KERNEL MATRIX RANK STRUCTURES WITH APPLICATIONS

by

Mikhail Lepilov

A Dissertation

Submitted to the Faculty of Purdue University In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Department of Mathematics West Lafayette, Indiana May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL STATEMENT OF COMMITTEE APPROVAL

Dr. Jianlin Xia, Chair

Department of Mathematics

Dr. Jie Shen

Department of Mathematics

Dr. Guang Lin

Department of Mathematics

Dr. Zhiqiang Cai

Department of Mathematics

Approved by:

Dr. Plamen Stefanov

To my family and friends.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my dissertation advisor, Jianlin Xia. He has always provided helpful guidance in every situation. More importantly, he has helped keep me focused on the most important tasks. But most importantly, he has been very kind to me. I have learned, as many people quickly learn in grade school, that this is quite rare among people who study mathematics. I would also like to thank the other members of thesis committee: Prof. Jie Shen, Prof. Guang Lin, and Prof. Zhiqiang Cai. Their suggestions have helped to shape my thesis into a far more readable state.

Next, I would like to thank Roberto "Misterobo" Ulloa. He has been a kindred spirit during our long time together in the program, and I immensely value his openness about living and learning as a human being on this planet. I would also like to thank Aashna "Aashnaba" Aggarwal for sharing her valuable perspective with me over several years, especially as it has initially been quite different from mine. Moreover, I would like to thank Lily, a.k.a. Lilybug, a.k.a. Lilycat, a.k.a. Lil Buggie, a.k.a. Buggie, a.k.a. Bubbie, a.k.a. Squibbie, a.k.a. Squibble, a.k.a. Bubble. She has been the gentlest cat I have encountered to date.

Furthermore, I would like to thank my friends far and wide that have allowed our friendship to continue blossoming, or at the very least have not allowed it to start wilting: Timothy Yi, Colin Ford, Pascal Jundt, Andre Frankenthal, Ilyas Yilgor, David Batuner, Aleksey Mafusalov, and Nick Golikov. I am glad for each person to have influenced me the way that they have, as people are inevitably wont to.

Finally, I would like to thank my family for their support over these years.

The order of these lists is not related to the relative importance of the people named, because in actuality each list is ordered west to east with respect to each person's present address, starting at the International Date Line. Thankfully, there were no ties. Lily's room is to the east of Aashna's, mind you.

TABLE OF CONTENTS

| LI | ST O | F TAB | LES | 7 | | | | |
|----|------|--|---|----|--|--|--|--|
| LI | ST O | F FIGU | URES | 8 | | | | |
| Al | BSTR | ACT | | 10 | | | | |
| 1 | INT | RODU | CTION | 11 | | | | |
| | 1.1 | Review | w of low-rank approximation | 14 | | | | |
| | | 1.1.1 | Algebraic low-rank compression techniques | 14 | | | | |
| | | 1.1.2 | Analytic low-rank compression techniques | 16 | | | | |
| | 1.2 | Review | w of HSS matrix approximation | 19 | | | | |
| 2 | HYF | PERFAS | ST HSS CONSTRUCTION OF SOME CAUCHY MATRICES | 22 | | | | |
| | 2.1 | Analy | tic far-field compression by the proxy point method | 25 | | | | |
| | 2.2 | Relati | onships among off-diagonal blocks of C | 32 | | | | |
| | 2.3 | 2.3 Hyperfast analytic hierarchical compression scheme | | | | | | |
| | | 2.3.1 | Compression at the leaf level | 38 | | | | |
| | | 2.3.2 | Compression at nonleaf levels | 41 | | | | |
| | | 2.3.3 | Other HSS generators | 43 | | | | |
| | 2.4 | Algori | thm analysis and performance | 44 | | | | |
| 3 | HYF | PERFAS | ST HSS CONSTRUCTION OF SOME TOEPLITZ MATRICES | 52 | | | | |
| | 3.1 | Gener | al proxy point method analysis for analytic kernels | 54 | | | | |
| | 3.2 | Sublin | ear Toeplitz kernel HSS generator construction | 59 | | | | |
| | | 3.2.1 | Constructing the HSS row generators | 60 | | | | |
| | | 3.2.2 | Constructing the remaining HSS generators | 64 | | | | |
| | | 3.2.3 | Number of proxy points required | 64 | | | | |
| | 3.3 | Nume | rical tests | 68 | | | | |
| | 3.4 | Furthe | er extensions | 70 | | | | |
| 4 | MUI | LTIDIM | IENSIONAL PROXY POINT METHOD | 71 | | | | |

| 4.1 | Error analysis for the proxy point method in \mathbb{C}^k | | | | | |
|-------|---|----|--|--|--|--|
| 4.2 | General HSS matrix construction and rank bounds | 77 | | | | |
| 4.3 | Further extensions | 82 | | | | |
| REFER | RENCES | 83 | | | | |

LIST OF TABLES

| 3.1 | The size <i>n</i> of the matrix T_n and the minimum number of proxy points <i>N</i> required to attain $\ (T_n) _{\mathbf{\hat{i}}_{n,1},[1,n]\setminus\mathbf{\hat{i}}_{n,1}} - \tilde{T}_n^{1,N}\ _F < 10^{-6}$. | 65 |
|-----|---|----|
| 3.2 | The size n of the matrix T_n and the minimum value of N such that the L -level HSS approximation constructed in Sections 3.2.1 and 3.2.2 with N proxy points approximates the topmost HSS block of T_n to a relative Frobenius norm error 10^{-10} . | 66 |
| 3.3 | The relative Frobenius norm errors of the <i>L</i> -level HSS approximation to T_n from Sections 3.2.1 and 3.2.2 using N proxy points | 69 |
| 3.4 | The relative Frobenius norm errors of the <i>L</i> -level HSS approximation to T_n from Sections 3.2.1 and 3.2.2 using N proxy points, as well as the numerical HSS rank s of T_n with tolerance 10^{-14} . | 69 |
| 4.1 | The relative Frobenius norm errors of the multidimensional proxy point approx- imation to the Gauss kernel matrix applied to $X, Y \subseteq \mathbb{R}^2$ giving varying values of R/r , and using varying numbers of proxy points N in each dimension | 76 |
| 4.2 | The relative Frobenius norm errors of the multidimensional proxy point approxi- mation to the Coulomb kernel matrix applied to $X, Y \subseteq \mathbb{R}^2$ giving varying values of R/r , and using varying numbers of proxy points N in each dimension | 76 |
| 4.3 | The relative Frobenius norm errors of the approximation to the far field of the topmost HSS row block of an $n \times n$ matrix A , viewed as a kernel matrix in the sense of Equation 4.1 with kernels g_1 and g_2 , with $n = 8192$ using N proxy points. | 81 |

LIST OF FIGURES

| 1.1 | The HSS block rows and columns of A where $L = 2$. The labeled green blocks with rounded corners correspond to the HSS tree depth $l = 1$; the labeled yellow blocks with sharp corners correspond to the HSS tree depth $l = 2$ | 20 |
|------|---|----|
| 1.2 | The HSS generator products of A placed into the blocks of A that they generate. | 21 |
| 2.1 | Illustration of the sets under consideration, where each cross (\times) is a point in X , each box (\Box) is a point in Y , $\mathbf{s} \subset X$ is the source set, $\mathbf{t} \subset Y$ is the target set, and \mathbf{z} is a set of proxy points. | 26 |
| 2.2 | Illustration of the regions of definition for the piecewise-defined functions γ_1 and γ_2 , along with points in the sets \mathbf{s} (x) and \mathbf{t} (D). | 29 |
| 2.3 | Illustration of the points $z_0 = \rho \omega^{k-1} e^{i\theta}$, $\tilde{z}_0 = \rho \omega^{k-1} e^{-i\theta}$, and $\hat{z}_0 = \rho \omega^{k-1}$, along with points in the sets s (×) and t (□), where the dotted black line illustrates the | 20 |
| 0.4 | | 29 |
| 2.4 | Partitioning of C_i and C_j . | 35 |
| 2.5 | The relationship between $C_{i_1}^{ } _{\{2:m\}}$ and $C_{i_1}^{-} _{\{:\}\times\{1:m-1\}}$, when i_1 is the leftmost node at level l of \mathcal{T} . | 37 |
| 2.6 | Illustration of the sets under consideration, the sparsification of \mathbf{s}_i into $\tilde{\mathbf{s}}_i$ for a leaf i, and the resulting source point subsets of X after one level of compression. | 39 |
| 2.7 | Partitioning of the HSS block C_i^- corresponding to Figure 2.6(a) | 39 |
| 2.8 | Forming $X_i = \mathbf{\hat{s}}_{c_1} \cup \mathbf{\hat{s}}_{c_2}$ and \mathbf{s}_i and sparsifying \mathbf{s}_i into $\mathbf{\tilde{s}}_i$ for a nonleaf node i | 42 |
| 2.9 | Operation count for Algorithm 1 with varying n , together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the flop count associated with forming the D , U , and R generators in the left plot is asymptotically $O(\log^2 n)$ and that the flop count as- sociated with forming the B generators in the right plot is asymptotically $O(\log^3 n)$. | 47 |
| 2.10 | Effective storage costs for Algorithm 1 with varying n , together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the effective storage cost for forming the D, U , and R generators in the left plot is $O(\log^2 n)$ and that the effective storage cost for forming the B generators in the right plot is $O(\log^3 n)$. | 48 |
| 2.11 | The number of exponential function evaluations in Algorithm 1 with varying n , together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the number of function evaluations for forming the D , U , and R generators in the left plot is $O(\log^2 n)$ and that the number of evaluations for forming the B generators in the right plot is $O(\log^2 n)$ | 40 |
| | 10 0 (105 10) | чЈ |

| 2.12 | The relative accuracies in the Frobenius norm for the HSS approximation to C using the generators constructed by the new algorithm. Here, the leaf level size was kept constant at 64, both the off-diagonal far-field rank r and the number of proxy points was 25. | 49 |
|------|--|----|
| 2.13 | The cardinalities of \mathbf{s}_i (·) and $\mathbf{\tilde{s}}_i$ (o) for $n = 4096$, $L = 6$, and $r = 25$. Here, $r = 25$ proxy points were also used in the far-field approximation | 50 |
| 2.14 | The cardinalities of X_i (·) and $\mathbf{\hat{s}}_i$ (o) for $n = 4096$, $L = 6$, and $r = 25$. Again, $r = 25$ proxy points were also used in the far-field approximation. | 51 |
| 3.1 | Symmetry of the proxy point contours (dotted lines) and their enclosed points \mathbf{s}_i (\Box) | 52 |
| 3.2 | Setup for a proxy point approximation to $k(X, Y)$. (The set Y is not pictured.) | 55 |
| 3.3 | Top: the near-field points $\tilde{\mathbf{i}}'_1$ (\circ), far-field points $\hat{\mathbf{i}}'_1$ (\bullet), proxy points (\times), and the points [9, 32] (\Box) involved in the approximation of the leaf HSS block $T_1^- _{\tilde{\mathbf{i}}'_1\times[1,n- \mathbf{i}_1]} = k$ ([1, 8], [9, 32]) for a matrix of size $n = 32$, number of HSS levels $L = 2$, and number of proxy points $N = 16$. Bottom: the resulting index set \mathbf{i}'_1 (\Box). (These are "cartoon illustrations" and are not actual results from such an approximation applied to a subblock of an actual matrix T .) | 62 |
| 3.4 | Top: the near-field points $\tilde{\mathbf{i}}'_3$ (\circ), far-field points $\hat{\mathbf{i}}'_3$ (\bullet), proxy points (\times), and the points [17, 32] (\Box) involved in the approximation of the height-2 HSS block $T_3^- _{\tilde{\mathbf{i}}'_3\times[1,n- \mathbf{i}_3]} = k\left(\tilde{\mathbf{i}}'_j, [17, 32]\right)$ for a matrix of size $n = 32$, number of HSS levels $L = 2$, and number of proxy points $N = 16$. Bottom: the resulting index set \mathbf{i}'_3 (\Box). (As noted in Figure 3.3 above, these are "cartoon illustrations" and are not reflective of actual numerical results.) | 63 |
| 4.1 | The definition of the separation R/r between points X (\circ , bottom left) and Y (\Box , top right), for $k = 2$ and $X, Y \subseteq \mathbb{R}^2$. (That is, $\operatorname{im}(w) = 0$ for all $w \in X, Y$.) The smallest box centered at \mathbf{c} containing X , as well as the largest box containing X but excluding Y centered at \mathbf{c} , are displayed (dashed squares). The point \mathbf{c} is the black cross (\times) | 75 |
| 4.2 | The sets $X \subseteq \mathbb{R}^2$ (blue, bottom left) and $Y \subseteq \mathbb{R}^2$ (red, top right) used in Example 4.1.2 for $R/r = 2$. Here, $ X = 4096$ and $ Y = 8192$. | 76 |
| 4.3 | The points X (\Box , blue) and Y (\circ , yellow) in our setup for viewing the top HSS row block (shown in grey) of a general matrix as a kernel matrix, for $n = 16$. In performing a proxy point approximation, as before, we only apply it to the "far field" of X (shown as squares with white interior) to ensure separation | 77 |

ABSTRACT

Many kernel matrices from differential equations or data science applications possess low or approximately low off-diagonal rank for certain key matrix subblocks; such matrices are referred to as rank-structured. Operations on rank-structured matrices like factorization and linear system solution can be greatly accelerated by converting them into hierarchical matrix forms, such as the hiearchically semiseparable (HSS) matrix form. The dominant cost of this conversion process, called HSS construction, is the low-rank approximation of certain matrix blocks. Low-rank approximation is also a required step in many other contexts throughout numerical linear algebra. In this work, a proxy point low-rank approximation method is detailed for general analytic kernel matrices, in both one and several dimensions. A new accuracy analysis for this approximation is also provided, as well as numerical evidence of its accuracy. The extension of this method to kernels in several dimensions is novel, and its new accuracy analysis makes it a convenient choice to use over existing proxy point methods. Finally, a new HSS construction algorithm using this method for certain Cauchy and Toeplitz matrices is given, which is asymptotically faster than existing methods. Numerical evidence for the accuracy and efficacy of the new construction algorithm is also provided.

1. INTRODUCTION

Large matrices feature prominently in various areas of sciences, engineering, and applied mathematics, such as in data science methods and differential equation solutions. When the dimensions of the matrix in question are large, it is already either impractical or impossible to do something as basic as to solve the corresponding linear system using the standard methods learned in a first-year linear algebra class. Not only does the time complexity of such "standard methods," such as Gaussian elimination without pivoting, scale cubically in the dimensions of the matrix, such methods also suffer from very poor numerical stability, potentially leading to errors rendering the computation useless. For a broad overview of the shortcomings of these naive algorithms and ways to address them, see [1].

Although more sophisticated algorithms that the above reference discusses, such as Gaussian elimination with partial pivoting or Householder QR factorization, have better stability properties, they are still not practical when the size of the matrix is very large. Hence, methods have been devised that leverage additional matrix structure for a substantial asymptotic speedup. Perhaps the most well-known example of additional matrix structure is "sparsity," or the property that a matrix has asymptotically fewer nonzero entries compared to its dimensions. Examples of algorithms that take advantage of sparsity are the multifrontal method [2], as well as the many iterative methods covered by the comprehensive survey in [3]. Another example of matrices which possess additional structure are Toeplitz matrices; at the outset of Chapter 3, we will give a brief overview of the development of fast algorithms for Toeplitz matrices and propose a new one. Yet another broad class of matrices for which fast algorithms have been devised, and which we will focus on in this work, is that of rank-structured matrices.

The term "rank structure" refers to either the low rank or the approximate low rank (the latter defined precisely in Section 1.1) of certain off-diagonal subblocks of a matrix. Here, "low rank" is usually taken to mean bounded by a polynomial in the logarithm of the matrix dimensions. Another term for rank-structured matrices is "data-sparse matrices." It should be noted that these terms are typically used in the context of dense—i.e. not sparse—matrices. Nevertheless, rank-structured matrix methods can be used to accelerate sparse matrix computations as well, as in [4].

There are many ways of leveraging the rank structure of a matrix, most frequently by using data structures such as SSS (sequentially semi-separable), HSS (hierarchically semiseparable), HODLR (hierarchical off-diagonal low-rank), or similar matrix forms. Such data structures, sometimes called hierarchical matrix representations, allow either a representation of or an approximation to any given matrix that allows for asymptotically faster (in the dimensions of the matrix) algorithms thereon, such as matrix-vector products, inversion, or solution of the associated linear systems. These representations also use asymptotically less storage. In exchange for these desirable properties, a modest cost is often paid in the form of converting a given matrix into such a data structure; this process is referred to as "construction."

The exact identities of the "certain off-diagonal subblocks" referred to above depend entirely on context: examples include so-called "HSS blocks" and "HODLR blocks" [5] (see below), as well as other subblocks [6]. A lot of matrices encountered in data science applications turn out to possess rank structure [7]. Furthermore, a lot of so-called kernel matrices with certain analytic and/or geometric properties possess rank structure. The first focus of our work here is to identify such rank structures and construct low-rank approximations to kernel matrices.

In recent years, a lot of research has focused on the HSS matrix form due to the existence of parallelizable and numerically stable algorithms that operate on HSS matrices. The first such algorithms have appeared in, for example, [8]–[10]; a broad overview of some of these ideas can be found in [11]. In particular, the stability advantage of rank-structured algorithms is analyzed in [12]. Our second focus here is on aspects of the HSS form representation of the aforementioned kernel matrices, which we will briefly review in Section 1.2. In particular, in Chapters 2 and 3, we will detail new HSS construction algorithms for certain classes of kernel matrices. Such an algorithm helps solve an existing bottleneck in the fastest HSS-based solvers for the relevant matrices. We will also propose a generalization of this algorithm in Chapter 4. In the literature, the term "kernel matrix" is sometimes vague, so it is useful to clarify what we mean. We will take it to mean, given some sets X, Y, Z, a matrix whose (i, j)th entry is $k(x_i, y_j)$ for some points $x_i \in X$, $y_j \in Y$ and for some function $k : X \times Y \to Z$. The function k is referred to as a "kernel function," and the sets X, Y, Z are usually subsets of the real or complex numbers. Of course, every matrix can be viewed as a kernel matrix if either k or the sets X, Y, Z do not have to satisfy any particular properties, so the interpretation of a matrix as a kernel matrix is frequently context-specific. (We will touch upon this question, in our context of complex-analytic functions k defined on complex domains, in Section 4.2.)

The rest of the work is structured as follows: in the remainder of this chapter, we will review concepts and methods related to the low-rank approximation of a matrix, in particular the proxy point method and the HSS form of a matrix. In Chapter 2, we will combine certain algebraic and analytic properties of some prevalent types of kernel matrices known as Cauchy matrices in a new algorithm to construct its HSS form. This construction is done with sublinear time complexity, which is asymptotically faster than any previously known methods. Some new analysis and a new algorithm are detailed. In Chapter 3, we come up with a new, more general proxy point analysis than any previously known; it is applicable to any complex-analytic function of one variable. We then use it to come up with low-rank approximations to and detail a similarly quick algorithm for the HSS construction of some Toeplitz matrices. Finally, in Chapter 4, we will generalize the proxy point analysis given in Chapter 3 to any analytic kernel function of any number of variables. We then sketch a generalization of the new sublinear-time HSS construction from Chapter 3 to certain classes of general matrices.

Throughout this work, we use the following notation: let $k : F \times G \to \mathbb{C}$ be a function and $X \subseteq F, Y \subseteq G$ be totally-ordered finite subsets of size r and s, respectively. Then by k(X, Y) we mean the r-by-s matrix A with entries $A_{j,k} = k(x_j, y_k)$, where x_j is the jth element of X and y_k is the kth element of Y. Furthermore, for $a, b \in \mathbb{N}$, define $[a, b] = \{j \in \mathbb{N} : a \leq j \leq b\}$. (In particular, with this notation we will never mean the closed interval on the real line from a to b.) Moreover, let C be an l-by-m matrix and $L \subseteq [1, l], M \subseteq [1, m]$. Then by $A_{L \times M}$ we mean the |L|-by-|M| "submatrix" B of A consisting of entries $B_{j,k} = A_{l_j,m_k}$, where l_j is the jth element of L and m_k is the kth element of M, ordered the usual way. Finally, we also

write $C|_L$ and $C|_{\{:\}\times M}$ mean the "submatrices" of C corresponding to the rows specified by L and columns specified by M, respectively.

1.1 Review of low-rank approximation

First, we define what we mean by "approximately low-rank."

Definition 1.1.1. Let $\tau > 0$, and let $A \in \mathbb{C}_{m \times n}$ for some $m, n \in \mathbb{N}$. Then the numerical rank of A with respect to τ is defined as $\min\{\operatorname{rank}(B) \mid B \in \mathbb{C}_{m \times n} ||A - B||_2 < \tau\}.$

With this definition in hand, we can say that a matrix A is "approximately low-rank" if its numerical rank with respect to some context-specific tolerance τ , itself often picked to be a small fraction of $||A||_2$, is bounded by a polynomial in the logarithm of the dimensions of A.

From the definition, it may not be immediately clear, given A and τ , which approximant B might satisfy $||A - B||_2 < \tau$. The well-known Eckart-Young-Mirsky Theorem answers this question:

Theorem 1.1.1 (Eckart-Young-Mirsky). Let $\tau > 0$, $A \in \mathbb{C}_{m \times n}$ for some $m, n \in \mathbb{N}$, and let $A = U\Sigma V^*$ be the singular value decomposition. Assume that all the singular values of Σ are distinct; an analogous statement follows otherwise. Then

$$\|A - U\Sigma_k V^*\|_f < \tau, \tag{1.1}$$

where f is either the 2-norm or the Frobenius norm, Σ_k is the matrix identical to Σ in its top k values and with zero values otherwise, and k is the numerical rank of A with respect to τ .

1.1.1 Algebraic low-rank compression techniques

Algebraic low-rank compression methods are designed to be applicable to any matrix A, without assuming any additional structure. As Theorem 1.1.1 suggests, one low-rank approximation technique we may apply is a truncated singular value decomposition; that

is, we approximate A by $U\Sigma_k V^*$ as in (1.1). However, recalling [1], the standard singular value decomposition algorithm for general $n \times n$ matrices scales poorly with respect to n, although faster algorithms exist. (This is especially true for certain subclasses of matrices; see, e.g. [13].) Instead, it is possible to settle for less-than-optimal general algebraic lowrank approximation that performs well in practice. One such approximation is the strong rank-revealing QR factorization [14]:

Proposition 1.1.2. Let $A \in \mathbb{C}_{m \times n}$ for $m \ge n$. Let $\sigma_k(A)$ denote the kth singular value of A. Then we may compute a factorization $A\Pi = QR$ where

$$R = \begin{pmatrix} U & V \\ 0 & W \end{pmatrix},$$

 $U \in \mathbb{C}_{k \times k}$ is upper triangular and $\sigma_k(A)p(k,n) \ge ||W||_2 \ge \sigma_k(A)$ in sub-cubic time if $k \ll n$. Here, p(k,n) is a low-degree polynomial in k and in n.

By multiplying $\begin{pmatrix} V \\ W \end{pmatrix}$ by the (relatively) quickly-invertible A^{-1} , we may find an approximation to the original matrix A with the desired numerical rank bound. A set of algorithms that compute such a factorization, with certain desirable stability guarantees, as well as the precise meaning of "sub-cubic" and "low-degree" above are given in [14]. Another algebraic low-rank approximation method is the so-called "skeletonization" method of [15]. Here, we pick $k \ll n$ columns of A, C, and $l \ll n$ rows of A, R, such that

$$A \approx CUR$$

U is computed to minimize the error $||A - CUR||_2$ given C and R. The exact identities of the rows and columns to be picked may be difficult to identify, but it is important to find ones such that U is well-conditioned; see [15]. It is a combination of this method and the QR factorization mentioned above that we will employ in Chapters 2 and 3.

