

INTERPOLATIVE DECOMPOSITION VIA PROXY POINTS FOR  
KERNEL MATRICES\*XIN XING<sup>†</sup> AND EDMOND CHOW<sup>‡</sup>

**Abstract.** In the construction of rank-structured matrix representations of dense kernel matrices, a heuristic compression method, called the proxy point method, has been used in practice to efficiently compute the low-rank approximation of certain kernel matrix blocks in the form of an interpolative decomposition. We present a long overdue error analysis for the proxy point method, rigorously proving the effectiveness of the method under specific conditions. The analysis also generalizes the method, allowing it to be applied to the construction of different types of rank-structured matrices with general kernel functions in low-dimensional spaces. Based on the analysis, a systematic and adaptive scheme for selecting the proxy points used in the method is developed, which can guarantee that the method is effective for any given kernel function under specific conditions.

**Key words.** interpolative decomposition, low-rank approximation, rank-structured matrix, proxy point method, kernel matrix

**AMS subject classifications.** 15A23, 15B99, 65F99

**DOI.** 10.1137/19M1258700

**1. Introduction.** Large dense kernel matrices appear in many scientific computing problems. The quadratic cost for matrix storage and matrix-vector multiplications is the main challenge in working with these matrices. A class of tools that reduces these costs is *rank-structured matrix* techniques, such as  $\mathcal{H}$  [21, 24],  $\mathcal{H}^2$  [22, 23], HSS [10], HODLR [1], directional  $\mathcal{H}^2$  [3, 5], and butterfly factorization [27]. The main idea of these techniques is to first locate the blocks of a kernel matrix that are numerically low-rank, and then represent these blocks in low-rank form. Representing a kernel matrix in a rank-structured matrix format generally requires expensive computation to construct low-rank approximations of kernel matrix blocks (referred to as *compressing* the blocks). For example, compressing the blocks via the singular value decomposition (SVD) or QR factorization can lead to prohibitive quadratic construction cost. As a result, efficient compression methods are critical for the construction and application of rank-structured matrix techniques.

For a class of fast direct solvers [11, 16, 25, 29] for kernel matrices based on HSS format, an acceleration technique called the *proxy surface method* has been proposed to efficiently compress specific kernel matrix blocks in a low-rank form called an interpolative decomposition [11, 20] (ID, to be explained in subsection 2.2), dramatically reducing the construction cost for HSS formats. Methods closely related to the proxy surface method also exist which differ in their selection of so-called proxy points [12, 32]. Together, all these methods, including the proxy surface method, have a general form that we refer to as *the proxy point method*. As to be discussed later in this section, while the proxy point method has many advantages, several important

---

\*Received by the editors April 29, 2019; accepted for publication (in revised form) by S. Le Borne November 14, 2019; published electronically January 30, 2020.

<https://doi.org/10.1137/19M1258700>

**Funding:** The work of the authors was supported by the National Science Foundation grant ACI-1609842.

<sup>†</sup>School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0765 (xxing33@gatech.edu).

<sup>‡</sup>School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332-4017 (echow@cc.gatech.edu).

problems about the method still remain unsolved, which limits its present application to the construction of HSS and  $\mathcal{H}^2$  matrices with specific kernel functions. In this paper, we address these problems and generalize the application of the method to the construction of different types of rank-structured matrices with kernel functions that are more general than those usually used.

Figure 1.1 gives a simple illustration of the proxy surface method which calculates an ID approximation of a kernel matrix block  $K(X_0, Y_0) = (K(x_i, y_j))_{x_i \in X_0, y_j \in Y_0}$  where  $K(x, y)$  is the Laplace kernel and  $X_0$  and  $Y_0$  are two sets of points lying in domains  $\mathcal{X}$  and  $\mathcal{Y}$ . The method first uniformly selects a small set of *proxy points*  $Y_p$  on the interior boundary of  $\mathcal{Y}$ . An ID approximation of  $K(X_0, Y_p)$  is then computed algebraically, where  $K(X_0, Y_p)$  is usually a much smaller matrix than  $K(X_0, Y_0)$ . Lastly, this ID approximation is used to efficiently construct an ID approximation of  $K(X_0, Y_0)$ . The proxy point method generalizes the proxy surface method, where the proxy points  $Y_p$  can be selected in the whole domain  $\mathcal{Y}$  and the domains  $\mathcal{X}$  and  $\mathcal{Y}$  are chosen differently for different kernel functions (see Figure 2.1 later in this paper).

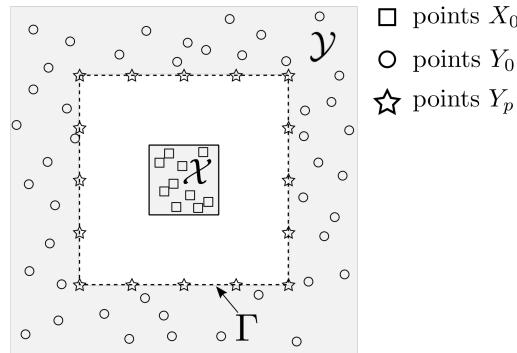


FIG. 1.1. Illustration of the proxy surface method. The proxy points  $Y_p$  are uniformly selected on the interior boundary  $\Gamma$  of  $\mathcal{Y}$ . The ID approximation of  $K(X_0, Y_0)$  is efficiently constructed via an algebraic ID approximation of  $K(X_0, Y_p)$ .

For  $K(X_0, Y_0)$  described above, the more commonly used efficient compression methods are analytic techniques based on a degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ , such as multipole expansion [19], polynomial interpolation [15, 17], Fourier series [38], and other special expansions [2, 13, 36, 37]. Compared with analytic techniques, the proxy point method only requires that a degenerate approximation of  $K(x, y)$  exists and does not need that approximation to be known explicitly (to be explained in section 4). The approximation accuracy of the method for  $K(X_0, Y_0)$  is directly controlled by controlling the accuracy of the ID approximation of  $K(X_0, Y_p)$ . Meanwhile, controlling the approximation accuracy of analytic techniques generally requires trial-and-error or algebraic recompression.

The proxy point method, however, has limited application due to several unsolved problems. First, the proper selection of proxy points  $Y_p$  is critical to controlling the approximation accuracy of the method, and such a selection varies for different kernel functions. However,  $Y_p$  is only heuristically selected in practice. Figure 1.2 shows a numerical example where the proxy point method with  $Y_p$  selected as in Figure 1.1 works well for the Laplace kernel but poorly for a Gaussian kernel. A more effective but still heuristic selection of  $Y_p$  for Gaussian kernels is suggested in [32]. Second, the

effectiveness of the method is, so far, only supported by numerical results. A rigorous explanation of how the proxy point method works and under what conditions is still missing.

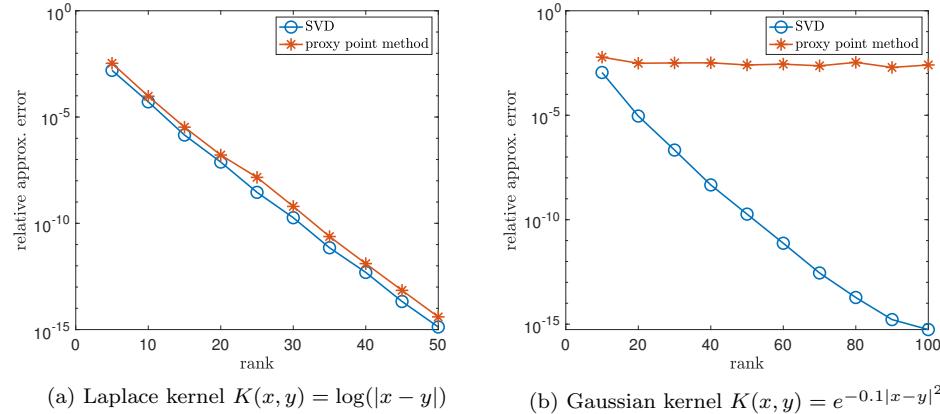


FIG. 1.2. *Relative approximation error of the proxy point method for  $K(X_0, Y_0)$  with different approximation ranks and with two different kernel functions. Let  $\mathcal{X} = [-1, 1]^2$ ,  $\mathcal{Y} = [-5, 5]^2 \setminus [-3, 3]^2$ , and  $\Gamma = \partial([-3, 3]^2)$  as illustrated in Figure 1.1. We randomly select 400 points in  $\mathcal{X}$  for  $X_0$  and 6400 points in  $\mathcal{Y}$  for  $Y_0$ , and uniformly select 300 points on  $\Gamma$  for  $Y_p$ .*

In this paper, the proxy point method is presented for a general kernel function  $K(x, y)$  with compact domains  $\mathcal{X}$  and  $\mathcal{Y}$  satisfying that  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  is smooth and can be represented by an accurate, low-degree degenerate approximation (to be explained in subsection 2.1). In this general problem setting, we provide a rigorous error analysis for the proxy point method. Using this error analysis, we further develop a systematic and adaptive scheme to select proxy points that can guarantee the method to be effective, paving the way for a wider application of the proxy point method in the construction of different types of rank-structured matrices.

## 2. Background.

**2.1. Low-rank kernel matrix blocks.** Let  $K(x, y)$  be a kernel function in a pair of domains  $\mathcal{X} \times \mathcal{Y}$ . In this paper, we always assume that domains  $\mathcal{X}$  and  $\mathcal{Y}$  are *compact*, i.e., closed and bounded, and that  $K(x, y)$  is *smooth* in  $\mathcal{X} \times \mathcal{Y}$  for the simplicity of analysis, which is mostly the case in practice. For two sets of points,  $X_0$  in  $\mathcal{X}$  and  $Y_0$  in  $\mathcal{Y}$ , a kernel matrix block  $K(X_0, Y_0)$  consists of the entries  $K(x_i, y_j)$  for all pairs of  $x_i \in X_0$  and  $y_j \in Y_0$ . The low-rank property of  $K(X_0, Y_0)$  is closely related to the existence of a low-degree degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ . A degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  is precisely defined as follows.

**DEFINITION 2.1.**  *$K(x, y)$  is said to have an  $r$ -term  $\varepsilon$ -expansion in  $\mathcal{X} \times \mathcal{Y}$  if there exist functions  $\{\psi_i(x)\}_{i=1}^r$  and  $\{\phi_i(y)\}_{i=1}^r$  such that*

$$(2.1) \quad \left| K(x, y) - \sum_{i=1}^r \psi_i(x) \phi_i(y) \right| \leq \varepsilon, \quad x \in \mathcal{X}, y \in \mathcal{Y}.$$

*The summation in (2.1) is called a degenerate approximation (also known as separated representation) of  $K(x, y)$  with degree  $r$  and accuracy  $\varepsilon$  in  $\mathcal{X} \times \mathcal{Y}$ .*

Let  $\Psi(x)$  and  $\Phi(y)$  denote the  $r$ -dimensional vectors of functions  $\{\psi_i(x)\}_{i=1}^r$  and  $\{\phi_i(y)\}_{i=1}^r$  from (2.1), respectively. The approximation (2.1) can be written as

$$K(x, y) = \Psi(x)^T \Phi(y) + O(\varepsilon).$$

Substituting all pairs of  $x_i \in X_0$  and  $y_j \in Y_0$  into the above equation gives

$$(2.2) \quad K(X_0, Y_0) \approx \Psi(X_0)^T \Phi(Y_0),$$

where  $\Psi(X_0) \in \mathbb{R}^{r \times |X_0|}$  denotes the matrix of column vectors  $\Psi(x_i)$  for all  $x_i \in X_0$  and  $\Phi(Y_0)$  is similarly defined. Equation (2.2) gives a rank- $r$  approximation  $\Psi(X_0)^T \Phi(Y_0)$  to  $K(X_0, Y_0)$  with  $O(\varepsilon)$  error.

