

FAST APPROXIMATE COMPUTATIONS WITH CAUCHY MATRICES AND POLYNOMIALS

VICTOR Y. PAN

ABSTRACT. Multipoint polynomial evaluation and interpolation are fundamental for modern symbolic and numerical computing. The known algorithms solve both problems over any field of constants in nearly linear arithmetic time, but the cost grows to quadratic for numerical solution. We fix this discrepancy: our new numerical algorithms run in nearly linear arithmetic time. At first we restate our goals as the multiplication of an $n \times n$ Vandermonde matrix by a vector and the solution of a Vandermonde linear system of n equations. Then we transform the matrix into a Cauchy structured matrix with some special features. By exploiting them, we approximate the matrix by a generalized hierarchically semiseparable matrix, which is a structured matrix of a different class. Finally we accelerate our solution to the original problems by applying the Fast Multipole Method to the latter matrix. Our resulting numerical algorithms run in nearly optimal arithmetic time when they perform the above fundamental computations with polynomials, Vandermonde matrices, transposed Vandermonde matrices, and a large class of Cauchy and Cauchy-like matrices. Some of our techniques may be of independent interest.

1. INTRODUCTION

1.1. Background and our progress. Multipoint polynomial evaluation and interpolation are fundamental for modern symbolic and numerical computing. The known FFT-based algorithms run in nearly linear arithmetic time, but need quadratic time if the precision of computing is restricted, e.g., to the IEEE standard double precision (cf. [BF00], [BEGO08]). Our algorithms solve the problems in nearly linear arithmetic time even under such a restriction.

At first we restate the original tasks as the problems of multiplication of a Vandermonde matrix by a vector and the solution of a nonsingular Vandermonde linear system of equations, then transform the input matrix into a matrix with the structure of Cauchy type, and finally apply the numerically stable FMM to a generalized HSS matrix that approximates the latter matrix.¹ “Historically HSS representation is just a special case of the representations commonly exploited in the FMM literature” [CDG06]. We refer the reader to the books [B10], [VVM], [EGH13], and the bibliographies therein for the FMM and the HSS matrices.

Received by the editor June 7, 2015 and, in revised form, April 20, 2016 and July 6, 2016.

2010 *Mathematics Subject Classification.* Primary 12Y05, 15A04, 47A65, 65D05, 68Q25.

Key words and phrases. Polynomial evaluation, rational evaluation, interpolation, Vandermonde matrices, transformation of matrix structures, Cauchy matrices, fast multipole method, HSS matrices, matrix compression.

Some results of this paper have been presented at ILAS’2013, Providence, RI, 2013, at CASC’2013, Berlin, Germany, 2013, and at CSR’2014, Moscow, Russia, 2014.

¹“HSS” and “FMM” are the acronyms for “Hierarchically Semiseparable” and “Fast Multipole Method”.

Our resulting fast algorithms apply to the following computational problems:

- multipoint polynomial evaluation and interpolation,
- multiplication by a vector of a Vandermonde matrix, its transpose, and, more generally, matrices with the structures of Cauchy or Vandermonde type,
- the solution of a linear system of equations with these coefficient matrices,
- rational interpolation and multipoint evaluation associated with Cauchy matrix computations.

Some of our techniques can be of independent interest (cf. their extension in [Pa16]).

As in the papers [MRT05], [CGS07], [XXG12], and [XXCB14], we count arithmetic operations in the field \mathbb{C} of complex numbers with no rounding errors, but our algorithms are essentially reduced to application of the celebrated algorithms of FFT and FMM, having stable numerical performance.

1.2. Related works and our techniques. *Our progress can be viewed as a new demonstration of the power of combining the transformation of matrix structures of [P90] with the FMM/HSS techniques.*

The paper [P90] has proposed some efficient techniques for the transformation of the four most popular matrix structures of Toeplitz, Hankel, Cauchy, and Vandermonde types into each other and then showed that these techniques enable us to readily extend any efficient algorithm for the inversion of a matrix having one of these structures to efficient inversion of the matrices having structures of the three other types. The papers [PSLT93] and [PZHY97] have extended these techniques to the acceleration of multipoint polynomial evaluation, but have not invoked the FMM and achieved only limited progress. Short Section 9.2 of [PRT92] has pointed out some potential benefits of combining FMM with the algorithm of the paper [G88], but has not developed that idea. The papers [P95] and [DGR96] applied FMM and some other advanced techniques in order to accelerate approximate polynomial evaluation at a set of real points.

The closest neighbors of our present study are the papers [MRT05], [CGS07], [XXG12], [XXCB14], and [P15]. The former four papers approximate the solution of Toeplitz, Hankel, Toeplitz-like, and Hankel-like linear systems of equations in nearly linear arithmetic time, versus the cubic time of the classical numerical algorithms and the previous record quadratic time of [GKO95]. All five papers [GKO95], [MRT05], [CGS07], [XXG12], and [XXCB14] begin with the transformation of an input matrix into a Cauchy-like one, by specializing the cited technique of [P90]. Then [GKO95] continued by exploiting the invariance of the Cauchy structure in row interchange, while the other four papers apply the numerically stable FMM in order to operate efficiently with HSS approximations of the basic Cauchy matrix.

We incorporate the powerful FMM/HSS techniques, but extend them nontrivially. The papers [GKO95], [MRT05], [CGS07], [XXG12], and [XXCB14] handle just the special Cauchy matrix $C = (\frac{1}{s_i - t_j})_{i,j=0}^{m-1,n-1}$ for which $m = n$, $\{s_0, \dots, s_{n-1}\}$ is the set of the n th roots of unity and $\{t_0, \dots, t_{n-1}\}$ is the set of the other $2n$ th roots of unity. Our fast Vandermonde multipliers and solvers bring us to a subclass of Cauchy matrices $C = (\frac{1}{s_i - t_j})_{i,j=0}^{m-1,n-1}$ rather than to a single matrix: we still assume that the knots t_0, \dots, t_{n-1} are equally spaced on the unit circle, but impose

no restriction on the knots s_0, \dots, s_{m-1} and arrive at the matrices

$$(1.1) \quad C_{\mathbf{s},f} = \left(\frac{1}{s_i - f\omega^j} \right)_{i,j=0}^{m-1,n-1},$$

for any complex numbers f, s_0, \dots, s_{m-1} and $\omega = \exp(2\pi\sqrt{-1}/n)$ denoting a primitive n th root of unity.

We call the matrices $C_{\mathbf{s},f}$ *CV matrices*, link them to Vandermonde matrices, and devise efficient approximation algorithms that multiply a CV matrix by a vector, solve a nonsingular CV linear system of equations, and hence perform multipoint polynomial evaluation and interpolation. In order to achieve this progress, we work with extended HSS matrices, associated with CV matrices via a proper partition of the complex plane: we bound the numerical rank of the off-block-tridiagonal blocks (rather than the off-block-diagonal blocks, as is customary) and allow distinct rectangular blocks to share row indices. Extension of the FMM/HSS techniques to such matrix classes was not straightforward and required additional care.