1.1.2 Analytic low-rank compression techniques

Analytic low-rank compression, on the other hand, assumes additional structure on the entries of A. Hence, analytic low-rank techniques are less widely applicable, but "instantly" yield an approximation $A \approx \Phi K$ for Φ and K of rank equal to the numerical rank of A (with respect to some tolerance). Often, as in the subject of this work, we assume that A is a kernel matrix with analytic structure. For concreteness, recalling the definition given above, we have

$$A = k(X, Y) = \begin{pmatrix} k(x_1, y_1) & k(x_1, y_2) & \dots & k(x_1, y_m) \\ k(x_2, y_1) & k(x_2, y_2) & \dots & k(x_2, y_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_l, y_1) & k(x_l, y_2) & \dots & k(x_l, y_m) \end{pmatrix}$$
(1.2)

for some $k : \mathbb{C}^2 \to \mathbb{C}$ with some regularity properties in each variable and $x_i, y_i \in \mathbb{C}$ for $1 \leq i \leq n$. (We focus on the 1-dimensional case in our brief outline of analytic methods here, but such ideas often generalize to the multidimensional case.) In particular, one way of compressing A for some suitably differentiable k is by using Taylor expansions: fixing, say, x_i , we have

$$k(x_{i}, y) \approx \tilde{k}(x_{i}, y) = \sum_{k=1}^{N} c_{x_{i},k} y^{k}$$

$$(1.3)$$

for some Taylor coefficients $c_{x_i,k}$. We could have, of course, instead taken the expansion in the first variable x instead of the second variable y. From (1.3), we then have the rank-k approximation

$$A \approx \begin{pmatrix} c_{x_{1},1} & c_{x_{1},2} & \cdots & c_{x_{1},k} \\ c_{x_{2},1} & c_{x_{2},2} & \cdots & c_{x_{2},k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{x_{l},1} & c_{x_{l},2} & \cdots & c_{x_{l},k} \end{pmatrix} \begin{pmatrix} y_{1} & y_{2} & \cdots & y_{m} \\ y_{1}^{2} & y_{2}^{2} & \cdots & y_{m}^{2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1}^{k} & y_{2}^{k} & \cdots & y_{m}^{k} \end{pmatrix}$$

In this work, however, we focus on another analytic low-rank approximation method called the proxy point method. The proxy point method is similar to the above, except for instead of using (1.3), we take an approximation of the form

$$k(x_{\mathbf{i}}, y_{\mathbf{j}}) \approx \tilde{k}(x_{\mathbf{i}}, y_{\mathbf{j}}) = \sum_{k=1}^{N} \Phi(x_{\mathbf{i}}, z_{k}) k(z_{k}, y_{\mathbf{j}})$$
(1.4)

for some function $\Phi : \mathbb{C}^2 \to \mathbb{C}$ related to k. The points $z_k \in \mathbb{C}$ for k = 1, ..., N are picked depending on the properties of k, X, and Y; for details, see [16]–[20]. This gives the approximation

$$A \approx \begin{pmatrix} \Phi(x_1, z_1) & \Phi(x_1, z_2) & \cdots & \Phi(x_1, z_k) \\ \Phi(x_2, z_1) & \Phi(x_2, z_2) & \cdots & \Phi(x_2, z_k) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(x_l, z_1) & \Phi(x_l, z_2) & \cdots & \Phi(x_l, z_k) \end{pmatrix} \begin{pmatrix} k(z_1, y_1) & k(z_1, y_2) & \cdots & k(z_1, y_m) \\ k(z_2, y_1) & k(z_2, y_2) & \cdots & k(z_2, y_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(z_k, y_1) & k(z_k, y_2) & \cdots & k(z_k, y_m) \end{pmatrix}$$

Oftentimes, the relationship (1.4) comes from the discretization of some integral representation of $k(x_i, y_j)$, so Φ encodes some kind of "Green's function" and quadrature weights. We give our precise context in Section 3.1. In the literature, various such integral representations are used implicitly or explicitly, see for example [21].

The convergence properties of these analytic low-rank approximations via Taylor series or proxy points depends on the analytic properties of the function k and the sets X, Y defining A as in (1.2). There have been multiple approaches to studying the error involved, of which we will recount the two most relevant analyses for the proxy point method. These have concentrated on analytic k and "well-separated" sets X and $Y: X, Y \subseteq \mathbb{C}$ are assumed to be such that there exist $c \in \mathbb{C}$ and $\gamma > 0$ such that $X \subseteq B(c, \gamma)$ and $\overline{B(c, \gamma)} \cap Y = \emptyset$.

The first proxy point analysis we wish to mention is given in [18] for $k = 1/(x - y)^d$; we restate a part of the result they obtain for d = 1. (A similar bound is shown in [18] for $k = 1/(x - y)^d$.)

Proposition 1.1.3. Let $X, Y \subseteq \mathbb{C}$ be finite and well-separated; let $r = \max_{x \in X} |c - x|$; and let $R = \min_{y \in Y} |c - y|$. Then, using the approximation (1.4) with Φ resulting from the discretization of the Cauchy integral formula for $f(z, y_j)$ for each j along $\partial B(c, \gamma)$ using the trapezoidal quadrature rule and N uniformly-selected points, we have

$$\left|\frac{k(x_{\rm i}, y_{\rm j})}{\tilde{k}(x_{\rm i}, y_{\rm j})}\right| - 1 \le \frac{1}{1 - (R/\gamma)^N} + \frac{1}{1 - (\gamma/r)^N}$$

Hence, to minimize the error given X and Y, we need to pick c to maximize R/r, and then to subsequently minimize γ by picking $\gamma = \sqrt{Rr}$. We will explore the geometric question of picking c and γ in our specific context in Section 2.1. Note that this bound explicitly shows the relationship between the number of proxy points N and the entrywise error.

In Section 3.1, we will also give a new bound that similarly depends on the separation ratio R/r in Proposition 1.1.3, and similarly shows an explicit relationship between N and the entrywise error, but applies to general analytic k. Furthermore, in our proof, we will use facts about Laurent series that lend themselves to greater generalization; we will consider such a generalization in Chapter 4. The analysis in [18] given for Proposition 1.1.3, on the other hand, used techniques that only applied to the specific kernel $k = 1/(x-y)^d$. However, it should be noted that the bound we will give in Section 3.1 is absolute, whereas this bound is relative.

We also pause to note that, in the context of analytic k and well-separated X, Y, the left factor of (1.4) remains unchanged for different k. This "kernel-free" property of the proxy point approximation will be useful for our HSS construction methods given in Chapters 3 and 4.

The second existing proxy point approximation error analysis we wish to mention is given in the "Cauchy FMM" scheme [22]. There, it is simply argued that, for analytic k in (1.4), $|k(x_i, y_j) - \tilde{k}(x_i, y_j)|$ is bounded by Ae^{-BN} for some A, B > 0. So, while elementwise exponential convergence is argued for several different kinds of contours resulting from the Cauchy integral representation (and not just for circles as in Proposition 1.1.3 and Section 3.1), the exact number of proxy points N is not related to the geometry of X and Y in a concrete way. Therefore, the necessity of our new analysis should already be clear, and it should hopefully become even clearer in Chapters 3 and 4.

1.2 Review of HSS matrix approximation

In this section, we review the data structure known as a hierarchically semiseparable (HSS) matrix form. Here we only give a brief outline necessary to understand the algorithms given in the subsequent chapters; more details can be found in [10].

Definition 1.2.1. Let A be a matrix. Assume without loss of generality that A is square with row/column size n equal to a power of two, and let $L < \log_2(n)$. Recursively partition in two the set of row/column indices of A for a total of $2^L - 1$ subsets. Specifically, for each $0 \le l \le L$, partition [1, n] into the 2^l sets

$$\mathcal{I}_{l} = \left\{ \left[1, \frac{n}{2^{l}}\right], \left[\frac{n}{2^{l}} + 1, \frac{n}{2^{l-1}}\right], \dots, \left[(2^{l} - 1)\frac{n}{2^{l}} + 1, n\right] \right\}.$$

Let $\mathcal{I} = \bigcup_{j=0}^{L} \mathcal{I}_l$, and impose a partial order on \mathcal{I} by set inclusion. We call \mathcal{I} the L-level HSS index set of A. Then its Hasse diagram \mathcal{T} is a perfect binary tree, called the HSS tree of A. Now, for each $1 \leq j \leq 2^L - 1$, define $\mathbf{i}_j \in \mathcal{I}$ to be the element corresponding to the jth vertex of \mathcal{T} in its postordered traversal. For each $1 \leq j \leq 2^L - 1$, define $A_j^- = A_{\mathbf{i}_j,[1,n]\setminus \mathbf{i}_j}$ and $A_j^{|} = A_{[1,n]\setminus \mathbf{i}_j,\mathbf{i}_j}$; these are called the jth HSS block row and jth HSS block column, respectively. (See Figure 1.1 for an example when L = 2.) The HSS rank of A is the maximum rank, over all $1 \leq j \leq 2^L - 1$, of A_j^- and $A_j^{|}$.

An L-level HSS form for A is a 6-tuple $\{\mathbf{D}, \mathbf{U}, \mathbf{R}, \mathbf{V}, \mathbf{W}, \mathbf{B}\}$, where:

- $\mathbf{U} = \{U_j\}_{1 \le j \le 2^L 2}, \ \mathbf{V} = \{V_j\}_{1 \le j \le 2^L 2}, \ and \ \mathbf{B} = \{B_j\}_{1 \le j \le 2^L 2} \ are \ sets \ of \ matrices;$
- D = {D_j}_{j∈I} is a set of matrices, where I is the set of postordered indices of leaves of *T*;
- and R = {R_j}_{j∈J} and W = {W_j}_{j∈J} are sets of matrices, where J is the set of postordered indices of vertices of T of depth at least two;

such that

1.
$$D_j = A_{\mathbf{i}_j, \mathbf{i}_j}$$
 for $j \in \mathbf{I}$;



(a) HSS block rows (b) HSS block columns

Figure 1.1. The HSS block rows and columns of A where L = 2. The labeled green blocks with rounded corners correspond to the HSS tree depth l = 1; the labeled yellow blocks with sharp corners correspond to the HSS tree depth l = 2.

- 2. $A_{\mathbf{i}_{j},\mathbf{i}_{sib(j)}} = U_{j}B_{j}V_{sib(j)}^{T}$ for $1 \leq j \leq 2^{L} 2$, where B_{j} is full-rank and sib(j) is the postordered index of the sibling of j;
- 3. and $U_{j} = \begin{pmatrix} U_{c_{1}(j)}R_{c_{1}(j)} \\ U_{c_{2}(j)}R_{c_{2}(j)} \end{pmatrix}$ and $V_{j} = \begin{pmatrix} V_{c_{1}(j)}W_{c_{1}(j)} \\ V_{c_{2}(j)}W_{c_{2}(j)} \end{pmatrix}$ for $1 \leq j \leq 2^{L} 2$, where $c_{1}(j)$ and $c_{2}(j)$ denote the postordered indices of the left and right children of the postordered jth vertex of \mathcal{T} , respectively.

Collectively, all of the matrices mentioned in this definition are called *HSS generators* of A. Note that we can find generators whose sizes can all be bounded by the HSS rank of A [10]; this is the main point constructing the HSS form of A and the reason for the efficiency of HSS algorithms. Figure 1.2 illustrates the various relationships of the HSS generators of A.

Finally, we say A has numerical HSS rank k with respect to a tolerance τ if the numerical rank of A_j^- and A_j^{\dagger} with respect to a tolerance τ is at most k over all $1 \leq j \leq 2^L - 1$. We define an *L*-level rank-k HSS approximation of A to be an *L*-level HSS form of A where we replace condition 2 in Definition 1.2.1 above with the following:

2'
$$A_{\mathbf{i}_j,\mathbf{i}_{\mathrm{sib}(j)}} \approx U_j B_j V_{\mathrm{sib}(j)}^T$$
 for $1 \leq j \leq 2^L - 2$, where B_j is a $k \times k$ matrix

| D_1 | $U_1B_1V_2^T$ | $-U_3B_3V_6^T$ | | D_1 | $U_1B_1V_2^T$ | $U_1 R_1 B_3 W_4^T V_4^T$ | $U_1 R_1 B_3 W_5^T V_5^T$ |
|-----------------|---------------|-----------------|---------------|---------------------------|---------------------------|---------------------------|---------------------------|
| $U_2 B_1 V_1^T$ | D_2 | | | $U_2 B_2 V_1^T$ | D_2 | $U_2 R_2 B_3 W_4^T V_4^T$ | $U_2 R_2 B_3 W_5^T V_5^T$ |
| U.B | $U_0 B_0 V^T$ | | $U_4B_4V_5^T$ | $U_4 R_4 B_6 W_1^T V_1^T$ | $U_4 R_4 B_6 W_2^T V_2^T$ | D_4 | $U_4 B_4 V_5^T$ |
| $U_6 D_6 V_3$ | | $U_5 B_5 V_4^T$ | D_5 | $U_5 R_5 B_6 W_1^T V_1^T$ | $U_5R_5B_6W_2^TV_2^T$ | $U_5B_5V_4^T$ | D_5 |

Figure 1.2. The HSS generator products of A placed into the blocks of A that they generate.

In the case that A is a Toeplitz matrix, for $1 \leq j \leq 2^L - 2$, existing methods of constructing any one of U_j , V_j , R_j , W_j , or B_j in general scale linearly in n, for at least some j.[23] In the next section, we outline an algorithm to construct any such generator with sublinear cost. This is useful depending on how the HSS form of A is subsequently used. For example, our method confers a speedup if only part of the output of a matrix-vector multiplication with A is needed.

2. HYPERFAST HSS CONSTRUCTION OF SOME CAUCHY MATRICES

Cauchy matrices—that is, kernel matrices with Cauchy kernel

$$k(x,y) = \frac{1}{x-y}.$$
 (2.1)

for values $x, y \in \mathbb{C}$ —frequently arise in some numerical computations, such as solutions of differential equations and integral equations, solutions of Toeplitz/Hankel/Vandermonde systems, kernel methods, and electrostatic potential evaluations. We consider some Cauchy matrices defined by the evaluation of (2.1) at uniform points $x_i, y_j \in \mathbb{C}$, i, j = 1, 2, ..., n on a circle. Specifically, the case with the following points plays an important role in Toeplitz matrix computations:

$$X \equiv \{x_i\}_{i=1:n}$$
 with $x_i = \omega^{2i-2}$, $Y \equiv \{y_j\}_{j=1:n}$ with $y_j = \omega^{2j-1}$, (2.2)

where $\omega = e^{\frac{\pi i}{n}}$, $\mathbf{i} = \sqrt{-1}$, and the Matlab notation 1 : *n* means the natural numbers 1, 2, ..., *n*. In fact, a Toeplitz matrix in Fourier space or in conjunction with displacement structures (see, e.g., [24]–[27]) is related to the Cauchy matrix

$$C = k(X, Y) \equiv (k(x_{i}, y_{j}))_{n \times n} = \left(\frac{1}{\omega^{2i-2} - \omega^{2j-1}}\right)_{n \times n},$$
(2.3)

where $(k(x_i, y_j))_{n \times n}$ similarly denotes the $n \times n$ matrix with the (i, j) entry $k(x_i, y_j)$.

To be specific, after multiplying by the $n \times n$ discrete Fourier transform (DFT) matrix, a Toeplitz system Tx = b can be turned into an equivalent system

$$\mathcal{C}\xi=\beta,$$

where $C = \mathcal{F}_n T D \mathcal{F}_n^H$, \mathcal{F}_n is an $n \times n$ DFT matrix, and D is a certain diagonal matrix. It turns out that C satisfies what is known as a "displacement equation":

$$A\mathcal{C} - \mathcal{C}B = UV, \tag{2.4}$$

where $U \in \mathbb{C}_{n \times 2}$ and $U \in \mathbb{C}_{2 \times n}$. (In general, $A, B \in \mathbb{C}_{n \times n}$ are not assumed a priori to take any particular structure, but in our case they are diagonal matrices with entries on the unit circle.) This implies that the matrix \mathcal{C} is Cauchy-like; that is, it has entries

$$C_{i,j} = \frac{G|_{\{i\}}H|_{\{:\}\times\{j\}}}{x_i - y_j},$$
(2.5)

where $U \in \mathbb{C}_{n \times 2}$, $U \in \mathbb{C}_{2 \times n}$, and x_i, y_j are as defined above. Displacement structures and their applications have been studied at length in the literature; for more details, see e.g. [24] and [9]. More recently, analytic properties of displacement structures, including those of Cauchy matrices, have been studied in [28].

For matrices satisfying displacement structure, a Gaussian elimination with partial pivoting algorithm may be designed to take asymptotically less time than $O(n^3)$ [24]. In fact, for Toeplitz systems, the particular displacement structure in (2.4) ensures a time complexity of $O(n^2)$. More recently, rank-structured Toeplitz methods, ensuring a sub-quadratic (so-called "superfast") time complexity, have proliferated over the past twenty years [9], [23], [29], [30]. In particular, such Toeplitz methods ensure greater control over the numerical stability in system solution and other operations [23], [30]. A key component of such methods is the efficient low-rank approximation to C, such as the randomized methods explored for the task in [30]. These methods use HSS approximations of C to construct HSS approximations to a Cauchy-like matrix corresponding to the Toeplitz matrix to be solved in Fourier space. Once such HSS approximations are obtained, their factorization can be quickly performed. The approximations can also be used for other computations related to Toeplitz matrices such as least squares solution [23] and eigenvalue solution [13], [31], [32].

Thus, the quick approximation of C by an HSS form is very useful for computations with Toeplitz matrices. In this chapter, we focus on ways to try and quickly find HSS approximations to C. As usual, the most expensive operation in constructing this HSS form is the low-rank approximation or compression of relevant off-diagonal blocks. In previous work, this has been done by either the direct algebraic rank-revealing factorizations mentioned in Section 1.1, as in [9], [29], or randomized compression, as in [23], [33]. The former has a total cost of about $O(n^2)$ flops or more, and the latter costs about O(n) flops. Such costs are needed to compress the off-diagonal block rows as well as columns at each level. Typically in hierarchical partitioning with L levels, there are $O(\frac{n}{2^l})$ such blocks at level $l = 0, 1, \ldots, L$, with level L corresponding to the finest level.

However, C in (2.3) is highly structured and only depends on n. It has previously been mentioned in [9], [33] that selected subblocks of C are closely related. Here, we would like to explore more connections among the off-diagonal blocks of C in order to extensively reuse computations in all compression steps. That is, we intend to reuse compression information across all the off-diagonal block rows at each hierarchical level of the HSS approximation, and further share information between off-diagonal block rows and columns. Accordingly, we show that we only need to approximate a single off-diagonal block row at each level, which is sufficient to produce a basis matrix for all the other off-diagonal block rows and columns at the same level. Consequently, the strategy significantly reduces the number of compression steps, from O(n) to $O(\log n)$.

Furthermore, we split the off-diagonal compression step into a near-field part and a farfield part. To avoid expensive rank-revealing factorizations, we approximate the far-field block using the proxy point method, and then perform an algebraic compression. Hence, such a method is sometimes called a "hybrid" compression method. The feasibility of the proxy point method for the far-field compression of the kernel matrix in question has been justified in [18]. In our case, for the kernel function (2.1), we will provide an optimal choice of a key parameter involved in the proxy point approximation. The optimal choice makes sure that the resulting low-rank approximation error is as small as possible.

We then give a hierarchical compression scheme to construct an HSS approximation to C. The scheme ensures that the far-field part of an off-diagonal block row at each hierarchical level meets two efficiency requirements. First, it satisfies a separation condition so that the far-field part is numerically low rank. Second, a near-field block row only involves a small number of rows, so that, combined with the above observation, a compact numerical basis matrix for the off-diagonal block row can be quickly found algebraically. Then, the aforementioned basis reuse idea and the proxy point method are integrated into this scheme to reach sublinear complexity for the HSS construction. Specifically, we can approximate Cby an HSS form in $O(r^3 \log^2 n)$ flops and with $O(r^2 \log^2 n)$ storage, where r is a constant rank bound for far-field low-rank approximation. This is a significant reduction from the previous roughly $O(n^2)$ or O(n) costs. Thus, we say that the approximation method is *hyperfast*. An algorithm is given and the sublinear complexity is confirmed numerically.

This chapter is structured as follows: in Section 2.1, we detail the application of the proxy point method to the far-field compression for the off-diagonal blocks of C. Section 2.2 reveals some intrinsic structural relationships among the off-diagonal blocks of C. The hierarchical compression scheme is then presented in Section 2.3. Lastly, the algorithm, its analysis, and some tests are given in Section 2.4.

The work presented in this chapter will appear in [34].

2.1 Analytic far-field compression by the proxy point method

In this section, we consider the low-rank approximation of the following subblock of C:

$$K = k(\mathbf{s}, \mathbf{t}) \equiv \left(k(x_{\mathbf{i}}, y_{\mathbf{j}})\right)_{x_{\mathbf{i}} \in \mathbf{s}, y_{\mathbf{j}} \in \mathbf{t}},\tag{2.6}$$

where k is given in (2.1), and

$$\mathbf{s} = \{\omega^{2(i-1)}, \ 1 \le i \le k\} \subset X, \quad \mathbf{t} = \{\omega^{2j-1}, \ k+s < j \le n-s-1\} \subset Y, \tag{2.7}$$
$$1 \le k < \frac{n}{2}, \quad 1 \le s < \frac{n-k}{2}.$$

Here, s is the number of points on each side of s that separate s from t. An illustration of the sets $X, Y, \mathbf{s}, \mathbf{t}$ is given in Figure 3.2. The reason why such sets are considered will become clear later in this section. For convenience, s and t are sometimes referred to as the source and target sets, respectively, and K is the interaction between s and t.



Figure 2.1. Illustration of the sets under consideration, where each cross (\times) is a point in X, each box (\Box) is a point in Y, $\mathbf{s} \subset X$ is the source set, $\mathbf{t} \subset Y$ is the target set, and \mathbf{z} is a set of proxy points.

The two sets \mathbf{s} and \mathbf{t} are well-separated in the sense that there exists a point $z \in \mathbb{C}$ such that for every $x \in \mathbf{s}$ and $y \in \mathbf{t}$,

$$\frac{\max_{x \in \mathbf{s}} |z - x|}{\min_{y \in \mathbf{t}} |z - y|} \le \delta < 1,$$
(2.8)

where δ is a constant often referred to as the separation ratio. This concept of separation is a basic tool in the fast multipole method [35] and also hierarchical matrix methods [36]. Using the analytic approximation bounds discussed in this work, it can be used to show that K is numerically low-rank.

We seek to quickly write a (column) basis matrix G in a low-rank approximation to K. This is an essential component in the HSS approximation of C in the next section. For this purpose, we use the proxy point method in [18], and which we will review and analyze more generally in Section 3.1, to directly produce G as follows:

$$K \approx GH^T$$
, with $G = k(\mathbf{s}, \mathbf{z}),$ (2.9)

where \mathbf{z} is a set of points located on a contour (called a proxy surface) that separates \mathbf{s} from \mathbf{t} in \mathbb{C} . See Figure 3.2(b) for an illustration. With G in (2.9), H is then computed appropriately; see [18] for more details. In particular, recall Proposition 1.1.3 (shown in [18]): if \mathbf{s} is located inside a circle with radius γ_1 and center z and \mathbf{t} is outside a circle with

radius $\gamma_2 > \gamma_1$ and the same center z, then a nearly optimal choice of proxy points can be obtained by choosing some quadrature points on a circle with radius $\gamma = \sqrt{\gamma_1 \gamma_2}$. With such a choice, the approximation error satisfies

$$\frac{\|K - GH^T\|_F}{\|K\|_F} \le \frac{2}{(\frac{\gamma_2}{\gamma_1})^N - 1} = O\left(\left(\frac{\gamma_1}{\gamma_2}\right)^N\right),\tag{2.10}$$

where N is the number of proxy points chosen.

It is important to note that this bound gives a relative bound for the proxy point error of the appropriate matrix subblock. However, as we mentioned before, some techniques used in obtaining this bound are specific to the Cauchy kernel, and so may not be readily generalized to other kernels. In Section 3.1, we give a more general bound for the proxy point error of a general complex-analytic kernel matrix, but our new error bound is absolute. Hence, we use the relative bound given in Proposition 1.1.3 and (2.10) for greater error control in this specific instance.

If we take $\gamma_1 = \max_{x \in \mathbf{s}} |z - x|$ and $\gamma_2 = \min_{y \in \mathbf{t}} |z - y|$, then (2.8) is just $\frac{\gamma_1}{\gamma_2} \leq \delta$ so that

$$\frac{\|K - GH^T\|_F}{\|K\|_F} = O(\delta^r).$$
(2.11)

Thus, the approximation error of the proxy point method decreases as $\frac{\gamma_1}{\gamma_2}$ get smaller. We then try to minimize $\frac{\gamma_1}{\gamma_2}$ (by moving the center point z in (2.8)). It is not immediately clear which center z gives such a minimum. To this end, we have the following proposition.