To locate the low-rank blocks of a kernel matrix, instead of algebraically checking the numerical rank of a block  $K(X_0, Y_0)$ , rank-structured matrix techniques check the pair of domains  $\mathcal{X}$  and  $\mathcal{Y}$  to decide whether there exists an accurate, low-degree degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ . Here, “low-degree” usually refers to  $O(1)$ -degree. More specifically, for common kernel functions used in practice, there exist geometric criteria for a pair of domains  $\mathcal{X}$  and  $\mathcal{Y}$  that can decide whether such a degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  exists. Such geometric criteria are referred to as *admissibility conditions* and a pair of domains that meet the criteria are said to be *admissible*. Figure 2.1 shows examples of admissible pairs of domains in three different situations.

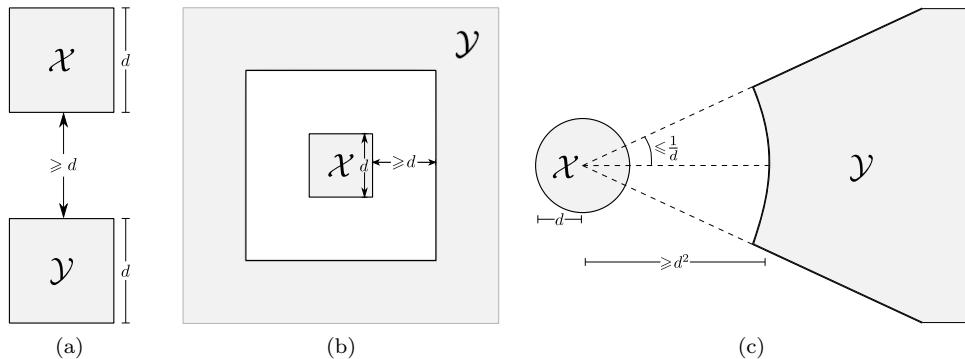


FIG. 2.1. Examples of admissible pairs of domains  $\mathcal{X} \times \mathcal{Y}$  for different kernel functions  $K(x, y)$ : (a)  $K(x, y) = \log(|x-y|)$  used in  $\mathcal{H}$  matrices [24], (b)  $K(x, y) = \log(|x-y|)$  used in  $\mathcal{H}^2$  matrices [23], (c) Two-dimensional (2D) Helmholtz kernel used in the fast directional multilevel algorithm [13]. Admissibility conditions generally have constraints on the relative size of certain geometric features of  $\mathcal{X}$  and  $\mathcal{Y}$  which is characterized by the parameter  $d$  in the figures.

Different rank-structured matrices use different admissibility conditions for a kernel function to locate the low-rank blocks of a kernel matrix. For example, admissibility conditions for nonoscillatory kernels used in  $\mathcal{H}$  and  $\mathcal{H}^2$  matrices are illustrated in Figure 2.1a and Figure 2.1b, respectively. For oscillatory Helmholtz kernels, more complicated admissibility conditions are used in the butterfly method [31], directional  $\mathcal{H}^2$  matrices [3], and the fast directional multilevel algorithm [13]. Figure 2.1c illustrates the admissible condition used in [13]. In the construction of a rank-structured matrix with a corresponding admissibility condition, a kernel matrix block  $K(X_0, Y_0)$  is approximated by a low-rank form if  $X_0$  and  $Y_0$  are in an admissible pair of domains. Compression of these kernel blocks dominates the construction cost.

We note that the above discussion of rank-structured matrices based on admissible domains mainly works for kernel matrices with points in low-dimensional spaces (e.g., 2D and three-dimensional [3D]). In high-dimensional spaces, admissibility conditions that guarantee the existence of low-degree degenerate approximations in a pair of domains are generally not available. In this case, the low-rank property of kernel matrix blocks usually relies on the existence of low-dimensional manifolds that support all the points. In practice, heuristic criteria based on the distances between points [28] have been used to locate low-rank blocks.

Like all the analytic techniques based on degenerate approximations, the proxy point method discussed in this paper requires an admissible pair of domains associated with the approximated kernel matrix block. Thus, it cannot be applied directly to high-dimensional problems. Existing methods for high-dimensional problems, e.g., [28, 33], all rely on purely algebraic compression techniques.

**2.2. Proxy surface and related methods.** An interpolative decomposition [11, 20] represents or approximates a matrix  $A \in \mathbb{R}^{n \times m}$  in the low-rank form  $UA_J$ , where  $U \in \mathbb{R}^{n \times k}$  has bounded entries,  $A_J \in \mathbb{R}^{k \times m}$  contains  $k$  rows of  $A$ , and  $k$  is the rank. An ID approximation defined this way is said to have error below the *error threshold*  $\varepsilon_0$  if the 2-norm of each row of  $A - UA_J$  is bounded by  $\varepsilon_0$ . Using an algebraic approach, an ID approximation with a given rank or a given error threshold can be calculated using the strong rank-revealing QR (SRRQR) decomposition [20] with typical complexity  $O(knm)$  (or  $O(knm \log(n))$  in rare cases). The matrix  $U$  obtained by this approach can have all its entries bounded by a prespecified parameter  $C_{qr} \geq 1$ . Specifically, an ID approximation of a kernel matrix  $K(X_0, Y_0)$  can be written as

$$K(X_0, Y_0) \approx UK(X_{\text{id}}, Y_0),$$

where  $K(X_{\text{id}}, Y_0)$  contains a subset of rows in  $K(X_0, Y_0)$  and  $X_{\text{id}}$  is a subset of points in  $X_0$ . The components  $U$  and  $X_{\text{id}}$  uniquely decide such an ID approximation.

To the best of our knowledge, only two specific examples of the proxy point method have been used in practice, the proxy surface method [11, 16, 25, 29] and a variant [12, 32] of the proxy surface method.

The proxy surface method illustrated in Figure 1.1 in the introduction is a specific case of the general proxy point method (to be described in Algorithm 3.1) simply with the proxy points  $Y_p$  selected on the boundary of  $\mathcal{Y}$ . The proxy surface method mainly works for kernel functions from potential theory, such as the Laplace and the Stokes kernel. The motivation and heuristic explanation [29] for the method are based on Green's identity from potential theory, where proxy points play the role of “equivalent charges” for the interactions (defined by the kernel function) between charges located at  $X_0$  and those at  $Y_0$ .

For Gaussian kernels, the proxy surface method works poorly as illustrated previously by Figure 1.2. In this case, a heuristic and effective variant method [32] is to select the proxy points  $Y_p$  in an annulus around the boundary of  $\mathcal{Y}$  and the remaining steps are exactly the same as the proxy surface method.

Existing works for the error analysis of the proxy surface and related methods are limited. Reference [34] provides an error analysis specifically for the proxy surface method for the 3D Laplace kernel. Reference [35] considers a variant of the proxy surface method with kernel functions in the specific form  $K(x, y) = \frac{1}{|x-y|^d}$  with  $x, y$  in the complex plane and  $d$  being a positive integer. Compared to these existing results, our error analysis in this paper is more general and can be applied to the proxy point method with broader sets of kernel functions, domains, and proxy points.

**3. Proxy point method.** Recall that the proxy surface method and its variant only work for specific kernel functions with an admissible pair of domains as exemplified in Figure 1.1. The proxy point method has a more general form than the proxy surface method and its variant and is described in Algorithm 3.1. It turns out that the proxy point method can calculate a good ID approximation of a kernel matrix block  $K(X_0, Y_0)$  for any smooth kernel function  $K(x, y)$  as long as the pair of compact domains  $\mathcal{X}$  and  $\mathcal{Y}$  enclosing  $X_0$  and  $Y_0$  is admissible for  $K(x, y)$ , i.e., there exists an accurate, low-degree degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ . The key is to select a set of proxy points  $Y_p$  in  $\mathcal{Y}$  properly and adaptively according to the kernel function  $K(x, y)$  and the domains  $\mathcal{X}$  and  $\mathcal{Y}$ .

---

**Algorithm 3.1** Proxy point method.

---

**Input:**  $\mathcal{X}, \mathcal{Y}, X_0, Y_0$ .

**Output:** ID approximation of  $K(X_0, Y_0)$ ,

$$(3.1) \quad K(X_0, Y_0) \approx UK(X_{\text{id}}, Y_0), \quad X_{\text{id}} \subset X_0.$$

**Step 1:** Select a set of proxy points  $Y_p$  in the domain  $\mathcal{Y}$ . Points in  $Y_p$  are independent of  $X_0$  and  $Y_0$ .

**Step 2:** Calculate  $U$  and  $X_{\text{id}}$  for (3.1) from an algebraic ID approximation of  $K(X_0, Y_p)$  using SRRQR with a given error threshold (or a given rank),

$$(3.2) \quad K(X_0, Y_p) \approx UK(X_{\text{id}}, Y_p).$$


---

This section provides a fundamental mathematical interpretation of how the proxy point method works and also connects the method with an existing analytic compression technique. Section 4 then rigorously studies the approximation error of the proxy point method in the general case, proving the effectiveness of the method. Lastly, section 5 discusses how to systematically select a proper set of proxy points for any given kernel function  $K(x, y)$  and domains  $\mathcal{X}$  and  $\mathcal{Y}$ .

**3.1. Algorithm interpretation.** The proxy point method can be interpreted as follows. In Algorithm 3.1, the ID approximations (3.1) and (3.2) can be viewed row-by-row as

$$(3.3) \quad K(X_0, Y_0) \approx UK(X_{\text{id}}, Y_0) \iff K(x_i, Y_0) \approx u_i^T K(X_{\text{id}}, Y_0), \quad x_i \in X_0,$$

$$(3.4) \quad K(X_0, Y_p) \approx UK(X_{\text{id}}, Y_p) \iff K(x_i, Y_p) \approx u_i^T K(X_{\text{id}}, Y_p), \quad x_i \in X_0,$$

where  $u_i^T$  denotes the  $i$ th row of  $U$ . For each  $x_i \in X_0$ , the above two row approximations are connected by the function approximation in the domain  $\mathcal{Y}$ ,

$$(3.5) \quad K(x_i, y) \approx u_i^T K(X_{\text{id}}, y), \quad y \in \mathcal{Y}.$$

Evaluating this function approximation at  $Y_0$  and  $Y_p$  gives the row approximations (3.3) and (3.4), respectively. Furthermore, it always holds that

$$\|K(x_i, Y_0) - u_i^T K(X_{\text{id}}, Y_0)\|_2 / \sqrt{|Y_0|} \leq \max_{y \in \mathcal{Y}} |K(x_i, y) - u_i^T K(X_{\text{id}}, y)|,$$

with any ID components  $U$  and  $X_{\text{id}}$ , which is based on the simple inequality  $\|v\|_2 \leq \sqrt{n} \max_i |v_i|$  for  $v \in \mathbb{R}^n$  and the fact that  $Y_0 \subset \mathcal{Y}$ .

From this analysis, a good ID approximation  $UK(X_{\text{id}}, Y_0)$  to  $K(X_0, Y_0)$  can be found by seeking  $U$  and  $X_{\text{id}}$  such that each function approximation defined in (3.5) has small error in  $\mathcal{Y}$ . To make the problem tractable, instead of considering the approximation (3.5) at every  $y \in \mathcal{Y}$ , the proxy point method considers it at the finite set of proxy points  $Y_p \subset \mathcal{Y}$ . With  $\mathcal{X} \times \mathcal{Y}$  being admissible for  $K(x, y)$ , it turns out that there exists a proper selection of  $Y_p$  such that the approximation (3.5) to each  $K(x_i, y)$  with any ID components  $U$  and  $X_{\text{id}}$  has maximum absolute error bounded by a small multiple of its root-mean-square error at  $Y_p$ , i.e.,

$$(3.6) \quad \max_{y \in \mathcal{Y}} |K(x_i, y) - u_i^T K(X_{\text{id}}, y)| \leq O(1) \left( \|K(x_i, Y_p) - u_i^T K(X_{\text{id}}, Y_p)\|_2 / \sqrt{|Y_p|} \right).$$

Such a selection of  $Y_p$  will be discussed in section 5.