The paper [P15] revisited the method of the transformation of matrix structures (traced back to [P90]), recalled its techniques in some detail, extended them, and finally outlined our present approach to polynomial interpolation and multipoint evaluation in order to demonstrate the power of that method once again. The paper included only one half of a page to HSS matrices and about as much to the reduction of the polynomial evaluation and interpolation to computations with CV matrices. No room has been left for the description of nontrivial computations with generalized HSS matrices (having cyclic block tridiagonal part), to which the original problems are reduced. Furthermore the competing fast algorithms for polynomial and rational interpolation and multipoint evaluation of [MB72], [H72], and [GGS87] have not been cited.

We fill this void by describing in some detail the omitted algorithms for generalized HSS computation, by linking polynomial and rational interpolation and multipoint evaluation to CV matrices, by demonstrating the inherent numerical instability of the algorithms of [MB72], [H72], and [GGS87], and by presenting some numerical tests, in particular for comparison of numerical stability of our algorithms with that of [MB72]. Also we more fully and more clearly cover the approximation of CV matrices by generalized HSS matrices.

1.3. Organization of our paper. In the next section we recall some basic results for matrix computations. In Section 3 we recall the problems of polynomial and rational evaluation and interpolation and represent them in terms of Vandermonde, Cauchy, and CV matrices. Sections 2 and 3 (on the Background) make up Part I of our paper.

Sections 4 and 5 (on the extended HSS matrices) make up Part II, where at first we recall the known algorithms for fundamental computations with HSS matrices and then extend the algorithms to generalized HSS matrices having cyclic block tridiagonal part. Part II can be read independently of Section 3.

Sections 6 and 7 (on computations with the CV matrices and extensions) make up Part III of the paper. In Section 6 we approximate a CV matrix by generalized HSS matrices and estimate the complexity of the resulting numerical computations with CV matrices. In Section 7 we comment on the extensions and implementation of our algorithms, in particular the extension to computations with Vandermonde

matrices and polynomials. The results of Section 6 imply our main results because we have already reduced polynomial interpolation and multipoint evaluation to computations with CV matrices in Part I and have elaborated upon fast computations with generalized HSS matrices in Part II.

Part III uses Section 3.2 and equations (3.2) and (3.4) of Part I (which support the cited reduction to CV matrices) and Theorem 5.1 and Corollary 5.1 of Part II (where we estimate the cost of computations with generalized HSS matrices), but otherwise can be read independently of Parts I and II.

Sections 8 and 9 make up Part IV of the paper. In Section 8 we report the results of our numerical tests. In Section 9 we briefly summarize our study.

Part I. Background

2. DEFINITIONS AND AUXILIARY RESULTS

2.1. Some basic definitions for matrix computations. $O = O_{m,n}$ is the $m \times n$ matrix filled with zeros. $I = I_n$ is the $n \times n$ identity matrix. M^T is the transpose of a matrix M , M^H is its Hermitian transpose.

$\text{diag}(B_0, \dots, B_{k-1}) = \text{diag}(B_j)_{j=0}^{k-1}$ is a $k \times k$ block diagonal matrix with diagonal blocks B_0, \dots, B_{k-1} .

Both $(B_0 \dots B_{k-1})$ and $(B_0 \mid \dots \mid B_{k-1})$ denote a $1 \times k$ block matrix with k blocks B_0, \dots, B_{k-1} .

$\|M\| = \|M\|_2$ denotes the spectral norm of a matrix M .

For an $m \times n$ matrix $M = (m_{i,j})_{i,j=0}^{m-1,n-1}$, write $|M| = \max_{i,j} |m_{i,j}|$, and so $\|M\| \leq \sqrt{mn} |M|$, but for a set \mathcal{S} we write $|\mathcal{S}|$ to denote its cardinality.

An $m \times n$ matrix U is *unitary* if $U^H U = I_n$ or $U U^H = I_m$, and then $\|U\| = 1$.

“ \ll ” stands for “much less” quantified in context.

2.2. Submatrices, rank, and generators. An $m \times n$ matrix M has a nonunique *generating pair* (F, G^T) of a length ρ if $M = F G^T$ for two matrices F of size $m \times \rho$ and G of size $n \times \rho$. The minimum length of a generating pair of a matrix is equal to its rank.

$\mathcal{R}(B)$ and $\mathcal{C}(B)$ are the index sets of the rows and columns of its submatrix B , respectively. For two sets $\mathcal{I} \subseteq \{1, \dots, m\}$ and $\mathcal{J} \subseteq \{1, \dots, n\}$, define the submatrix $B = M(\mathcal{I}, \mathcal{J}) = (m_{i,j})_{i \in \mathcal{I}, j \in \mathcal{J}}$ such that $\mathcal{R}(B) = \mathcal{I}$ and $\mathcal{C}(B) = \mathcal{J}$. Write $M(\mathcal{I}, \cdot) = M(\mathcal{I}, \mathcal{J})$ if $\mathcal{J} = \{1, \dots, n\}$. Write $M(\cdot, \mathcal{J}) = M(\mathcal{I}, \mathcal{J})$ if $\mathcal{I} = \{1, \dots, m\}$.

Theorem 2.1. *A matrix M has rank at least ρ if and only if it has a nonsingular $\rho \times \rho$ submatrix $M(\mathcal{I}, \mathcal{J})$.*

If $\text{rank}(M) = \rho$, then $M = M(\cdot, \mathcal{I})M(\mathcal{I}, \mathcal{J})^{-1}M(\mathcal{J}, \cdot)$.

The theorem defines two generating pairs $(M(\cdot, \mathcal{I}), M(\mathcal{I}, \mathcal{J})^{-1}M(\mathcal{J}, \cdot))$ and $(M(\cdot, \mathcal{I})M(\mathcal{I}, \mathcal{J})^{-1}, M(\mathcal{J}, \cdot))$ and a *generating triple* $(M(\cdot, \mathcal{I}), M(\mathcal{I}, \mathcal{J})^{-1}, M(\mathcal{J}, \cdot))$ of a length ρ for a matrix M . We call such pairs and triples *generators*. One can obtain some generators of the minimum length for a given matrix by computing its SVD $U \Sigma V$ or its less costly rank revealing factorizations such as ULV and URV factorizations in [CGS07], [XXG12], and [XXCB14], where the matrices U and V are unitary, Σ is diagonal, and L and R are triangular (cf. [GL13, Section 5.6.8]). For efficient alternative techniques, some of which use randomization or heuristics; see [GOS08], [GT01], [HMT11], [LWMRT], [M11], [M11a], [PQY15], [T00], [W14], [XXG12], and the references therein.

2.3. Small-norm approximation and perturbation. Hereafter we deal with perturbations within a positive tolerance ξ . (One may think of machine epsilon, but in this paper we just assume that ξ is small in context.)

A matrix \tilde{M} is a ξ -approximation of a matrix M if $\|\tilde{M} - M\| \leq \xi\|M\|$.

A ξ -generator of a matrix M is a generator of its ξ -approximation.

The ξ -rank of a matrix M is the integer $\min_{\|\tilde{M}-M\|\leq\xi\|M\|} \text{rank}(\tilde{M})$.