Proposition 2.1.1. Consider **s** and **t** as in (2.7). Let $g : \mathbb{C} \to \mathbb{R}$ be the function defined by

$$g(z) = \frac{\gamma_1(z)}{\gamma_2(z)}, \quad with \quad \gamma_1(z) = \max_{x \in \mathbf{s}} |z - x|, \quad \gamma_2(z) = \min_{y \in \mathbf{t}} |z - y|.$$

Then

$$\arg\min(g) = \omega^{k-1},$$
$$\min_{z \in \mathbb{C}} g(z) = \sin \frac{\pi(k-1)}{2n} / \sin \frac{\pi(k+2(s+1))}{2n} < \cos \frac{(2s+3)\pi}{2n} < 1.$$
(2.12)

Proof. Since g(0) = 1 and $g(\omega^{k-1}) < 1$, we have $\arg\min(g) \neq 0$. Let the map $\mathcal{A} : \mathbb{C} \to [0, n)$ be defined by

$$\mathcal{A}(z) = \frac{\arg(z)}{2\pi}n,\tag{2.13}$$

which can be thought of as the argument of z scaled to the interval [0, n). Let

$$\Theta_{1} = \left\{ z \mid \frac{k-1}{2} \leq \mathcal{A}(z) < \frac{n}{2}, \ z \neq 0 \right\},\$$

$$\Theta_{2} = \left\{ z \mid 0 \leq \mathcal{A}(z) < \frac{k-1}{2} \quad \text{or} \quad k-1+\frac{n}{2} \leq \mathcal{A}(z) < n, \ z \neq 0 \right\},\$$

$$\Omega_{1} = \left\{ z \mid \frac{k-1}{2} \leq \mathcal{A}(z) < k+s, \ z \neq 0 \right\},\$$

$$\Omega_{2} = \left\{ z \mid 0 \leq \mathcal{A}(z) < \frac{k-1}{2} \quad \text{or} \quad n-s-1 \leq \mathcal{A}(z) < n, \ z \neq 0 \right\}.$$

Figure 2.2 illustrates these regions. Note that

$$\gamma_{1}(z) = \begin{cases} |1-z|, & z \in \Theta_{1}, \\ |\omega^{2(k-1)}-z|, & z \in \Theta_{2}, \\ |\omega^{2k_{z}}-z|, & \text{otherwise,} \end{cases} \quad \gamma_{2}(z) = \begin{cases} |\omega^{2(k+s)+1}-z|, & z \in \Omega_{1}, \\ |\omega^{2(n-s-1)-1}-z|, & z \in \Omega_{2}, \\ |\omega^{2j_{z}}-z|, & \text{otherwise,} \end{cases}$$
(2.14)

where $k_z = \lfloor \mathcal{A}(z) - \frac{n}{2} \rfloor$ and $j_z = \lfloor \mathcal{A}(z) - \frac{1}{2} \rfloor + \frac{1}{2}$ (with tie-breaking by rounding up). This can be conveniently understood with the aid of Figure 2.2. (On Θ_1 and Θ_2 , $\gamma_1(z)$ returns the distance between the input z and the red and blue crosses, respectively. On Ω_1 and Ω_2 , $\gamma_2(z)$ returns the distance between z and the red and blue squares, respectively.)

In particular, we see by the above that g(z) > 1 for all $z \in \mathbb{C} \setminus \{\Omega_1 \cup \Omega_2 \cup \{0\}\}$. Thus,

$$\arg\min(g)\in\Omega_1\cup\Omega_2.$$

Let $z_0 \in \Omega_1$. If we write $z_0 = \rho \omega^{k-1} e^{i\theta}$ for some $r, \theta \in \mathbb{R}$ with $\rho \ge 0$, we have $\tilde{z}_0 = r \omega^{k-1} e^{-i\theta} \in \Omega_2 \cup \left\{ z \mid a(z) = \frac{k-1}{2}, \ z \ne 0 \right\}$. Then

$$g\left(z_0\right) = g\left(\tilde{z}_0\right).$$



Figure 2.2. Illustration of the regions of definition for the piecewise-defined functions γ_1 and γ_2 , along with points in the sets \mathbf{s} ($_{\times}$) and \mathbf{t} ($_{\Box}$).

See Figure 2.3(a), where $g(z_0)$ is the length of the solid blue line divided by the length of the solid red line and $g(\tilde{z}_0)$ is the length of the dashed blue line divided by the length of the dashed red line. Hence,

$$\min_{z\in\Omega_1\cup\Omega_2}g(z)=\min_{z\in\Omega_1}g(z).$$



Figure 2.3. Illustration of the points $z_0 = \rho \omega^{k-1} e^{i\theta}$, $\tilde{z}_0 = \rho \omega^{k-1} e^{-i\theta}$, and $\hat{z}_0 = \rho \omega^{k-1}$, along with points in the sets \mathbf{s} (x) and \mathbf{t} (\Box), where the dotted black line illustrates the relevant symmetry.

Next, let $\hat{z}_0 = \rho \omega^{k-1}$. Since $\gamma_1(z_0) \leq \gamma_1(\hat{z}_0), \gamma_2(z_0) \geq \gamma_2(\hat{z}_0)$, we have $g(z) \leq g(\hat{z}_0)$. See Figure 2.3(b), where $g(\hat{z}_0)$ is the length of the dashed red line divided by the length of the dashed blue line. Thus,

$$\arg\min(g) \in \left\{ z \mid \mathcal{A}(z) = \frac{k-1}{2}, \ z \neq 0 \right\}.$$

Then from (2.14), the positive minima of g occur at the positive minima of the following function:

$$h(\rho) = \frac{\left|1 - \rho \omega^{k-1}\right|}{\left|\omega^{2(k+s)+1} - \rho \omega^{k-1}\right|} = \frac{\left|1 - \rho e^{\beta \mathbf{i}}\right|}{\left|e^{\alpha \mathbf{i}} - \rho e^{\beta \mathbf{i}}\right|},$$

where $\alpha = \frac{2\pi(k+s)+\pi}{n}$ and $\beta = \frac{\pi(k-1)}{n}$. Since $h(\rho)$ is positive, we can just consider

$$h^{2}(\rho) = \frac{\left(\rho \sin \beta\right)^{2} + \left(1 - \rho \cos \beta\right)^{2}}{\left(\sin \alpha - \rho \sin \beta\right)^{2} + \left(\cos \alpha - \rho \cos \beta\right)^{2}}.$$

Since the denominator above does not vanish, by the quotient rule, the positive minima of $h^2(\rho)$ occur at the positive zeros of the function

$$\tilde{h}(\rho) = [2\rho \sin^2 \beta - 2(1 - \rho \cos \beta) \cos \beta]$$

$$\cdot [(\sin \alpha - \rho \sin \beta)^2 + (\cos \alpha - \rho \cos \beta)^2]$$

$$- [-2(\sin \alpha - h \sin \beta) \sin \beta) - 2(\cos \alpha - h \cos \beta) \cos \beta]$$

$$\cdot [\rho^2 \sin^2 \beta + (1 - \rho \cos \beta)^2]$$

$$= 2(\rho^2 - 1)(\cos \beta - \cos (\alpha - \beta)).$$

The only such zero is at $\rho = 1$, so this is the only positive minimum argument of $h^2(\rho)$. Thus, $\arg\min(g) = e^{\frac{\pi i(k-1)}{n}} = \omega^{k-1}$. With the choice of the center $z = \omega^{k-1}$, we have

$$\gamma_1 = \left| 1 - \omega^{k-1} \right| = \sqrt{2(1 - \cos \beta)} = 2 \sin \frac{(k-1)\pi}{2n},$$

$$\gamma_2 = \left| \omega^{2(k+s)+1} - \omega^{k-1} \right| = \sqrt{2(1 - \cos(\alpha - \beta))} = 2 \sin \frac{(k+2(s+1))\pi}{2n}$$

Then $\min_{z \in \mathbb{C}} g(z)$ in (2.12) is obtained.

Finally, by the definition of the sets **s** and **t**, we have $k < \frac{n}{2}$ and $s + 1 \le \frac{n-k}{2}$. Then

$$\frac{(k-1)\pi}{2n} \in \left[0, \frac{\pi}{4}\right), \quad \frac{(k+2(s+1))\pi}{2n} \in \left(0, \frac{\pi}{2}\right).$$

Also, note that $\frac{(k+2(s+1))\pi}{2n} - \frac{(k-1)\pi}{2n} = \frac{(2s+3)\pi}{2n} \in (0, \frac{\pi}{2})$. Then

$$\sin \frac{(k-1)\pi}{2n} = \sin \frac{(k+2(s+1)-(2s+3))\pi}{2n}$$
$$< \sin \frac{(k+2(s+1))\pi}{2n} \cos \frac{(2s+3)\pi}{2n}$$

This yields the inequality in (2.12).

This result shows how to choose the optimal center z in (2.8) so as to make the error in (2.10) as small as possible. With the choice of the optimal z in the theorem, we can ensure that \mathbf{s} and \mathbf{t} in (2.7) are well-separated so as to obtain δ in (2.8) as a constant smaller than 1 and independent of n. More specifically, we have the following result.

Corollary 2.1.2. For **s** and **t** in (2.7), suppose $s = \frac{k}{2} = o(n)$. With the optimal z in Theorem 2.1.1, we have

$$\frac{\max_{x \in \mathbf{s}} |z - x|}{\min_{y \in \mathbf{t}} |z - y|} \sim \frac{1}{2} \quad as \quad n \to \infty.$$
(2.15)

Proof. Note that we have

$$\sin\frac{\pi(k-1)}{2n} / \sin\frac{\pi(k+2(s+1))}{2n} \sim \frac{k}{k+2s}$$

as $k \to \infty$, so the corollary follows by the previous proposition.

Note that the result can be made more general. For convenience, we introduce the following definition.

Definition 2.1.3. For subsets $\mathbf{s} \in X$ and $\mathbf{t} \in Y$, with the notation in (2.13), the argument span of \mathbf{s} and the argument gap between \mathbf{s} and \mathbf{t} are respectively,

$$\operatorname{span}_{\mathcal{A}}(\mathbf{s}) = \max_{x_1, x_2 \in \mathbf{s}} (|\mathcal{A}(x_1) - \mathcal{A}(x_2)| \mod n),$$
$$\operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t}) = \min_{x \in \mathbf{s}, y \in \mathbf{t}} (|\mathcal{A}(x) - \mathcal{A}(y)| \mod n).$$

In this definition, $\operatorname{span}_{\mathcal{A}}(\mathbf{s})$ and $\operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t})$ are given in terms of the scaled argument. For example, for \mathbf{s} and \mathbf{t} in (2.7),

$$\operatorname{span}_{\mathcal{A}}(\mathbf{s}) = k, \quad \operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t}) > s.$$

Thus, s controls the argument gap between s and t. In general, for subsets $s \in X$ and $t \in Y$, as long as

$$\operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t}) \ge \frac{1}{2} \operatorname{span}_{\mathcal{A}}(\mathbf{s}),$$
 (2.16)

then the separation ratio between \mathbf{s} and \mathbf{t} is $\frac{1}{2}$ or smaller with center z chosen as above. (With some technicalities, a precise statement can be made and is skipped here.) This definition will come in handy in the next section.

This discussion indicates that, as long as $\operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t})$ is large enough in comparison with $\operatorname{span}_{\mathcal{A}}(\mathbf{s})$, \mathbf{s} and \mathbf{t} would be well-separated. Then the interaction matrix between \mathbf{s} and \mathbf{t} is numerically low rank. Accordingly, with the proxy point method, we can use $r = O(|\log \tau|)$ proxy points to make the error in (2.11) bounded by any accuracy τ . In fact, for \mathbf{s} and \mathbf{t} in (2.7), following the study in [18], we select the proxy points to be the uniform quadrature points used in the trapezoidal rule on a circle with radius γ , where we take the nearly optimal choice $\gamma = \sqrt{\gamma_1 \gamma_2}$. Thus, we choose

$$\gamma = 2\sqrt{\sin\frac{(k-1)\pi}{2n}\sin\frac{(k+2(s+1))\pi}{2n}}.$$

2.2 Relationships among off-diagonal blocks of C

Another key idea in our hyperfast rank-structured approximation of C is to fully explore the algebraic relationships among the off-diagonal blocks of C. As in Section 1.2, we suppose n is a power of 2, so that C can be hierarchically partitioned up to $L = O(\log_2 n)$ times with uniform block sizes at each level, and let the HSS tree \mathcal{T} be defined as before. We explore the relationships among the off-diagonal block rows and columns at each depth of the tree so as to save low-rank compression cost in the rank-structured approximation. Let a node i of \mathcal{T} correspond to the *k*th block row at its depth *l*. That is, i corresponds to the subblock of *C* with index set

$$\mathbf{i}_{i} = \left\{ (k-1)\frac{n}{2^{l}} + 1, (k-1)\frac{n}{2^{l}} + 2, \dots, k\frac{n}{2^{l}} \right\}.$$
(2.17)

As before, the HSS block row and column corresponding to node i are, respectively,

$$C_{i}^{-} = C|_{i \times ([1,n] \setminus i)}, \quad C_{i}^{|} = C|_{([1,n] \setminus i) \times i}$$
 (2.18)

Suppose j is the node at depth l of \mathcal{T} that corresponds to the (k + 1)st block row at that depth, and so has row index set

$$\mathbf{i}_{j} = \left\{ k \frac{n}{2^{l}} + 1, k \frac{n}{2^{l}} + 2, \dots, (k+1) \frac{n}{2^{l}} \right\}.$$
(2.19)

In [9], it is pointed out that the subblock $C|_{\mathbf{i}_{i}\times\mathbf{i}_{j}}$ of C_{i}^{-} and the subblock $C|_{\mathbf{i}_{j}\times((k+1)\frac{n}{2l}+1:(k+2)\frac{n}{2l})}$ of C_{j}^{-} differ by just the constant scalar $\omega^{-\frac{n}{2l}}$. This follows directly from the explicit definition of C. Thus, if a low-rank approximation is obtained for $C|_{\mathbf{i}_{i}\times\mathbf{i}_{j}}$, it can be reused for $C|_{\mathbf{i}_{j}\times((k+1)\frac{n}{2l}+1:(k+2)\frac{n}{2l})}$, which is referred to as a shifting relation.

Here, we would like to systematically generalize this shifting relation and also give an intuitive justification. We first look at the block rows at the same level and then the block columns. The following lemma directly follows from the structure of C and provides a convenient tool to study the relationships among the HSS blocks of C.

Lemma 2.2.1. Let C_1 and C_2 be circulant matrices with the first row

$$\left(\begin{array}{ccc} \frac{1}{1-\omega} & \frac{1}{1-\omega^3} & \cdots & \frac{1}{1-\omega^{2n-1}} \end{array}\right), \quad \left(\begin{array}{cccc} \frac{1}{1-\omega} & \frac{1}{\omega^{2n-2}-\omega} & \cdots & \frac{1}{\omega^2-\omega} \end{array}\right),$$

respectively. Then

$$C = \Lambda \mathcal{C}_1 = \mathcal{C}_2 \Lambda,$$

where $\Lambda = \operatorname{diag}(1, \omega^2, \dots, \omega^{2(n-1)}).$

The relationship among the HSS block rows/columns at each level can be shown as follows.

Proposition 2.2.2. Let i and j be two nodes at the same level of T corresponding to HSS block rows C_i^- and C_j^- , respectively. Then there exists nonzero scalars μ_1 and μ_2 and permutation matrices P_1 and P_2 such that

$$C_{\rm i}^- = \mu_1 C_{\rm j}^- P_1, \quad C_{\rm i}^{|} = \mu_2 P_2 C_{\rm j}^{|}.$$
 (2.20)

Proof. For the first equality in (2.20), it suffices to show the result for the two nearby HSS blocks C_i^- and C_j^- corresponding to the row index sets \mathbf{i}_i in (2.17) and \mathbf{i}_j in (2.19), respectively. Once this is shown, the result then holds for all the HSS blocks at the same depth l. Following the notation in (2.17) and (2.19), we may consider the sets

$$[1,n] \setminus \mathbf{i}_{i} = \mathbf{J}_{i,1} \cup \mathbf{J}_{i,2} \cup \mathbf{J}_{i,3}, \quad [1,n] \setminus \mathbf{i}_{j} = \mathbf{J}_{j,1} \cup \mathbf{J}_{j,2} \cup \mathbf{J}_{j,3},$$

where

$$\begin{aligned} \mathbf{J}_{i,1} &= \left\{ 1: (k-1)\frac{n}{2^l} \right\}, \quad \mathbf{J}_{i,2} &= \left\{ k\frac{n}{2^l} + 1: n - \frac{n}{2^l} \right\}, \quad \mathbf{J}_{i,3} &= \left\{ n - \frac{n}{2^l} + 1: n \right\}, \\ \mathbf{J}_{j,1} &= \left\{ 1: \frac{n}{2^l} \right\}, \quad \mathbf{J}_{j,2} &= \left\{ \frac{n}{2^l} + 1: k\frac{n}{2^l} \right\}, \quad \mathbf{J}_{j,2} &= \left\{ k\frac{n}{2^l} + 1: n \right\}. \end{aligned}$$

The partitions are illustrated in Figure 2.4. According to Lemma 2.2.1,

$$C_{\mathbf{i}}^{-} = \begin{pmatrix} C|_{\mathbf{i}_{i} \times \mathbf{J}_{i,1}} & C|_{\mathbf{i}_{i} \times \mathbf{J}_{i,2}} & C|_{\mathbf{i}_{i} \times \mathbf{J}_{i,3}} \end{pmatrix}$$

$$= \Lambda|_{\mathbf{i}_{i} \times \mathbf{i}_{i}} \begin{pmatrix} C_{1}|_{\mathbf{i}_{i} \times \mathbf{J}_{i,1}} & C_{1}|_{\mathbf{i}_{i} \times \mathbf{J}_{i,2}} & C_{1}|_{\mathbf{i}_{i} \times \mathbf{J}_{i,3}} \end{pmatrix},$$

$$C_{\mathbf{j}}^{-} = \begin{pmatrix} C|_{\mathbf{i}_{j} \times \mathbf{J}_{j,1}} & C|_{\mathbf{i}_{j} \times \mathbf{J}_{j,2}} & C|_{\mathbf{i}_{j} \times \mathbf{J}_{j,3}} \end{pmatrix}$$

$$= \Lambda|_{\mathbf{i}_{j} \times \mathbf{i}_{j}} \begin{pmatrix} C_{1}|_{\mathbf{i}_{j} \times \mathbf{J}_{j,1}} & C_{1}|_{\mathbf{i}_{j} \times \mathbf{J}_{j,2}} & C_{1}|_{\mathbf{i}_{j} \times \mathbf{J}_{j,3}} \end{pmatrix}.$$

$$(2.21)$$



Figure 2.4. Partitioning of C_i^- and C_j^- .

It is clear that $\Lambda|_{\mathbf{i}_i \times \mathbf{i}_i} = \mu_1 \Lambda|_{\mathbf{i}_j \times \mathbf{i}_j}$ with $\mu_1 = \omega^{-n/2^{l-1}}$. Also, due to the circulant structure of \mathcal{C}_1 , we have

$$\mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,1}} = \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,2}}, \quad \mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,2}} = \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,3}}, \quad \mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,3}} = \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,1}}.$$

Thus, there exists a permutation matrix P_1 such that

$$\left(\begin{array}{ccc} \mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,1}} & \mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,2}} & \mathcal{C}_1|_{\mathbf{i}_i\times\mathbf{J}_{i,3}} \end{array}\right) = \left(\begin{array}{ccc} \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,1}} & \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,2}} & \mathcal{C}_1|_{\mathbf{i}_j\times\mathbf{J}_{j,3}} \end{array}\right) P_1.$$

(2.21)-(2.22) then lead to

$$C_{\mathbf{i}}^{-} = \mu_1 \Lambda |_{\mathbf{i}_{\mathbf{j}} \times \mathbf{i}_{\mathbf{j}}} \left(\begin{array}{cc} \mathcal{C}_1 |_{\mathbf{i}_{\mathbf{j}} \times \mathbf{J}_{\mathbf{j},1}} & \mathcal{C}_1 |_{\mathbf{i}_{\mathbf{j}} \times \mathbf{J}_{\mathbf{j},2}} & \mathcal{C}_1 |_{\mathbf{i}_{\mathbf{j}} \times \mathbf{J}_{\mathbf{j},3}} \end{array} \right) P_1 = \mu_1 C_{\mathbf{j}}^{-} P_1.$$

Next, for the second equality in (2.20), we may consider the HSS block rows of C^T . According to Lemma 2.2.1, $C^T = \Lambda C_2^T$. Since C_2^T is still a circulant matrix, the result can then be shown as above.

This proposition shows that the HSS block rows (columns) are related by scalar multiples and column (row) permutations. Moreover, we can further relate the HSS block rows to the HSS block columns. By Proposition 2.2.2, we only need to relate the topmost HSS block row and the leftmost HSS block column at a level l. For convenience, for an index set \mathbf{i}_i associated with a node $i \in \mathcal{T}$, we partition $[1, n] \setminus \mathbf{i}_i$ as

$$[1,n] \setminus \mathbf{i}_{\mathbf{i}} = \mathbf{i}_{\mathbf{i}}^{<} \cup \mathbf{i}_{\mathbf{i}}^{>},$$

where $\mathbf{i}_{i}^{<}$ and $\mathbf{i}_{i}^{>}$ include all indices of [1, n] smaller and larger than those in \mathbf{i}_{i} , respectively. Then for $C_{i}^{-} = C|_{\mathbf{i}_{i} \times ([1,n] \setminus \mathbf{i}_{i})} = \begin{pmatrix} C|_{\mathbf{i}_{i} \times \mathbf{i}_{i}^{<}} & C|_{\mathbf{i}_{i} \times \mathbf{i}_{i}^{>}} \end{pmatrix}$, let P be a permutation matrix such that

$$C_{\mathbf{i}}^{-}P = \left(\begin{array}{cc} C|_{\mathbf{i}_{\mathbf{i}}\times\mathbf{i}_{\mathbf{i}}^{>}} & C|_{\mathbf{i}_{\mathbf{i}}\times\mathbf{i}_{\mathbf{i}}^{<}} \end{array} \right).$$
(2.23)

Then we have the following proposition.

Proposition 2.2.3. Let i be a node at level l of \mathcal{T} corresponding to the HSS block row $C_i^$ and HSS block column C_i^{\mid} . Let m be the row size of C_i^{\mid} . Then there exists a nonzero constant μ such that

$$(P^T C_{\mathbf{i}}^{|})|_{\{2:m\}} = \mu \left((C_{\mathbf{i}}^- P)|_{\{:\} \times \{1:m-1\}} \right)^T,$$

where P is the permutation matrix in (2.23).

Proof. Let i_1 be the leftmost node at the same level l as i. According to Proposition 2.2.2, $C_{i_1}^-$ and C_i^- are related by a scalar multiple and a column permutation. In fact, it can be further shown that, with the permutation matrix P in (2.23), we have

$$C_{i_1}^- = \mu_1 C_i^- P, \quad C_{i_1}^{\dagger} = \mu_2 P^T C_i^{\dagger}, \tag{2.24}$$

where μ_1 and μ_2 are nonzero scalars.

Now for $1 \leq j < k \leq n$, since

$$C_{k,j} = \frac{1}{\omega^{2k-2} - \omega^{2j-1}} = \left(-\frac{1}{\omega}\right) \frac{1}{\omega^{2j-2} - \omega^{2k-3}} = -\frac{1}{\omega}C_{j,k-1},$$

we have

$$C_{i_1}^{\dagger}|_{\{2:m\}} = -\frac{1}{\omega} (C_{i_1}^{-}|_{\{:\}\times\{1:m-1\}})^T.$$
(2.25)

(See Figure 2.5 for an illustration.)

Then (2.24) and (2.25) together lead to our result.

The implication of Proposition 2.2.2 is that, once we obtain a low-rank approximation to one HSS block row (or column) at level l, we can reuse its column (row) basis matrix for all the other HSS block rows (or columns) at level l. Proposition 2.2.3 further means that, the


Figure 2.5. The relationship between $C_{i_1}^{|}|_{\{2:m\}}$ and $C_{i_1}^{-}|_{\{:\}\times\{1:m-1\}}$, when i_1 is the leftmost node at level l of \mathcal{T} .

HSS block column C_i^{\dagger} and row C_i^{-} are also closely related. With the exception of one row in C_i^{\dagger} and one column in C_i^{-} , the remaining subblocks can also share basis information. In the next subsection, we will take advantage of these relations to design our hyperfast HSS approximation for C.