Assuming that we have a set of proxy points  $Y_p$  satisfying (3.6), the proxy point method calculates  $U$  and  $X_{\text{id}}$  from (3.2) using SRRQR, which has the approximation error  $\|K(x_i, Y_p) - u_i^T K(X_{\text{id}}, Y_p)\|_2$  for each row bounded by a specified error threshold. According to (3.6), the resulting  $U$  and  $X_{\text{id}}$  define a good function approximation (3.5) at points in  $Y_p$  and thus in  $\mathcal{Y}$ . As a result, the error of the ID approximation  $UK(X_{\text{id}}, Y_0)$  to  $K(X_0, Y_0)$  is controlled by the error of the ID approximation  $UK(X_{\text{id}}, Y_p)$  to  $K(X_0, Y_p)$  as

$$\|K(x_i, Y_0) - u_i^T K(X_{\text{id}}, Y_0)\|_2 / \sqrt{|Y_0|} \leq O(1) \left( \|K(x_i, Y_p) - u_i^T K(X_{\text{id}}, Y_p)\|_2 / \sqrt{|Y_p|} \right).$$

As to be shown in section 5, the number of proxy points  $Y_p$  needed to satisfy (3.6) can be as small as the degree of a putative degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  with a given accuracy. Therefore,  $K(X_0, Y_p)$  is usually a much smaller matrix than  $K(X_0, Y_0)$ , and the proxy point method in Algorithm 3.1 can be much faster than the direct ID approximation of  $K(X_0, Y_0)$  using SRRQR alone.

**3.2. Connection with pseudoskeleton approximation.** The general proxy point method is closely related to the degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  in a pseudoskeleton form [18], i.e.,

$$(3.7) \quad K(x, y) \approx K(x, Y_{pt}) K(X_{pt}, Y_{pt})^\dagger K(X_{pt}, y), \quad x \in \mathcal{X}, y \in \mathcal{Y},$$

where  $X_{pt}$  and  $Y_{pt}$  are “pivot” points selected in  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively, and the dagger symbol “ $\dagger$ ” denotes the pseudoinverse of a matrix. Pseudoskeleton approximations with different choices of  $X_{pt}$  and  $Y_{pt}$  are used in many fast matrix-vector multiplication algorithms such as the kernel independent fast multipole method [36, 37], the fast directional multilevel algorithm [13], and the butterfly method [31]. Compression techniques such as adaptive cross approximation [4] and skeletonized interpolation [9] are also in this form.

The proxy point method turns out to be equivalent to the combination of a pseudoskeleton approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  and an algebraic recompression. For two sets of points,  $X_0$  in  $\mathcal{X}$  and  $Y_0$  in  $\mathcal{Y}$ , a pseudoskeleton approximation (3.7) defines a low-rank approximation of  $K(X_0, Y_0)$  as

$$(3.8) \quad K(X_0, Y_0) \approx K(X_0, Y_{pt}) K(X_{pt}, Y_{pt})^\dagger K(X_{pt}, Y_0).$$

Then consider a recompression of the above low-rank approximation via computing an ID approximation  $UK(X_{\text{id}}, Y_{pt})$  to the factor  $K(X_0, Y_{pt})$  using SRRQR. This step

is analogous to the computation of (3.2) in the proxy point method. With this ID approximation, the approximation (3.8) is written as

$$\begin{aligned} K(X_0, Y_0) &\approx UK(X_{\text{id}}, Y_{pt})K(X_{pt}, Y_{pt})^\dagger K(X_{pt}, Y_0) \\ &\approx UK(X_{\text{id}}, Y_0), \end{aligned}$$

where the second approximation is obtained by substituting  $X_{\text{id}}$  and  $Y_0$  into (3.7). This final ID approximation  $UK(X_{\text{id}}, Y_0)$  is exactly the result (3.1) computed by the proxy point method if  $Y_p$  is selected as  $Y_{pt}$ .

The main difference between the proxy point method and the above recompressed pseudoskeleton approximation is that there is no need for  $X_{pt}$  and  $K(X_{pt}, Y_{pt})^\dagger$  in the proxy point method. The matrix  $K(X_{pt}, Y_{pt})$  is usually close to singular and calculation of its pseudoinverse can be numerically unstable. In practice, [37] applies a Tikhonov regularization to compute  $K(X_{pt}, Y_{pt})^\dagger$ , and [9] applies a backward stable algorithm to compute  $K(X_{pt}, Y_{pt})^\dagger K(X_{pt}, y)$  for a given point  $y$ .

**3.3. Examples of proxy points.** In the proxy point method, the key component is a set of proxy points  $Y_p$  that satisfies (3.6). Clearly, the selection of proxy points in  $\mathcal{Y}$  should be related to  $K(x, y)$  and  $\mathcal{X} \times \mathcal{Y}$ . Based on the above connection between the proxy point method and the pseudoskeleton approximation, it is natural to use pivot points  $Y_{pt}$  from existing pseudoskeleton approximation methods as the corresponding proxy points  $Y_p$ . Figure 3.1 shows three examples of proxy points borrowed from existing pseudoskeleton approximation methods. The effectiveness of some of these proxy point selections can be rigorously justified by the error analysis in section 4.

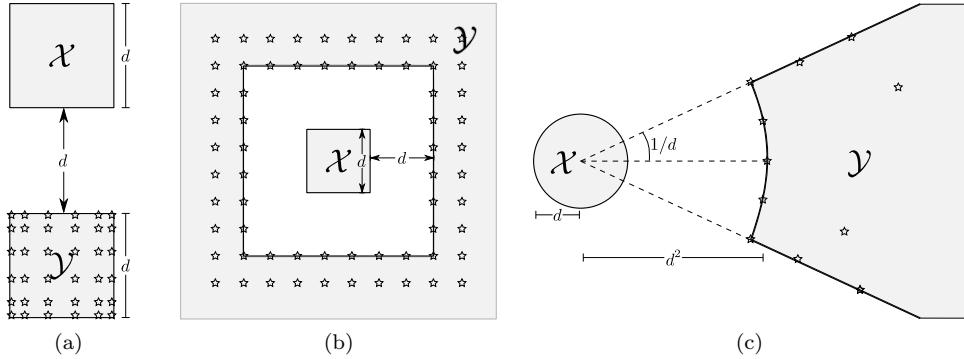


FIG. 3.1. Examples of proxy points (marked as stars) borrowed from existing pseudoskeleton approximation methods for different kernel functions and corresponding admissible domain pairs: (a) For smooth kernels that can be well approximated by polynomials in  $\mathcal{Y}$ , proxy points are selected as a tensor grid of Chebyshev points in  $\mathcal{Y}$  based on [9]. (b) For smooth radial basis functions, proxy points are selected as multiple layers of uniform grid points around the interior boundary of  $\mathcal{Y}$  based on [36]. (c) For oscillatory kernels, proxy points are selected via the pivoted QR decomposition of a kernel matrix defined by points densely selected in  $\mathcal{X}$  and  $\mathcal{Y}$  based on [13] (cf. Figure 3 in [14]).

**4. Error analysis.** In this section, we present a rigorous error analysis for the proxy point method, which proves the effectiveness of the method and also provides theoretical guidance for selecting the proxy points as discussed in section 5. Recall that we always assume that  $\mathcal{X}$  and  $\mathcal{Y}$  are compact and  $K(x, y)$  is smooth in  $\mathcal{X} \times \mathcal{Y}$ .

Using the resulting  $U$  and  $X_{\text{id}}$  from Algorithm 3.1, denote the error of the function approximation (3.5) to each  $K(x_i, y)$  as

$$(4.1) \quad e_i(y) = K(x_i, y) - u_i^T K(X_{\text{id}}, y), \quad x_i \in X_0, \quad y \in \mathcal{Y}.$$

With this notation, the  $i$ th row of the error matrix  $K(X_0, Y_0) - UK(X_{\text{id}}, Y_0)$  for the ID approximation (3.1) is exactly  $e_i(Y_0)$ . Similarly, the  $i$ th row of the error matrix  $K(X_0, Y_p) - UK(X_{\text{id}}, Y_p)$  for the ID approximation (3.2) is  $e_i(Y_p)$ .

For an arbitrary set of points  $Y_0$  in  $\mathcal{Y}$ , the best upper bound for  $\|e_i(Y_0)\|_2$  is

$$(4.2) \quad \|e_i(Y_0)\|_2 \leq \sqrt{|Y_0|} \max_{y \in \mathcal{Y}} |e_i(y)| = \sqrt{|Y_0|} \|e_i(y)\|_\infty,$$

where equality holds when  $|e_i(y)|$  reaches the same maximum in  $\mathcal{Y}$  for all points in  $Y_0$ . On the other hand,  $\|e_i(Y_p)\|_2$  is bounded by the error threshold specified for the ID approximation (3.2) of  $K(X_0, Y_p)$ . Thus, the following error analysis for the proxy point method seeks an upper bound for  $\|e_i(y)\|_\infty$  in terms of  $\|e_i(Y_p)\|_2$ , i.e., an inequality in the form of (3.6).

For analysis purposes, consider a generic  $r$ -term  $\varepsilon$ -expansion of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ ,

$$(4.3) \quad K(x, y) = \sum_{j=1}^r \psi_j(x) \phi_j(y) + R_r(x, y), \quad x \in \mathcal{X}, \quad y \in \mathcal{Y},$$

where the remainder  $|R_r(x, y)|$  is bounded by  $\varepsilon$ . Assume that functions in  $\{\phi_j(y)\}_{j=1}^r$  are linearly independent. Such an expansion (4.3) shows that, for any  $x \in \mathcal{X}$ , the function  $K(x, y)$  in the single variable  $y \in \mathcal{Y}$  is close to the  $r$ -dimensional function space spanned by  $\{\phi_j(y)\}_{j=1}^r$  with distance less than  $\varepsilon$ , i.e.,

$$(4.4) \quad \text{dist}(K(x, y), \text{span}(\{\phi_j(y)\}_{j=1}^r)) = \inf_{f \in \text{span}(\{\phi_j(y)\}_{j=1}^r)} \|K(x, y) - f(y)\|_\infty \leq \varepsilon.$$

Being a linear combination of  $K(x_i, y)$  and  $\{K(x_j, y)\}_{x_j \in X_{\text{id}}}$ , each error function  $e_i(y)$  is also close to  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  with small distance,

$$(4.5) \quad \begin{aligned} \text{dist}(e_i(y), \text{span}(\{\phi_j(y)\}_{j=1}^r)) &\leq (1 + \|u_i\|_1) \max_{x \in X_0} \text{dist}(K(x, y), \text{span}(\{\phi_j(y)\}_{j=1}^r)) \\ &\leq (1 + C_{\text{qr}}|Y_p|)\varepsilon. \end{aligned}$$

The first inequality above is based on the triangular inequality. The second inequality is based on the facts that entries of  $U$  computed in (3.2) are bounded by the prespecified parameter  $C_{\text{qr}}$  and that  $|X_{\text{id}}|$  computed in (3.2) is no greater than  $|Y_p|$ .