A matrix M is *ill-conditioned* if its rank exceeds its numerical rank.

3. POLYNOMIAL AND RATIONAL EVALUATION AND INTERPOLATION
AS OPERATIONS WITH STRUCTURED MATRICES

3.1. Four classes of structured matrices. Cauchy and Vandermonde matrices. Recall the four classes of highly popular structured matrices, that is, *Toeplitz* matrices $T = (t_{i-j})_{i,j=0}^{m-1,n-1}$, *Hankel* matrices $H = (h_{i+j})_{i,j=0}^{m-1,n-1}$, *Vandermonde* matrices $V = V_{\mathbf{s}} = (s_i^j)_{i,j=0}^{m-1,n-1}$, and *Cauchy* matrices $C = C_{\mathbf{s},\mathbf{t}} = \left(\frac{1}{s_i-t_j}\right)_{i,j=0}^{m-1,n-1}$. (Some authors call the transpose V^T a Vandermonde matrix.) The mn entries of such a structured $m \times n$ matrix are defined by at most $m + n$ parameters.

These classes have been extended to the four more general classes of matrices having structures of Toeplitz, Hankel, Vandermonde, and Cauchy types. Each such an $m \times n$ matrix is naturally defined by its *displacement generator* FH where F and G are $m \times d$ and $d \times n$ matrices, respectively, and where $d \ll \min\{m, n\}$, that is, $\min\{m, n\}$ exceeds greatly the integer d (cf. [P01], [P15]).

We mostly work with Vandermonde and Cauchy matrices and next recall some of their basic properties.

The scalars $s_0, \dots, s_{m-1}, t_0, \dots, t_{n-1}$ define the Vandermonde and Cauchy matrices $V_{\mathbf{s}}$ and $C_{\mathbf{s},\mathbf{t}}$, and we call them *knots*. If we shift the knots of a Cauchy matrix or scale them by a constant, we arrive at a Cauchy matrix again: $aC_{as,at} = C_{\mathbf{s},\mathbf{t}}$ for $a \neq 0$ and $C_{\mathbf{s}+a\mathbf{e},\mathbf{t}+a\mathbf{e}} = C_{\mathbf{s},\mathbf{t}}$ for $\mathbf{e} = (1, \dots, 1)^T$.

Theorem 3.1. (i) An $m \times n$ Vandermonde matrix $V_{\mathbf{s}} = (s_i^j)_{i,j=0}^{m-1,n-1}$ has full rank if and only if all m knots s_0, \dots, s_{m-1} are distinct. (ii) An $m \times n$ Cauchy matrix $C_{\mathbf{s},\mathbf{t}} = \left(\frac{1}{s_i-t_j}\right)_{i,j=0}^{m-1,n-1}$ is well defined and has full rank if and only if all its $m + n$ knots are distinct.

The four cited matrix structures have quite distinct features. In particular the matrix structure of Cauchy type is invariant in row and column interchange, in contrast to the structures of Toeplitz and Hankel types. This structure is stable in shifting and scaling its basic knots unlike the structure of Vandermonde type.

The paper [P90], however, has transformed the matrices of any of the four classes into the matrices of the three other classes simply by means of multiplication by Hankel, Vandermonde, and transposed or inverse Vandermonde matrices. Then the paper has showed that such transforms *readily extend any efficient matrix inversion algorithm for matrices of one of the four classes to the matrices of the three other classes*, and similarly for the computation of determinants and the solution of linear systems of equations.

Presently we apply a simple specialization of this general technique for devising efficient approximation algorithms for Vandermonde matrix computations linked to polynomial evaluation and interpolation.

3.2. Four computational problems.

Problem 1. Multipoint polynomial evaluation or Vandermonde-by-vector multiplication.

INPUT: $m + n$ complex scalars $p_0, \dots, p_{n-1}; s_0, \dots, s_{m-1}$.

OUTPUT: m complex scalars v_0, \dots, v_{m-1} satisfying $v_i = p(s_i)$ for $p(x) = p_0 + p_1x + \dots + p_{n-1}x^{n-1}$ and $i = 0, \dots, m-1$ or equivalently $V\mathbf{p} = \mathbf{v}$ for $V = V_{\mathbf{s}} = (s_i^j)_{i,j=0}^{m-1,n-1}$, $\mathbf{p} = (p_j)_{j=0}^{n-1}$, and $\mathbf{v} = (v_i)_{i=0}^{m-1}$.

Problem 2. Polynomial interpolation or the solution of a Vandermonde linear system.

INPUT: $2n$ complex scalars $v_0, \dots, v_{n-1}; s_0, \dots, s_{n-1}$, the last n of them distinct.

OUTPUT: n complex scalars p_0, \dots, p_{n-1} satisfying the above equations for $m = n$.

Problem 3. Multipoint rational evaluation or Cauchy-by-vector multiplication.

INPUT: $2m + n$ complex scalars $s_0, \dots, s_{m-1}; t_0, \dots, t_{n-1}; v_0, \dots, v_{m-1}$.

OUTPUT: m complex scalars v_0, \dots, v_{m-1} satisfying $v_i = \sum_{j=0}^{n-1} \frac{u_j}{s_i - t_j}$ for $i = 0, \dots, m-1$ or equivalently $C\mathbf{u} = \mathbf{v}$ for $C = C_{\mathbf{s},\mathbf{t}} = \left(\frac{1}{s_i - t_j} \right)_{i,j=0}^{m-1,n-1}$, $\mathbf{u} = (u_j)_{j=0}^{n-1}$, and $\mathbf{v} = (v_i)_{i=0}^{m-1}$.

Problem 4. Rational interpolation or the solution of a Cauchy linear system of equations.

INPUT: $3n$ complex scalars $s_0, \dots, s_{n-1}; t_0, \dots, t_{n-1}; v_0, \dots, v_{n-1}$, the first $2n$ of them distinct.

OUTPUT: n complex scalars u_0, \dots, u_{n-1} satisfying the above equations for $m = n$.

3.3. The arithmetic complexity of Problems 1–4. The algorithm of [MB72] solves Problem 1 by using $O((m+n)\log^2(n)\log(\log(n)))$ arithmetic operations. This complexity bound has been extended to the solution of Problems 2 in [H72], 3 in [GGS87], and 4 (see equation (3.1) below) and is within a factor of $\log(n)\log(\log(n))$ from the optimum [BM75].

The cited algorithms supporting this bound require extended precision of computing and fail already for the input polynomials of moderate degree if the precision is restricted to the IEEE standard double precision (cf. Table 8.8). The approach relies heavily on computing with extended precision. Already the fast polynomial division algorithm requires computations with high precision for the worst case input, and the problem is aggravated in the recursive fan-in processes of polynomial multiplication and division in the algorithms of [MB72], [H72], and [GGS87]. Moreover, the following argument demonstrates that we must add at least n bits of precision when these algorithms compute the Lagrange auxiliary polynomial with the roots s_0, \dots, s_{n-1} .

Problem 5. Computation of the polynomial coefficients from its roots.

