0537E-14$
4	193	0.3816	0.3792	99	$8.9381E-15$	$7.5528E-14$



(a) $d = 1$

(b) $d = 2$

Fig. 7. *Example 3: Representative points (+ shapes) and proxy points (× shapes).*

a technique in [3]), while SRRQR factorizations have full stability measurements and produce column basis matrices with controlled norms. On the other hand, this also reflects that hybrid compression is a practical method.

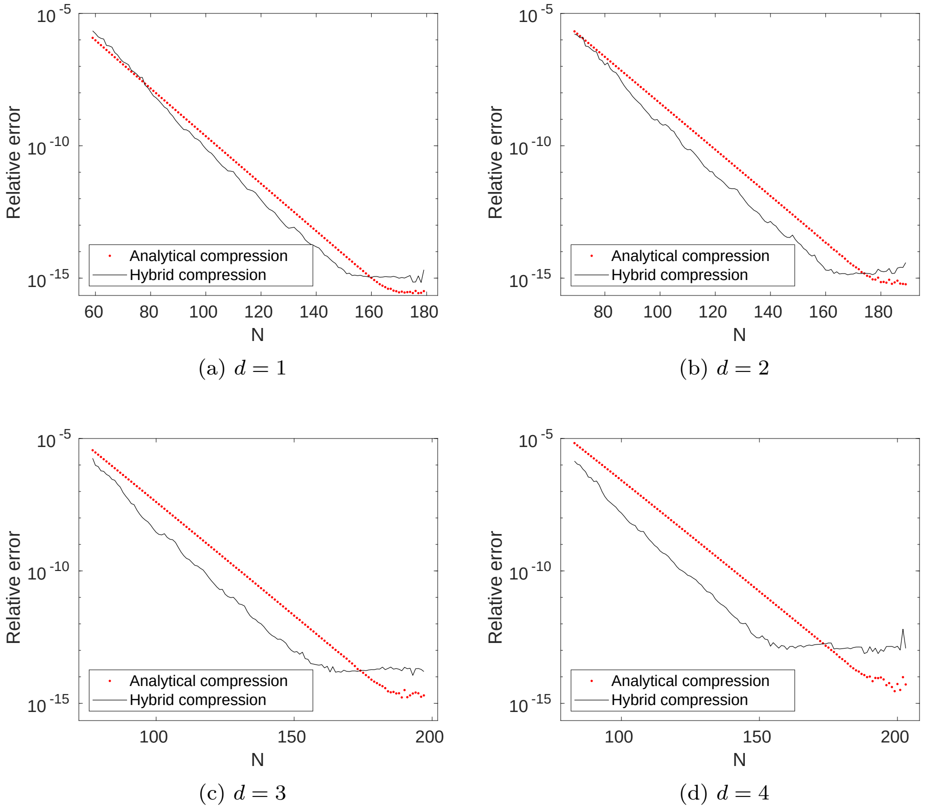
Also in Figure 7 for $d = 1, 2$, we plot the proxy points as well as the representative points $\hat{X}$ produced by the hybrid approximation with $\gamma = \tilde{\gamma}^*$.

In our next set of tests, we vary the number of proxy points N for the analytical compression step and check its effect on the hybrid low-rank approximation error. For each N, the radius of the proxy surface γ is set to be $\tilde{\gamma}^*$. The results are shown in Figure 8. The approximation error for the analytical compression decays exponentially as predicted by Propositions 3.1 and 3.3 (until N reaches the values indicated in Table 1; after that point, it stops decaying due to floating point errors).

5. Discussions. The proxy point method has some attractive features similar to some methods used for data analysis such as the Nyström method and the pseudo-input approximation [8, 13, 26, 40, 46]. For kernel matrices, both the proxy point method and the Nyström method construct low-rank basis matrices directly based on selections of reference points and evaluations of the original kernel function.

However, there are some key differences between the two methods.