To estimate  $\|e_i(y)\|_\infty$  in terms of  $\|e_i(Y_p)\|_2$ , the idea is to first approximate  $e_i(y)$  in the function space  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  based on its values at points in  $Y_p$  and then estimate  $\|e_i(y)\|_\infty$  in this finite-dimensional space. To begin with, consider finding an approximation of  $e_i(y)$  in  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  as

$$(4.6) \quad e_i(y) \approx c_1 \phi_1(y) + \cdots + c_r \phi_r(y) = c^T \Phi(y),$$

where  $c = (c_1, c_2, \dots, c_r)^T$  and  $\Phi(y) = (\phi_1(y), \phi_2(y), \dots, \phi_r(y))^T$ . The coefficient vector  $c$  is selected to minimize the function approximation error at  $Y_p$ , i.e.,

$$c = \arg_{v \in \mathbb{R}^r} \min \|e_i(Y_p) - v^T \Phi(Y_p)\|_2,$$

where  $\Phi(Y_p) \in \mathbb{R}^{r \times |Y_p|}$  denotes the matrix of column vectors  $\Phi(y_j)$  for all  $y_j$  in  $Y_p$ . For the uniqueness of  $c$  and thus the uniqueness of the approximant  $c^T \Phi(y)$ , a necessary condition that  $Y_p$  needs to satisfy is

$$(4.7) \quad \text{rank}(\Phi(Y_p)) = r,$$

meaning that row vectors of  $\Phi(Y_p)$  are linearly independent. Under this condition,  $c$  is solved as  $c^T = e_i(Y_p)\Phi(Y_p)^\dagger$  and  $e_i(y)$  is approximated as

$$e_i(y) \approx e_i(Y_p)\Phi(Y_p)^\dagger\Phi(y).$$

Let  $S_{Y_p}(y) = (s_1(y), s_2(y), \dots, s_{|Y_p|}(y))^T$  denote the vector  $\Phi(Y_p)^\dagger\Phi(y)$  of dimension  $|Y_p|$ . The above approximation process of  $e_i(y)$  can be generalized as a linear operator  $\mathcal{L} : C^\infty(\mathcal{Y}) \rightarrow \text{span}(\{\phi_j(y)\}_{j=1}^r)$  in the function space  $C^\infty(\mathcal{Y})$  that contains all the smooth functions in  $\mathcal{Y}$ ,

$$(4.8) \quad \mathcal{L}f(y) = \sum_{y_j \in Y_p} f(y_j)s_j(y) = f(Y_p)S_{Y_p}(y), \quad f \in C^\infty(\mathcal{Y}).$$

This operator defines a generalized interpolation process such that  $\mathcal{L}f(y)$  is the unique function in  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  whose values at points in  $Y_p$  are equal to the projection of  $f(Y_p)$  onto the vector space  $\text{span}(\{\phi_j(Y_p)\}_{j=1}^r)$ , i.e.,  $\mathcal{L}f(Y_p) = f(Y_p)\Phi(Y_p)^\dagger\Phi(Y_p)$ . Also, it holds that  $\mathcal{L}f(y) = f(y)$  for any  $f(y)$  in  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  based on (4.7).

We then estimate  $\|e_i(y)\|_\infty$  based on its approximation  $\|\mathcal{L}e_i(y)\|_\infty$ ,

$$\begin{aligned} \|e_i(y)\|_\infty &\leq \|e_i(y) - \mathcal{L}e_i(y)\|_\infty + \|\mathcal{L}e_i(y)\|_\infty \\ &\leq \min_{f \in \text{span}(\{\phi_j(y)\}_{j=1}^r)} (\|e_i(y) - f(y)\|_\infty + \|f(y) - \mathcal{L}e_i(y)\|_\infty) + \|\mathcal{L}e_i(y)\|_\infty \\ &= \min_{f \in \text{span}(\{\phi_j(y)\}_{j=1}^r)} (\|e_i(y) - f(y)\|_\infty + \|\mathcal{L}f(y) - \mathcal{L}e_i(y)\|_\infty) + \|\mathcal{L}e_i(y)\|_\infty \\ &\leq \min_{f \in \text{span}(\{\phi_j(y)\}_{j=1}^r)} (\|e_i(y) - f(y)\|_\infty + \|\mathcal{L}\|_\infty\|f(y) - e_i(y)\|_\infty) + \|\mathcal{L}e_i(y)\|_\infty \\ (4.9) \quad &= (1 + \|\mathcal{L}\|_\infty)\text{dist}(e_i(y), \text{span}(\{\phi_j(y)\}_{j=1}^r)) + \|\mathcal{L}e_i(y)\|_\infty, \end{aligned}$$

where the operator norm  $\|\mathcal{L}\|_\infty$  is defined as

$$(4.10) \quad \|\mathcal{L}\|_\infty = \max_{f \in C^\infty(\mathcal{Y}), f \neq 0} \frac{\|\mathcal{L}f\|_\infty}{\|f\|_\infty} = \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1.$$

The second representation of  $\|\mathcal{L}\|_\infty$  above is based on (4.8) and Hölder's inequality

$$|\mathcal{L}f(y)| \leq \|f(Y_p)\|_\infty \|S_{Y_p}(y)\|_1 \leq \|f\|_\infty \|S_{Y_p}(y)\|_1, \quad y \in \mathcal{Y},$$

where equality can hold true for some  $f \in C^\infty(\mathcal{Y})$  and thus the second equality in (4.10) holds true. Also,  $\mathcal{L}e_i(y) = e_i(Y_p)S_{Y_p}(y)$  can be bounded as

$$(4.11) \quad \|\mathcal{L}e_i(y)\|_\infty \leq \|e_i(Y_p)\|_2 \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2.$$

Lastly, substituting (4.5), (4.10), and (4.11) into (4.9),  $\|e_i(y)\|_\infty$  is bounded as

$$(4.12) \quad \|e_i(y)\|_\infty \leq (1 + \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1)(1 + C_{qr}|Y_p|)\varepsilon + \|e_i(Y_p)\|_2 \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2.$$

We note that the above estimation of  $\|e_i(y)\|_\infty$  in terms of  $\|e_i(Y_p)\|_2$  works for any  $r$ -term  $\varepsilon$ -expansion (4.3) of  $K(x, y)$  and for any  $Y_p$  that satisfies the condition (4.7). Noting that the upper bound (4.12) does not rely on  $\{\psi_j(x)\}_{j=1}^r$  in the expansion (4.3), (4.12) can be further sharpened by fixing  $\{\phi_j(x)\}_{j=1}^r$  and varying  $\{\psi_j(x)\}_{j=1}^r$

to reduce  $\varepsilon$  that appears in the upper bound. Specifically, the minimal accuracy  $\varepsilon_*$  of an expansion (4.3) using a fixed set of functions  $\{\phi_j(y)\}_{j=1}^r$  can be defined as

$$(4.13) \quad \varepsilon_* = \sup_{x \in \mathcal{X}} \inf_{f \in \text{span}(\{\phi_j(y)\}_{j=1}^r)} \|K(x, y) - f(y)\|_\infty,$$

which describes the maximum distance between all the functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  and the function space  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$ .

Combining the upper bound (4.12), the minimal accuracy  $\varepsilon_*$  in (4.13), and the inequality  $\|e_i(Y_0)\|_2 / \sqrt{|Y_0|} \leq \|e_i(y)\|_\infty$  from (4.2), the error bound for the proxy point method can be summarized as follows.

**THEOREM 4.1** (error bound for the proxy point method). *Consider two compact domains  $\mathcal{X}$  and  $\mathcal{Y}$  and a kernel function  $K(x, y)$  being smooth in  $\mathcal{X} \times \mathcal{Y}$ . Given a set of linearly independent functions  $\{\phi_j(y)\}_{j=1}^r$  in  $\mathcal{Y}$ , the minimal accuracy  $\varepsilon_*$  of all the possible degenerate approximations of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$  using  $\{\phi_j(y)\}_{j=1}^r$  is defined in (4.13). If the set of proxy points  $Y_p$  satisfies the condition  $\text{rank}(\Phi(Y_p)) = r$ , the ID approximation of  $K(X_0, Y_0)$  calculated by the proxy point method with  $Y_p$  has error in the  $i$ th row  $e_i(Y_0)$  bounded as*

$$(4.14) \quad \frac{\|e_i(Y_0)\|_2}{\sqrt{|Y_0|}} \leq \left(1 + \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1\right) (1 + C_{qr}|Y_p|) \varepsilon_* + \|e_i(Y_p)\|_2 \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2.$$

We note that Theorem 4.1 works for any set of linearly independent functions  $\{\phi_j(y)\}_{j=1}^r$  which will be referred to as a set of *basis functions*. The assumption that  $\mathcal{X} \times \mathcal{Y}$  is admissible for  $K(x, y)$ , i.e., there exists an accurate, low-degree degenerate approximation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ , guarantees that there exists a small set of basis functions  $\{\phi_j(y)\}_{j=1}^r$  such that the minimal accuracy  $\varepsilon_*$  is negligible compared to  $\|e_i(Y_p)\|_2$ . With such  $\{\phi_j(y)\}_{j=1}^r$ , the first term of the upper bound (4.14) is negligible if  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1$  is of scale  $O(1)$ . Theorem 4.1 then shows that

$$\|e_i(Y_0)\|_2 / \sqrt{|Y_0|} \leq O(1) \|e_i(Y_p)\|_2 / \sqrt{|Y_p|}$$

if  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  and  $|Y_p|$  are also of scale  $O(1)$ , which proves the effectiveness of the proxy point method. The remaining problem becomes how to select a small set of proxy points  $Y_p$  so that  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  and  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1$  are of scale  $O(1)$ .

## 5. Proxy point selection.

**5.1. Guidelines for selecting proxy points.** Recall that the ultimate goal of a proper selection of the proxy points  $Y_p$  is to guarantee that  $\|e_i(y)\|_\infty$  is bounded by a small multiple of  $\|e_i(Y_p)\|_2 / \sqrt{|Y_p|}$ . Assume that  $\|e_i(Y_p)\|_2 \leq \varepsilon_{\text{id}}$  from the error threshold  $\varepsilon_{\text{id}}$  specified for the ID approximation of  $K(X_0, Y_p)$ . Based on Theorem 4.1, the guidelines for selecting proxy points are established as follows.

To simplify our discussion, the upper bound (4.14) is slightly loosened as

$$(5.1) \quad \|e_i(y)\|_\infty \lesssim \left( \varepsilon_* |Y_p|^{\frac{3}{2}} + \|e_i(Y_p)\|_2 \right) \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2,$$

using  $\|S_{Y_p}(y)\|_1 \leq \sqrt{|Y_p|} \|S_{Y_p}(y)\|_2$ . This upper bound relies on two important components: basis functions  $\{\phi_j(y)\}_{j=1}^r$  and proxy points  $Y_p \subset \mathcal{Y}$ . Fixing a set of basis functions  $\{\phi_j(y)\}_{j=1}^r$ , (5.1) gives the following guidelines for selecting proxy points:

1.  $Y_p$  should satisfy  $\text{rank}(\Phi(Y_p)) = r$  as required by Theorem 4.1.
2.  $Y_p$  should have  $O(1)r$  points for the efficiency of the proxy point method.
3.  $Y_p$  should make  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  of scale  $O(1)$ .

To make  $\|e_i(y)\|_\infty$  bounded by  $O(1)\varepsilon_{\text{id}}$  via (5.1), the basis functions  $\{\phi_j(y)\}_{j=1}^r$  used in the above guidelines need to satisfy

$$(5.2) \quad \varepsilon_* |Y_p|^{\frac{3}{2}} \sim \varepsilon_* r^{\frac{3}{2}} \leq O(1)\varepsilon_{\text{id}}.$$

The number of basis functions  $r$  should also be small so that only a small number of proxy points is selected following the guidelines. In other words, we need to use a small set of basis functions  $\{\phi_j(y)\}_{j=1}^r$  (small  $r$ ) whose span is close to all the functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  (small  $\varepsilon_*$ ). As to be discussed next, such a set of basis functions can be selected heuristically based on analytic properties of the kernel function. The basis functions can also be selected in a numerical way, which is the approach used by the proposed proxy point selection scheme in this paper.

**5.2. Selection of basis functions.** General expansion techniques such as Taylor expansion, interpolation, and Fourier series are commonly used to construct degenerate approximations of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ , which correspond to approximating functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  using specific basis functions, e.g., polynomials and trigonometric polynomials. If we know that  $K(x, y)$  in the single variable  $y \in \mathcal{Y}$  can be well approximated using a small number of polynomials or trigonometric polynomials, then these basis functions can be used in the guidelines for selecting proxy points. The number of such basis functions  $r$  can be selected by trial-and-error or by analytic estimation of the minimal accuracy  $\varepsilon_*$  in terms of  $r$ .

Another idea that does not require a priori knowledge of a degenerate approximation of  $K(x, y)$  is to use a finite subset of  $\{K(x, y)\}_{x \in \mathcal{X}}$  as the basis functions. Denote such a subset as  $\{K(x_j, y)\}_{x_j \in X_p}$  associated with a set of points  $X_p$  in  $\mathcal{X}$ . Based on the assumption that  $\mathcal{X}$  and  $\mathcal{Y}$  are compact and  $K(x, y)$  is smooth in  $\mathcal{X} \times \mathcal{Y}$ , it can be proved that there always exists such a finite subset that can approximate all the functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  with a given accuracy. The basic idea is that we can select a finite set of points  $X_p \subset \mathcal{X}$  such that, for any  $x \in \mathcal{X}$ , there exists some  $x_* \in X_p$  whose distance to  $x$  is below an arbitrarily small threshold. Then,  $K(x_*, y)$  can approximate  $K(x, y)$  well in  $\mathcal{Y}$  and thus so can a linear combination of  $\{K(x_j, y)\}_{x_j \in X_p}$ . We skip the detailed proof here. In practice, the basis functions  $\{K(x_j, y)\}_{x_j \in X_p}$  with the minimal accuracy  $\varepsilon_*$  approximately below a given threshold  $\varepsilon_p$  can be numerically selected as follows.