2.3 Hyperfast analytic hierarchical compression scheme

In this section, we showcase our hyperfast HSS approximation of C using a combination of several techniques. Again, we partition C in the manner of Section 1.2. During the construction of the HSS approximation for C, each HSS block row and column is compressed so as to find the generators. In almost all existing HSS approximation methods, the HSS blocks are compressed individually, leading to the complexity of at least O(n). In the following subsections, we show how to construct an HSS approximation to C in sublinear complexity by exploring certain compression strategies for the off-diagonal blocks of C and using the algebraic structures of the previous section. The main tools are as follows:

- Extract a row basis matrix for one HSS block row at each hierarchical level via the analytical far-field compression
- Ensure the near-field block row size is small enough at each hierarchical level

• Share the row basis across the HSS block rows at the entire level and also extend to the HSS block columns

We recall our assumption on n that allows us to partition C hierarchically with L levels of block rows, so that the associated HSS tree \mathcal{T} is a perfect binary tree. Also, corresponding to a leaf node $i \in \mathcal{T}$, each HSS block row $C_i^- = k(X_i, Y_i)$ is the interaction between subsets $X_i \subset X$ and $Y_i \subset Y$ of sizes $\frac{n}{2^L}$ and $n - \frac{n}{2^L}$, respectively. For example, if i = 1 which corresponds to the topmost HSS block row at the leaf level, we have $X_i = \{\omega^{2k-2} \mid 1 \leq k \leq \frac{n}{2^L}\}$ and $Y_i = \{\omega^{2k-1} \mid \frac{n}{2^L} + 1 \leq k \leq n\}$.

2.3.1 Compression at the leaf level

For a node i at the leaf level or level L, first consider the case where i corresponds to the topmost HSS block row. (That is, consider i = 1.) Partition X_i into

$$X_{\mathbf{i}} = \mathbf{s}_{\mathbf{i}} \cup \bar{\mathbf{s}}_{\mathbf{i}} \quad \text{with}$$

$$\mathbf{s}_{\mathbf{i}} = \left\{ \omega^{2k-2} \mid \frac{n}{2^{L+2}} \le k < \frac{3n}{2^{L+2}} \right\}, \quad \bar{\mathbf{s}}_{\mathbf{i}} = X_{\mathbf{i}} \setminus \mathbf{s}_{\mathbf{i}}.$$

$$(2.26)$$

Since we are considering the interaction between X_i and Y_i , we may consider \mathbf{s}_i as the "farfield" subset of X_i and $\mathbf{\bar{s}}_i$ the "near-field" subset of X_i . See Figure 2.6(a) for an illustration and see Figure 2.7 for the partitioning of the corresponding HSS block C_i^- . Since it is the leaf level, the sizes of both \mathbf{s}_i and $\mathbf{\bar{s}}_i$ are $\frac{n}{2^{L+1}}$ and are typically set to be small multiples of a desired numerical rank r.

Note that the partitioning in (2.26) makes $\operatorname{span}_{\mathcal{A}}(\mathbf{s}_i)$ to be at most half of $\operatorname{span}_{\mathcal{A}}(X_i)$. $\bar{\mathbf{s}}_i$ has two pieces, each has argument span about 1/4 of $\operatorname{span}_{\mathcal{A}}(X_i)$. It can be verified that, if we set $\mathbf{s} = \bar{\mathbf{s}}_i$ and $\mathbf{t} = Y_i$ in Definition 2.1.3, then (2.16) holds and \mathbf{s} and \mathbf{t} are well-separated. Thus, the proxy point method analysis of Section 2.1 ensures a good approximation to the far-field interaction $C_{i,2}^- \equiv k(\mathbf{s}_i, Y_i)$. For convenience, let $P^{(L)}$ be a row permutation matrix such that

$$C_{\mathbf{i}}^{-} = P^{(L)} \begin{pmatrix} C_{\mathbf{i},1}^{-} \\ C_{\mathbf{i},2}^{-} \end{pmatrix} \equiv \begin{pmatrix} k(\bar{\mathbf{s}}_{\mathbf{i}}, Y_{\mathbf{i}}) \\ k(\mathbf{s}_{\mathbf{i}}, Y_{\mathbf{i}}) \end{pmatrix}.$$



Figure 2.6. Illustration of the sets under consideration, the sparsification of \mathbf{s}_i into $\mathbf{\tilde{s}}_i$ for a leaf i, and the resulting source point subsets of X after one level of compression.



Figure 2.7. Partitioning of the HSS block C_i^- corresponding to Figure 2.6(a).

Then, apply the proxy point method to $C_{i,2}^-$ with the set \mathbf{z}_i of $r = \mathcal{O}(|\log \tau|)$ proxy points obtain the following approximation with relative accuracy τ (see (2.9)):

$$C_{\mathbf{i},2}^- \approx G_{\mathbf{i}} H_{\mathbf{i}}^T$$
, with $G_{\mathbf{i}} = k(\mathbf{s}_{\mathbf{i}}, \mathbf{z}_{\mathbf{i}})$. (2.27)

Now, apply a strong rank-revealing factorization [14] to G_i to get a factorization of the form

$$G_{\mathbf{i}} \approx \Pi^{(L)} \begin{pmatrix} I \\ E^{(L)} \end{pmatrix} G_{\mathbf{i}}|_{\tilde{\mathbf{i}}_{\mathbf{i}}}, \qquad (2.28)$$

where $\Pi^{(L)}$ is a permutation matrix, $E^{(L)}$ has entries satisfying certain bounds as in [14] to ensure stability, and $K_i|_{\tilde{\mathbf{i}}_i}$ consists of r selected rows of K_i corresponding to a row index set $\tilde{\mathbf{i}}_i$. In theory, any rank-revealing QR factorization may suffice, but in practice the stability guarantees of the SRRQR factorization are important to maintain good accuracy with a large number of proxy points. This factorization is also called an interpolative decomposition [37] or structure preserving rank-revealing (SPRR) factorization [33]. As mentioned in [33], this results in the approximation

$$C_{\mathbf{i},2}^{-}(=k(\mathbf{s}_{\mathbf{i}},Y_{\mathbf{i}})) \approx \Pi^{(L)} \begin{pmatrix} I \\ E^{(L)} \end{pmatrix} G_{\mathbf{i}}|_{\mathbf{\tilde{i}}_{\mathbf{i}}} H_{\mathbf{i}}^{T} \approx \Pi^{(L)} \begin{pmatrix} I \\ E^{(L)} \end{pmatrix} C_{\mathbf{i},2}^{-}|_{\mathbf{\tilde{i}}_{\mathbf{i}}}.$$
 (2.29)

Thus, $C_{i,2}^{-}|_{\tilde{\mathbf{i}}_{i}}$ serves as an approximate row basis matrix for $C_{i,2}^{-}$. Such a procedure of using the proxy point method and the SPRR factorization to get (2.27)-(2.29) is also called a "hybrid" compression scheme [18]. We pause to note that we have $C_{i,2}^{-}|_{\tilde{\mathbf{i}}_{i}} = k(\tilde{\mathbf{s}}_{i}, \mathbf{t}_{i})$ for a subset $\tilde{\mathbf{s}}_i$ of \mathbf{s}_i . That is, (2.29) sparsifies \mathbf{s}_i into $\tilde{\mathbf{s}}_i$. In such circumstances, $\tilde{\mathbf{s}}_i$ is sometimes called a skeleton [38], [39] or representative set [40].

Accordingly, we have

$$C_{\mathbf{i}}^{-} \approx P^{(L)} \begin{pmatrix} I \\ \Pi^{(L)} \begin{pmatrix} I \\ E^{(L)} \end{pmatrix} \end{pmatrix} \begin{pmatrix} C_{\mathbf{i},1} \\ C_{\mathbf{i},2}^{-}|_{\mathbf{\tilde{i}}} \end{pmatrix}.$$
(2.30)

(Here, we abuse notation and use I to denote identity matrices of different sizes.) For convenience, let $\mathbf{\overline{i}}_i$ be the row index of $C_{i,1}^-$ in C_i^- and let $\mathbf{\widehat{i}}^{(L)} = \mathbf{\overline{i}}_i \cup \mathbf{\widetilde{i}}_i$. Then we may rewrite (2.30) as

$$C_{\rm i}^- \approx U^{(L)} C_{\rm i}^-|_{\hat{\mathbf{j}}^{(L)}}$$
 with $U^{(L)} = \hat{P}^{(L)} \begin{pmatrix} I \\ \hat{E}^{(L)} \end{pmatrix}$, (2.31)

where $\hat{P}^{(L)} = P^{(L)} \begin{pmatrix} I \\ \Pi^{(L)} \end{pmatrix}$ is a permutation matrix, $\hat{E}^{(L)} = \begin{pmatrix} 0 & E^{(L)} \end{pmatrix}$, and the identity block in $\hat{P}^{(L)}$ and the zero block in $\hat{E}^{(L)}$ both have column sizes equal to $|\bar{\mathbf{s}}|$.

(2.31) essentially means that the entire X_i set is then sparsified to

$$\mathbf{\hat{s}}_{i} \equiv \mathbf{\bar{s}}_{i} \cup \mathbf{\tilde{s}}_{i}, \tag{2.32}$$

which corresponds to the index set $\mathbf{\hat{i}}^{(L)}$. This is illustrated in Figure 2.6(b). (2.31) may then also be written as

$$C_{\rm i}^-(=k(X_{\rm i},Y_{\rm i})) \approx U^{(L)}k(\hat{\mathbf{s}}_{\rm i},Y_{\rm i}).$$
 (2.33)

Note that

$$\mathbf{s}_{i}| = |\bar{\mathbf{s}}_{i}| = \frac{n}{2^{L+1}}, \quad |\mathbf{\hat{s}}_{i}| = \frac{n}{2^{L+1}} + r.$$
 (2.34)

At this point, we set the HSS generator $U_i = U^{(L)}$. If i is any other node at the leaf level not corresponding to the topmost HSS block row, we note that by Proposition 2.2.2, all of the displays in the previous case are still valid for the same matrix $U^{(L)}$ and row index set $\mathbf{i}^{(L)}$. Hence, we may set the HSS generator U_i to $U^{(L)}$ in this case as well. (Note that our notation $\Pi^{(L)}$, $U^{(L)}$, $\hat{E}^{(L)}$, and $\mathbf{i}^{(L)}$ above is therefore justified.) Also, when \mathbf{s}_i for each leaf i is sparsified, the entire set X gets sparsified accordingly, as illustrated in Figure 2.6(c).

2.3.2 Compression at nonleaf levels

For a node i at a nonleaf level l < L, by induction, we may assume that we have computed $U^{(l+1)}$. Let c_1 and c_2 be the children of i.

First, suppose that i corresponds to the topmost HSS block row at level l. With i in (2.33) set to be c_1 and c_2 , respectively, we have that $C_{c_1}^- \approx U^{(l+1)}k(\mathbf{\hat{s}}_{c_1}, Y_{c_1})$ and $C_{c_2}^- \approx U^{(l+1)}k(\mathbf{\hat{s}}_{c_2}, Y_{c_2})$. Then

$$C_{i}^{-}(=k(X_{i},Y_{i})) \approx \begin{pmatrix} U^{(l+1)} \\ U^{(l+1)} \end{pmatrix} \begin{pmatrix} k(\hat{\mathbf{s}}_{c_{1}},Y_{i}) \\ k(\hat{\mathbf{s}}_{c_{2}},Y_{i}) \end{pmatrix}$$

$$\equiv \begin{pmatrix} U^{(l+1)} \\ U^{(l+1)} \end{pmatrix} k(X_{i},Y_{i}),$$
(2.35)

where $X_i = \mathbf{\hat{s}}_{c_1} \cup \mathbf{\hat{s}}_{c_2}$. See Figure 2.8(a). Following the HSS construction procedure in [10], the compression of C_i^- can then be done on $k(X_i, Y_i)$, which corresponds to the rows of $C_i^$ with index set $\mathbf{\hat{i}}^{(l+1)} \cup (\mathbf{\hat{i}}^{(l+1)} + \frac{n}{2^{l+1}})$. Similarly to before, partition X_i as

$$X_{i} = \mathbf{s}_{i} \cup \bar{\mathbf{s}}_{i} \quad \text{with}$$
$$\mathbf{s}_{i} = X_{i} \cap \left\{ \omega^{2k-2} \mid \frac{n}{2^{l+2}} \le k < \frac{3n}{2^{l+2}} \right\}, \quad \bar{\mathbf{s}}_{i} = X_{i} \setminus \mathbf{s}_{i}. \tag{2.36}$$

Again, when the interaction between X_i and Y_i are considered as in $k(X_i, Y_i)$, \mathbf{s}_i and $\bar{\mathbf{s}}_i$ may be considered as the "far-field" and "near-field" subsets of X_i , respectively. Note that the reason why we take the intersection with $\left\{\omega^{2k-2} \mid \frac{n}{2^{l+2}} \leq k < \frac{3n}{2^{l+2}}\right\}$ in (2.36) is for the purpose of separation distance as in (2.26). This still ensures that, if we set $\mathbf{s} = \mathbf{s}_i$ and $\mathbf{t} = Y_i$ in Definition 2.1.3, then $\operatorname{gap}_{\mathcal{A}}(\mathbf{s}, \mathbf{t})$ and $\operatorname{span}_{\mathcal{A}}(\mathbf{s})$ satisfy (2.16), so that \mathbf{s} and \mathbf{t} are well-separated. See Figure 2.8(b).



Figure 2.8. Forming $X_i = \mathbf{\hat{s}}_{c_1} \cup \mathbf{\hat{s}}_{c_2}$ and \mathbf{s}_i and sparsifying \mathbf{s}_i into $\mathbf{\tilde{s}}_i$ for a nonleaf node i.

Then just as in the previous subsection, an analytic compression step can be applied to $k(\mathbf{s}_i, Y_i)$. The set \mathbf{s}_i is then sparsified to $\tilde{\mathbf{s}}_i$ of size $|\tilde{\mathbf{s}}_i| = r$. See Figure 2.8(c). Similarly to (2.29) and (2.33), we then obtain

$$k(\mathbf{s}_{i}, Y_{i}) \approx \Pi^{(l)} \begin{pmatrix} I \\ E^{(l)} \end{pmatrix} k(\tilde{\mathbf{s}}_{i}, Y_{i}),$$
 (2.37)

$$k(X_{\rm i}, Y_{\rm i}) \approx U^{(l)} k(\mathbf{\hat{s}}_{\rm i}, Y_{\rm i}), \quad \text{with} \quad U^{(l)} = \hat{P}^{(l)} \begin{pmatrix} I \\ \hat{E}^{(l)} \end{pmatrix},$$

$$(2.38)$$

where $\mathbf{\hat{s}}_{i}$ is given as in (2.32) and corresponds to an index set $\mathbf{\hat{i}}^{(l)}$ in $k(X_{i}, Y_{i})$, $\hat{P}^{(l)}$ is a permutation matrix, and $\hat{E}^{(l)} = \begin{pmatrix} 0 & E^{(l)} \end{pmatrix}$ has bounded entries.

Note that

$$|\mathbf{s}_{\mathbf{i}}| = |\mathbf{\bar{s}}_{\mathbf{i}}| = \frac{n}{2^{L+1}} + r(L-l), \quad |\mathbf{\hat{s}}_{\mathbf{i}}| = \frac{n}{2^{L+1}} + r(L-l+1).$$
(2.39)

The reason for this is as follows. For i at level L, (2.34) holds. For i at level L - 1 with children c_1 and c_2 , $\bar{\mathbf{s}}_i$ is formed by $\frac{n}{2^{l+2}}$ points of $\bar{\mathbf{s}}_{c_1}$, $\frac{n}{2^{l+2}}$ points of $\bar{\mathbf{s}}_{c_2}$, and r points from $\tilde{\mathbf{s}}_{c_1}$ and $\tilde{\mathbf{s}}_{c_2}$. (Note that the points in $\tilde{\mathbf{s}}_{c_1}$ and $\tilde{\mathbf{s}}_{c_2}$ are sampled from \mathbf{s}_{c_1} and \mathbf{s}_{c_2} , respectively, in the same way, and together their contributions to $\bar{\mathbf{s}}_i$ has r points.) Then, for i at any level l, (2.39) can be shown by induction. In addition, the size of the near-field set $\bar{\mathbf{s}}_i$ in (2.39) is reasonably small.

Now, from (2.35) and the HSS construction process in [10], we can set the HSS generators R_{c_1}, R_{c_2} as $\begin{pmatrix} R_{c_1} \\ R_{c_2} \end{pmatrix} = U^{(l)}$. Again from Proposition 2.2.2, for any other node i at the same level l, the same R_{c_1}, R_{c_2} generators are used.

Also, for the purpose of extracting B generators later, we introduce the following index set (as a column vector):

$$\mathbf{j}^{(l)} = \begin{pmatrix} \mathbf{j}^{(l+1)} \\ \mathbf{j}^{(l+1)} + \frac{n}{2^{l+1}} \end{pmatrix} \Big|_{\mathbf{\hat{i}}^{(l)}}, \qquad (2.40)$$

where $\mathbf{j}^{(L)} = \mathbf{\hat{i}}^{(L)}$. $\mathbf{j}^{(l)}$ is used to keep track of the index set of $\mathbf{\hat{s}}_i$ in X or the row index set of $k(\mathbf{\hat{s}}_i, Y_i)$ in C.

This process is performed until every non-root level of \mathcal{T} is visited. At that point, we get all the U, R generators in the HSS approximation.

2.3.3 Other HSS generators

Now, as discussed in Section 2.2, Proposition 2.2.3 means C_i^- and $(C_i^{\mid})^T$ are closely related. In fact, other than few near-field columns in C_i^- and rows in $(C_i^{\mid})^T$, the remaining subblocks only differ by a permutation. That is, the far-field compression for C_i^- can essentially be used for the far-field compression of $(C_i^{\mid})^T$. Specifically, if we replace C_i^- in (2.27)–(2.29) by $(C_i^{\dagger})^T$, the results stay the same. Accordingly, we can obtain the V, W generators by setting for each leaf i,

$$V_{\rm i} = U^{(L)}.$$

Similarly, for each nonleaf i at level l with children c_1, c_2 , we set

$$\left(\begin{array}{c} W_{c_1} \\ W_{c_2} \end{array}\right) = U^{(l)}.$$

Note that the generators are the same for all the nodes at the same level due to Proposition 2.2.2.

Next, due to the forms of $U^{(L)}$ in (2.31) and $U^{(l)}$ in (2.38), we can then use the index sets defined in (2.40) to pick, for a left node i at level l,

$$B_{\mathbf{i}} \equiv B^{(l,1)} = C|_{\mathbf{j}^{(l)} \times (\mathbf{j}^{(l)} + \frac{n}{2^l})},$$

and for a right node i at level l,

$$B_{i} \equiv B^{(l,2)} = C|_{(\mathbf{j}^{(l)} + \frac{n}{2^{l}}) \times \mathbf{j}^{(l)}}.$$

Lastly, for the (leaf-level) D generators, we need only to store $D_1 = k(X_1, X_1)$. Lemma 2.2.1 means that this can be used to obtain the D_i generators for any other leaf i.

Overall, we need only to obtain the matrices $U^{(l)}, B^{(l,1)}, B^{(l,2)}$ at each level l as well as D_1 at the leaf level. They are sufficient to write out all the HSS generators of the HSS approximation. The basis generators U, V, W, R generators are provided by $U^{(l)}$ which is defined by a permutation matrix and a matrix $E^{(l)}$ like in (2.31) or (2.38). For D_1 and each B generator, it just needs to store the associated index sets.

2.4 Algorithm analysis and performance

Algorithm 1 details the construction process. At each level of the HSS tree \mathcal{T} , one HSS block is compressed. The corresponding point set is sparsified. The compression information

is then used for the other HSS blocks at the same level. Such a scheme clearly leads to an overall sublinear complexity with a sublinear storage, as verified by the following proposition.

Algorithm 1.

function $(D_1, U_1, R^{(l)}, B^{(l)}) = \mathsf{HyperHSS}(\mathcal{T}, \tau)$ $\triangleright \mathcal{T}$: HSS tree with L levels $r \leftarrow O(|\log_2 \tau|)$ \triangleright Number of proxy points per level $i \leftarrow 1$ $X_{\mathbf{i}} \leftarrow \left(\omega^{2k-2}\right)_{0 \le k < \frac{n}{2L}}$ \triangleright Indices ordered by increasing k $D_{i} \leftarrow C|_{\mathbf{s}_{i} \times \mathbf{s}_{i}}$ for l = L, L - 1, ..., 1 do \triangleright Levelwise traversal of \mathcal{T} if i > 1 then \triangleright If i is not a leaf $c_1 \leftarrow \text{left child of i}$ $X_{i} \leftarrow \left(\mathbf{\hat{s}}_{c_{1}}, \mathbf{\hat{s}}_{c_{1}} + \frac{n}{2^{l+1}}\right)$ end if $\mathbf{s}_{i} \leftarrow X_{i} \cap \left\{ \omega^{2k-2} \mid \frac{n}{2^{l+2}} \leq k < \frac{3n}{2^{l+2}} \right\}$ \triangleright Far field $\bar{\mathbf{s}}_i \leftarrow X_i \setminus \mathbf{s}_i$ \triangleright Near field $c \leftarrow \omega^{2^{L-l} - \frac{1}{2}}$ \triangleright Center for proxy points $\gamma_1 \leftarrow |c - \omega^{2^{L-l-1}}|, \gamma_2 \leftarrow |c-1|$ $\mathbf{z} \leftarrow$ uniformly-spaced points on circle with center c and radius $\sqrt{\gamma_1 \gamma_2}$ \triangleright Proxy points $k(\mathbf{s}_{i}, \mathbf{z}) \approx U^{(l)}k(\mathbf{\tilde{s}}_{i}, \mathbf{z})$ \triangleright Rank-*r* SPRR factorization to get $\mathbf{\tilde{s}}_i \subset \mathbf{s}_i$ $\mathbf{\hat{s}}_i \leftarrow \mathbf{\bar{s}}_i \cup \mathbf{\tilde{s}}_i$ \triangleright Reordered counterclockwise in \mathbb{C} starting at 1 if l = L then $U_{i} \leftarrow U^{(l)}$ else

$$\begin{pmatrix} R_{c_1} \\ R_{c_2} \end{pmatrix} \leftarrow U^{(l)}$$
end if

 $\triangleright c_1, c_2$: children of i

 $\mathbf{j}_i \leftarrow \text{index set corresponding to } \mathbf{\hat{s}}_i \text{ in } X$ $B_{\mathbf{i}} \leftarrow C|_{\mathbf{j} \times (2^{L-l} + \mathbf{j}_{\mathbf{i}})}, B_{\mathbf{i}+1} \leftarrow C|_{(2^{L-l} + \mathbf{j}_{\mathbf{i}}) \times \mathbf{j}_{\mathbf{i}}}$ $i \leftarrow parent of i in T$ end for end function

end

Proposition 2.4.1. Let $L = O(\log n)$ and let r be the number of proxy points used in each far-field compression step of the algorithm. Then the algorithm described above constructs the HSS approximation in $O(r^3 \log^2 n)$ flops with $O(r^2 \log^2 n)$ storage for the HSS generators if the B generators are not explicitly formed. An extra $O(r^2 \log^3 n)$ cost and $O(r^2 \log^3 n)$ storage are needed for the B generators. In addition, the B generators at level l have sizes O(r(L-l)).

Proof. At each level l, associated with the leftmost node i, the cost to form G_i like in (2.27) is $r\left(\frac{n}{2^{L+1}}+r(L-l)\right)$ flops, since G_i has size $|\mathbf{s}_i| \times r$ with $|\mathbf{s}_i|$ in (2.39). The SPRR factorization of K_i costs $O\left(r^2\left(\frac{n}{2^{L+1}}+r(L-l)\right)\right)$. The total compression cost at all levels is then

$$\sum_{l=1}^{L} O\left(r^2 \left(\frac{n}{2^{L+1}} + r(L-l)\right)\right) = O(r^3 L^2).$$

The storage is mainly for $E^{(l)}$ in (2.37) and for some index vectors and looks like

$$\sum_{l=1}^{L} O\left(r\left(\frac{n}{2^{L+1}} + r(L-l)\right) \right) = O(r^2 L^2).$$

The *B* generators at level *l* have sizes $|\mathbf{j}^{(l)}|$, which is also $|\mathbf{\hat{i}}^{(l)}|$. Now, $|\mathbf{\hat{i}}^{(l)}| = |\mathbf{\hat{s}}_i| = O(r(L-l))$ from (2.39). If additionally we explicitly form *B* generators, then the extra cost is

$$\sum_{l=1}^{L} O\left((r(L-l))^2 \right) = O(r^2 L^3).$$

(Note that the cost for forming D_1 is only $O(r^2)$.)