INPUT: n complex scalars s_0, \dots, s_{n-1} .

OUTPUT: the coefficients of the polynomial $l(x) = \prod_{i=0}^{n-1} (x - s_i)$.

In order to observe the need for the precision increase, notice that the constant coefficient has absolute value $\prod_{j=0}^{n-1} |s_j|$, which turns into 2^n if, say, $s_i = 2$ for all

i , but the coefficient of $x^{\lfloor n/2 \rfloor}$ has the order of 2^n even if $s_i = 1$ for all i . The restriction of using bounded (e.g., double) precision of computing rules out using the cited fast algorithms, and the known double precision algorithms for Problems 1–4 require quadratic arithmetic time (cf. [BF00], [BEGO08]).

This pessimistic outcome, however, does not apply to the important special case where the knots s_i are the n th roots of 1, that is, where $s_i = \omega^i$ for $\omega = \omega_n = \exp(2\pi\sqrt{-1}/n)$, $i = 0, \dots, n - 1$. In this case, $V_s = (\omega^{ij})_{i,j=0}^{m-1, n-1}$ and Problems 1 (for $m = n$) and 2 turn into the computation of the forward and inverse *discrete Fourier transforms*, respectively. Hereafter we use the acronyms *DFT* and *IDFT* and write $\Omega = \frac{1}{\sqrt{n}}(\omega^{ij})_{i,j=0}^{n-1}$. Notice that $\Omega = \Omega^T$ and $\Omega^H = \Omega^{-1} = \frac{1}{\sqrt{n}}(\omega^{-ij})_{i,j=0}^{n-1}$ are unitary matrices. Based on FFT, one can perform the DFT and IDFT, that is, can solve Problems 1 and 2 in this special case, by using bounded precision of computing and involving only $O(n \log(n))$ arithmetic operations [P01, Problem 2.4.2].

3.4. Cauchy–Vandermonde links and their impact on Problems 1 and 2.

The following equation, traced to [K68] on [P01, p. 110], links Problems 1 and 2 to Cauchy matrices,

$$(3.1) \quad C_{\mathbf{s}, \mathbf{t}} = \text{diag}(l(s_i)^{-1})_{i=0}^{m-1} V_{\mathbf{s}} V_{\mathbf{t}}^{-1} \text{diag}(l'(t_j))_{j=0}^{n-1}, \quad l(x) = \prod_{j=0}^{n-1} (x - t_j).$$

For $\mathbf{t} = f \cdot (\omega^j)_{j=0}^{n-1}$, $f \neq 0$, the knots t_j are the scaled n th roots of 1, $l(x) = x^n - f^n$, $l'(x) = nx^{n-1}$, $V_{\mathbf{t}} = \sqrt{n} \Omega \text{diag}(f^j)_{j=0}^{n-1}$, $V_{\mathbf{t}}^{-1} = \frac{1}{\sqrt{n}} \text{diag}(f^{-j})_{j=0}^{n-1} \Omega^H$. Likewise for $\mathbf{s} = e \cdot (\omega^i)_{i=0}^{m-1}$, $e \neq 0$, the knots s_i are the scaled n th roots of 1,

$$V_{\mathbf{s}} = \sqrt{n} \Omega \text{diag}(e^i)_{i=0}^{m-1} \quad \text{and} \quad V_{\mathbf{s}}^{-1} = \frac{1}{\sqrt{n}} \text{diag}(e^{-j})_{j=0}^{m-1} \Omega^H.$$

Write $C_{\mathbf{s}, f} = (\frac{1}{s_i - f\omega^j})_{i,j=0}^{m-1, n-1}$ for $f \neq 0$ and $C_{e, \mathbf{t}} = (\frac{1}{e\omega^i - t_j})_{i,j=0}^{m-1, n-1}$ for $e \neq 0$ and obtain from (3.1) that

$$(3.2) \quad V_{\mathbf{s}} = \frac{f^{1-n}}{\sqrt{n}} \text{diag} \left(s_i^n - f^n \right)_{i=0}^{m-1} C_{\mathbf{s}, f} \text{diag}(\omega^j)_{j=0}^{n-1} \Omega \text{diag}(f^j)_{j=0}^{n-1},$$

$$(3.3) \quad V_{\mathbf{t}}^{-1} = \frac{1}{\sqrt{n}} \text{diag}(e^{-i})_{i=0}^{m-1} \Omega^H \text{diag}(l(e^i))_{i=0}^{m-1} C_{e, \mathbf{t}} \text{diag} \left(\frac{1}{l'(t_j)} \right)_{j=0}^{n-1}, \quad \text{and}$$

$$(3.4) \quad V_{\mathbf{s}}^{-1} = \sqrt{n} \text{diag}(f^{-j})_{j=0}^{n-1} \Omega^H \text{diag}(\omega^{-j})_{j=0}^{n-1} C_{\mathbf{s}, f}^{-1} \text{diag} \left(\frac{f^{n-1}}{s_i^n - f^n} \right)_{i=0}^{m-1} \quad \text{for } m = n.$$

These expressions link Vandermonde matrices and their inverses to the $m \times n$ CV matrices $C_{\mathbf{s}, f}$ of equation (1.1) and the $n \times m$ *CV^T matrices* $C_{e, \mathbf{t}} = -C_{\mathbf{t}, e}^T = (\frac{1}{e\omega^i - t_j})_{i,j=0}^{n-1, m-1}$ (for $e \neq 0$), that is, Cauchy matrices with an arbitrary knot set $\mathcal{T} = \{t_0, \dots, t_{n-1}\}$ and with the knot set $\mathcal{S} = \{s_i = e\omega^i, i = 0, \dots, m - 1\}$. More details on the subjects of this section can be found in [Pb].

Part II. Extended HSS matrices

4. QUASISEPARABLE AND HSS MATRICES

4.1. Quasiseparable matrices and generators.

Definition 4.1. Suppose that an $m \times n$ matrix M is represented as a $k \times k$ block matrix with a block diagonal $\widehat{\Sigma} = (\Sigma_0, \dots, \Sigma_{k-1})$. Let $\chi(\widehat{\Sigma})$ denote the overall number of the entries of all its k diagonal blocks $\Sigma_0, \dots, \Sigma_{k-1}$ and let $\chi(\widehat{\Sigma}) \ll mn$, that is, let mn greatly exceed $\chi(\widehat{\Sigma})$. Furthermore let l and u denote the maximum ranks of the sub- and superdiagonal blocks of the matrix M , respectively. Then the matrix M is (l, u) -quasiseparable. By replacing ranks with ξ -ranks we define a (ξ, l, u) -quasiseparable matrix.

The definition generalizes the class of banded matrices and their inverses: a matrix having a lower bandwidth l and an upper bandwidth u as well as its inverse (if defined) are (l, u) -quasiseparable.