First sample domain  $\mathcal{Y}$  to obtain a set of uniformly distributed points  $Y_1$  with high point density. Similarly, sample domain  $\mathcal{X}$  to obtain a set of points  $X_1$  which is larger than the expected size of  $X_p$  (through trial-and-error as explained below). Compute an ID approximation of  $K(X_1, Y_1)$  using SRRQR and define  $X_p$  as the subset of  $X_1$  that corresponds to the row subset of this ID approximation, i.e.,

$$(5.3) \quad K(X_1, Y_1) \approx U_1 K(X_p, Y_1).$$

The error threshold for this ID approximation is set to  $\varepsilon_p \sqrt{|Y_1|}$ . This selection of  $X_p$  satisfies the condition that  $K(x, Y_1)$  for any  $x \in X_1$  can be approximated by a linear combination of  $\{K(x_j, Y_1)\}_{x_j \in X_p}$  with root-mean-square error bounded by  $\varepsilon_p$ . Since  $K(x, y)$  is smooth in  $\mathcal{X} \times \mathcal{Y}$  and  $X_1$  and  $Y_1$  have high point densities in  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively, we can expect that functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  can be approximated by linear combinations of  $\{K(x_j, y)\}_{x_j \in X_p}$  with error  $O(\varepsilon_p)$ . The above selection process can start with a relatively small set of points  $X_1$ . If the ID approximation (5.3) has full rank, i.e.,  $X_p = X_1$ , we can double the size of  $X_1$  and repeat the selection process.

To make  $\varepsilon_*$  satisfy the condition (5.2), we can conservatively set  $\varepsilon_p$  a few orders of magnitude smaller than  $\varepsilon_{\text{id}}$  while taking arithmetic rounding errors into account. This selection of basis functions is summarized by Step 1 in Algorithm 5.1.

**5.3. Selection of proxy points.** Assume that we have a set of basis functions  $\{\phi_j(y)\}_{j=1}^r$  selected according to the above discussion. Following the above guidelines for selecting proxy points, we seek exactly  $r$  proxy points such that  $\text{rank}(\Phi(Y_p)) = r$  and  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  is of scale  $O(1)$ . If  $|Y_p| = r$  and  $\text{rank}(\Phi(Y_p)) = r$ , it holds that  $S_{Y_p}(y) = \Phi(Y_p)^{-1}\Phi(y)$  where the pseudoinverse becomes the exact inverse. Then the operator  $\mathcal{L}$  defined in (4.8) is exactly the interpolation operator in  $\text{span}(\{\phi_j(y)\}_{j=1}^r)$  where  $Y_p$  is the set of interpolation nodes and entries of  $S_{Y_p}(y)$  are the corresponding Lagrangian functions.

In numerical approximation theory, the norm of  $\mathcal{L}$ ,  $\|\mathcal{L}\|_\infty = \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1$ , is called the *Lebesgue constant* and is used to characterize the quality of interpolation nodes  $Y_p$ . For polynomials and trigonometric polynomials in a regular box domain, selections of interpolation nodes leading to small Lebesgue constant have been well studied [30]. Since  $\|S_{Y_p}(y)\|_2 \leq \|S_{Y_p}(y)\|_1$ , these interpolation nodes can be directly used as proxy points. For example, if  $\{\phi_j(y)\}_{j=1}^r$  are polynomials in a 2D box  $\mathcal{Y}$  of degree up to  $k - 1$  for each variable ( $r = k^2$ ), we can select  $Y_p$  as a  $k \times k$  tensor grid of Chebyshev nodes in  $\mathcal{Y}$  as illustrated in Figure 3.1a, which has  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  that is well bounded, i.e.,

$$\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2 \leq \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_1 \approx \left( \frac{2}{\pi} \log(k+1) \right)^2.$$

Furthermore, we can also select  $Y_p$  as a  $k \times k$  tensor grid of Gauss–Lobatto nodes which gives an even smaller scaling factor,  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2 = 1$  (see [7]).

In this paper, we only consider the general case with basis functions of the form  $\{K(x_j, y)\}_{x_j \in X_p}$  and with irregular domain  $\mathcal{Y}$ , where analytic choices of interpolation nodes with small Lebesgue constant are not available. In this case, we introduce a numerical method to select  $r$  proxy points  $Y_p$  in  $\mathcal{Y}$  such that  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  is of scale  $O(1)$ .

First, we sample domain  $\mathcal{Y}$  to obtain a set of uniformly distributed points  $Y_2$  with high point density. Then, we find an  $r$ -point subset  $Y_p$  of  $Y_2$  that bounds  $\|S_{Y_p}(y)\|_2 = \|\Phi(Y_p)^{-1}\Phi(y)\|_2$  at any point  $y \in Y_2$ . To do this, compute a numerically exact SRRQR decomposition of  $\Phi(Y_2) \in \mathbb{R}^{r \times |Y_2|}$  as

$$(5.4) \quad \Phi(Y_2)P = Q(R_1 \ R_2),$$

where  $P$  is a permutation matrix,  $Q$  is an orthogonal matrix, and  $R_1$  is an  $r \times r$  upper-triangular matrix. Let  $Y_p$  be the subset of points in  $Y_2$  that corresponds to the columns of  $R_1$  in the decomposition, i.e.,  $\Phi(Y_p) = QR_1$ . It is the key feature of SRRQR that, with (5.4), the matrix

$$\Phi(Y_p)^{-1}\Phi(Y_2) = (QR_1)^{-1}Q(R_1 \ R_2)P^T = (I_r \ R_1^{-1}R_2)P^T$$

has all its entries bounded by the prespecified parameter  $C_{\text{qr}} \geq 1$ . Thus,  $\|S_{Y_p}(y)\| \leq C_{\text{qr}}\sqrt{r}$  for any  $y \in Y_2$ . Since  $Y_2$  is uniformly distributed and has high point density in  $\mathcal{Y}$ , this selection of  $Y_p$  satisfies the condition that  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2 \lesssim C_{\text{qr}}\sqrt{r}$ .

This selection of proxy points is summarized by Step 2 in Algorithm 5.1. Also, we note that this selection approach is closely related to the pseudoskeleton approximation method [13] and the numerical construction of a special set of interpolation nodes called *Fekete points* [6].

**5.4. Summary of the selection scheme.** Algorithm 5.1 summarizes the overall systematic numerical scheme above to select the basis functions and the proxy points. A heuristic “densification” step is included in Algorithm 5.1 in order to improve the quality of the set of selected proxy points. The motivation for this additional step is explained below.

---

**Algorithm 5.1** Proxy point selection scheme.

---

**Input:**  $K(x, y)$ ,  $\mathcal{X}$ ,  $\mathcal{Y}$ ,  $\varepsilon_p$ .

**Output:** Proxy points  $Y_p$ .

**Step 1:** Selection of basis functions  $\{\phi_j(y)\}_{j=1}^r$ .

- Sample domains  $\mathcal{X}$  and  $\mathcal{Y}$  to obtain two sets of uniformly distributed points  $X_1$  and  $Y_1$  with high point density, respectively.
- Compute an ID approximation (5.3) of  $K(X_1, Y_1)$  using SRRQR with error threshold  $\varepsilon_p \sqrt{|Y_1|}$ .
- Set  $\{\phi_j(y)\}_{j=1}^r$  as  $\{K(x_j, y)\}_{x_j \in X_p}$  with  $X_p$  defined by (5.3).

**Step 2:** Selection of proxy points  $Y_p$ .

- Sample domain  $\mathcal{Y}$  to obtain a set of uniformly distributed points  $Y_2$  with high point density.
- Compute an exact SRRQR decomposition (5.4) of  $\Phi(Y_2)$ .
- Set  $Y_p$  as the subset of points in  $Y_2$  that corresponds to the columns of  $R_1$  in (5.4).

**Step 3:** Heuristic densification of  $Y_p$ . For each point  $y_j \in Y_p$ , calculate the distance  $d_j$  between  $y_j$  and its nearest neighbor in  $Y_p$ . Randomly select one (or more) extra point in the ball centered at  $y_j$  with radius  $d_j/3$  and add it to  $Y_p$ .

---

In Algorithm 5.1, Step 1 obtains a set of basis functions  $\{K(x_j, y)\}_{x_j \in X_p}$  that can approximate functions in  $\{K(x, y)\}_{x \in \mathcal{X}}$  with error  $O(\varepsilon_p)$ . Step 2 then selects  $r = |X_p|$  number of proxy points  $Y_p$  that satisfies  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2 \lesssim C_{qr} \sqrt{r}$ . By (5.1), the proxy points  $Y_p$  obtained by Step 1 and Step 2 satisfy the error bound

$$(5.5) \quad \|e_i(y)\|_\infty \lesssim \left( \varepsilon_p r^{\frac{3}{2}} + \|e_i(Y_p)\|_2 \right) C_{qr} \sqrt{r},$$

which justifies the effectiveness of the selected proxy points if  $\varepsilon_p r^{\frac{3}{2}}$  is negligible or of similar scale as the error threshold  $\varepsilon_{id}$  specified for  $\|e_i(Y_p)\|_2$ .

However, note that  $\varepsilon_p$  is the root-mean-square error threshold for the algebraic ID approximation (5.3). Due to arithmetic rounding errors,  $\varepsilon_p$  cannot be of smaller scale than machine precision  $\varepsilon_{\text{machine}}$ . Thus, the error bound (5.5) suggests that the approximation error of the proxy point method with  $Y_p$  selected by Step 1 and Step 2 in Algorithm 5.1 may stagnate around  $\varepsilon_{\text{machine}} r^{\frac{3}{2}}$  if the specified error threshold  $\varepsilon_{id}$  is close to machine precision. Such error stagnation is indeed observed in some of our preliminary tests, where the smallest relative approximation error of the proxy point method can only reach around  $10^{-13} \sim 10^{-10}$  with double precision ( $10^{-16}$ ) arithmetic. To tackle this possible error stagnation for extremely small error threshold  $\varepsilon_{id}$ , a heuristic densification of  $Y_p$  by Step 3 in Algorithm 5.1 is added where extra proxy points are added. This densification step turns out to be experimentally effective as illustrated by the numerical results in the next section.

In terms of computational cost, Algorithm 5.1 is expensive due to the large number of sample points in  $X_1$ ,  $Y_1$ , and  $Y_2$ . Reusing the proxy points selected by Algorithm 5.1

is critical for practical applications of Algorithm 5.1. Luckily, in most rank-structured matrix applications, the kernel function is translationally invariant, i.e.,

$$K(x, y) = k(x - y) \quad \text{for some univariate function } k(\cdot).$$

As a result, the proxy points selected by Algorithm 5.1 can be reused by simple translations for different admissible pairs of domains in the construction of rank-structured matrices (see the numerical test in subsection 6.5).

Lastly, in practice and for the numerical experiments in the next section, the sample points  $X_1$ ,  $Y_1$ , and  $Y_2$  in Algorithm 5.1 are randomly and uniformly sampled from  $\mathcal{X}$  and  $\mathcal{Y}$  with their numbers of points heuristically decided for efficiency and simplicity. However, to rigorously justify the arguments about  $X_1$ ,  $Y_1$ , and  $Y_2$  in subsection 5.2 and subsection 5.3, the point densities of these sets should depend on the magnitude of the variation of  $K(x, y)$  in  $\mathcal{X} \times \mathcal{Y}$ , about which we skip more detailed discussions. Possible approaches for reducing the sizes of these sample point sets are the “weakly admissible meshes” used in [6] and the tensor grid of Chebyshev nodes used in [9]. Another heuristic but usually effective approach, similar to the idea used in [13], is to initially apply Algorithm 5.1 with small  $X_1$ ,  $Y_1$ , and  $Y_2$  and recursively enlarge the three sets of points if needed.