1. The Nyström method is typically used to seek low-rank approximations for *square* kernel matrices of the form $K^{(X,X)}$, which corresponds to interactions within the same set X. $K^{(X,X)}$ is often heuristically considered to be of low numerical rank (with modest accuracies) in data science and machine learning applications. On the other hand, the proxy point method deals with rectangular kernel matrices $K^{(X,Y)}$ for two different and well-separated sets X and Y. If $K^{(X,X)}$ is considered, then FMM or $\mathcal{H}/\mathcal{H}^2$/HSS matrix strategies are first applied to generate well-separated subsets. That is, X is first hierarchically partitioned into subsets X_i. Then the proxy point method



(a) $d = 1$

(b) $d = 2$

(c) $d = 3$

(d) $d = 4$

FIG. 8. *Example 3: Accuracies with $\gamma = \tilde{\gamma}^*$ and varying N.*

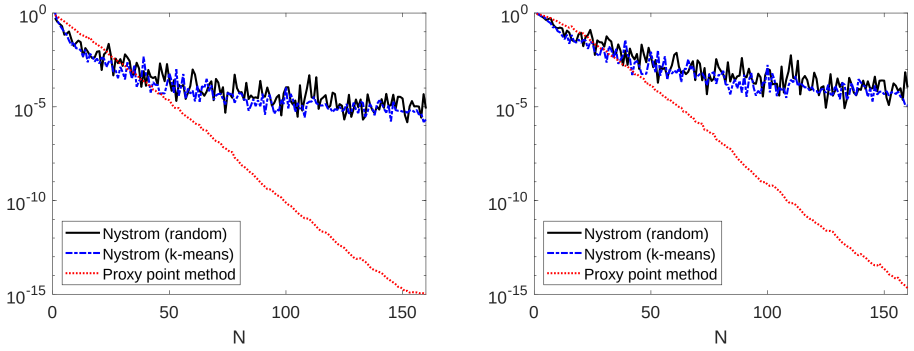
can be applied to $K^{(X_i,X_j)}$ for well-separated X_i and X_j. That is, in the matrix form, the proxy point method compresses appropriate *off-diagonal blocks* of $K^{(X,X)}$. Such an off-diagonal compression idea leads to so-called rank-structured matrices that have been extensively studied in the field of fast solvers for some linear systems, PDEs, and integral equations. (The Nyström method may also be applied to well-separated sets, but it is hard to guarantee high accuracies. See the last point below.)

2. Due to the different natures of the applications that the two methods are applied to, their accuracy requirements are typically quite different. For kernel methods such as the support vector machine or Gaussian process regression, the Nyström method produces modest accuracies (such as $\mathcal{O}(10^{-3}) \sim \mathcal{O}(10^{-1})$) which are good enough for making reasonable predictions in the model. The proxy point method considers interactions between well-separated sets instead of the entire set. For some applications, the separation of sets can be used to analytically justify the low-rankness with any specified accuracy. The proxy point method helps to conveniently compress the off-diagonal blocks of $K^{(X,X)}$ so as to quickly obtain accurate rank-structured matrix approximations to $K^{(X,X)}$ that are suitable for fast and reliable direct factorizations, inversions, eigenvalue solutions, etc.
3. Since the Nyström method often selects points based on techniques such as sampling and clustering, the accuracy analysis is typically probabilistic [8, 55, 56]. The proxy point method here uses a deterministic way to select

proxy points. The proxy point selection and basis matrix computation are supported by analytical justifications with guaranteed controllable accuracies. The analysis enables us to rigorously quantify the error behaviors and to optimize parameters. Of course, this also means that such rigorous analysis is typically nontrivial and is feasible for specific kernels on a case-by-case basis (although the method has been successfully applied to many different types of kernels in practice). Studies for many other kernels still need to be performed, and this paper serves as a starting point for such studies. In addition, as mentioned in Remark 4.2, the hybrid error analysis in Theorem 4.1 is not restricted to specific kernels or proxy point selection methods.

4. The Nyström method may be applied to data points in high dimensions, while the proxy point method focuses on data points in low-dimensional spaces that are often encountered in the solutions of some linear systems, eigenvalue problems, PDEs, and integral equations. For example, the proxy point method are useful for direct solutions of Cauchy/Cauchy-like/Toeplitz/Vandermonde linear systems [34, 39, 49] and FMM accelerations of Hermitian eigenvalue problems [17, 45], where the data points under consideration are on some lines or curves. For some FMM techniques and PDE/integral equation solutions, the points are in one-, two-, or three-dimensional spaces [12, 32, 33, 35, 52, 53].
5. The Nyström method may be extended to well-separated sets X and Y. However, there is no guarantee that a specified high accuracy can be reached. For example, we may obtain an initial approximate column basis matrix $K^{(X,\hat{Y})}$ by selecting a subset $\hat{Y}$ from Y. $K^{(X,\hat{Y})}$ can be used like $K^{(X,Z)}$ in section 4 to obtain an approximation just like (4.2d). (We use this scheme so that its cost is nearly the same as our method. We may also select points from both X and Y in the Nyström method, but the accuracy in the following test is even lower.)