From this, we can see that the resulting HSS approximation has B generators with sizes $O(r(L-l)) = O(\log n)$. This reflects the off-diagonal ranks of the HSS form. With the

compression of each far-field blocks satisfying the accuracy in (2.11), it is convenient to apply a result in [31] to obtain a global approximation error, which is roughly $O(\tau r^{\log n})$.



Figure 2.9. Operation count for Algorithm 1 with varying n, together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the flop count associated with forming the D, U, and R generators in the left plot is asymptotically $O(\log^2 n)$ and that the flop count associated with forming the B generators in the right plot is asymptotically $O(\log^3 n)$.

To illustrate the performance of the algorithm, we apply it to C with matrix sizes $n = 2^7, 2^8, \ldots, 2^{62}$. The SPRR factorization is based on an implementation from [41]. In Figure 2.9, we compare the number of floating point operations required to construct the HSS generators using Algorithm 1 for a Cauchy matrix of size n. The left plot shows the operation count associated with forming the D, U, and R generators; the right plot shows the count associated with forming the B generators. We compare the count for the D, U, R generators and the count for the B generators with the provided $O(\log^2 n)$ and $O(\log^3 n)$ trendlines, respectively; it is evident that the former is asymptotically proportional to $\log^2 n$ and that the latter is asymptotically proportional to $\log^3 n$.

Similarly, in the left plot of Figure 2.10, we provide the effective storage costs for the constructed D, U, R, generators. This count includes storing the E matrices and permutation vectors at each level, rather than the full U and R generators, of which significant subblocks are simply permutation matrices. (Counting the full U and R generator storage cost would



Figure 2.10. Effective storage costs for Algorithm 1 with varying n, together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the effective storage cost for forming the D, U, and R generators in the left plot is $O(\log^2 n)$ and that the effective storage cost for forming the B generators in the right plot is $O(\log^3 n)$.

introduce another factor of $\log n$.) In the right plot, we count the effective storage costs for the *B* generators. In these counts, we set both the number of proxy points and the far-field rank to be 25. In comparing the storage count for the *D*, *U*, and *R* generators and the storage count for the *B* generators to the provided $O(\log^2 n)$ and $O(\log^3 n)$ trendlines, respectively, we again see that both counts appear to be asymptotically proportional to $\log^2 n$ and $\log^3 n$. So, again their sum is asymptotically proportional to $\log^3 n$.

In Algorithm 1, we also require a number of function evaluations to precompute certain points selected from the unit circle. For completeness, we count these function evaluations in Figure 2.11. Again, the evaluation count associated with forming the D, U, and Rgenerators is counted in the left plot, and the evaluation count associated with forming the B generators is counted in the right plot. In comparing them to the provided $O(\log^2 n)$ and $O(\log^3 n)$ trendlines, respectively, these again appear to asymptotically grow as $\log^2 n$ and $\log^3 n$. Together, the three Figures 2.9 through 2.11 corroborate the cost calculation in the preceding proposition, since r is kept constant at 25 and L is $O(\log n)$.

Also, in Figures 2.13 and 2.14, we plot \mathbf{s}_i , $\mathbf{\tilde{s}}_i$, X_i , and $\mathbf{\hat{s}}_i$ versus the height of each corresponding HSS tree node, for a Cauchy matrix of size 4096 and 6 levels of compression. We again take 25 to be the number of proxy points and the off-diagonal far-field rank. The



Figure 2.11. The number of exponential function evaluations in Algorithm 1 with varying n, together with trendlines (shown in circles \circ) that are $O(\log^2 n)$ in the left plot and $O(\log^3 n)$ in the right plot. The figures suggest that the number of function evaluations for forming the D, U, and R generators in the left plot is $O(\log^2 n)$ and that the number of evaluations for forming the B generators in the right plot is $O(\log^3 n)$.



Figure 2.12. The relative accuracies in the Frobenius norm for the HSS approximation to C using the generators constructed by the new algorithm. Here, the leaf level size was kept constant at 64, both the off-diagonal far-field rank r and the number of proxy points was 25.

expected linear growth of the sizes of \mathbf{s}_i , $\mathbf{\tilde{s}}_i$, X_i , as well as the expected constant size of the sparsified far field $\mathbf{\hat{s}}_i$, shows that our implementation of the algorithm validates the assumptions on the computational and storage costs at each level in the proof of the above proposition. Finally, to illustrate the accuracy of this algorithm, in Figure 2.12 we show that the relative accuracy in the Frobenius norm for the HSS approximation to C constructed using Algorithm 1 stays roughly within one order of magnitude using the same number of proxy points and off-diagonal far-field rank at each level. Here, we also take both to be 25. This indicates that Algorithm 1 seems to continue producing accurate approximations as ngrows.



Figure 2.13. The cardinalities of \mathbf{s}_i (·) and $\mathbf{\tilde{s}}_i$ (o) for n = 4096, L = 6, and r = 25. Here, r = 25 proxy points were also used in the far-field approximation.



Figure 2.14. The cardinalities of X_i (·) and $\hat{\mathbf{s}}_i$ (o) for n = 4096, L = 6, and r = 25. Again, r = 25 proxy points were also used in the far-field approximation.

3. HYPERFAST HSS CONSTRUCTION OF SOME TOEPLITZ MATRICES

The ideas of the previous chapter concerned a particular kind of Cauchy matrix C. In this chapter, we show how to extend these ideas to another class of "kernel matrices" defined on a set of points with regular geometry as follows. Since the compression step for each HSS row block C_i^- only relies on the factor K_i in (2.27), it is possible to apply the idea of reusing the same row basis \tilde{U}_l , as in (2.31) and (2.38) across all the HSS block rows at the same depth to some other kernel matrices, as long as the relative geometry of the proxy points and enclosed points is the same due to symmetry. See Figure 3.1 for two examples: (a) gives the symmetry in use in the last chapter, and (b) gives the symmetry we will use in this chapter.



Figure 3.1. Symmetry of the proxy point contours (dotted lines) and their enclosed points \mathbf{s}_i (\Box).

In particular, let A = k(X, Y), for some finite sets $X, Y \subseteq \mathbb{C}$, and let X_i, Y_i be the points associated with the ith HSS block row A_i^- . Suppose that for each $y \in Y_i$, k(z, y) is complexanalytic in z on a region containing a circular contour around a subset \mathbf{s}_i (the "far field") of X_i . Let \mathbf{z}_i be the associated set of proxy points. Then, suppose the corresponding proxy point approximation looks like

$$k(\mathbf{s}_{i}, Y_{i}) \approx \lambda(\mathbf{s}_{i}, \mathbf{z}_{i})k_{0}(\mathbf{z}_{i}, Y_{i}),$$

where λ is the Cauchy kernel and k_0 depends on k and i. If this approximation yields an identical $\lambda(\mathbf{s}_i, \mathbf{z}_i)$ for each i, we may apply the basis reuse idea from our earlier discussion. This is exactly what we intend to do in this chapter, by considering Toeplitz matrices as kernel matrices.

Toeplitz matrices arise frequently in many areas of computation and engineering such as digital signal processing and differential equation solutions. In Chapter 2, we mentioned several algorithms have been devised that exploit the displacement structure of general Toeplitz matrices to perform various matrix operations faster than the counterpart "naive" algorithms applicable to general matrices. We have seen examples of the resulting fast and superfast algorithms in [24], [42]. We have also seen that the resulting Cauchy-like intermediate matrices in such algorithms can be quickly approximated by structured matrices as in [9], [24], [29]. Similarly, in digital signal processing, it has become well-known that the multiplication of Toeplitz convolution matrices with a given signal can be accelerated by applying FFTs and performing the equivalent operation in the frequency domain.[43], [44]

As we saw in the last chapter, the central idea of such algorithms over the past few decades has become to apply fast Fourier transforms (FFTs) and then solve the equivalent Cauchylike system in the frequency space Recalling the methods mentioned in the last chapter, we observe that, after certain speedups that may be obtained using randomized techniques, the dominant cost in structured matrix frequency-domain Toeplitz algorithms becomes the application of FFTs, as in [10], [30], [37]. Hence, in theory, general HSS algorithms can potentially achieve a speedup for matrix operations whenever a matrix is both Toeplitz and has low off-diagonal rank before the application of FFTs.[10] In such algorithms, the dominant cost becomes the construction of the structured approximant; thus, bringing this cost down is a worthwhile endeavor.

In this chapter, we show that for Toeplitz matrices whose Toeplitz vector is generated by a univalent map applied to the positive integers, we are able to reduce the HSS construction time complexity from $O(k^2n)$ [23] to $O(\log^5 n)$ in the size n of a square matrix with offdiagonal rank bound k. This is done in much the same way as in the previous chapter. While the new algorithm is less widely applicable, it may nevertheless be applied to certain important classes of matrices, such as those arising from a convolution of a digital signal with a large Gaussian filter [44]. In addition, since this new scheme does not rely on Fourier space representation, it has the advantage of preserving the rank structure of any diagonal or rank-structured summand that may be added to the Toeplitz matrix, such as when localizing eigenvalues; examples of the necessity of such summands include [31], [32].

The key ideas in this new construction scheme largely remain the same as in the previous chapter: the use of the proxy point method in the process of obtaining an interpolative decomposition; the separation of the relevant points into near- and far-field sets; and the reuse of the resulting approximate basis matrix factors for all the HSS blocks at a given HSS depth. To guide the process of obtaining these approximate basis factors, and to understand when it is applicable, we perform a new analysis of the error introduced by the proxy point method when applied to complex-analytic kernel matrices. In the case that the proxy point method is used to approximate off-diagonal blocks of Toeplitz matrices with Toeplitz vector generated by a complex-analytic univalent map, this error is then shown to be small enough and to increase slowly enough in n to allow our construction algorithm to be performed in sublinear time with respect to n.

The rest of this chapter is structured as follows: in Section 3.1, we perform the new proxy point error analysis for general complex-analytic functions. The HSS construction algorithm for the Toeplitz matrices under our consideration is detailed in Section 3.2; an explicit bound on the number of proxy points required in this construction is given in Section 3.2.3. Finally, in Section 3.3, we perform some numerical tests of the new algorithm. This work will appear in [45].

3.1 General proxy point method analysis for analytic kernels

First, we recall the proxy point method and provide a new analysis for the error introduced in its application. As mentioned earlier in Section 2.1, the analysis to follow here is more general but provides an absolute bound, rather than the relative bound found in [18]. (See Chapter 1.)

Let $D, E \subseteq \mathbb{C}$ be open balls with center c and radii r and R, respectively; let $X \subseteq D$ and $Y \subseteq \mathbb{C}$ be finite sets; and let $k : \mathbb{C}^2 \to \mathbb{C}$ be a function such that, for each $y \in Y$, k(z, y) is

an analytic function of z on the annulus $E \setminus \overline{D}$. Then, for each $x \in X, y \in Y$, by the Cauchy integral formula, we have

$$k(x,y) = \frac{1}{2\pi i} \int_C \frac{k(\zeta,y)}{\zeta - x} d\zeta$$

where C the boundary of an open ball with center c radius \sqrt{Rr} . (See Figure 3.2 (a).)



(a) The contour C (red), the finite set X (blue), and the boundaries of the open balls D and E; the shaded green region shows where k(y, z) is assumed to be analytic in z for each $y \in Y$

(b) The finite sets X (blue) and $Z = \bigcup_{j=1}^{N} z_j$ (red), as well as the boundaries of the open balls D and E

Figure 3.2. Setup for a proxy point approximation to k(X, Y). (The set Y is not pictured.)

Using the trapezoidal rule with N points to approximate this integral, we have

$$k(x,y) = \frac{\sqrt{Rr}}{N} \sum_{j=1}^{N} \left(\frac{1}{z_j - x}\right) \left(\omega^{j} k(z_j, y)\right) + \epsilon,$$

where $z_j = c + \sqrt{Rr}\omega^j$, $\omega = e^{\frac{2\pi}{N}}$, and $\epsilon \in \mathbb{C}$ is ideally very small in magnitude. (See Figure 3.2 (b).) To find a bound for $|\epsilon|$, we prove the following fact; the proof is partly based on the proof of Theorem 2.2 in [46]:

Proposition 3.1.1. Let $D, E \subseteq \mathbb{C}$ be open balls with center c and radii r and R, respectively; let $X \subseteq D$ and $Y \subseteq \mathbb{C}$ be finite sets; and let $k : \mathbb{C}^2 \to \mathbb{C}$ be a function such that, for each $y \in Y$, k(z, y) is an analytic function of z on E. Then for each $x \in X, y \in Y$, we have

$$\left|k(x,y) - \frac{\sqrt{Rr}}{N} \sum_{j=1}^{N} \left(\frac{1}{z_j - x}\right) \left(\omega^j k(z_j,y)\right)\right| \le K \frac{\max_{z \in \partial F} |k(z,y)|}{\left(\frac{R}{r}\right)^{\frac{N}{4}} - 1},$$

where $K = \left(2\frac{\sqrt[4]{\frac{R}{r}}}{\sqrt[4]{\frac{R}{r}-1}}\right)$ and F is the open ball with center c and radius $r\left(\frac{R}{r}\right)^{\frac{3}{4}}$.

Proof. Let $A = \overline{B\left(0, \sqrt[4]{\frac{R}{r}}\right)} \setminus B\left(0, \left(\sqrt[4]{\frac{R}{r}}\right)^{-1}\right)$, and for each $x \in X, y \in Y$, let $k_{x,y} : A \to \mathbb{C}$ be defined by

$$k_{x,y}(z) = \frac{k\left(c + z\sqrt{Rr}, y\right)\left(z\sqrt{Rr}\right)}{c + z\sqrt{Rr} - x}$$

By our assumption on k, $k_{x,y}$ is analytic on an open set containing the compact annulus A, so we have the Laurent series expansion $k_{x,y}(z) = \sum_{p=-\infty}^{\infty} a_p z^p$. In particular, we have

$$\sum_{j=1}^{N} \left(\frac{1}{N}\right) k_{x,y} \left(\omega^{j}\right) = \sum_{j=1}^{N} \left(\frac{1}{N}\right) \sum_{p=-\infty}^{\infty} a_{p} \left(\omega^{j}\right)^{p}$$
$$= \sum_{p=-\infty}^{\infty} \sum_{j=1}^{N} \left(\frac{1}{N}\right) a_{p} \left(\omega^{j}\right)^{p}$$
$$= \sum_{p=-\infty}^{\infty} a_{Np},$$

where the last line follows from the fact that $\sum_{j=1}^{N} \left(\frac{1}{N}\right) (\omega^{j})^{p}$ is 0 if p is not a multiple of N and 1 if p is a multiple of N. In addition, by the definition of Laurent series coefficients and the Cauchy integral formula, respectively, we have

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} k_{x,y} \left(e^{i\xi} \right) d\xi = k(x,y).$$

Hence, we get

$$\left| k(x,y) - \sum_{j=1}^{N} \left(\frac{\sqrt{Rr}}{N} \right) \frac{\omega^{j} k\left(z_{j}, y \right)}{z_{j} - x} \right| = \left| \frac{1}{2\pi} \int_{0}^{2\pi} k_{x,y} \left(e^{i\xi} \right) d\xi - \sum_{j=1}^{N} \left(\frac{1}{N} \right) k_{x,y} \left(\omega^{j} \right) \right|$$
$$= \left| \left(\sum_{p=-\infty}^{-1} a_{Np} \right) + \left(\sum_{p=1}^{\infty} a_{Np} \right) \right|$$
$$\leq \left(\sum_{p=-\infty}^{-1} |a_{Np}| \right) + \left(\sum_{p=1}^{\infty} |a_{Np}| \right).$$

Now, let F' be the annulus $B\left(c, r\left(\frac{R}{r}\right)^{\frac{3}{4}}\right) \setminus B\left(c, r\sqrt[4]{\frac{R}{r}}\right)$. For each $p \in \mathbb{Z}$ with $p \neq 0$, we have

$$\begin{aligned} |a_p| &\leq \left| \frac{1}{2\pi} \int_{|\zeta| = \sqrt[4]{r}} \frac{k_{x,y}(\zeta)}{\zeta^{p+1}} d\zeta \right|, \left| \frac{1}{2\pi} \int_{|\zeta| = \left(\sqrt[4]{r}\right)^{-1}} \frac{k_{x,y}(\zeta)}{\zeta^{p+1}} d\zeta \\ &\leq \frac{\max_{z \in A} \left(|k_{x,y}(z)| \right)}{\left(\sqrt[4]{r}\right)^{|p|}} \\ &\leq \frac{\left(\max_{z \in F'} |k(z,y)|\right) \left(\max_{z \in F'} \left| \frac{z-c}{z-x} \right|\right)}{\left(\sqrt[4]{r}\right)^{|p|}} \\ &\leq \left(\frac{\sqrt[4]{r}}{\sqrt[4]{r}-1}\right) \frac{\left(\max_{z \in F'} |k(z,y)|\right)}{\left(\sqrt[4]{r}\right)^{|p|}} \\ &\leq \left(\frac{K}{2}\right) \frac{\left(\max_{y \in Y, z \in F} |k(z,y)|\right)}{\left(\sqrt[4]{r}\right)^{|p|}} \\ &\leq \left(\frac{K}{2}\right) \frac{\max_{y \in Y, z \in \partial F} |k(z,y)|}{\left(\sqrt[4]{r}\right)^{|p|}}, \end{aligned}$$

with the last two inequalities by the maximum modulus principle since k(z, y) is holomorphic on *E*. Therefore,

$$\begin{aligned} \left| k(x,y) - \sum_{j=1}^{N} \left(\frac{\sqrt{Rr}}{N} \right) \frac{\omega^{j} k\left(z_{j}, y \right)}{z_{j} - x} \right| &\leq \left(\sum_{p=-\infty}^{-1} |a_{Np}| \right) + \left(\sum_{p=1}^{\infty} |a_{Np}| \right) \\ &\leq \left(\frac{K}{2} \right) \left(2 \sum_{p=1}^{\infty} \frac{1}{\left(\sqrt[4]{\frac{R}{r}} \right)^{Np}} \right) \max_{y \in Y, z \in \partial F} |k(z,y)| \\ &\leq K \frac{\max_{y \in Y, z \in \partial F} |k(z,y)|}{\left(\frac{R}{r} \right)^{\frac{N}{4}} - 1}, \end{aligned}$$

as claimed.

For a discussion of similar bounds, see [46]. However, note that the bounds given there and elsewhere in the numerical analysis literature do not simultaneously and explicitly bound the proxy point error for all values of an enclosed set X for each $y \in Y$; hence, we may use our new bound to bound the entrywise error across an entire kernel matrix block. This property will allow us to use this bound to guarantee applicability of the HSS construction method in Sections 3.2.1 and 3.2.2. This proof also provides a reason for the heuristic, shown in Proposition 1.1.3 ([18]) for the Cauchy kernel, that in the setup above we should pick C to have radius \sqrt{Rr} . Therefore, for the kernel matrix k(X, Y), we have a low-rank approximation

$$k(X,Y) \approx UV,\tag{3.1}$$

where

$$U = \begin{pmatrix} \frac{1}{z_{1}-x_{1}} & \frac{1}{z_{2}-x_{1}} & \cdots & \frac{1}{z_{N}-x_{1}} \\ \frac{1}{z_{1}-x_{2}} & \frac{1}{z_{2}-x_{2}} & \cdots & \frac{1}{z_{N}-x_{2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{z_{1}-x_{l}} & \frac{1}{z_{2}-x_{l}} & \cdots & \frac{1}{z_{N}-x_{l}} \end{pmatrix};$$

$$V = \begin{pmatrix} \omega\sqrt{Rr}k(y_{1},z_{1}) & \omega\sqrt{Rr}k(y_{2},z_{1}) & \cdots & \omega\sqrt{Rr}k(y_{m},z_{1}) \\ \omega^{2}\sqrt{Rr}k(y_{1},z_{2}) & \omega^{2}\sqrt{Rr}k(y_{2},z_{2}) & \cdots & \omega^{2}\sqrt{Rr}k(y_{m},z_{2}) \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{N}\sqrt{Rr}k(y_{1},z_{N}) & \omega^{N}\sqrt{Rr}k(y_{2},z_{N}) & \cdots & \omega^{N}\sqrt{Rr}k(y_{m},z_{N}) \end{pmatrix};$$

 x_1, \ldots, x_l are the elements of X; and y_1, \ldots, y_m are the elements of Y. Furthermore, by Proposition 3.1.1, we have the bound

$$||k(X,Y) - UV||_2 \le lmK \frac{\max_{y \in Y, z \in \partial F}(k(z,y))}{\left(\frac{R}{r}\right)^{\frac{N}{4}} - 1},$$

where F is an open ball with center c and radius $r\left(\frac{R}{r}\right)^{\frac{3}{4}}$. Hence, for a given 2-norm tolerance τ of the proxy point approximation to k(X, Y), we only need to use $O\left(\log lm + \log \tau\right)$ proxy points as long as the assumption on the analyticity of k holds.

As we noted in Section 1.1.2, the left factor U does not at all depend on k. Hence, the quality of the approximation (3.1) depends entirely on the function maximum term considered in Proposition 3.1.1. Coming up with a condition on k to ensure a slow enough growth of this maximum will be our task in Section 3.2.3. There, we show that univalence of a function related to the definition of k serves this purpose.

3.2 Sublinear Toeplitz kernel HSS generator construction

In this section, we detail our sublinear HSS construction algorithm for Toeplitz matrices arising from univalent maps applied to a regular grid. This method is almost the same as that given in Algorithm 1, with the only difference being the geometry of the proxy points involved. For completeness, we detail the approximation construction algorithm in 3.2.1 and 3.2.2; the analysis of the number of proxy points necessary for a good approximation is shown in 3.2.3.

Let

$$T = \begin{pmatrix} t_0 & t_{-1} & \dots & t_{-(n-1)} \\ t_1 & t_0 & \dots & t_{-(n-2)} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \dots & t_0 \end{pmatrix},$$

be an $n \times n$ real- or complex-valued Toeplitz matrix where the entries may be expressed as $t_i = f_1(-i)$ for $-n \leq i \leq -1$ and $t_i = f_2(i)$ for $1 \leq i \leq n$ for some injective ("univalent") functions $f_1 \in \mathcal{O}\left(B\left(-\frac{n}{2}, \frac{n}{2}\right)\right)$ and $f_2 \in \mathcal{O}\left(B\left(\frac{n}{2}, \frac{n}{2}\right)\right)$. (Here, by $\mathcal{O}\left(B\left(-\frac{n}{2}, \frac{n}{2}\right)\right)$, we mean the set of holomorphic functions on the open ball in \mathbb{C} with both center and radius $\frac{n}{2}$.) Such matrices occur in [47][48], among many other more general contexts discussed in the introduction. In [47], for example, the functions f_1, f_2 are defined by $-f_1(z) = f_2(z) = (1-z)\log\left(\frac{z-1}{z}\right) + (z+1)\log\left(\frac{z}{z+1}\right)$. To illustrate the application of this method, we will deal with the symmetric case $f = f_1 = f_2$. The non-symmetric case is handled similarly (see Section 3.2.2). Since we are constructing generators for approximations to the off-diagonal blocks of T, we may assume without loss of generality that $t_0 = 0$. Furthermore, we assume that n is a power of two greater than 8.