In order to operate with (l, u) -quasiseparable matrices efficiently, one exploits their representation with *quasiseparable generators*, demonstrated by the following 4×4 example and defined below in general form,

$$(4.1) \quad M = \begin{pmatrix} \Sigma_0 & S_0T_1 & S_0B_1T_2 & S_0B_1B_2T_3 \\ P_1Q_0 & \Sigma_1 & S_1T_2 & S_1B_2T_3 \\ P_2A_1Q_0 & P_2Q_1 & \Sigma_2 & S_2T_3 \\ P_3A_2A_1Q_0 & P_3A_2Q_1 & P_3Q_2 & \Sigma_3 \end{pmatrix}.$$

By generalizing this example we arrive at the following definition.

Definition 4.2 (Cf. Table 4.1). Suppose that an $m \times n$ matrix M is represented as a $k \times k$ block matrix with a block diagonal $\widehat{\Sigma} = (\Sigma_0, \dots, \Sigma_{k-1})$ such that $\chi(\widehat{\Sigma}) \ll mn$. (We reuse these assumptions of Definition 4.1.)

Furthermore suppose that a set $\{\mathcal{I}_1, \dots, \mathcal{I}_k\}$ partitions the set $\{1, \dots, m\}$; a set $\{\mathcal{J}_1, \dots, \mathcal{J}_k\}$ partitions the set $\{1, \dots, n\}$, and there exists a six-tuple $\{P_i, Q_h, S_h, T_i, A_g, B_g\}$ such that $M(\mathcal{I}_i, \mathcal{J}_h) = P_iA_{i-1} \cdots A_{h+1}Q_h$ and $M(\mathcal{I}_h, \mathcal{J}_i) = S_hB_{h+1} \cdots B_{i-1}T_i$ for $0 \leq h < i < k$.

Here $P_i, Q_h,$ and A_g are $|\mathcal{I}_i| \times l_i, l_{h+1} \times |\mathcal{J}_h|,$ and $l_{g+1} \times l_g$ matrices, respectively, and S_h, T_i and B_g are $|\mathcal{I}_h| \times u_{h+1}, u_i \times |\mathcal{J}_i|,$ and $u_g \times u_{g+1}$ matrices, respectively, for $g = 1, \dots, k - 2, h = 0, \dots, k - 2, i = 1, \dots, k - 1$.

Then the six-tuple $\{P_i, Q_h, S_h, T_i, A_g, B_g\}$ is an (l, u) -quasiseparable generator of the matrix M , and the integers $l = \max_g \{l_g\}$ and $u = \max_h \{u_h\}$ are the *lower and upper lengths* or *orders* of this generator.

TABLE 4.1. The sizes of quasiseparable generators of Definition 4.2

P_i	Q_h	A_g	S_h	T_i	B_g
$ \mathcal{I}_i \times l_i$	$l_{h+1} \times \mathcal{J}_h $	$l_{g+1} \times l_g$	$ \mathcal{I}_h \times u_{h+1}$	$u_i \times \mathcal{J}_i $	$u_g \times u_{g+1}$

Theorem 4.1 (Cf. [B10], [VVM], [X12], [EGH13], and the bibliographies therein). *A matrix M is (l, u) -quasiseparable if and only if it has a (nonunique) representation via (l, u) -quasiseparable generators.*

By virtue of this theorem one can redefine the (l, u) -quasiseparable matrices as those representable with the families of quasiseparable generators $\{P_h, Q_i, A_g\}$ and $\{S_h, T_i, B_g\}$ that have lower and upper orders l and u , respectively. Definitions 4.1 and 4.2 provide two useful insights into the properties of these matrices. The third equivalent definition in Section 4.4 (cf. Theorem 4.5) provides yet another insight and is linked to the study of the Cauchy matrix $C_{1, \omega_{2n}}$ in [CGS07], [XXG12], [XXCB14]. Various definitions, equivalent or closely related to those above, have been introduced by a number of authors (cf. [VVM], [B10], [EGH13], and the references therein). In particular the related study of H -matrices and H^2 -matrices in [H99], [T00], [BH02], [GH03], [B09], [B10], and references therein, was the basis for the software libraries HLib, www.hlib.org, and H2Lib, <http://www.h2lib.org/>, <https://github.com/H2Lib/H2Lib>, developed at the Max Planck Institute for Mathematics in the Sciences.

4.2. Operations with quasiseparable matrices: definitions and demonstration. Next we cover some basic operations with matrices represented with (l, u) -quasiseparable generators.

Definition 4.3. Given diagonal blocks $\Sigma_q, q = 0, \dots, k - 1$, of an (l, u) -quasiseparable matrix M and (l, u) -quasiseparable generators for all its sub- and super-diagonal blocks, let $\alpha(M)$ and $\beta(M)$ denote the arithmetic cost of computing the vectors $M\mathbf{u}$ and $M^{-1}\mathbf{u}$, respectively, maximized over all normalized vectors \mathbf{u} , $|\mathbf{u}| = 1$, and minimized over all algorithms. Write $\beta(M) = \infty$ if the matrix M is singular. $\alpha_\xi(M)$ and $\beta_\xi(M)$ replace the bounds $\alpha(M)$ and $\beta(M)$, respectively, provided that instead of the evaluation of the vectors $M\mathbf{u}$ and $M^{-1}\mathbf{u}$, respectively, we approximate them within the error bounds $\xi\|M\mathbf{u}\|$ and $\xi\|M^{-1}\mathbf{u}\|$, respectively.

The straightforward algorithm supports the following bound.

Theorem 4.2. $\alpha(M) \leq 2(m + n)\rho - \rho - m$ where a generating pair of length ρ defines an $m \times n$ matrix M .

The following estimates for computations with quasiseparable matrices extend the well-known estimates in the case of banded matrices.

Theorem 4.3 ([DV98], [H99], [EG02]). *Suppose that an (l, u) -quasiseparable matrix M of size $m \times n$ is defined by its $m_q \times n_q$ diagonal blocks $\Sigma_q, q = 0, \dots, k - 1$, such that $\sum_{q=0}^{k-1} m_q = m, \sum_{q=0}^{k-1} n_q = n$, and $s = \sum_{q=0}^{k-1} m_q n_q = O((l + u)(m + n))$ and by the generators of length at most l and at most u for its sub- and super-diagonal blocks, respectively.*

- (i) *Then $\alpha(M) \leq 2 \sum_{q=0}^{k-1} ((m_q + n_q)(l + u) + s) + 2l^2k + 2u^2k = O((l + u)(m + n))$ and*
- (ii) *$\beta(M) = O(\sum_{q=0}^{k-1} ((l + u)^2(l + u + n_q)n_q + n_q^3))$ if $m_q = n_q$ for all q and if the matrix M is nonsingular.*

Example 4.1 (Cf. Figures 2 and 3). Let us multiply by a vector \mathbf{v} the matrix M of equation (4.1).

- (i) At first, view it as a 2×2 block matrix with diagonal blocks $\overline{\Sigma}_1 = \begin{pmatrix} \Sigma_0 & S_0T_1 \\ P_1Q_0 & \Sigma_1 \end{pmatrix}$ and $\overline{\Sigma}_2 = \begin{pmatrix} \Sigma_2 & S_2T_3 \\ P_3Q_2 & \Sigma_3 \end{pmatrix}$; multiply the blocks $\begin{pmatrix} S_0B_1T_2 & S_0B_1B_2T_3 \\ S_1T_2 & S_1B_2T_3 \end{pmatrix}$ and $\begin{pmatrix} P_2A_1Q_0 & P_2Q_1 \\ P_3A_2A_1Q_0 & P_3A_2Q_1 \end{pmatrix}$ by two subvectors of the vector \mathbf{v} .