To compare the Nyström scheme in the last item above with the proxy point method for well-separated sets, we apply them to the data sets used in Example 3 by selecting the same number of points N to obtain hybrid compression. In the Nyström method, we try both random sampling with replacement and k-means clustering for selecting reference points like in [56] based on some routines from [54]. The relative approximation errors for the cases $d = 1$ and 2 are plotted in Figure 9. The approxi-



(a) $d = 1$ (b) $d = 2$

FIG. 9. *Relative approximation errors (in Frobenius norm) of the Nyström method and the proxy point method, where the Nyström method uses random sampling or k-means clustering for selecting reference points.*

mation accuracy from the Nyström method initially improves with increasing N, but the accuracy improvement gets very slow and almost stagnates. In comparison, the errors from the proxy point method decrease all the way to near machine precision.

6. Conclusions.

The proxy point method is a very simple and convenient strategy for computing low-rank approximations for kernel matrices evaluated at well-separated sets. In this paper, we present an intuitive way of explaining the method. Moreover, we provide rigorous approximation error analysis for the kernel function approximation and low-rank kernel matrix approximation in terms of a class of important kernels. Based on the analysis, we show how to choose nearly optimal locations of the proxy points. The work can serve as a starting point to study the proxy point method for more general kernels. Some possible strategies in future work will be based on other kernel expansions or Cauchy FMM ideas [28]. Various results here are already applicable to more general kernels and other approximation methods. We also hope this work can draw more attention from researchers in the field of matrix computations to study and utilize such an elegant method.

Acknowledgments. The authors would like to thank Steven Bell at Purdue University for some helpful discussions and thank the referees for valuable comments.

REFERENCES

[1] C. R. Anderson, *An implementation of the fast multipole method without mltipoles*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 923–947.

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

[3] D. Cai and J. Xia, *A Stable Matrix Version of the Fast Multipole Method*, submitted, 2020.

[4] R. H. Chan, J. Xia, and X. Ye, *Fast Direct Solvers for Linear Third-Order Differential Equations*, preprint, 2016.

[5] S. Chandrasekaran, M. Gu, and T. Pals, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 603–622.

[6] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, *A superfast algorithm for Toeplitz systems of linear equations*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1247–1266.

[7] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1389–1404.

[8] P. Drineas and M. W. Mahoney, *On the Nyström method for approximating a Gram matrix for improved kernel-based learning*, J. Machine Learning, 6 (2005), pp. 2153–2175.

[9] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

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

[11] J. R. Gilbert and S.-H. Teng, *Matlab Mesh Partitioning and Graph Separator Toolbox*, https://sites.cs.ucsb.edu/~gilbert/cs140/matlab/graphs/meshpart/meshpart.htm.

[12] A. Gillman, P. M. Young, and P. G. Martinsson, *A direct solver with O(N) complexity for integral equations on one-dimensional domains*, Front. Math. China, 7 (2009), pp. 217–247.

[13] A. Gittens and M. W. Mahoney, *Revisiting the Nyström method for improved large-scale machine learning*, J. Machine Learning, 16 (2016), pp. 1–65.

[14] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, in Structured Matrices in Mathematics, Computer Science, and Engineering, Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 47–52.

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

[16] M. Gu, *Subspace iteration randomization and singular value problems*, SIAM J. Sci. Comput., 37 (2015), pp. A1139–A1173.

[17] M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 79–92.

[18] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

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

[20] W. Hackbusch, B. Khoromskij, and S. Sauter, *On $\mathcal{H}^2$ matrices*, in Lectures on Applied Mathematics, Springer, Berlin, 2000, pp. 9–29.

[21] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[22] K. L. Ho and L. Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J. Sci. Comput., 34 (2012), pp. A2507–A2532.