3.2.1 Constructing the HSS row generators

Let $L \leq \log_2(n) - 2$ be the number levels in the desired HSS approximation to T. Let s be a bound for the numerical HSS rank of T; we assume specifically that s is $O(\log n)$. (The analysis in Section 3.2.3 can actually be used to show that s is $O(\log^2 n)$, but that is not our focus here.) For each $1 \leq i, j \leq n$ with $i \neq j$, we have $T_{i,j} = f(|j - i|)$. Hence, we may consider an HSS block T_j^- to be the kernel matrix $k(\mathbf{i}_j, [1, n] \setminus \mathbf{i}_j)$, where k is defined by k(x, y) = f(|x - y|). Directly finding a low-rank factorization for T_j^- , for example as when j = 1 in the first step in the HSS construction algorithm in [10], is already prohibitively expensive with at least O(n) flops. Instead, we may follow a similar list of steps as in Section 2.3:

• If j is not leaf of \mathcal{T} , we assume we have performed this list of steps on its children $c_1(j)$ and $c_2(j)$ to obtain sets of indices $\mathbf{i}'_{c_1(j)}, \mathbf{i}'_{c_2(j)} \subseteq \mathbf{i}_j$. If j is a leaf, we define $c_1(j) = c_2(j) = \mathbf{j}$ and $\mathbf{i}'_j = \mathbf{i}_j$. Then, we define $\mathbf{i}'_j = \mathbf{i}'_{c_1(j)} \cup \mathbf{i}'_{c_2(j)}$ and apply a proxy point approximation to $(T_j^-)_{\mathbf{i}'_j \times [1,n-|\mathbf{i}_j|]}$. However, since we only assumed that f is analytic on $B\left(\frac{n}{2}, \frac{n}{2}\right)$, by Equation 3.1, the ratio $\frac{R}{r}$ in this case could be as large $\frac{1}{n}$, and therefore the number of proxy points N required to obtain a reasonably good approximation may be prohibitively large. Hence, we first separate \mathbf{i}_j into the "near-field" and "farfield" subsets $\mathbf{\hat{i}}_j$ and $\mathbf{\tilde{i}}_j = \mathbf{i}_j \setminus \mathbf{\hat{i}}_j$, respectively, where $\mathbf{\hat{i}}_j$ is the subset of \mathbf{i}_j consisting of its first and last $\frac{|\mathbf{i}_j|}{4}$ values, respectively, ordered the usual way. We then define $\mathbf{\hat{i}}'_j = \mathbf{\hat{i}}_j \cap \mathbf{\tilde{i}}'_j$, $\mathbf{\tilde{i}}'_j = \mathbf{\tilde{i}}_j \cap \mathbf{\tilde{i}}'_j$, $T_{j,1}^- = k\left(\mathbf{\hat{i}}'_j, [1,n] \setminus \mathbf{i}_j\right)$, and $T_{j,2}^- = k\left(\mathbf{\tilde{i}}'_j, [1,n] \setminus \mathbf{i}_j\right)$; and we apply a proxy point approximation to only the far-field subblock: $T_{j,2}^- \approx \tilde{U}_j \tilde{V}_j$. For this approximation, we use a circular contour with center $\frac{1}{2} (\min(\mathbf{i}_j) + \max(\mathbf{i}_j))$ and radius $\frac{\sqrt{2}}{2} (\max(\mathbf{i}_j) - \min(\mathbf{i}_j) + 1)$ to obtain $\frac{R}{r} = 2$. (See Figure 3.3 and Figure 3.4.)

We thus have

$$(T_{j}^{-})|_{\tilde{i}_{j}^{\prime} \times [1,n-|\tilde{i}_{j}|]} = \Pi_{i} \begin{pmatrix} T_{j,1}^{-} \\ T_{j,2}^{-} \end{pmatrix} = \Pi_{i} \begin{pmatrix} I & 0 \\ 0 & \tilde{U}_{i} \end{pmatrix} \begin{pmatrix} T_{j,1}^{-} \\ \tilde{V}_{i} \end{pmatrix},$$

where Π_i is a permutation matrix.

• Next, we find a strong rank-revealing QR factorization

$$\tilde{U}_{j} = \overline{U}_{j} \left(\Pi_{j}^{\prime T} \tilde{U}_{j} \right) |_{[1,s] \times [1,N]},$$

where $\overline{U}_{j} = \begin{pmatrix} I & E_{j} \end{pmatrix}^{T}$ and Π'_{j} is a permutation matrix. We then have

$$T_{\mathbf{j},2}^{-} \approx \overline{U}_{\mathbf{j}} \left(\Pi_{\mathbf{j}}^{\prime T} \tilde{U}_{\mathbf{j}} \right)_{[1,s] \times [1,N]} \tilde{V}_{\mathbf{j}} \approx \overline{U}_{\mathbf{j}} \left(\Pi_{\mathbf{j}}^{\prime T} T_{\mathbf{j},2}^{-} \right)_{[1,s] \times [1,n] \setminus \mathbf{i}_{\mathbf{j}}},$$



Figure 3.3. Top: the near-field points $\mathbf{\tilde{i}}'_1(\circ)$, far-field points $\mathbf{\hat{i}}'_1(\bullet)$, proxy points (\times) , and the points $[9,32](\Box)$ involved in the approximation of the leaf HSS block $T_1^-|_{\mathbf{\tilde{i}}'_1\times[1,n-|\mathbf{i}_1|]} = k([1,8],[9,32])$ for a matrix of size n = 32, number of HSS levels L = 2, and number of proxy points N = 16. Bottom: the resulting index set $\mathbf{i}'_1(\Box)$. (These are "cartoon illustrations" and are not actual results from such an approximation applied to a subblock of an actual matrix T.)

 \mathbf{SO}

$$\begin{split} \left(T_{j}^{-}\right)|_{\vec{i}_{j}' \times [1,n-|\mathbf{i}_{j}|]} &\approx \Pi_{j} \begin{pmatrix} T_{j,1}^{-} \\ T_{j,2}^{-} \end{pmatrix} \\ &\approx U_{j} \begin{pmatrix} \left(\Pi_{j}^{T}T_{j}^{-}|_{\vec{i}_{j}' \times [1,n] \setminus \mathbf{i}_{j}}\right)|_{[1,|\hat{\mathbf{i}}_{j}'|] \times [1,n-|\mathbf{i}_{j}|]} \\ &\left(\Pi_{j}^{T}T_{j}^{-}|_{\vec{i}_{j}' \times [1,n] \setminus \mathbf{i}_{j}}\right)|_{[|\hat{\mathbf{i}}_{j}'|+1,|\hat{\mathbf{i}}_{j}'|+s] \times [1,n-|\mathbf{i}_{j}|]} \end{pmatrix} \\ &= U_{j}T^{-}|_{\mathbf{i}_{j}' \times [1,n] \setminus \mathbf{i}_{j}}, \end{split}$$



Figure 3.4. Top: the near-field points $\tilde{\mathbf{i}}'_3$ (\circ), far-field points $\hat{\mathbf{i}}'_3$ (\bullet), proxy points (\times), and the points [17, 32] (\Box) involved in the approximation of the height-2 HSS block $T_3^-|_{\tilde{\mathbf{i}}'_3\times[1,n-|\mathbf{i}_3|]} = k\left(\tilde{\mathbf{i}}'_j,[17,32]\right)$ for a matrix of size n = 32, number of HSS levels L = 2, and number of proxy points N = 16. Bottom: the resulting index set \mathbf{i}'_3 (\Box). (As noted in Figure 3.3 above, these are "cartoon illustrations" and are not reflective of actual numerical results.)

where $\mathbf{i}'_{j} \subseteq \mathbf{i}_{j}$ is of size $\left|\mathbf{\hat{i}}'_{j}\right| + s$ and

$$U_{j} = \Pi_{j} \begin{pmatrix} I & 0 \\ 0 & \Pi'_{j} \begin{pmatrix} I \\ E_{j} \end{pmatrix} \end{pmatrix}.$$

Now, if j is a leaf, this last display is precisely the HSS generator. If j is not a leaf, we set $R_{c_1(j)} = U_j|_{\left(\mathbf{i}'_j \cap \mathbf{i}_{c_1(j)}\right) \times \left[1, \left|\mathbf{\tilde{i}}'_j\right| + s\right]} \text{ and } R_{c_2(j)} = U_j|_{\left(\mathbf{i}'_j \cap \mathbf{i}_{c_2(j)}\right) \times \left[1, \left|\mathbf{\tilde{i}}'_j\right| + s\right]}.$

3.2.2 Constructing the remaining HSS generators

Now, note that for each j at the leaf level in \mathcal{T} , each matrix $(T_j^{-})|_{\mathbf{i}'_{c_1(j)}\cup\mathbf{i}'_{c_2(j)}\times[1,n-|\mathbf{i}_j|]}$ used to obtain the generator U_j yields the same U_j regardless of the specific value of j. This is the symmetry we mentioned at the beginning of the chapter, so as we mentioned earlier, it is a very similar situation to Chapter 2. Hence, \mathbf{i}'_j is the same for any leaf-level j, so we can show by induction on L that for each j at the same depth of \mathcal{T} , U_j and \mathbf{i}'_j are the same. This shows that again, we only need to perform the above steps once at each depth of \mathcal{T} to obtain all the HSS row generators U_j for a leaf-level j and R_j for j with depth(j) $\leq L-2$. Furthermore, because the above steps do not depend on the specific function k(x, y) = f(|x-y|) as long as f satisfies the analyticity condition, the above steps also construct the HSS column generators V_j and W_j . So, we set $V_j = U_j$ for a leaf-level j and $W_j = R_j$ for j with depth(j) $\leq L-2$. This last fact shows why our assumption that $f_1 = f_2$ at the beginning of this section confers no loss of generality. Finally, for each $j \in \mathcal{T}$, we set $B_j = T_{\mathbf{i}'_j \times \mathbf{i}'_{\mathrm{sib}(j)}}$.

So far, we have not mentioned how many proxy points are required for the far-field approximation at each level in the above construction method; we will explore this issue in the next section. We note here, however, that if the number of proxy points is $O(\log n)$, then because the above procedure is almost exactly the same as in Section sec:hss, the flop count of this method is the same. Therefore, we again get a total of $O(\log^5 n)$ flops. We will show that this is indeed the case—that is, a sublinear bound on the number of proxy points required holds—in the next section whenever f satisfies certain conditions.

3.2.3 Number of proxy points required

First, as before, let \mathcal{T}, \mathcal{I} be the HSS tree and HSS index set of T, respectively. Let $\mathbf{j} \in \mathcal{T}$ have corresponding index set $\mathbf{i}_{\mathbf{j}} \in \mathcal{I}$. We define $\mathbf{\hat{i}}_{\mathbf{j}}$ to be the subset of $\mathbf{i}_{\mathbf{j}}$ missing its least and greatest $\frac{\mathbf{i}_{\mathbf{j}}}{4}$ elements, ordered the usual way. We also define $\tilde{T}_{n}^{\mathbf{j},N}$ to be the *N*-point proxy point approximation (in the first variable) to the subblock $T|_{\mathbf{\hat{i}}_{\mathbf{j}},[1,n]\setminus\mathbf{i}_{\mathbf{j}}} = k(\mathbf{\hat{i}}_{\mathbf{j}},[1,n]\setminus\mathbf{i}_{\mathbf{j}})$ with center $\frac{1}{2}(\min(\mathbf{i}_{\mathbf{j}}) + \max(\mathbf{i}_{\mathbf{j}}))$ and radius $\frac{1}{2}(\max(\mathbf{i}_{\mathbf{j}}) - \min(\mathbf{i}_{\mathbf{j}}) + 1)$.

Next, we show with Example 3.2.1 that for general $f \in \mathcal{O}\left(B\left(\frac{n}{2}, \frac{n}{2}\right)\right)$, this approximation need not have good convergence properties.

Example 3.2.1. For $n \ge 8$, let $T_n \in \mathbb{R}_{n \times n}$ have entries $(T_n)_{i,j} = \cos\left(\frac{\pi}{4}|j-i|\right)$, and let $\mathcal{I}_n = \{\mathbf{i}_{n,1}, \mathbf{i}_{n,2}, \mathbf{i}_{n,3}\}$ be its one-level HSS index set, indexed the usual way. Then the associated function $f(z) = f_1(z) = f_2(z) = \cos\left(\frac{\pi z}{4}\right)$ is holomorphic on $B\left(\frac{n}{2}, \frac{n}{2}\right)$. Table 3.1 shows the minimum number of points N required for $\tilde{T}_n^{1,N}$ to approximate $(T_n)|_{\mathbf{i}_{n,1},[1,n]\setminus\mathbf{i}_{n,1}}$ to a given tolerance. Note that even for such small matrix sizes and large tolerance, the number of proxy points required already scales linearly with n. It is also worth noting that the rank of T_n is at most 8 for all n and every off-diagonal block.

Table 3.1. The size *n* of the matrix T_n and the minimum number of proxy points *N* required to attain $\|(T_n)|_{\mathbf{\hat{i}}_{n,1},[1,n]\setminus\mathbf{\hat{i}}_{n,1}} - \tilde{T}_n^{1,N}\|_F < 10^{-6}$.

| n | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
|---|----|----|----|----|----|----|----|----|----|
| N | 21 | 27 | 34 | 39 | 47 | 53 | 59 | 65 | 72 |

A few words about the poor performance in Example 3.2.1 are in order. By Proposition 3.1.1, for each $y \in Y = [1, n] \setminus \mathbf{i}_{n,1} = \left[\frac{n}{2} + 1, n\right], k(z, y) = f(|z - y|)$ must not be too large in absolute value for all $z \in \partial F = \partial B\left(\frac{n}{4} + \frac{1}{2}, \frac{\sqrt[4]{8}n}{8}\right)$ in order for a small number of proxy points to be sufficient. But in this case, we may observe that, if $y = \frac{n}{2} + 1$, the maximum of $f(|y - z|) = \cos\left(\frac{\pi}{4}|y - z|\right)$ along $z \in \partial F$ grows exponentially in n. In particular, even though cosine is bounded on the real line, its growth along the one-dimensional line z(t) = t + it (as a real vector space, for real t) is exponential. Hence, the growth of N with respect to n shown in Table 3.1 gives evidence that f with large values on $B\left(\frac{n}{2}, \frac{n}{2}\right)$ may require a lot of proxy points for an accurate approximation.

On the other hand, if f is bounded on the real line and univalent on $B\left(\frac{n}{2}, \frac{n}{2}\right)$, we show in Example 3.2.2 that we do seem to have good proxy point convergence for the HSS approximation outlined in Sections 3.2.1 and 3.2.2.

Example 3.2.2. For $n \ge 8$, let $T_n \in \mathbb{R}_{n \times n}$ have entries $(T_n)_{i,j} = \cos\left(\frac{\pi|j-i|}{n}\right)$. Then the associated function $f(z) = f_1(z) = f_2(z) = \cos\left(\frac{\pi z}{n}\right)$ is univalent on $B\left(\frac{n}{2}, \frac{n}{2}\right)$ and bounded on the real line. Table 3.2 shows the minimum number of proxy points required for the sublinear HSS construction method to yield a given approximation tolerance for the topmost HSS row block.

Table 3.2. The size n of the matrix T_n and the minimum value of N such that the *L*-level HSS approximation constructed in Sections 3.2.1 and 3.2.2 with N proxy points approximates the topmost HSS block of T_n to a relative Frobenius norm error 10^{-10} .

| n | 2048 | 4096 | 4096 | 8192 | 8192 | 8192 | 16384 | 16384 | 16384 | 16384 |
|---|------|------|------|------|------|------|-------|-------|-------|-------|
| L | 1 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 3 | 4 |
| N | 26 | 27 | 27 | 28 | 28 | 28 | 28 | 28 | 28 | 28 |

Example 3.2.2 gives strong numerical evidence that the proxy-point approximation has good enough convergence properties to be used in practice, even despite global HSS error accumulation. We now show that good proxy point convergence is true for general univalent f.

Lemma 3.2.3. Let \mathcal{I} be an HSS index set for an $n \times n$ matrix, where n is a power of 2; let $\mathbf{i} \in \mathcal{I}$; and let l be the height of \mathbf{i} . Define k(x, y) = f(|y - x|) for some $f \in \mathcal{O}\left(B\left(\frac{n}{2}, \frac{n}{2}\right)\right)$; let $x \in \hat{\mathbf{i}}$; let $y \in [1, n] \setminus \mathbf{i}$; and let $N \in \mathbb{N}$. Then

$$\left| k(x,y) - \sum_{j=1}^{N} \left(\frac{\left(\sqrt[4]{8}\right) 2^{l-1}}{N} \right) \frac{\omega^{j} k(z_{j},y)}{z_{j}-x} \right| < 14 \frac{\max_{z \in \partial F} \left(|f(y-z)| \right)}{2^{\frac{N}{4}} - 1},$$
(3.2)

where $z_{j} = c + (\sqrt[4]{8}) 2^{l-1} \omega^{j}$, F is the open ball with center c and radius $(\sqrt[4]{8}) 2^{l-1}$, and $c = \frac{1}{2} (\max(\mathbf{i}) - \min(\mathbf{i}) + 1)$.

Proof. This is a fairly straightforward application of Proposition 3.1.1, where we set $X = \hat{\mathbf{i}}$; $Y = [1, n] \setminus \mathbf{i}$; and D and E to be the open balls with center c and radii $R = 2^{l-1}$ and $r = 2^{l}$, respectively. We thus get $K = 2\frac{\sqrt[4]{2}}{\sqrt[4]{2-1}} < 14$.

By the maximum modulus principle and Lemma 3.2.3, if $\max_{z \in \partial B\left(\frac{n+1}{2}, \frac{n}{2}-1\right)} |f(z)|$ has a sufficiently small bound with respect to n, we would need only $O\left(\log n\right) + |\log(\epsilon)|$ proxy points to obtain an entrywise proxy point approximation with tolerance ϵ at every height of the HSS tree. We may obtain such a bound to combine with Lemma 3.2.3 if f is univalent on $B\left(\frac{n}{2}, \frac{n}{2}\right)$, and if f and its derivative does not grow too quickly quickly with respect to n along the real axis.

Proposition 3.2.4. Let $f \in \mathcal{O}\left(B\left(\frac{n}{2}, \frac{n}{2}\right)\right)$ be bounded and univalent (hence conformal) on $B\left(\frac{n}{2}, \frac{n}{2}\right)$. Then for $z \in \partial B\left(\frac{n+1}{2}, \frac{n}{2} - 1\right)$,

$$|f(z)| \le \left(\frac{n}{2}\right)^3 \left| f'\left(\frac{n}{2}\right) \right| + \left| f\left(\frac{n}{2}\right) \right|.$$

Proof. Define the functions $h : \mathbb{D} \to \mathbb{C}$ and $g : \mathbb{D} \to \mathbb{C}$

$$h(z) = \left(\frac{n}{2}\right)z + \frac{n}{2}, \text{ and}$$
$$g(z) = \frac{(f \circ h)(z) - (f \circ h)(0)}{(f \circ h)'(0)}$$

•

Then g is schlicht, so by the growth theorem, we have $|g(z)| \leq \frac{|z|}{(1-|z|)^2}$. Thus, for $z \in \partial B\left(\frac{n+1}{2}, \frac{n}{2} - 1\right)$,

$$|(f \circ h)(z) - (f \circ h)(0)| \le \frac{|z| |(f \circ h)'(0)|}{(1 - |z|)^2}.$$

Therefore, we have

$$\begin{split} |(f \circ h)(z)| &\leq \frac{|z| \left| (f \circ h)'(0) \right|}{(1 - |z|)^2} + |(f \circ h)(0)| \\ &\leq \left(\frac{n}{2}\right)^2 \left| (f \circ h)'(0) \right| + |(f \circ h)(0)| \\ &= \left(\frac{n}{2}\right)^2 \left| f'\left(\frac{n}{2}\right) h'(0) \right| + \left| f\left(\frac{n}{2}\right) \right| \\ &\leq \left(\frac{n}{2}\right)^2 \left| f'\left(\frac{n}{2}\right) \right| |h'(0)| + \left| f\left(\frac{n}{2}\right) \right| \\ &= \left(\frac{n}{2}\right)^2 \left(\frac{n}{2}\right) \left| f'\left(\frac{n}{2}\right) \right| + \left| f\left(\frac{n}{2}\right) \right|, \end{split}$$

so the result follows from the definition of h.

We observe that the independence of this bound from x, y, and l (as defined in Lemma 3.2.3) is the main point of the statement. With this proposition in hand, we may obtain an error bound for the proxy point approximation of an off-diagonal "far-field" row block.

Corollary 3.2.5. Let $T \in \mathbb{C}$ be the $n \times n$ matrix with entries $T_{i,j} = f(|j-i|)$, where $f \in \mathcal{O}\left(B\left(\frac{n}{2}, \frac{n}{2}\right)\right)$ is injective (hence conformal) on $B\left(\frac{n}{2}, \frac{n}{2}\right)$. Let \mathcal{I} be the HSS index set of T, and let $\mathbf{i}_j \in \mathcal{I}$. Then

$$\left\|T\right|_{\mathbf{\hat{i}}_{j},[1,n]\setminus\mathbf{\hat{i}}_{j}}-\tilde{T}^{\mathbf{j},N}\right\|_{F} \leq \left(\frac{7n^{2}}{2^{\frac{N}{4}+1}-2}\right)\left(\frac{n^{3}}{8}\left|f'\left(\frac{n}{2}\right)\right|+\left|f\left(\frac{n}{2}\right)\right|\right).$$

Proof. By Lemma 3.2.3, the maximum modulus principle, and Proposition 3.2.4, in that order, we have that for each $1 \le u \le |\mathbf{\hat{i}}_j|$ and $1 \le v \le |[1, n] \setminus \mathbf{i}_j|$,

$$\begin{split} \left| \left(T|_{\hat{\mathbf{i}}_{j},[1,n]\backslash \mathbf{i}_{j}} \right)_{u,v} - \left(\tilde{T}^{j,N} \right)_{u,v} \right| &< 14 \frac{\max_{y \in [1,n]\backslash \mathbf{i}, z \in \partial F} \left(|f(y-z)| \right)}{2^{\frac{N}{4}} - 1} \\ &\leq 14 \frac{\max_{z \in \partial B\left(\frac{n+1}{2}, \frac{n}{2} - 1\right)} \left(|f(z)| \right)}{2^{\frac{N}{4}} - 1} \\ &\leq \frac{14}{2^{\frac{N}{4}} - 1} \left(\frac{n^{3}}{8} \left| f'\left(\frac{n}{2}\right) \right| + \left| f\left(\frac{n}{2}\right) \right| \right). \end{split}$$

Since $|\mathbf{\hat{i}}_{j}|, |[1, n] \setminus \mathbf{i}_{j}| \leq \frac{n}{2}$, the result follows by summing over all u and v.

Thus, to obtain a given proxy point approximation tolerance ϵ for any level, we need $O(\log n) + O(\left|f\left(\frac{n}{2}\right)\right|) + O(\left|f'\left(\frac{n}{2}\right)\right|) + O(\left|\log \epsilon\right|)$ proxy points. In practice, f and its derivative are often bounded on the real line, as in Examples 3.3.1 and 3.3.2 below.

3.3 Numerical tests

First, we note that although univalence of f is a sufficient condition, it is not strictly necessary in practice to enable the use of our sublinear Toeplitz HSS construction algorithm. Example 3.3.1 illustrates this.

Example 3.3.1. For $n \ge 8$, let $T_n \in \mathbb{R}_{n \times n}$ have entries $(T_n)_{i,j} = (|j-i| - \frac{n}{2})^2$, so the associated function $f(z) = f_1(z) = f_2(z) = (z - \frac{n}{2})^2$ is not univalent on $B(\frac{n}{2}, \frac{n}{2})$. Table 3.3 lists the relative approximation tolerance for various HSS approximations of T from Sections 3.2.1 and 3.2.2. (For the scheme as outlined there, we set s = 28. Each matrix involved has a relative off-diagonal numerical rank of 3 with respect to the tolerance 10^{-14} .) Note that relatively small values of N result in a good approximation.

| n | | 2048 | 2048 | 2048 | 2048 | 8192 | 8192 | 8192 |
|--------------------------------|--------------------------------------|-------------|--------|--------|--------|--------|--------|--------|
| L | | 2 | 2 | 4 | 4 | 4 | 4 | 6 |
| N 32 | | | 48 | 32 | 48 | 32 | 48 | 32 |
| rel. err. $(e10^{-13})$ 5.4863 | | | 2.0441 | 2.9697 | 9.3656 | 7.7119 | 3.2532 | 3.3541 |
| | $\begin{array}{c c}n\\ L\end{array}$ | | 8192 | 16384 | 16384 | 16384 | 16384 | |
| | | | 6 | 6 | 6 | 7 | 7 | |
| | N | | | 32 | 48 | 32 | 48 | |
| | rel. err. (el | $10^{-13})$ | 1.0675 | 6.9370 | 2.9239 | 3.4362 | 1.0933 | |

Table 3.3. The relative Frobenius norm errors of the *L*-level HSS approximation to T_n from Sections 3.2.1 and 3.2.2 using N proxy points.

On the other hand, the conditions of Proposition 3.2.5 provides a wide class of functions for which our sublinear HSS construction algorithm is guaranteed to work.