(ii) Then multiply the blocks S_0T_1 , P_1Q_0 , S_2T_3 , and P_3Q_2 of the matrices $\overline{\Sigma_1}$ and $\overline{\Sigma_2}$ of smaller sizes by four subvectors of the vector \mathbf{v} .

Perform the computations at both stages fast if the given generators of the blocks have small length.

(iii) Then multiply the four diagonal blocks Σ_1 , Σ_2 , Σ_3 , and Σ_4 by four subvectors of the vectors \mathbf{v} . Perform these computations fast because the four blocks have a small overall number of entries.

(iv) Finally obtain the vector $M\mathbf{v}$ by properly summing the products.

4.3. Fast multiplication with recursive merging of diagonal blocks: outline. In Example 4.1 we multiply the matrix M by a vector by using generators for only 6 out of its 22 sub- and super-diagonal blocks. Next we extend the above demonstration to multiplication of a general quasiseparable matrix M by a vector by using a small fraction of all generators.

Definition 4.4. Suppose that $M = (M_0 \mid \dots \mid M_{k-1})$ is a $1 \times k$ block matrix with k block columns M_q , each partitioned into a *diagonal block* Σ_q and a *neutered block column* N_q , $q = 0, \dots, k-1$ (cf. our Figures 1–3 and [MRT05, Section 1]). Such a matrix is ρ -*neutered* if its every neutered block column N is represented as $N = FH$ or $N = FSH$ where F of size $h \times r$, S of size $r \times r$, and H of size $r \times k$ are its *generator matrices* and $r \leq \rho$. Call such a pair or triple a *length r generator of the neutered block N* and call r its *length*. A ξ -approximation of such a matrix is called (ξ, ρ) -*neutered*.

In Figure 1 the diagonal blocks are black and the neutered block columns are gray or white.

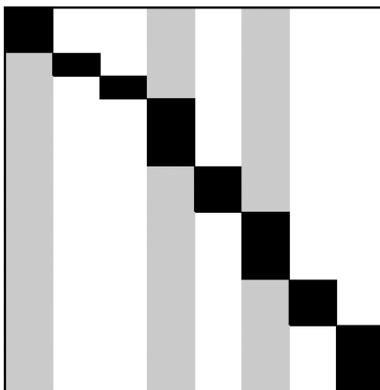


FIGURE 1

In Figure 2 the diagonal blocks from Figure 1 (marked by black color) are merged pairwise into their diagonal unions, each made up of four blocks. Two of them (from Figure 1) are marked by black color, and the two other by gray color. The new neutered block columns are either white or gray, but their gray color is lighter. The new (larger) diagonal blocks of Figure 2 are merged pairwise into the diagonal blocks of Figure 3, each made up of two black and two gray blocks, and its two neutered block columns are white.

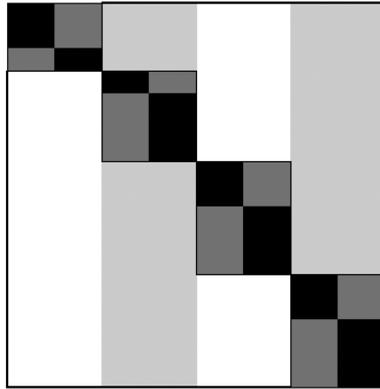


FIGURE 2

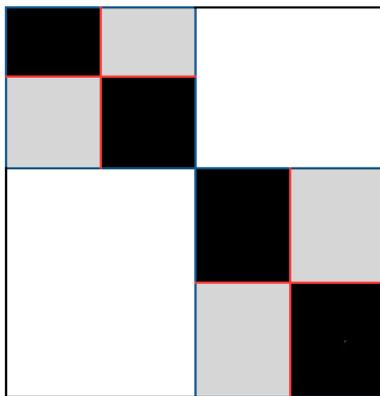


FIGURE 3

Theorem 4.4. *Suppose that an $m \times n$ matrix M is a ρ -neutered $k \times k$ block matrix and that we are given k generators of length at most ρ for all its k -neutered block columns as well as all the $\chi(\widehat{\Sigma})$ entries in the k -diagonal blocks $\Sigma_0, \dots, \Sigma_{k-1}$. Then*

$$\alpha(M) \leq 2\chi(\widehat{\Sigma}) + (2m + 2n - 1)k\rho = O(\chi(\widehat{\Sigma}) + (m + n)k\rho).$$

Proof. Multiply the diagonal blocks by vectors in the straightforward way and multiply the neutered block columns by vectors by using the representation with generators.

Formally, write $M = M' + \text{diag}(\Sigma_q)_{q=0}^{k-1}$. Notice that $\alpha(M) \leq 2\chi(\widehat{\Sigma}) + \alpha(M') + m$. The neutered block columns of the matrix M share their entries with the matrix M' , whose other entries are zeros. So the k pairs $(F_0, G_0), \dots, (F_{k-1}, G_{k-1})$ together form a single generating pair of a length at most $k\rho$ for the matrix M' . Therefore $\alpha(M') \leq (2m + 2n - 1)k\rho - m$ by virtue of Theorem 4.2. \square

The upper bound on $\alpha(M)$ of Theorem 4.4 is sufficiently small unless the integers k or $\chi(\widehat{\Sigma})$ are large. Unfortunately we cannot bound both of these integers at once, but we can circumvent the problem by applying the algorithm of Theorem 4.4 recursively. We begin with a partition of the matrix M defined by a few diagonal

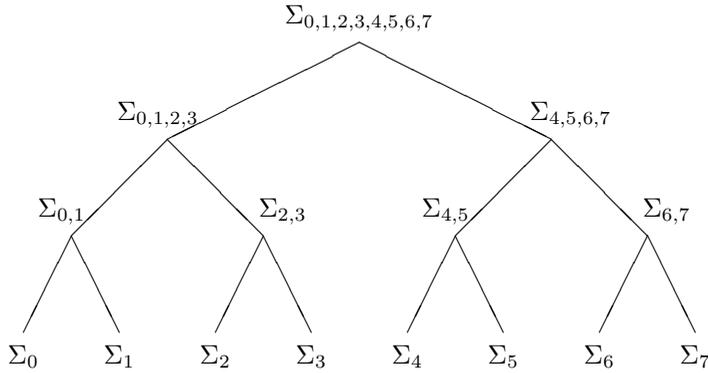


FIGURE 4

blocks that are ρ -neutered matrices themselves. Then we multiply neutered block columns fast (by using their generators), partition the diagonal blocks into smaller diagonal blocks and neutered block columns, and apply the same techniques recursively until we decrease the overall number of entries of the remaining diagonal blocks below a fixed tolerance bound of order $m + n$ or $(m + n)\rho$.