**6. Numerical experiments.** In this section, we provide numerical tests to illustrate the effectiveness of the proxy point selection scheme (Algorithm 5.1) and the performance of the proxy point method (Algorithm 3.1) for general kernel functions and corresponding admissible domain pairs. The parameter  $C_{qr}$  for SRRQR used in the proxy point method and in the proxy point selection scheme is set to 2. In all the tests of Algorithm 5.1,  $X_1$  contains 1500 points sampled in  $\mathcal{X}$ ,  $Y_1$  contains 10000 points sampled in  $\mathcal{Y}$ ,  $Y_2$  is the same set of points as  $Y_1$ , and the parameter  $\varepsilon_p$  is set to  $10^{-14}$ . All these sets of sample points,  $X_1$  and  $Y_1$ , are randomly and uniformly sampled in their corresponding domains.

**6.1. Basic tests.** Consider the following four different problem settings:

1.  $K(x, y) = 1/\sqrt{1 + |x - y|^2}$ ,  $\mathcal{X} = [-1, 1]^2$ ,  $\mathcal{Y} = [3, 5] \times [-1, 1]$ .
2.  $K(x, y) = e^{-|x-y|^2}$ ,  $\mathcal{X} = [-1, 1]^2$ ,  $\mathcal{Y} = [-7, 7]^2 \setminus [-3, 3]^2$ .
3.  $K(x, y) = e^{2\pi i|x-y|}$ ,  $\mathcal{X} = B_2(0, 2)$ ,  $\mathcal{Y} = \{y \in \mathbb{R}^2 : \theta(y, l) \leq 1/2, 4 \leq |y| \leq 16\}$  with  $l = (1, 0)$ .  
(Note:  $B_d(0, a)$  denotes the  $d$ -dimensional ball centered at the origin with radius  $a$ , and  $\theta(y, l)$  denotes the angle between  $y$  and  $l$ .)
4.  $K(x, y) = e^{2\pi i|x-y|}/|x - y|$ ,  $\mathcal{X} = B_3(0, 2)$ ,  $\mathcal{Y} = \{y \in \mathbb{R}^3 : \theta(y, l) \leq 1/2, 4 \leq |y| \leq 8\}$  with  $l = (1, 0, 0)$ .

For each problem setting above, the domains  $\mathcal{X}$  and  $\mathcal{Y}$  and the corresponding proxy points  $Y_p$  selected by Algorithm 5.1 are plotted in Figures 6.1 to 6.4.

Two sets of points  $X_0$  and  $Y_0$  are randomly and uniformly selected in  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively, with an average 100 points per unit of area in 2D or 50 points per unit of volume in 3D. For different approximation ranks, the relative approximation error of the proxy point method with the selected  $Y_p$  for  $K(X_0, Y_0)$  is also plotted in the figure for each of the problem settings. The relative error of the intermediate ID approximation of  $K(X_0, Y_p)$  in the proxy point method is plotted as well.

As can be observed from the results, the proxy point method with proxy points selected by Algorithm 5.1 has approximation error close to those of SVD and the ID approximation using SRRQR. Also, the intermediate ID approximation of  $K(X_0, Y_p)$  has similar relative error as the ID approximation of  $K(X_0, Y_0)$ . This shows that the accuracy of the final ID approximation in the proxy point method can be controlled by controlling the accuracy of the ID approximation of  $K(X_0, Y_p)$ .

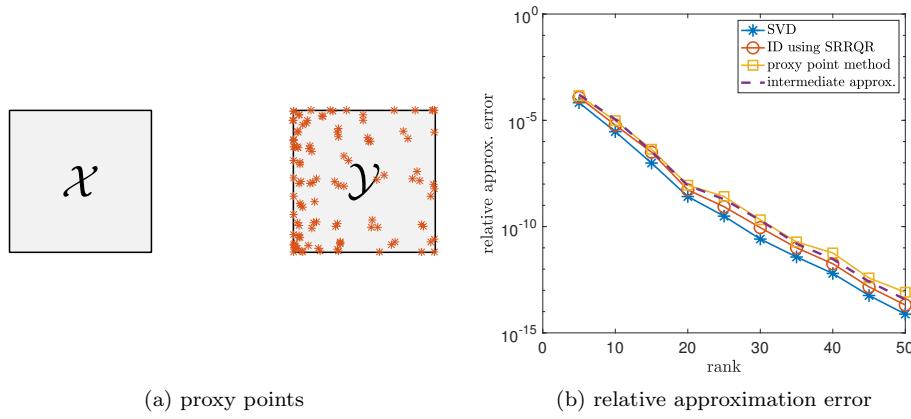


FIG. 6.1. Basic test for problem setting 1:  $K(x, y) = 1/\sqrt{1 + |x - y|^2}$ ,  $|X_0| = 400$ ,  $|Y_0| = 400$ , and  $|Y_p| = 118$ . In this and the following three figures, “intermediate approx.” refers to the relative error of the ID approximation of  $K(X_0, Y_p)$  in the proxy point method.

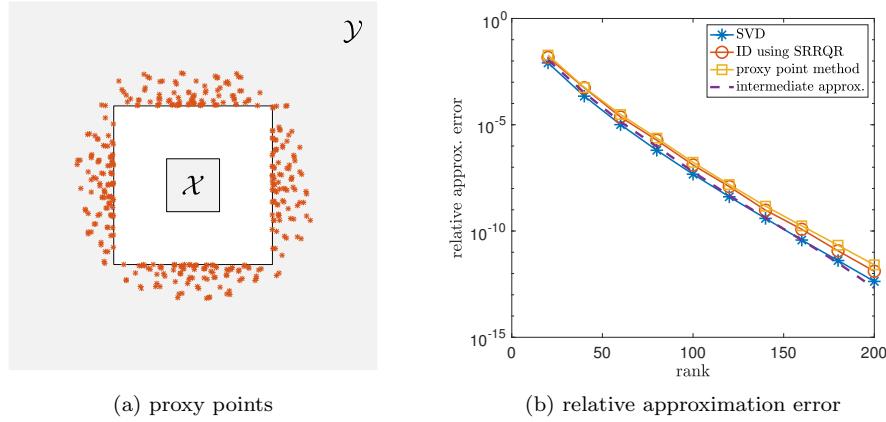


FIG. 6.2. Basic test for problem setting 2:  $K(x, y) = e^{-|x-y|^2}$ ,  $|X_0| = 400$ ,  $|Y_0| = 16000$ , and  $|Y_p| = 384$ .

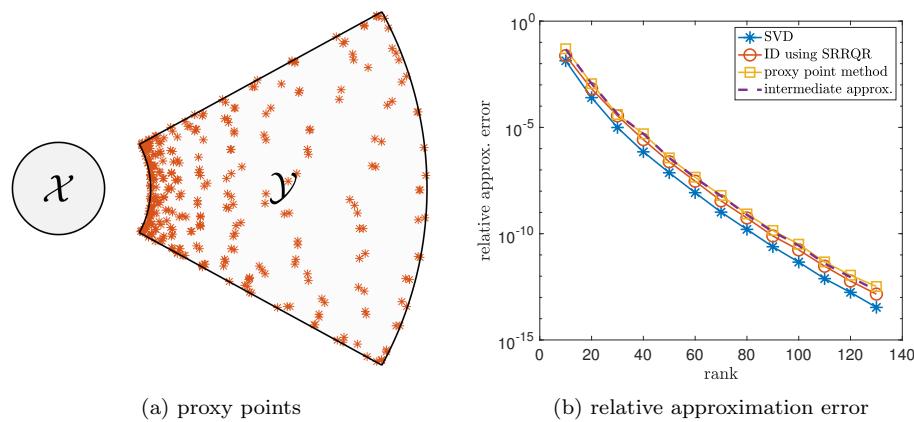


FIG. 6.3. Basic test for problem setting 3:  $K(x, y) = e^{2\pi i|x-y|}$ ,  $|X_0| = 1300$ ,  $|Y_0| = 6000$ , and  $|Y_p| = 330$ .

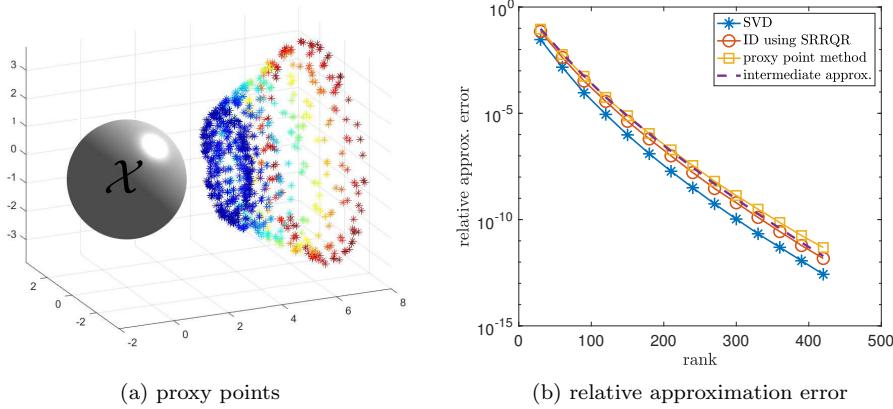


FIG. 6.4. Basic test for problem setting 4:  $K(x, y) = e^{2\pi i|x-y|}/|x-y|$ ,  $|X_0| = 1700$ ,  $|Y_0| = 22500$ , and  $|Y_p| = 870$ . Domain  $\mathcal{Y}$  is not plotted. The selected proxy points are colored in (a) based on their  $x$ -axis coordinates for better visualization. Most of the proxy points are near the boundary of  $\mathcal{Y}$  excluding the outer hemispherical surface, i.e.,  $|y| = 8$ .

**6.2. Error bound for  $\|e_i(y)\|_\infty$ .** The proxy point selection scheme in Algorithm 5.1 and the effectiveness of the proxy point method are both based on Theorem 4.1 and the derived error bound (5.1) for  $\|e_i(y)\|_\infty$ . In this subsection, we numerically study the tightness of this error bound.

Consider  $K(x, y) = e^{-|x-y|^2}$ ,  $\mathcal{X} = [-1, 1]^2$ , and  $\mathcal{Y} = [-7, 7]^2 \setminus [-3, 3]^2$ . We randomly and uniformly select 400 points in  $\mathcal{X}$  for  $X_0$ . We fix the error threshold  $\varepsilon_{\text{id}} = 10^{-6}$  for the ID approximation of  $K(X_0, Y_p)$  with any  $Y_p$ . In order to test the proxy point method with a given number of proxy points, we use a simple modification of Algorithm 5.1. Specifically, given an integer  $r$ , we select a set of  $r$  basis functions  $\{K(x_j, y)\}_{x_j \in X_p}$  with  $X_p$  computed by a rank- $r$  ID approximation of  $K(X_1, Y_1)$  in Step 1 of Algorithm 5.1. Exactly  $r$  proxy points  $Y_p$  can be then selected by Step 2 of Algorithm 5.1 without densification.

Using this selection scheme, we vary  $r$  and select the corresponding  $r$  proxy points  $Y_p$ . The error function  $e_i(y)$  for each  $x_i \in X_0$  is then defined by the ID approximation of  $K(X_0, Y_p)$  with error threshold  $\varepsilon_{\text{id}}$ . For different  $Y_p$ , Figure 6.5 plots  $\max_{x_i \in X_0} \|e_i(y)\|_\infty$  and its upper bound derived from (5.1), i.e.,

$$(6.1) \quad \max_{x_i \in X_0} \|e_i(y)\|_\infty \lesssim (r^{\frac{3}{2}} \varepsilon_* + \varepsilon_{\text{id}}) \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2,$$

where  $\varepsilon_*$  and  $\max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2$  are numerically estimated for each set of proxy points  $Y_p$  based on the associated set of basis functions  $\{K(x_j, y)\}_{x_j \in X_p}$ . Although having a large gap, the upper bound (6.1) captures the changing trend of  $\max_{x_i \in X_0} \|e_i(y)\|_\infty$ . Further numerical tests show that the large gap between the upper bound (6.1) and the actual value  $\max_{x_i \in X_0} \|e_i(y)\|_\infty$  is mainly due to the loose estimate of  $\|e_i(y) - \mathcal{L}e_i(y)\|_\infty$  in (4.9), i.e.,

$$\|e_i(y) - \mathcal{L}e_i(y)\|_\infty \leq (1 + \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2)(1 + C_{\text{qr}}|Y_p|)\varepsilon_* \lesssim r^{\frac{3}{2}} \varepsilon_* \max_{y \in \mathcal{Y}} \|S_{Y_p}(y)\|_2,$$

which is used to derive the first term in (6.1).