[23] J. Kestyn, E. Polizzi, and P. T. P. Tang, *FEAST eigensolver for non-Hermitian problems*, SIAM J. Sci. Comput., 38 (2016), pp. S772–S799.

[24] N. Kishore Kumar and J. Schneider, *Literature survey on low rank approximation of matrices*, Linear Multilinear Algebra, 65 (2017), pp. 2212–2244.

[25] R. Kress, *Linear Integral Equations*, 3rd ed., Springer, Berlin, 2014.

[26] S. Kumar, M. Mohri, and A. Talwalkar, *Sampling methods for the Nyström method*, J. Machine Learning, 13 (2002), pp. 981–1006.

[27] X. Liu, J. Xia, and M. V. de Hoop, *Parallel randomized and matrix-free direct solvers for large structured dense linear systems*, SIAM J. Sci. Comput., 38 (2016), pp. S508–S538.

[28] P.-D. Létourneau, C. Cecka, and E. Darve, *Cauchy fast multipole method for general analytic kernels*, SIAM J. Sci. Comput., 36 (2014), pp. A396–A426.

[29] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 697–702.

[30] J. Makino, *Yet another fast multipole method without multipoles–pseudoparticle multipole method*, J. Comput. Phys., 151 (1999), pp. 910–920.

[31] P.-G. Martinsson, G. Q. Ortí, N. Heavner, and R. van de Geijn, *Householder QR factorization with randomization for column pivoting (HQRRP)*, SIAM J. Sci. Comput., 39 (2017), pp. C96–C115.

[32] P. G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comput. Phys., 205 (2005), pp. 1–23.

[33] 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.

[34] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.

[35] E. Michielssen and A. Boag, *A multilevel matrix decomposition algorithm for analyzing scattering from large structures*, IEEE Trans. Antennas Propagation, 44 (1996), pp. 1086–1093.

[36] V. Minden, K. L. Ho, A. Damle, and L. Ying, *A recursive skeletonization factorization based on strong admissibility*, Multiscale Model. Simul., 15 (2017), pp. 768–796.

[37] L. Miranian and M. Gu, *Strong rank-revealing LU factorizations*, Linear Algebra Appl., 367 (2003), pp. 1–16.

[38] M. O'Neil and V. Rokhlin, *A New Class of Analysis-Based Fast Transforms*, Technical report, Yale University, YALE/DCS/TR1384, 2007.

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

[40] E. Snelson and Z. Ghahramani, *Sparse Gaussian processes using pseudo-inputs*, in Advances in Neural Information Processing Systems 18, 2005, pp. 1257–1264.

[41] E. M. Stein and R. Shakarchi, *Complex Analysis*, Princeton University Press, Princeton, NJ, 2003.

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

[43] L. N. Trefethen and J. A. C. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.

[44] E. E. Tyrtyshnikov, *Mosaic-skeleton approximations*, Calcolo, 33 (1996), pp. 47–57.

[45] 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.

[46] C. Williams and M. Seeger, *Using the Nyström method to speed up kernel machines*, in Advances in Neural Information Processing Systems 13, 2001, pp. 682–688.

[47] J. Xia, *Randomized sparse direct solvers*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 197–227.

[48] 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.

[49] J. Xia, Y. Xi, and M. Gu, *A superfast structured solver for Toeplitz linear systems via randomized sampling*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 837–858.

[50] X. Xing and E. Chow, *An Efficient Method for Block Low-Rank Approximations for Kernel Matrix Systems*, preprint, 2018.

[51] X. Ye, J. Xia, R. H. Chan, S. Cauley, and V. Balakrishnan, *A fast contour-integral eigensolver for non-hermitian matrices*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1268–1297.

[52] L. Ying, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457.

[53] 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.

[54] K. Zhang, *Improved Nystrom Kernel Low-Rank Approximation*, MATLAB Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/38422-improved-nystrom-kernel-low-rank-approximation.

[55] K. Zhang and J. T. Kwok, *Block-quantized kernel matrix for fast spectral embedding*, in Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 1097–1104.

[56] K. Zhang, I. W. Tsang, and J. T. Kwok, *Improved Nyström low-rank approximation and error analysis*, in Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 1232–1239.