We can begin with $k = 2$ and $\chi(\widehat{\Sigma}) \approx 0.5n^2$ and then double the integer k and roughly halve the integer $\chi(\widehat{\Sigma})$ in every recursive step. Then overall we deal with only $O(m + n)$ neutered block columns and their generators and therefore multiply the matrix M by a vector by using $O((m + n)\rho)$ arithmetic operations in all these recursive steps, thus matching the cost bounds in part (i) of Theorem 4.3.

4.4. HSS and balanced HSS matrices and the cost of basic operations with them. Let us supply formal definitions and formal derivation of the latter estimates by applying the recursive process in the opposite direction, where at first the integer k is large and then is recursively doubled, while the diagonal blocks are small at first and then are merged recursively pairwise.

Definition 4.5. Fix two positive integers l and q such that $l + q \leq k$ and merge the l block columns $M_q, M_{q+1}, \dots, M_{q+l-1}$, the l diagonal blocks $\Sigma_q, \Sigma_{q+1}, \dots, \Sigma_{q+l-1}$, and the l neutered block columns $N_q, N_{q+1}, \dots, N_{q+l-1}$ into their union $M_{q,l} = M(\cdot, \bigcup_{j=0}^{l-1} \mathcal{C}(\Sigma_{q+j}))$, their diagonal union $\Sigma_{q,l}$, and their neutered union $N_{q,l}$, respectively, such that $\mathcal{R}(\Sigma_{q,l}) = \bigcup_{j=0}^{l-1} \mathcal{R}(\Sigma_{q+j})$ and every block column $M_{q,l}$ is partitioned into the diagonal union $\Sigma_{q,l}$ and the neutered union $N_{q,l}$.

Define recursive merging of all diagonal blocks $\Sigma_0, \dots, \Sigma_{k-1}$ by a binary tree whose leaves are associated with these blocks and whose every internal vertex is the union of its two children (see Figure 4). For every vertex v define the sets $L(v)$ and $R(v)$ of its left and right descendants, respectively. If $0 \leq |L(v)| - |R(v)| \leq 1$ for all vertices v , then the binary tree is balanced and identifies balanced merging of its leaves, in our case the diagonal blocks. We can uniquely define a balanced tree with n leaves by removing the $2^{l(n)} - n$ rightmost leaves of the complete binary tree that has $2^{l(n)}$ leaves for $l(n) = \lceil \log_2(n) \rceil$. All leaves of the resulting heap structure with n leaves lie in its two lowest levels.

Definition 4.6. (i) A block matrix is a *balanced ρ -HSS matrix* if it is ρ -neutered throughout the process of balanced merging of its diagonal blocks, that is, if all neutered unions of its neutered block columns involved into this process have ranks at most ρ . This is a *ρ -HSS matrix* if it is ρ -neutered throughout any process of recursive merging of its diagonal blocks.

(ii) By replacing ranks with ξ -ranks we define *balanced (ξ, ρ) -HSS matrices* and *(ξ, ρ) -HSS matrices*.

Fact 4.1. (i) Let a matrix be ρ_j -neutered at the j th step of recursive balanced merging for every j . Then this is a balanced ρ -HSS matrix for $\rho = \max_j \rho_j$.

(ii) Likewise, let a matrix be (ξ_j, ρ_j) -neutered at the j th step of recursive balanced merging for every j . Then this is a balanced (ξ, ρ) -HSS matrix for $\xi = \max_j \xi_j$ and $\rho = \max_j \rho_j$.

Theorem 4.5. (i) *Every (l, u) -quasiseparable matrix M is an $(l + u)$ -HSS matrix.*

(ii) *Every ρ -HSS matrix is (ρ, ρ) -quasiseparable.*

Proof. A neutered block column N_q can be partitioned into its block sub- and super-diagonal parts L_q and U_q , respectively, and so $\text{rank}(N_q) \leq \text{rank}(L_q) + \text{rank}(U_q)$. This implies that $\text{rank}(N_q) \leq l + u$ for $q = 0, \dots, k - 1$ if the matrix M is (l, u) -quasiseparable, and part (i) is proven.

Next consider the union N of any set of neutered block columns of a matrix M . It turns into a neutered block column at some stage of appropriate recursive merging. Therefore $\text{rank}(N) \leq \rho$ where M is a ρ -HSS matrix. Now, for every off-diagonal block B of a matrix M , define the set of its neutered block columns that share some column indices with the block B and then notice that the block B is a submatrix of the neutered union of this set. Therefore $\text{rank}(B) \leq \text{rank}(N) \leq \rho$, and we obtain part (ii). \square

By combining Theorems 4.3 and 4.5 we obtain the following results.

Corollary 4.1. *Assume a ρ -HSS matrix M given with $m_q \times n_q$ diagonal blocks Σ_q , $q = 0, \dots, k - 1$, and write $m = \sum_{q=0}^{k-1} m_q$, $n = \sum_{q=0}^{k-1} n_q$, and $s = \sum_{q=0}^{k-1} m_q n_q$. Then*

- (i) $\alpha(M) < 2s + 4\rho^2 k + 4 \sum_{q=0}^{k-1} (m_q + n_q)\rho = O((m + n)\rho + s)$ and
- (ii) $\beta(M) = O(\sum_{q=0}^{k-1} ((\rho + n_q)\rho^2 n_q + n_q^3))$ if $m_q = n_q$ for all q and if $\det(M) \neq 0$.

For a balanced ρ -HSS matrix M we only have a little weaker representation than in Theorem 4.1, and so the proof of the estimates of Corollary 4.1 for $\alpha(M)$ and $\beta(M)$ does not apply, but next we extend these bounds. Unlike Theorem 4.3 and Corollary 4.1, we allow $m_q \neq n_q$ for all q .

Theorem 4.6. *Assume a balanced ρ -HSS matrix M with $m_q \times n_q$ diagonal blocks Σ_q , $q = 0, \dots, k - 1$, having $s = \sum_{q=0}^{k-1} m_q n_q$ entries overall and write $l = \lceil \log_2(k) \rceil$, $m = \sum_{q=0}^{k-1} m_q$, $n = \sum_{q=0}^{k-1} n_q$, $m_+ = \max_{q=0}^{k-1} m_q$, $n_+ = \max_{q=0}^{k-1} n_q$, and $s \leq \min\{m_+ n, m n_+\}$.*

(i) *Then*

$$(4.2) \quad \alpha(M) < 2s + (m + 4(m + n)\rho)l.$$

(ii) *If $m = n$ and if the matrix M is nonsingular, then*

$$(4.3) \quad \beta(M) = O(n_+ s + (n_+^2 + \rho n_+ + l\rho^2)n + (k\rho + n)\rho^2).$$

(iii) *The same bounds (4.2) and (4.3) hold for the transpose of a balanced ρ -HSS matrix M matrix having $n_q \times m_q$ diagonal blocks Σ_q for $q = 0, \dots, k - 1$.*

Proof. Let us readily prove part (i) by just counting the arithmetic operations involved in recursive merging.