**6.3. Comparison of different selections of proxy points.** As illustrated in the introduction, an improper selection of proxy points  $Y_p$  can lead to a much

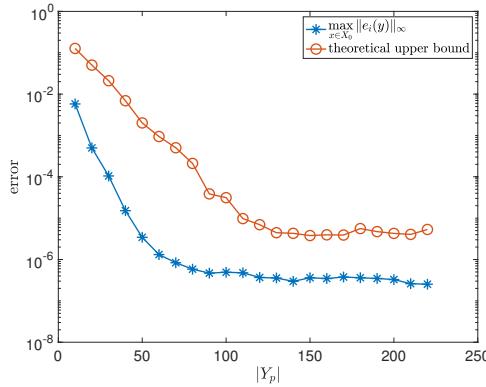


FIG. 6.5. Values of  $\max_{x_i \in X_0} \|e_i(y)\|_\infty$  and its upper bound (6.1) for different numbers of proxy points selected by a simple modification of Algorithm 5.1.

larger approximation error in the proxy point method when compared to SVD. In this subsection, we compare the proposed proxy point selection scheme in Algorithm 5.1 with existing heuristic selection schemes.

Consider again  $K(x, y) = e^{-|x-y|^2}$ ,  $\mathcal{X} = [-1, 1]^2$ , and  $\mathcal{Y} = [-7, 7]^2 \setminus [-3, 3]^2$ . Algorithm 5.1 obtains 384 proxy points with densification, as illustrated previously in Figure 6.2. We test three other heuristic choices of 384 proxy points: (1) uniform selection on the surface  $\partial[-3, 3]^2$ , (2) uniform and random selection in the annulus  $[-3.2, 3.2]^2 \setminus [-3, 3]^2$ , (3) uniform and random selection in another wider annulus  $[-3.5, 3.5]^2 \setminus [-3, 3]^2$ . The heuristic selection of proxy points in an annulus was suggested in [32] for Gaussian kernels.

We randomly and uniformly select 400 points in  $\mathcal{X}$  for  $X_0$  and 16000 points in  $\mathcal{Y}$  for  $Y_0$ . The relative approximation errors of the proxy point method with the above four sets of proxy points for  $K(X_0, Y_0)$  are plotted in Figure 6.6a. We apply the same test to another Gaussian kernel  $K(x, y) = e^{-0.1|x-y|^2}$  with the same domains  $\mathcal{X}$  and  $\mathcal{Y}$ , where 194 proxy points are selected by Algorithm 5.1. The corresponding relative approximation errors are plotted in Figure 6.6b.

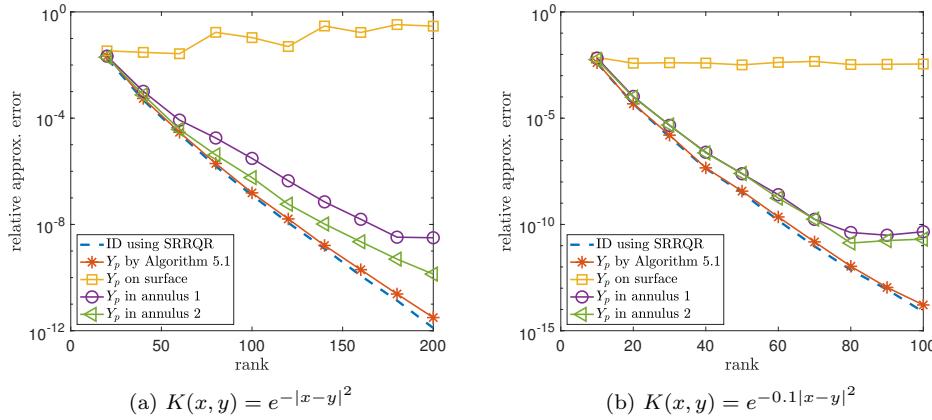


FIG. 6.6. Relative approximation error of the proxy point method with different selections of proxy points for two kernel functions. In the legend, “annulus 1” refers to  $[-3.2, 3.2]^2 \setminus [-3, 3]^2$  and “annulus 2” refers to  $[-3.5, 3.5]^2 \setminus [-3, 3]^2$ .

As can be observed, Algorithm 5.1 outperforms the three heuristic selection schemes. With these heuristic schemes, the relative approximation error can stop decreasing or even increase when the approximation rank becomes large. One part of the reason for the increasing approximation error is the numerical instability of computing the ID approximation of  $K(X_0, Y_p)$  for a large approximation rank. With an improper selection of proxy points  $Y_p$  (especially the selection on the surface),  $K(X_0, Y_p)$  can have numerical rank smaller than the actual rank needed for an ID approximation of  $K(X_0, Y_0)$  to obtain a given accuracy. As a result, the ID approximation of  $K(X_0, Y_p)$  computed by SRRQR with a given approximation rank larger than the numerical rank of  $K(X_0, Y_p)$  becomes numerically unstable.

**6.4. Comparison with algebraic compression methods.** In this subsection, we compare the proxy point method with three algebraic compression methods: (1) ID using SRRQR, (2) adaptive cross approximation (ACA) with partial pivoting [4], and (3) a pseudoskeleton approximation method based on random column sampling and the pivoted QR decomposition that is similarly presented in [26].

Consider the same test settings as in the previous subsection:  $K(x, y) = e^{-|x-y|^2}$ ,  $\mathcal{X} = [-1, 1]^2$ ,  $\mathcal{Y} = [-7, 7]^2 \setminus [-3, 3]^2$ , 384 proxy points selected by Algorithm 5.1, 400 points in  $\mathcal{X}$  for  $X_0$ , and 16000 points in  $\mathcal{Y}$  for  $Y_0$ . For different approximation ranks, Figure 6.7 plots the relative approximation errors and running times of the four different compression methods. As can be observed, the proxy point method has slightly better approximation accuracy and much less computational cost compared to ACA and the pseudoskeleton approximation method. However, it is worth noting that the accuracy difference between these tested methods is also related to the actual distribution of points  $Y_0$  in the domain  $\mathcal{Y}$ . Meanwhile, all these methods and tests are implemented in Matlab without code optimization. These facts could also affect the test results. More tests are needed for a comprehensive comparison of these methods but these are outside the scope of this paper.

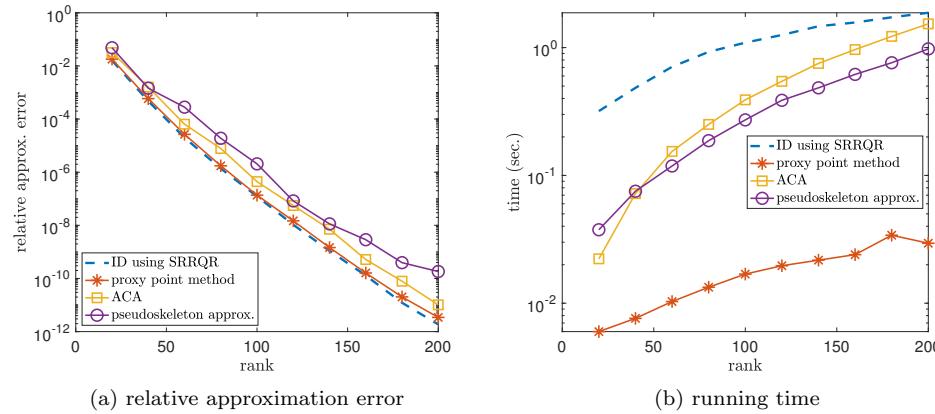


FIG. 6.7. *Relative approximation error and running time of the proxy point method and three different algebraic compression methods. The running time of the proxy point method in (b) does not include the running time of the proxy point selection by Algorithm 5.1, which is 5.9 seconds.*

We note that the running time of the proxy point method in Figure 6.7 does not include the running time of the proxy point selection by Algorithm 5.1, which is 5.9 seconds. As discussed in subsection 5.4, the proxy point selection by Algorithm 5.1 should be viewed as a precomputation step, and the selected proxy points are used

repeatedly when applying the proxy point method in rank-structured matrix construction (see the numerical test in subsection 6.5). Thus, it is reasonable to ignore the running time of the proxy point selection when comparing the proxy point method with other algebraic compression methods.

**6.5. Application to  $\mathcal{H}^2$  matrix construction.** We now apply the proxy point method to efficiently construct  $\mathcal{H}^2$  matrix representations of kernel matrices in the form  $K(X, X)$  defined by a nonoscillatory kernel function  $K(x, y)$  and randomly generated points  $X$ . Readers can refer to [8, 25] for details of  $\mathcal{H}^2$  matrix construction that has kernel matrix blocks compressed in the form of an ID.

In  $d$ -dimensional space ( $d = 2$  or  $d = 3$ ), we uniformly and randomly select  $N$  points in a square or cubical box with edge length  $L = N^{\frac{1}{d}}$  for  $X$ . This box, enclosing all the points, is recursively partitioned into smaller boxes. Specifically, a box is uniformly subdivided into  $2^d$  smaller boxes if it contains more than 300 points. With uniformly distributed points, we partition all the boxes at every level until the finest level. At the  $k$ th level of the recursive partitioning, the original box is partitioned into  $2^{dk}$  boxes with edge length  $L/2^k$ . Such a recursive partitioning has  $O(\log N)$  levels in total.

$\mathcal{H}^2$  matrix construction at the  $k$ th level needs to compute ID approximations of  $2^{dk}$  number of kernel matrix blocks. These blocks share the same form  $K(X_0, Y_0)$  where, for some box  $\mathcal{B}$  at the  $k$ th level,  $X_0$  is a set of points lying in  $\mathcal{B}$  and  $Y_0$  is a set of points lying in all the other boxes that are at least one box away from  $\mathcal{B}$ . Assuming that  $K(x, y)$  is translationally invariant, we only need to select proxy points  $Y_p$  using Algorithm 5.1 for  $K(x, y)$  with domains

$$\mathcal{X} = \left[ -\frac{L}{2^{k+1}}, \frac{L}{2^{k+1}} \right]^d \quad \text{and} \quad \mathcal{Y} = \left[ -(L - \frac{L}{2^{k+1}}), L - \frac{L}{2^{k+1}} \right]^d \setminus \left[ -\frac{3L}{2^{k+1}}, \frac{3L}{2^{k+1}} \right]^d.$$

We can then apply the proxy point method to each  $K(X_0, Y_0)$  at the  $k$ th level with  $Y_p$  properly shifted according to the position of  $\mathcal{B}$  relative to  $\mathcal{X}$ .

We test two different problem settings:  $K(x, y) = 1/\sqrt{1 + |x - y|^2}$  in 2D and  $K(x, y) = (1 + 0.01|x - y|)e^{-0.01|x - y|}$  in 3D. The proxy point method is used to compute all the ID approximations in the construction of the  $\mathcal{H}^2$  matrix, where all the intermediate ID approximations of  $K(X_0, Y_p)$  in the method are computed with relative error threshold  $\tau = 10^{-6}$ . Here, an ID approximation  $UA_J$  to a matrix  $A$  meets a relative error threshold  $\tau$  if the 2-norm of each row of  $A - UA_J$  is bounded by  $\tau$  times the maximum of the 2-norm of each row of  $A$ . For comparison, the  $\mathcal{H}^2$  matrix construction with all the ID approximations computed using SRRQR with relative error threshold  $\tau = 10^{-6}$  is also tested. All the tests are implemented in Matlab.