Example 3.3.2. Since $f_1(z) = \frac{n}{z}$ and $f_2(z) = -\frac{n}{z}$ are univalent on $B\left(\frac{n}{2}, \frac{n}{2}\right)$, the method from Sections 3.2.1 and 3.2.2 should work to find the HSS generators of T_n , the Cauchy kernel matrix evaluated at n equidistant points in [-1, 1], sublinearly.

Table 3.4 lists the relative approximation tolerance for various HSS approximations to the matrix $T_n \in \mathbb{R}_{n \times n}$ with off-diagonal values $(T_n)_{i,j} = \frac{n}{j-i}$ and diagonal values equal to 0. The maximum relative off-diagonal numerical rank s is also listed; for this experiment, we set s = 28 for each matrix.

| U | imerical | cal HSS rank s of T_n with tolerance 10^{-14} . | | | | | | | |
|-------------------------|-----------------------------|--|--------|--------|--------|--------|--------|--------|--------|
| | n | | 2048 | 2048 | 2048 | 2048 | 8192 | 8192 | 8192 |
| | <i>s</i> 26 | | 26 | 26 | 26 | 26 | 30 | 30 | 30 |
| | L | L 2 | | 2 | 4 | 4 | 4 | 4 | 6 |
| | N | | 32 | 48 | 32 | 48 | 32 | 48 | 32 |
| | rel. err. $(e10^{-14})$ 7.1 | | | 1.7926 | 5.9208 | 1.1841 | 8.1024 | 2.1102 | 6.1210 |
| | | $\begin{array}{ c c c } n & & \\ \hline s & & \end{array}$ | | 8192 | 16384 | 16384 | 16384 | 16384 | |
| | | | | 30 | 33 | 33 | 33 | 33 | |
| | $\frac{L}{N}$ | | | 6 | 6 | 6 | 7 | 7 | |
| | | | | 48 | 32 | 48 | 32 | 48 | |
| rel. err. $(e10^{-14})$ | | | 1.2407 | 9.4705 | 2.5062 | 6.1585 | 1.2521 | | |
| | | | | | | | | | |

Table 3.4. The relative Frobenius norm errors of the *L*-level HSS approximation to T_n from Sections 3.2.1 and 3.2.2 using N proxy points, as well as the numerical HSS rank s of T_n with tolerance 10^{-14} .

Again, we pause to note that even after global error accumulation associated with an HSS tree of depth 6 and 7 in Examples 3.3.2 and 3.3.1, the relative error is still quite low. This gives evidence that the asymptotic error decay regime from Proposition 3.2.4 holds well enough in practice: note that the function in Example 3.3.1 is even increasing on B(n/2, n/2) for increasing n.

3.4 Further extensions

We may be able to combine the proxy point method used in this section to approximate the HSS generators of the blocks of the Cauchy-like matrix C associated with T (see the beginning of Chapter 2) directly. This likely requires certain analytic or algebraic conditions on T: namely, the generators of C may be obtained from (2.5) interpreted as an analytic kernel applied to well-separated sets, as well as to a "near field." However, the geometric properties of the points of \mathbb{C}^2 encoded by the matrices G and H in (2.5), namely their separation, require conditions on T to be desirable. Such conditions may be difficult to elucidate; work in this direction has been undertaken in [28], [49]. It is possible that certain Toeplitz matrices or other matrices with displacement structure that come from specific applications have such desirable properties.

4. MULTIDIMENSIONAL PROXY POINT METHOD

The new proxy point method analysis detailed in Chapter 3 relies on the Cauchy integral formula, as well as the power series representation of and the maximum modulus principle for complex-analytic functions. One may wonder whether the proxy point method analysis may be extended to apply to points and kernels on \mathbb{C}^k for $k \geq 2$, since both the Cauchy integral formula and the required power series properties readily generalize to complexanalytic functions of several variables. The answer, as we will show, is yes; we detail this generalization in Section 4.1.

In addition, the method applied to accelerate HSS construction of Toeplitz matrices viewed a Toeplitz matrix as a kernel matrix applied two identical sets of points, since the values of a Toeplitz matrix can be thought of as a "one-parameter" set, varying only with the difference of their row/column ordinals. There, we used a complex contour to separate points on the real line and then used appropriate growth properties of the underlying function to guarantee off-diagonal low rank. Since the values of any matrix can be viewed as a "two-parameter" set, varying with the row/column ordinals, one may also wonder if we may generalize this method somehow to a suitable class of underlying two-variable functions, using the Cauchy integral formula in several variables. We will analyze one such generalization in Section 4.2.

4.1 Error analysis for the proxy point method in \mathbb{C}^k

First, we give a very straightforward generalization of Proposition 3.1.1 to several complex variables. To our knowledge, a similar idea has only been mentioned in the "Cauchy Fast Multipole Method" scheme introduced in [22]; see Section 2.2 of that paper. However, the convergence analysis of even the one-variable case given in that paper is largely heuristic, in that it does not quantitatively relate a global error to an aspect of the geometry of the points. That is, an analog to the ratio R/r, as defined in Proposition 3.1.1, does not play a role in bounding the number of proxy points N for all matrix entries simultaneously in the analysis of that paper. Hence, such analysis is not as well-suited for rank bounds, as in Corollary 3.2.5. Furthermore, no convergence analysis of the multidimensional case is given at all. By contrast, we will focus on quantitatively relating the global error to R/r and the number of proxy points N, even in the multidimensional case.

The proof of the following error bound works in exactly the same way as the one given in Proposition 3.1.1, but we provide it here for completeness. For simplicity, we only consider the case when the radii in each dimension are equal, and when the number of proxy points in each dimension is equal to N, for a total of N^k proxy points when dealing with kernel functions on \mathbb{C}^k . The general case follows similarly.

Proposition 4.1.1. Let $D, E \subseteq \mathbb{C}^k$ be hyperdisks with center \mathbf{c} and radii r and R, respectively; let $X \subseteq D$ and $Y \subseteq \mathbb{C}^k$ be finite sets; and let $k : \mathbb{C}^{2k} \to \mathbb{C}$ be a function such that, for each $\mathbf{y} \in Y$, $k(\mathbf{z}, \mathbf{y})$ is an analytic function of \mathbf{z} on E. Then for each $\mathbf{x} \in X, \mathbf{y} \in Y$, we have

$$\left| k(\mathbf{x}, \mathbf{y}) - \left(\frac{\sqrt{Rr}}{N}\right)^k \sum_{1 \le j_1, \dots, j_k \le N} \frac{\omega^{j_1} \cdots \omega^{j_k} k\left((z_{j_1, 1}, \dots, z_{j_k, k}), \mathbf{y}\right)}{(z_{j_1, 1} - x_1) \cdots (z_{j_k, k} - x_k)} \right| \\ \le \left(\frac{2(R/r)^{1/4}}{((R/r)^{1/4} - 1)((R/r)^{N/4} - 1)}\right)^k \max_{z \in \partial F} |k(\mathbf{z}, \mathbf{y})|,$$

where F is the polydisk with center **c** and radius $r\left(\frac{R}{r}\right)^{\frac{3}{4}}$, and where $z_{j,l} = c_l + \sqrt{Rr}\omega^j$ for $1 \le l \le k$ and $1 \le j \le N$.

Proof. Let $A = \overline{\mathbb{D}^k(\mathbf{0}, (R/r)^{1/4})} \setminus \mathbb{D}^k(\mathbf{0}, (R/r)^{-1/4})$, and for each $\mathbf{x} \in X, \mathbf{y} \in Y$, let $k_{\mathbf{x}, \mathbf{y}} : A \to \mathbb{C}$ be defined by

$$k_{\mathbf{x},\mathbf{y}}(z) = \frac{k\left(\mathbf{c} + \mathbf{z}\sqrt{Rr}, \mathbf{y}\right) \prod_{l=1}^{k} \left(z_l \sqrt{Rr}\right)}{\prod_{l=1}^{k} \left(c_l + z_l \sqrt{Rr} - x_l\right)}$$

By our assumption on k, $k_{\mathbf{x},\mathbf{y}}$ is analytic on an open multicircular domain containing the compact set A, so we have the k-variable Laurent series expansion

$$k_{\mathbf{x},\mathbf{y}}(z_1,\ldots,z_k) = \sum_{\alpha_1,\ldots,\alpha_k \in \mathbb{Z}} c_{\alpha_1,\ldots,\alpha_k} \prod_{l=1}^k z_l^{\alpha_l}$$
for $(z_1, \ldots, z_k) \in A$. In particular, since convergence is uniform on A, we have

$$\sum_{1 \le j_1, \dots, j_k \le N} \left(\frac{1}{N}\right)^k k_{\mathbf{x}, \mathbf{y}} \left(\omega^{j_1}, \dots, \omega^{j_k}\right) = \sum_{1 \le j_1, \dots, j_k \le N} \left(\frac{1}{N}\right)^k \sum_{\alpha_1, \dots, \alpha_k \in \mathbb{Z}} c_{\alpha_1, \dots, \alpha_k} \prod_{l=1}^k (\omega^{j_l})^{\alpha_l}$$
$$= \sum_{\alpha_1, \dots, \alpha_k \in \mathbb{Z}} \sum_{1 \le j_1, \dots, j_k \le N} \left(\frac{1}{N}\right)^k c_{\alpha_1, \dots, \alpha_k} \prod_{l=1}^k (\omega^{j_l})^{\alpha_l}$$
$$= \sum_{\alpha_1, \dots, \alpha_k \in \mathbb{Z}} c_{\alpha_1, \dots, \alpha_k} \prod_{l=1}^k (\omega^{j_l})^{\alpha_l},$$

where the last line follows from the fact that, for each $i = 1, ..., k, 1 \le j_1, ..., \hat{j_i}, ..., j_k \le N$, and $\alpha_1, ..., \alpha_k \in \mathbb{Z}$, we have

$$\sum_{\mathbf{j}_{i}=1}^{N} \left(\frac{1}{N}\right)^{k} \Pi_{l=1}^{k} \left(\omega^{\mathbf{j}_{l}}\right)^{\alpha_{l}} = 0$$

if α_i is not a multiple of N and

$$\left(\frac{1}{N}\right)^{k-1} \prod_{1 \le l \le N, \ l \ne i} \left(\omega^{j_l}\right)^{\alpha_l}$$

if α_i is a multiple of N. In addition, by the definition of Laurent series coefficients and the Cauchy integral formula, respectively, we have

$$a_0 = \frac{1}{(2\pi)^k} \int_0^{2\pi} \cdots \int_0^{2\pi} k_{\mathbf{x},\mathbf{y}} \left(e^{\mathbf{i}\xi_1}, \dots, e^{\mathbf{i}\xi_k} \right) d\xi_k \cdots \xi_1 = k(\mathbf{x}, \mathbf{y}).$$

Hence, we get

$$\begin{vmatrix} k(\mathbf{x},\mathbf{y}) - \left(\frac{\sqrt{Rr}}{N}\right)^k \sum_{1 \le j_1, \dots, j_k \le N} \frac{\omega^{j_1} \cdots \omega^{j_k} k\left((z_{j_1,1}, \dots, z_{j_k,k}), \mathbf{y}\right)}{(z_{j_1,1} - x_{j_1}) \cdots (z_{j_k,k} - x_{j_k})} \end{vmatrix}$$
$$= \left| \frac{1}{(2\pi)^k} \int_0^{2\pi} \cdots \int_0^{2\pi} k_{\mathbf{x},\mathbf{y}} \left(e^{\mathbf{i}\xi_1}, \dots, e^{\mathbf{i}\xi_k} \right) d\xi_k \cdots \xi_1 - \sum_{1 \le j_1, \dots, j_k \le N} \left(\frac{1}{N} \right)^k k_{\mathbf{x},\mathbf{y}} \left(\omega^{j_1}, \dots, \omega^{j_k} \right) \right|$$
$$= \left| \sum_{\alpha_1, \dots, \alpha_k \in \mathbb{Z}, \ (\alpha_1, \dots, \alpha_k) \neq \mathbf{0}} c_{N\alpha_1, \dots, N\alpha_k} \right|.$$

Now, let F' be the multicircular region $\overline{\mathbb{D}^k\left(\mathbf{c}, r\left(R/r\right)^{3/4}\right)} \setminus \mathbb{D}^k\left(\mathbf{c}, r\left(R/r\right)^{1/4}\right)$. For each $\alpha_1, \ldots, \alpha_k \in \mathbb{Z}$ with $(\alpha_1, \ldots, \alpha_k) \neq \mathbf{0}$, we have

$$|c_{\alpha_1,\dots,\alpha_k}| \le \frac{1}{(2\pi)^k} \left| \int_{|\xi_1| = ((R/r)^{1/4})^{e_1}} \cdots \int_{|\xi_k| = ((R/r)^{1/4})^{e_k}} \frac{k_{\mathbf{x},\mathbf{y}}(\xi_1,\dots,\xi_k)}{\prod_{l=1}^k (\xi_l)^{\alpha_l+1}} d\xi_k \cdots d\xi_1 \right|$$

for any $e_1, \ldots, e_k = \pm 1$. Thus,

$$\begin{aligned} |c_{\alpha_{1},...,\alpha_{k}}| &\leq \frac{\max_{\mathbf{z}\in A} |k_{\mathbf{x},\mathbf{y}}(\mathbf{z})|}{\Pi_{l} = 1^{k} \left((R/r)^{1/4} \right)^{|\alpha_{l}|}} \\ &\leq \frac{\max_{\mathbf{z}\in F'} |k(\mathbf{z},\mathbf{y})| \max_{\mathbf{z}\in F'} \left| \Pi_{l=1}^{k} \left(\frac{z_{l}-c_{l}}{z_{l}-x_{l}} \right) \right|}{\Pi_{l=1}^{k} \left((R/r)^{1/4} \right)^{|\alpha_{l}|}} \\ &\leq \left(\frac{(R/r)^{1/4}}{(R/r)^{1/4} - 1} \right)^{k} \frac{\max_{\mathbf{z}\in F'} |k(\mathbf{z},\mathbf{y})|}{\Pi_{l=1}^{k} \left((R/r)^{1/4} \right)^{|\alpha_{l}|}} \\ &\leq \left(\frac{(R/r)^{1/4}}{(R/r)^{1/4} - 1} \right)^{k} \frac{\max_{\mathbf{z}\in F} |k(\mathbf{z},\mathbf{y})|}{\Pi_{l=1}^{k} \left((R/r)^{1/4} \right)^{|\alpha_{l}|}} \\ &\leq \left(\frac{(R/r)^{1/4}}{(R/r)^{1/4} - 1} \right)^{k} \frac{\max_{\mathbf{z}\in \partial F} |k(\mathbf{z},\mathbf{y})|}{\Pi_{l=1}^{k} \left((R/r)^{1/4} \right)^{|\alpha_{l}|}}, \end{aligned}$$

with the last inequality by the maximum modulus principle since $k(\mathbf{z}, \mathbf{y})$ is analytic as a function of \mathbf{z} on E. Therefore,

$$\begin{vmatrix} k(\mathbf{x}, \mathbf{y}) - \left(\frac{\sqrt{Rr}}{N}\right)^{k} \sum_{1 \le j_{1}, \dots, j_{k} \le N} \frac{\omega^{j_{1}} \cdots \omega^{j_{k}} k\left((z_{j_{1}, 1}, \dots, z_{j_{k}, k}), \mathbf{y}\right)}{(z_{j_{1}, 1} - x_{j_{1}}) \cdots (z_{j_{k}, k} - x_{j_{k}})} \end{vmatrix}$$
$$\leq \left| \sum_{\alpha_{1}, \dots, \alpha_{k}, \ (\alpha_{1}, \dots, \alpha_{k}) \neq \mathbf{0}} \left(\frac{(R/r)^{1/4}}{(R/r)^{1/4} - 1}\right)^{k} \frac{\max_{\mathbf{z} \in \partial F} |k(\mathbf{z}, \mathbf{y})|}{\prod_{l=1}^{k} \left((R/r)^{1/4}\right)^{|\alpha_{l}|}} \right|$$
$$\leq \left(\frac{2(R/r)^{1/4}}{((R/r)^{1/4} - 1)((R/r)^{N/4} - 1)}\right)^{k} \max_{z \in \partial F} |k(\mathbf{z}, \mathbf{y})|,$$

as claimed.

The analysis above makes clear that any two sets X and Y must be well-separated in each coordinate. That is, since we are relying on a hyperdisk in \mathbb{C}^k to separate X and Y, we must have that the *l*th coordinate of all points in Y is at least a certain constant away from the *l*th coordinate of all points in X. If $X, Y \subseteq \mathbb{R}^2$, for example, this condition means that we must be able to draw two squares around X, the first the smallest square that contains X, given \mathbf{c} , and the second the largest square that contains X but no points of Y, given that same \mathbf{c} . See Figure 4.1.



Figure 4.1. The definition of the separation R/r between points X (\circ , bottom left) and Y (\Box , top right), for k = 2 and $X, Y \subseteq \mathbb{R}^2$. (That is, $\operatorname{im}(w) = 0$ for all $w \in X, Y$.) The smallest box centered at **c** containing X, as well as the largest box containing X but excluding Y centered at **c**, are displayed (dashed squares). The point **c** is the black cross (\times).

So far, it is not clear if, in practice, the multidimensional proxy point method allows for computational cost savings, or if commonly-used kernels grow too quickly along the set ∂F considered in Proposition 4.1.1 to give a reasonable approximation. The next example should make clear that the proxy point method remains effective for k = 2 and for the commonlyused Coulomb $(k(x, y) = \frac{1}{|y-x|})$ potential and Gaussian $(k(x, y) = e^{(y-x)^2})$ kernels.

Example 4.1.2. Tables 4.1 and 4.2 give relative Frobenius norm errors for the Gaussian and Coulomb potential kernels, for configurations of points X and Y giving varying values of R/r, as well as for varying numbers of proxy points in each dimension. The points used for R/r = 2 are displayed in Figure 4.2. Here, |X| = 4096 and |Y| = 8192.

The results of Example 4.1.2 who that, even for such moderately-sized matrices, the multidimensional proxy point method outlined in this section allows for computational cost savings. However, we caution that the number of proxy points is listed to be what it is in

Table 4.1. The relative Frobenius norm errors of the multidimensional proxy point approximation to the Gauss kernel matrix applied to $X, Y \subseteq \mathbb{R}^2$ giving varying values of R/r, and using varying numbers of proxy points N in each dimension.

| R/r | 2 | 2 | 2 | 3 | 3 | 3 |
|-----------|----------------|------------------|-----------------|----------------|-------------------|-----------------|
| N | 32 | 48 | 56 | 32 | 48 | 56 |
| rel. err. | $1.95e10^{-6}$ | $5.95 e 10^{-9}$ | $3.48e10^{-10}$ | $7.11e10^{-6}$ | $1.60 e 10^{-12}$ | $1.28e10^{-14}$ |

Table 4.2. The relative Frobenius norm errors of the multidimensional proxy point approximation to the Coulomb kernel matrix applied to $X, Y \subseteq \mathbb{R}^2$ giving varying values of R/r, and using varying numbers of proxy points N in each dimension.

| R/r | 2 | 2 | 2 | 3 | 3 | 3 |
|-----------|----------------|------------------|-----------------|------------------|-------------------|-------------------|
| N | 32 | 48 | 56 | 32 | 48 | 56 |
| rel. err. | $1.19e10^{-6}$ | $3.80 e 10^{-9}$ | $2.15e10^{-10}$ | $1.80 e 10^{-9}$ | $2.10 e 10^{-13}$ | $2.45 e 10^{-15}$ |



Figure 4.2. The sets $X \subseteq \mathbb{R}^2$ (blue, bottom left) and $Y \subseteq \mathbb{R}^2$ (red, top right) used in Example 4.1.2 for R/r = 2. Here, |X| = 4096 and |Y| = 8192.

each dimension. That is, "N=32" in Tables 4.1 and 4.2 refers to 32^2 proxy points, already a large number in two dimensions. Hence, the multidimensional proxy point scheme becomes impractical, even with large matrices, when the dimension of the domain of the underlying kernel function grows beyond two or three, unless the matrix in question is spectacularly large. Similar numbers of proxy points have been required in more traditional proxy point schemes, such as the one studied in [21]; hence, there is evidence that proxy point methods may not be appropriate for kernels acting on large-dimensional data sets.

4.2 General HSS matrix construction and rank bounds

In Section 3.2, we considered an $n \times n$ Toeplitz matrix as a kernel matrix evaluated at the difference of two sets of points in \mathbb{R} , each set of points being the integers [1, n]. In this section, we extend this idea to a general $n \times n$ matrix: we interpret each entry above the diagonal as a two-dimensional kernel matrix evaluated at a function applied to two sets of points in \mathbb{R}^2 . Define the function $p: [1, n] \to \mathbb{R}^2$ by p(j) = ((-n - 1)/2 + j, j - 1) and the sets X, Y = p([1, n]). In a similar manner as before, define $k((w_1, w_2), (z_1, z_2)) = (w_1 + z_1, w_2 - z_2)$. Then the (i, j)th entry of the matrix A is taken to be

$$A_{i,j} = k(p(i), p(j)) = f(p_1(i) + p_1(j), p_2(i) - p_2(j)) = f(-n - 1 + i + j, j - i)$$
(4.1)

for some $f : \mathbb{C}^2 \to \mathbb{C}$. See Figure 4.3.



Figure 4.3. The points $X (\Box, \text{blue})$ and $Y (\circ, \text{yellow})$ in our setup for viewing the top HSS row block (shown in grey) of a general matrix as a kernel matrix, for n = 16. In performing a proxy point approximation, as before, we only apply it to the "far field" of X (shown as squares with white interior) to ensure separation.

As an illustration, for n = 8, we have:

$$A = \begin{pmatrix} k\left((\frac{-7}{2},0),(\frac{-7}{2},0)\right) & k\left((\frac{-5}{2},1),(\frac{-7}{2},0)\right) & k\left((\frac{-3}{2},2),(\frac{-7}{2},0)\right) & k\left((\frac{1}{2},3),(\frac{-7}{2},0)\right) & k\left((\frac{1}{2},4),(\frac{-7}{2},0)\right) & k\left((\frac{3}{2},5),(\frac{-7}{2},0)\right) & k\left((\frac{5}{2},6),(\frac{-7}{2},0)\right) & k\left((\frac{7}{2},7),(\frac{-7}{2},0)\right) \\ k\left((\frac{-7}{2},0),(\frac{-5}{2},1)\right) & k\left((\frac{-5}{2},1),(\frac{-5}{2},1)\right) & k\left((\frac{-3}{2},2),(\frac{-5}{2},1)\right) & k\left((\frac{1}{2},3),(\frac{-5}{2},1)\right) & k\left((\frac{1}{2},4),(\frac{-5}{2},1)\right) & k\left((\frac{3}{2},5),(\frac{-5}{2},1)\right) & k\left((\frac{5}{2},6),(\frac{-7}{2},1)\right) & k\left((\frac{7}{2},7),(\frac{-5}{2},1)\right) \\ k\left((\frac{-7}{2},0),(\frac{-3}{2},2)\right) & k\left((\frac{-5}{2},1),(\frac{1}{2},2)\right) & k\left((\frac{-3}{2},2),(\frac{-3}{2},2)\right) & k\left((\frac{1}{2},3),(\frac{-3}{2},2)\right) & k\left((\frac{1}{2},3),(\frac{-3}{2},2)\right) & k\left((\frac{1}{2},3),(\frac{-3}{2},2)\right) & k\left((\frac{3}{2},5),(\frac{-3}{2},2)\right) & k\left((\frac{5}{2},6),(\frac{-3}{2},2)\right) & k\left((\frac{7}{2},7),(\frac{-3}{2},2)\right) \\ k\left((\frac{-7}{2},0),(\frac{-1}{2},3)\right) & k\left((\frac{-5}{2},1),(\frac{1}{2},4)\right) & k\left((\frac{-3}{2},2),(\frac{-1}{2},3)\right) & k\left((\frac{1}{2},3),(\frac{1}{2},3)\right) & k\left((\frac{1}{2},3),(\frac{1}{2},3)\right) & k\left((\frac{3}{2},5),(\frac{1}{2},3)\right) & k\left((\frac{5}{2},6),(\frac{1}{2},3)\right) & k\left((\frac{7}{2},7),(\frac{1}{2},3)\right) \\ k\left((\frac{-7}{2},0),(\frac{3}{2},5)\right) & k\left((\frac{-5}{2},1),(\frac{3}{2},5)\right) & k\left((\frac{-3}{2},2),(\frac{5}{2},6)\right) & k\left((\frac{1}{2},3),(\frac{5}{2},5)\right) & k\left((\frac{1}{2},3),(\frac{5}{2},5)\right) & k\left((\frac{5}{2},6),(\frac{5}{2},6)\right) & k\left((\frac{7}{2},7),(\frac{5}{2},6)\right) \\ k\left((\frac{-7}{2},0),(\frac{5}{2},6)\right) & k\left((\frac{-5}{2},1),(\frac{5}{2},6)\right) & k\left((\frac{-3}{2},2),(\frac{5}{2},6)\right) & k\left((\frac{1}{2},3),(\frac{5}{2},6)\right) & k\left((\frac{5}{2},6),(\frac{5}{2},6)\right) & k\left((\frac{7}{2},7),(\frac{5}{2},6)\right) \\ k\left((\frac{-7}{2},0),(\frac{5}{2},0),(\frac{5}{2},0)\right) & k\left((\frac{-5}{2},1),(\frac{5}{2},0)\right) & k\left((\frac{-7}{2},2),(\frac{5}{2},0)\right) & k\left((\frac{1}{2},3),(\frac{5}{2},6)\right) & k\left((\frac{1}{2},3),(\frac{5}{2},6)\right) & k\left((\frac{5}{2},6),(\frac{5}{2},6)\right) & k\left((\frac{7}{2},7),(\frac{5}{2},6)\right) \\ k\left((\frac{-7}{2},0),(\frac{5}{2},0)\right) & k\left((\frac{-5}{2},1),(\frac{5}{2},0)\right) & k\left((\frac{-7}{2},2),(\frac{5}{2},0)\right) & k\left((\frac{1}{2},2),(\frac{5}{2},0)\right) & k\left((\frac{1}{2},2),(\frac{5}{2},0)\right) \\ k\left((\frac{-7}{2},0),(\frac{5}{2},0)\right) & k\left((\frac{-5}{2},1),(\frac{5}{2},0)\right) & k\left((\frac{-7}{2},2),(\frac{5}{2},0)\right) \\ k\left((\frac{-7}{2},0),(\frac{5}{2},0)\right) & k\left((\frac{-5}{2},1),(\frac{5}{2},0)\right) & k\left((\frac{-7}{2},2),(\frac$$

which gives

$$A = \begin{pmatrix} f(-7,0) & f(-6,1) & f(-5,2) & f(-4,3) & f(-3,4) & f(-2,5) & f(-1,6) & f(0,7) \\ f(-6,-1) & f(-5,0) & f(-4,1) & f(-3,2) & f(-2,3) & f(-1,4) & f(0,5) & f(1,6) \\ f(-5,-2) & f(-4,-1) & f(-3,0) & f(-2,1) & f(-1,2) & f(0,3) & f(1,4) & f(2,5) \\ f(-4,-3) & f(-3,-2) & f(-2,-1) & f(-1,0) & f(0,1) & f(1,2) & f(2,3) & f(3,4) \\ f(-3,-4) & f(-2,-3) & f(-1,-2) & f(0,-1) & f(1,0) & f(2,1) & f(3,2) & f(4,3) \\ f(-2,-5) & f(-1,-4) & f(0,-3) & f(1,-2) & f(2,-1) & f(3,0) & f(4,1) & f(5,2) \\ f(-1,-6) & f(0,-5) & f(1,-4) & f(2,-3) & f(3,-2) & f(4,-1) & f(5,0) & f(6,1) \\ f(0,-7) & f(1,-6) & f(2,-5) & f(3,-4) & f(4,-3) & f(5,-2) & f(6,-1) & f(7,0) \end{pmatrix}$$

This setup is chosen so that each off-diagonal block of the matrix A corresponds to a kernel matrix evaluated at two sets of points that may be separated by a polydisk. Since any matrix may be written in this way (for example, take f to be a polynomial interpolant for the values of A), and since a lot of matrices do not have low off-diagonal rank, we may expect to require f to belong to a restrictive subclass of functions on $\mathbb{D}^2((0, n/2), n/2)$ to obtain growth bounds similar to 3.2.4. This is indeed the case: the two-variable variable analog of holomorphic, univalent functions on the unit disk are the biholomorphic starlike and convex functions, so our conditions on f need to be adjusted accordingly. A thorough reference on the subject of generalizing classical growth bounds on univalent functions of one complex variable to the several variable case is found in [50].

First, we provide a fairly straightforward generalization of Proposition 3.2.4 for component functions of starlike maps in the several variable case.

Proposition 4.2.1. Let f_1 be a component of a biholomorphic map $f : \mathbb{D}^2((0, n/2), n/2) \to \mathbb{C}^2$ that is starlike with respect to (0, n/2). Then, for $z \in \partial \mathbb{D}^2((0, (n+1)/2), (n/2) - 1)$,

$$|f_1(\mathbf{z})| \le n^2(n-1) \|J_f(0,n/2)\|_2 + |f(0,n/2)|_{\infty}$$

Proof. Define $h: \mathbb{D}^2 \to \mathbb{C}^2$ and $g: \mathbb{D}^2 \to \mathbb{C}^2$ by

$$h(z_1, z_2) = ((n/2)z_1, (n/2)z_2 + (n/2)),$$
 and
 $g(z_1, z_2) = ((f \circ h)(z_1, z_2) - (f \circ h)(0, 0)) (J_{f \circ h}(0, 0))^{-1}$

Then $|g(\mathbf{z})|_{\infty} \leq \frac{|\mathbf{z}|_{\infty}}{(1-|\mathbf{z}|_{\infty})^2}$ by the growth theorem for starlike biholomorphic maps, so for $z \in \partial \mathbb{D}^2 \left((0, (n+1)/2), (n/2) - 1 \right),$

$$\left|(f \circ h)(\mathbf{z}) - (f \circ h)(\mathbf{0})\right|_{\infty} \leq \frac{2|\mathbf{z}|_{\infty}}{(1 - |\mathbf{z}|_{\infty})^2} \left\|J_{f \circ h}(\mathbf{0})\right\|_2.$$

Hence,

$$\begin{aligned} |(f \circ h)(\mathbf{z})|_{\infty} &\leq \frac{2|\mathbf{z}|_{\infty}}{(1-|\mathbf{z}|_{\infty})^{2}} \left\| J_{f \circ h}(\mathbf{0}) \right\|_{2} + |(f \circ h)(\mathbf{0})|_{\infty} \\ &\leq 2n(n-1) \left\| J_{f \circ h}(\mathbf{0}) \right\|_{2} + |(f \circ h)(\mathbf{0})|_{\infty} \\ &= 2n(n-1) \left\| J_{f}(h(\mathbf{0})) J_{h}(\mathbf{0}) \right\|_{2} + |(f \circ h)(\mathbf{0})|_{\infty} \\ &= 2n(n-1) \left\| J_{f}(h(\mathbf{0})) \right\|_{2} \left\| J_{h}(\mathbf{0}) \right\|_{2} + |(f \circ h)(\mathbf{0})|_{\infty} \\ &= n^{2}(n-1) \left\| J_{f}(h(\mathbf{0})) \right\|_{2} + |(f \circ h)(\mathbf{0})|_{\infty} \\ &= n^{2}(n-1) \left\| J_{f}(0,n/2) \right\|_{2} + |f(0,n/2)|_{\infty}. \end{aligned}$$

Since $|f_1(\mathbf{z})| \leq |f(\mathbf{z})|_{\infty}$, the result follows.

Starlikeness is tedious to verify; a well-known characterization was given by Suffridge in [51]. In practice, numerical evidence suggests that this scheme could work in at least a few cases; we give numerical results for two variants of $f(\mathbf{z})$ for which this scheme appears to work in Example 4.2.4 below.

With this growth bound in hand for a appropriate functions f, we get the following lemma in analogy with 3.2.3.

Lemma 4.2.2. Let $A \in \mathbb{C}_{n \times n}$, for n a power of two, have entries $A_{i,j} = f(i + j - n - 1, j - i)$ for some analytic $f : \mathbb{D}^2((0, n/2), n/2) \to \mathbb{C}$. Let \mathcal{I} be the HSS index set for A; let $\mathbf{i} \in \mathcal{I}$; and let *l* be the height of **i**. Define $k((x_1, x_2), (y_1, y_2)) = f(x_1 + y_1, x_2 - y_2)$; let $i \in \hat{i}$; let $j \in [1, n] \setminus i$, and let $N \in \mathbb{N}$. Then

$$\begin{aligned} \left| A(\mathbf{i},\mathbf{j}) - \sum_{1 \le \mathbf{j}_1, \mathbf{j}_2 \le N} \left(\frac{(\sqrt[4]{2})2^{l-1}}{N} \right)^2 \frac{\omega^{\mathbf{j}_1} \omega^{\mathbf{j}_2} k\left(p(\mathbf{j}), \left(z_{\mathbf{j}_1,1}, z_{\mathbf{j}_2,2} \right) \right)}{(z_{\mathbf{j}_1,1} - (p(\mathbf{i}))_1) \left(z_{\mathbf{j}_2,2} - (p(\mathbf{i}))_2 \right)} \right| \\ < \frac{159}{(n^{N/4} - 1)^2} \max_{\mathbf{j} \in [1,n] \setminus \mathbf{i}, \, \mathbf{z} \in \partial F} \left| f\left((p(\mathbf{j}))_1 + z_1, p(\mathbf{j})_2 - z_2 \right) \right|, \end{aligned}$$

where $z_{s,1} = c_1 + (\sqrt[4]{2})2^{l-1}\omega^s$, $z_{s,2} = c_2 + (\sqrt[4]{2})2^{l-1}\omega^s$ for $1 \le s \le N$, and where $c_1 = -(n/2) + \frac{1}{2}(\min(\mathbf{i}) + \max(\mathbf{i}) - 1)$ and $c_2 = \frac{1}{2}(\min(\mathbf{i}) + \max(\mathbf{i})) - 1$.

Proof. This is again a straightforward application of Proposition 4.1.1, where we set

$$X = \{((-n-1)/2 + \mathbf{i}, \mathbf{i} - 1) \mid \mathbf{i} \in \mathbf{i}\},\$$
$$Y = \{((-n-1)/2 + \mathbf{j}, \mathbf{j} - 1) \mid \mathbf{j} \in [1, n] \setminus \mathbf{i}\},\$$

and D and E to be the open balls with center **c** and radii $R = 2^{l+1}$ and $r = 2^{l}$, respectively. We thus get $K = \left(\frac{2\sqrt[4]{2}}{\sqrt[4]{2}-1}\right)^2 < 159$.

Finally, we may use this, as before, to bound the approximation error to the far field of an HSS block of A using the proxy point method as above.

Corollary 4.2.3. Let $A \in \mathbb{C}$ be the $n \times n$ matrix with entries $A_{i,j} = f_1 (i + j - n - 1, j - i)$ for some f_1 a component function of a biholomorphic starlike (with respect to (0, n/2)) map f on $\mathbb{D}^2((0, n/2), n/2)$. Let \mathcal{I} be the HSS index set of A, and let $\mathbf{i} \in \mathcal{I}$. Then

$$\left\|A|_{\mathbf{\hat{i}},[1,n]\setminus\mathbf{i}} - \tilde{A}^{\mathbf{j},N}\right\|_{F} \le \left(\frac{n^{2}}{4}\right) \left(\frac{159}{(2^{N/4} - 1)^{2}}\right) \left(n^{3} \left\|J_{f}\left(0, n/2\right)\right\|_{2} + |f\left(0, n/2\right)|_{\infty}\right).$$

Proof. By Lemma 4.2.2, the maximum modulus principle, and Proposition 4.2.1, we again have that for each $1 \le u \le |\mathbf{\hat{i}}|$ and $1 \le v \le |[1, n] \setminus \mathbf{i}|$,

$$\begin{split} \left| \left(A|_{\mathbf{\hat{i}},[1,n]\setminus\mathbf{\hat{i}}} \right)_{u,v} - \left(\tilde{A}^{\mathbf{j},N} \right)_{u,v} \right| &\leq 159 \frac{\max_{z \in \partial \mathbb{D}^2((0,(n+1)/2),n/2-1)} |f(z)|}{(2^{N/4}-1)^2} \\ &\leq \frac{159}{(2^{N/4}-1)^2} \left(n^3 \left\| J_f(0,n/2) \right\|_2 + |f(0,n/2)|_{\infty} \right) \end{split}$$

Since $|\mathbf{\hat{i}}|, |[1, n] \setminus \mathbf{i}| \leq \frac{n}{2}$, the result follows by summing over all u and v.

Example 4.2.4. In Table 4.3, we show two examples of f (as in Equation 4.1) for which the proxy point approximation scheme detailed above yields a relatively small error. In particular, in this example, we set n = 8192 and f to be one of

$$g_1(z_1, z_2) = \sqrt{((z_1 + z_2 - 1)/2 + 1024)^2 + ((z_2 - z_1 - 1)/2)^2}$$
 or

$$g_2(z_1, z_2) = 1/(4096 + z_1) + 1/(4096 + z_2).$$

Note that using Theorem 3 in [51], it can be shown that g_2 is a component of a starlike map. To see this, for example, define

$$F(z_1, z_2) = \left(\frac{1}{4096 + z_1} + \frac{1}{4096 + z_2}, \frac{1}{4096 + z_1} - \frac{1}{4096 + z_2}\right).$$

Since this is a linear transformation of convex coordinate functions on $\mathbb{D}^2((0, n/2), n/2)$, it is convex, hence starlike. Thus, at least in the case of g_2 , Corollary 4.2.3 applies to give an asymptotic guarantee of the efficacy of our approximation scheme.

Table 4.3. The relative Frobenius norm errors of the approximation to the far field of the topmost HSS row block of an $n \times n$ matrix A, viewed as a kernel matrix in the sense of Equation 4.1 with kernels g_1 and g_2 , with n = 8192 using N proxy points.

| Kernel: | N = 32 | N = 48 | N = 56 |
|---------|------------------|------------------|-------------------|
| g_1 | $3.88e10^{-6}$ | $1.24 e 10^{-8}$ | $7.22 e 10^{-10}$ |
| g_2 | $4.64 e 10^{-6}$ | $1.47 e 10^{-8}$ | $8.51e10^{-10}$ |

Example 4.2.4 gives good evidence that we are able to, apply the sublinear HSS construction technique outlined in Chapter 3 to the "kernel matrix" defined by the function g_2 , with points defined via p as in (4.1). More theoretically, Lemma 4.2.2 suggests that we may perform a sublinear HSS construction for matrices A that are defined by a component function f (as in Equation 4.1) of any biholomorphic starlike map on the relevant region. The HSS construction using a hybrid approximation as in Section 3.2 works exactly as before, so it is not detailed here. Again, the key property enabling such a construction is the identical geometry between the proxy points involved and each appropriate subset of the points in X. Although the specific functions of Example 4.2.4 do not have significance in any application that we are aware of, the broader techniques illustrated may find application.

4.3 Further extensions

We may be able to generalize some ideas in this chapter to improve the bounds we obtained from Proposition 4.1.1. In particular, we may be able to apply a linear transformation to the sets X and Y in order to obtain a better value for R/r in the construction of our bounding hyperdisk, at the possible expense of increasing the maximum obtained by an accordingly-transformed value of f. Such techniques, if successful, would in particular help broaden the applicability of the scheme outlined in Section 4.2. Furthermore, we may be able to apply a similar multidimensional proxy point method to kernel functions of high-dimensional data via the tensor decomposition thereof, as long as such decompositions involve well-separated points.

Furthermore, we may be able to explore ways in which we can redefine f, X, and Y as above using transformations other than a linear transformation. As an example in one dimension, we may think of the Cauchy matrix studied in Chapter 2 instead as the "exponentialdifference" kernel $1/(e^n = e^m)$, applied to suitable values $n, m \in \mathbb{R}$. The Cauchy contour would then change compared to before, and so would the maximum taken by the inequality in Proposition 3.1.1. (Such more general contours were used in the Cauchy FMM scheme of [22].) The question is then whether or not the bound in Proposition 3.1.1 may be made tighter in this manner. Numerical evidence suggests that it may not, at least in the onevariable case, but there may be a difference in several variables.

REFERENCES

[1] L. Trefethen and D. Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[2] I. S. Duff and J. K. Reid, "The multifrontal solution of indefinite sparse symmetric linear," vol. 9, no. 3, pp. 302–325, 1983.

[3] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Second. Society for Industrial and Applied Mathematics, 2003.

[4] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, "Superfast multifrontal method for large structured linear systems of equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1382–1411, 2010.

[5] S. Massei, L. Robol, and D. Kressner, "Hm-toolbox: Matlab software for hodlr and hss matrices," *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. C43–C68, 2020.

[6] R. Vandebril, M. Van Barel, and N. Mastronardi, "Matrix computations and semiseparable matrices: Linear systems," in Johns Hopkins University Press, 2007, pp. 481–501.

[7] M. Udell and A. Townsend, "Why are big data matrices approximately low rank?" *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 144–160, 2019.

[8] S. Chandrasekaran, M. Gu, and T. Pals, "A fast ulv decomposition solver for hierarchically semiseparable representations," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 3, pp. 603–622, 2006.

[9] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, "A superfast algorithm for toeplitz systems of linear equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 4, pp. 1247–1266, 2008.

[10] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, "Fast algorithm for hierarchically semiseparable matrices," *Numerical Linear Algebra with Applications*, vol. 17, no. 6, pp. 953–976, 2010.

[11] S. Ambikasaran, "Fast algorithms for dense numerical linear algebra and applications," Ph.D. dissertation, Stanford University, 2013.

[12] Y. Xi and J. Xia, "On the stability of some hierarchical rank structured matrix algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1279–1303, 2016.

[13] X. Ou and J. Xia, "Superdc: Stable superfast divide-and-conquer eigenvalue decomposition," submitted. [Online]. Available: https://arxiv.org/abs/2108.04209.

[14] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing qr factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[15] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, "A theory of pseudoskeleton approximations," *Linear Algebra and its Applications*, vol. 261, no. 1, pp. 1–21, 1997.

[16] P. G. Martinsson and V. Rokhlin, "A fast direct solver for boundary integral equations in two dimensions," *Journal of Computational Physics*, vol. 205, no. 1, pp. 1–23, 2005.

[17] X. Xing and E. Chow, "Interpolative decomposition via proxy points for kernel matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 41, no. 1, pp. 221–243, 2020.

[18] X. Ye, J. Xia, and L. Ying, "Analytical low-rank compression via proxy point selection," *SIAM Journal on Matrix Analysis and Applications*, vol. 41, no. 3, pp. 1059–1085, 2020.

[19] L. Ying, "Short note: A kernel independent fast multipole algorithm for radial basis functions," *Journal of Computational Physics*, vol. 213, no. 2, pp. 451–457, 2006.

[20] L. Ying, G. Biros, and D. Zorin, "A kernel-independent adaptive fast multipole algorithm in two and three dimensions," *Journal of Computational Physics*, vol. 196, no. 2, pp. 591–626, 2004.

[21] X. Xing and E. Chow, "Error analysis of an accelerated interpolative decomposition for 3d laplace problems," *Applied and Computational Harmonic Analysis*, vol. 49, no. 1, pp. 316–327, 2020.

[22] P. D. Létourneau, C. Cecka, and E. Darve, "Cauchy fast multipole method for general analytic kernels," *SIAM Journal on Scientific Computing*, vol. 36, no. 2, A396–A426, 2014.

[23] Y. Xi, J. Xia, S. Cauley, and V. Balakrishnan, "Superfast and stable structured solvers for toeplitz least squares via randomized sampling," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 44–72, 2014.

[24] I. Gohberg, T. Kailath, and V. Olshevsky, "Fast gaussian elimination with partial pivoting for matrices with displacement structure," *Mathematics of Computation*, vol. 64, no. 212, pp. 1557–1576, 1995.

[25] G. Heinig, "Inversion of generalized cauchy matrices and other classes of structured matrices," in *Linear Algebra for Signal Processing*, A. Bojanczyk and G. Cybenko, Eds., Springer New York, 1995, pp. 63–81.

[26] T. Kailath, S. Y. Kung, and M. Morf, "Displacement ranks of matrices and linear equations," *Journal of Mathematical Analysis and Applications*, vol. 68, no. 2, pp. 395–407, 1979.

[27] V. Pan, "On computations with dense structured matrices," *Mathematics of Computation*, vol. 55, no. 191, pp. 179–190,

[28] B. Beckermann and A. Townsend, "On the singular values of matrices with displacement structure," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 4, pp. 1227–1248, 2017.

[29] P. G. Martinsson, V. Rokhlin, and M. Tygert, "A fast algorithm for the inversion of general toeplitz matrices," *Computers and Mathematics with Applications*, vol. 50, no. 5, pp. 741–752, 2005.

[30] J. Xia, Y. Xi, and M. Gu, "A superfast structured solver for toeplitz linear systems via randomized sampling," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 3, pp. 837–858, 2012.

[31] J. Vogel, J. Xia, S. Cauley, and V. Balakrishnan, "Superfast divide-and-conquer method and perturbation analysis for structured eigenvalue solutions," *SIAM Journal on Scientific Computing*, vol. 38, no. 3, A1358–A1382, 2016.

[32] Y. Xi, J. Xia, and R. Chan, "A fast randomized eigensolver with structured ldl factorization update," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 974– 996, 2014.

[33] J. Xia, "On the complexity of some hierarchical structured matrix algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 2, pp. 388–410, 2012.

[34] M. Lepilov and J. Xia, "Hyperfast has construction of some cauchy matrices," submitted.

[35] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, 1987.

[36] W. Hackbusch, "A sparse matrix arithmetic based on h-matrices. part i: Introduction to h-matrices.," *Computing*, vol. 62, pp. 89–108, 1999.

[37] E. Liberty, F. Woolfe, P. G. Martinsson, V. Rokhlin, and M. Tygert, "Randomized algorithms for the low-rank approximation of matrices," *Proceedings of the National Academy of Sciences*, vol. 104, no. 51, pp. 20167–20172, 2007.

[38] E. Corona, P. Martinsson, and D. Zorin, "An o(n) direct solver for integral equations on the plane," *Applied and Computational Harmonic Analysis*, vol. 38, no. 2, pp. 284–317, 2015.

[39] K. L. Ho and L. Ying, "Hierarchical interpolative factorization for elliptic operators: Integral equations," *Communications on Pure and Applied Mathematics*, vol. 69, no. 7, pp. 1314–1353, 2015.

[40] J. Xia, "Multi-layer hierarchical structures," CSIAM Transactions on Applied Mathematics, vol. 2, no. 2, pp. 263–296, 2021.

[41] X. Xing, *Strong rank-revealing qr decomposition*, https://www.mathworks.com/matlab central/fileexchange/69139-strong-rank-revealing-qr-decomposition, version 1.0.3, 2021.

[42] M. Gu, "Stable and efficient algorithms for structured systems of linear equations," *SIAM Journal on Matrix Analysis and Applications*, vol. 19, no. 2, pp. 279–306, 1998.

[43] A. K. Jain, "Fundamentals of digital image processing," in Prentice-Hall, 1988, pp. 25–26.

[44] W. Burger and M. J. Burge, "The discrete fourier transform in 2d," in *Digital Image Processing: An Algorithmic Introduction Using Java.* Springer London, 2016, pp. 481–501.

[45] M. Lepilov and J. Xia, "Analysis and applications of the proxy point method for matrices defined by analytic kernels," preprint.

[46] L. N. Trefethen and J. A. C. Weideman, "The exponentially convergent trapezoidal rule," *SIAM Review*, vol. 56, no. 3, pp. 385–458, 2014.

[47] R. Bilato, O. Maj, and M. Brambilla, "An algorithm for fast hilbert transform of real functions," *Advances in Computational Mathematics*, vol. 40, no. 5–6, pp. 1159–1168, 2014.

[48] H. Wang, K. Wang, and T. Sircar, "A direct o(nlog2n) finite difference method for fractional diffusion equations," *Journal of Computational Physics*, vol. 229, no. 21, pp. 8095–8104, 2010.

[49] D. Rubin, A. Townsend, and H. Wilber, "Bounding zolotarev numbers using faber rational functions," preprint. [Online]. Available: https://arxiv.org/abs/1911.11882.

[50] S. Gong, "The growth theorem for holomorphic starlike mappings," in *Convex and Starlike Mappings in Several Complex Variables*. Springer Netherlands, 1998, pp. 78–94.

[51] T. J. Suffridge, "The principle of subordination applied to functions of several variables," *Pacific Journal of Mathematics*, vol. 33, no. 1, pp. 241–248, 1970.