Figures 6.8 and 6.9 plot the construction time, storage cost, and relative error of  $\mathcal{H}^2$  matrix constructions for the two problem settings, respectively. For a constructed  $\mathcal{H}^2$  matrix, its relative error is measured as the relative error of 100 randomly chosen entries of the product of the  $\mathcal{H}^2$  matrix with a random vector compared with the entries of the exact product of the original kernel matrix with the vector. The reported relative error results are averaged over 20 independent tests.

The runtime of selecting proxy points is significant but its asymptotic complexity is only  $O(\log N)$  since Algorithm 5.1 is applied only once at each level of  $\mathcal{H}^2$  matrix construction. The proxy point method leads to nearly linear  $\mathcal{H}^2$  matrix construction, which can also be justified theoretically if we assume that the number of proxy points  $Y_p$  selected at each level is  $O(1)$ . Also, the storage cost and relative error of  $\mathcal{H}^2$  matrices constructed by the proxy point method are close to those by SRRQR, indicating

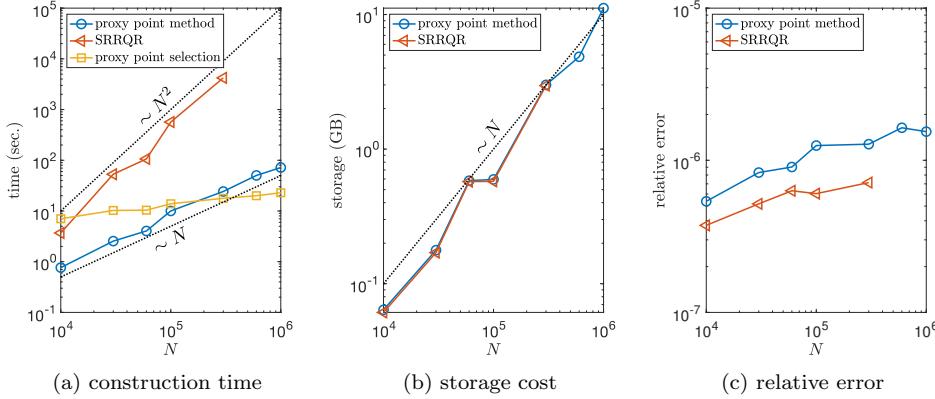


FIG. 6.8. Results of the  $\mathcal{H}^2$  matrix construction for  $K(x,y) = 1/\sqrt{1+|x-y|^2}$  in 2D: (a) construction time of  $\mathcal{H}^2$  matrices, (b) storage cost of the constructed  $\mathcal{H}^2$  matrices, (c) relative error of the constructed  $\mathcal{H}^2$  matrices. In the legend, “proxy point method” and “SRRQR” refer to the corresponding ID approximation methods used in the  $\mathcal{H}^2$  matrix construction. In (a), “proxy point selection” refers to the total runtime of selecting the proxy points by Algorithm 5.1 for  $\mathcal{H}^2$  matrix construction. Reference lines for linear and quadratic scaling with  $N$  are also plotted.

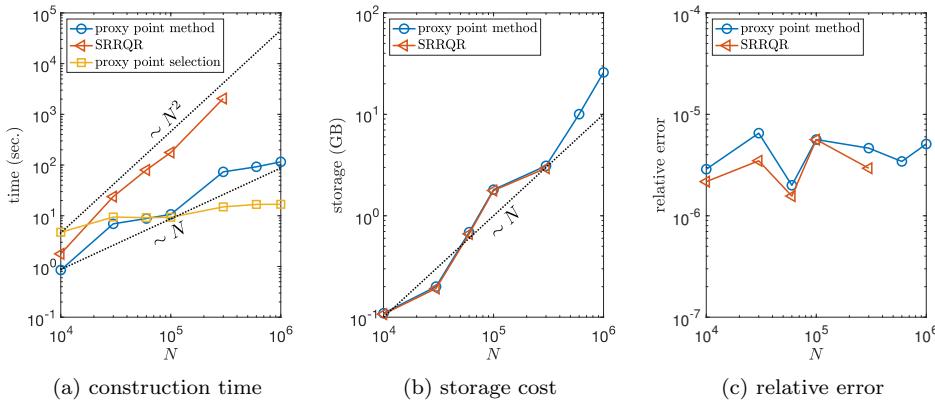


FIG. 6.9. Results of the  $\mathcal{H}^2$  matrix construction for  $K(x,y) = (1+0.01|x-y|)e^{-0.01|x-y|}$  in 3D: (a) construction time of  $\mathcal{H}^2$  matrices, (b) storage cost of the constructed  $\mathcal{H}^2$  matrices, (c) relative error of the constructed  $\mathcal{H}^2$  matrices.

that the proxy point method is as effective as SRRQR in terms of the approximation rank and accuracy for ID approximations in  $\mathcal{H}^2$  matrix construction.

**7. Conclusion.** In this paper, we illustrate the effectiveness of the proxy point method in a general problem setting, i.e., any kernel function  $K(x,y)$  and two compact domains  $\mathcal{X}$  and  $\mathcal{Y}$  satisfying that  $\mathcal{X} \times \mathcal{Y}$  is admissible for  $K(x,y)$  and  $K(x,y)$  is smooth in  $\mathcal{X} \times \mathcal{Y}$ , based on a rigorous error analysis. The selection of proxy points plays a critical role to control the approximation accuracy of the method. While only heuristic selections are used presently in practice, Algorithm 5.1 is proposed to systematically select a proper set of proxy points for any given kernel function and corresponding admissible domain pair. The effectiveness of this selection scheme is justified both theoretically and numerically.

Based on the selection scheme, the proxy point method can now be used to accelerate the construction of all types of rank-structured matrices with translationally invariant kernel functions in low-dimensional spaces by simply replacing the existing compression methods by the proxy point method. The kernel function being translationally invariant allows reusing the proxy points and thus amortizes the expensive cost of Algorithm 5.1, which is presently the main limitation for practical applications of Algorithm 5.1. For kernel functions that are not translationally invariant, the error analysis for the proxy point method is still valid and thus may be used to help design more efficient selection schemes for specific kernel functions.

## REFERENCES

- [1] S. AMBIKASARAN AND E. DARVE, *An  $\mathcal{O}(N \log N)$  fast direct solver for partial hierarchically semi-separable matrices*, J. Sci. Comput., 57 (2013), pp. 477–501.
- [2] M. BEBENDORF, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.
- [3] M. BEBENDORF, C. KUSKE, AND R. VENN, *Wideband nested cross approximation for Helmholtz problems*, Numer. Math., 130 (2015), pp. 1–34.
- [4] M. BEBENDORF AND S. RJASANOW, *Adaptive low-rank approximation of collocation matrices*, Comput., 70 (2003), pp. 1–24.
- [5] S. BÖRM, *Directional  $\mathcal{H}^2$ -matrix compression for high-frequency problems*, Numer. Linear Algebra Appl., 24 (2017), p. e2112.
- [6] L. BOS, S. DE MARCHI, A. SOMMARIVA, AND M. VIANELLO, *Computing multivariate Fekete and Leja points by numerical linear algebra*, SIAM J. Numer. Anal., 48 (2010), pp. 1984–1999.
- [7] L. BOS, M. TAYLOR, AND B. WINGATE, *Tensor product Gauss-Lobatto points are Fekete points for the cube*, Math. Comput., 70 (2001), pp. 1543–1547.
- [8] D. CAI, E. CHOW, L. ERLANDSON, Y. SAAD, AND Y. XI, *SMASH: Structured matrix approximation by separation and hierarchy*, Numer. Linear Algebra Appl., 25 (2018), p. e2204.
- [9] L. CAMBIER AND E. DARVE, *Fast low-rank kernel matrix factorization using skeletonized interpolation*, SIAM J. Sci. Comput., 41 (2019), pp. A1652–A1680.
- [10] 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.
- [11] 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.
- [12] E. CORONA, P. G. MARTINSSON, AND D. ZORIN, *An  $\mathcal{O}(N)$  direct solver for integral equations on the plane*, Appl. Comput. Harmonic Anal., 38 (2015), pp. 284–317.
- [13] B. ENGQUIST AND L. YING, *Fast directional multilevel algorithms for oscillatory kernels*, SIAM J. Sci. Comput., 29 (2007), pp. 1710–1737.
- [14] B. ENGQUIST AND L. YING, *Fast directional algorithms for the Helmholtz kernel*, J. Comput. Appl. Math., 234 (2010), pp. 1851–1859.
- [15] W. FONG AND E. DARVE, *The black-box fast multipole method*, J. Comput. Phys., 228 (2009), pp. 8712–8725.
- [16] A. GILLMAN AND A. BARNETT, *A fast direct solver for quasi-periodic scattering problems*, J. Comput. Phys., 248 (2013), pp. 309–322.
- [17] Z. GIMBUTAS AND V. ROKHLIN, *A generalized fast multipole method for nonoscillatory kernels*, SIAM J. Sci. Comput., 24 (2003), pp. 796–817.
- [18] S. A. GOREINOV, E. E. TYRTYSHNIKOV, AND N. L. ZAMARASHKIN, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.
- [19] L. GREENGARD AND V. ROKHLIN, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer., 6 (1997), pp. 229–269.
- [20] M. GU AND S. EISENSTAT, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
- [21] W. HACKBUSCH, *A sparse matrix arithmetic based on  $\mathcal{H}$ -matrices. Part I: Introduction to  $\mathcal{H}$ -matrices*, Comput., 62 (1999), pp. 89–108.
- [22] W. HACKBUSCH AND S. BÖRM, *Data-sparse approximation by adaptive  $\mathcal{H}^2$ -matrices*, Comput., 69 (2002), pp. 1–35.
- [23] W. HACKBUSCH, B. KHOROMSKIJ, AND S. A. SAUTER, *On  $\mathcal{H}^2$ -matrices*, Lect. Appl. Math., (2000), pp. 9–29.
- [24] W. HACKBUSCH AND B. N. KHOROMSKIJ, *A sparse  $\mathcal{H}$ -matrix arithmetic. Part II: Application to multi-dimensional problems*, Comput., 64 (2000), pp. 21–47.

- [25] 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.
- [26] N. KISHORE KUMAR AND J. SCHNEIDER, *Literature survey on low rank approximation of matrices*, Linear Multilinear Algebra, 65 (2017), pp. 2212–2244.
- [27] Y. LI, H. YANG, E. R. MARTIN, K. L. HO, AND L. YING, *Butterfly factorization*, Multiscale Model. Simul., 13 (2015), pp. 714–732.
- [28] W. B. MARCH, B. XIAO, AND G. BIROS, *ASKIT: Approximate skeletonization kernel-independent treecode in high dimensions*, SIAM J. Sci. Comput., 37 (2015), pp. A1089–A1110.
- [29] 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.
- [30] G. MASTROIANNI AND G. MILOVANOVIC, *Interpolation Processes: Basic Theory and Applications*, Springer Science & Business Media, New York, 2008.
- [31] E. MICHELSSEN AND A. BOAG, *A multilevel matrix decomposition algorithm for analyzing scattering from large structures*, IEEE Trans. Antennas Propagation, 44 (1996), pp. 1086–1093.
- [32] V. MINDEN, A. DAMLE, K. L. HO, AND L. YING, *Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations*, Multiscale Model. Simul., 15 (2017), pp. 1584–1611.
- [33] E. REBROVA, G. CHÁVEZ, Y. LIU, P. GHYSELS, AND X. S. LI, *A study of clustering techniques and hierarchical matrix formats for kernel ridge regression*, in Proceedings of the 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE, 2018, pp. 883–892.
- [34] X. XING AND E. CHOW, *Error analysis of an accelerated interpolative decomposition for 3D Laplace problems*, Appl. Comput. Harmonic Anal., to appear, 2019.
- [35] X. YE, J. XIA, AND L. YING, *Analytical Low-Rank Compression via Proxy Point Selection*, Preprint, arXiv:1903.08821, 2019.
- [36] L. YING, *A kernel independent fast multipole algorithm for radial basis functions*, J. Comput. Phys., 213 (2006), pp. 451–457.
- [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.
- [38] B. ZHANG, J. HUANG, N. P. PITSIANIS, AND X. SUN, *A Fourier-series-based kernel-independent fast multipole method*, J. Comput. Phys., 230 (2011), pp. 5807–5821.