With no loss of generality assume that the $(l - 1)$ st (that is, final) stage of a balanced merging process has produced a 2×2 block representation

$$M = \begin{pmatrix} \overline{\Sigma}_0^{(l)} & \overline{S_{01}^{(l)} T_1^{(l)}} \\ \overline{S_{10}^{(l)} T_0^{(l)}} & \overline{\Sigma}_1^{(l)} \end{pmatrix}$$

where $\overline{\Sigma}_j^{(l)}$ is an $\overline{m}_j^{(l)} \times \overline{n}_j^{(l)}$ matrix, $\overline{T}_j^{(l)}$ is an $\overline{n}_j^{(l)} \times \overline{\rho}^{(l)}_j$ matrix, $\overline{\rho}_j^{(l)} \leq \rho$, $j = 0, 1$, $\overline{m}_1^{(l)} + \overline{m}_2^{(l)} = m$, and $\overline{n}_1^{(l)} + \overline{n}_2^{(l)} = n$. Clearly $\alpha(M) \leq m + \sum_{j=0}^1 \alpha(\overline{\Sigma}_j^{(l)}) + \sum_{j=0}^1 \alpha(\overline{T}_j^{(l)}) + \alpha(\overline{S_{01}^{(l)}}) + \alpha(\overline{S_{10}^{(l)}})$.

Apply Theorem 4.2 and obtain that $\sum_{j=0}^1 \alpha(\overline{T}_j^{(l)}) + \alpha(\overline{S_{01}^{(l)}}) + \alpha(\overline{S_{10}^{(l)}}) < 4(m+n)\rho$.

The second last stage of the balanced merging process produces a similar 2×2 block representation for each of the diagonal blocks $\overline{\Sigma}_j^{(l)}$, $j = 0, 1$. Therefore $\sum_{j=0}^1 \alpha(\overline{\Sigma}_j^{(l)}) < m + 4(m+n)\rho + \sum_{j=0}^{k(1)} \alpha(\overline{\Sigma}_j^{(1-1)})$ where $\overline{\Sigma}_0^{(1-1)}, \dots, \overline{\Sigma}_{k(1)-1}^{(1-1)}$ are the diagonal blocks output at the second last merging stage (cf. Figures 3 and 4).

By recursively going back through the merging process, obtain that $\alpha(M) < (m + 4(m+n)\rho)l + \sum_{j=0}^{k-1} \alpha(\Sigma_j)$. Here $\Sigma_q = \overline{\Sigma}_q^{(0)}$ is an $m_q \times n_q$ matrix for $m_q = \overline{m}_q^{(0)}$, $n_q = \overline{n}_q^{(0)}$, $q = 0, \dots, k - 1$. Hence $\sum_{q=0}^{k-1} \alpha(\Sigma_q) < 2 \sum_{q=0}^{k-1} m_q n_q = 2s$, implying (4.2).

Part (ii) of the theorem has been supported by the merging and compression algorithm of [CGS07]. The algorithm has been presented and analyzed in [CGS07] (cf. also [XXG12] and [XXCB14]) for the subclass of balanced ρ -HSS matrices, approximating the special matrix $(\frac{1}{\omega - f\omega})_{i,j=0}^{n-1}$ for $\omega = \exp(2\pi\sqrt{-1}/n)$ and $f = \exp(\pi\sqrt{-1}/n)$, denoting primitive n th and $2n$ th roots of 1, respectively, but both the algorithm and its analysis are readily extended, and bound (4.3) follows. All the proofs can be equally applied when rows of the matrix M replace its columns and vice versa, and this implies part (iii). □

Corollary 4.2. *Under the assumptions of parts (i)–(iii) of Theorem 4.6 suppose that $k\rho = O(n)$ and $n_+ + \rho = O(\log(n))$. Then $\alpha(M) = O((m+n)\log^2(n))$ and $\beta(M) = O(n\log^3(n))$.*

For our application to computations with CV matrices we must estimate $\alpha(M)$ and $\beta(M)$ for a more general class of matrices M defined in the next section. (Such a matrix has cyclic block tridiagonal part with a sufficiently small overall number of entries, say, $O((m+n)\log(m+n))$, such that all blocks of the matrix M not overlapping this part have small rank, say, $O(\log(m+n))$.) The algorithms supporting Theorem 4.6 and Corollary 4.2 are quite readily extended to these matrices in the next section.

5. EXTENSION FROM DIAGONAL TO TRIDIAGONAL BLOCKS

Example 5.1. The following matrix has eight square or rectangular diagonal blocks $\Sigma_0, \dots, \Sigma_7$ and becomes block tridiagonal if we glue its lower and upper boundaries,

$$(5.1) \quad M = \begin{pmatrix} \Sigma_0 & B_0 & O & O & O & O & O & A_0 \\ A_1 & \Sigma_1 & B_1 & O & O & O & O & O \\ O & A_2 & \Sigma_2 & B_2 & O & O & O & O \\ O & O & A_3 & \Sigma_3 & B_3 & O & O & O \\ O & O & O & A_4 & \Sigma_4 & B_4 & O & O \\ O & O & O & O & A_5 & \Sigma_5 & B_5 & O \\ O & O & O & O & O & A_6 & \Sigma_6 & B_6 \\ B_7 & O & O & O & O & O & A_7 & \Sigma_7 \end{pmatrix}.$$

Define the eight *tridiagonal blocks*,

$$\begin{aligned} \Sigma_0^{(c)} &= \begin{pmatrix} B_7 \\ \Sigma_0 \\ A_1 \end{pmatrix}, \quad \Sigma_1^{(c)} = \begin{pmatrix} B_0 \\ \Sigma_1 \\ A_2 \end{pmatrix}, \quad \Sigma_2^{(c)} = \begin{pmatrix} B_1 \\ \Sigma_2 \\ A_3 \end{pmatrix}, \quad \Sigma_3^{(c)} = \begin{pmatrix} B_2 \\ \Sigma_3 \\ A_4 \end{pmatrix}, \\ \Sigma_4^{(c)} &= \begin{pmatrix} B_3 \\ \Sigma_4 \\ A_5 \end{pmatrix}, \quad \Sigma_5^{(c)} = \begin{pmatrix} B_4 \\ \Sigma_5 \\ A_6 \end{pmatrix}, \quad \Sigma_6^{(c)} = \begin{pmatrix} B_5 \\ \Sigma_6 \\ A_7 \end{pmatrix}, \quad \text{and} \quad \Sigma_7^{(c)} = \begin{pmatrix} B_6 \\ \Sigma_7 \\ A_0 \end{pmatrix}. \end{aligned}$$

Here $\Sigma_1^{(c)}, \Sigma_2^{(c)}, \Sigma_3^{(c)}, \Sigma_4^{(c)}, \Sigma_5^{(c)}$, and $\Sigma_6^{(c)}$ are six blocks of the matrix M of (5.1), while $\Sigma_0^{(c)}$ and $\Sigma_7^{(c)}$ consist of two pairs of its blocks. Each pair, however, turns into a single block if we glue together the lower and upper boundaries of the matrix M . With the diagonal block Σ_q and the *tridiagonal block* $\Sigma_q^{(c)}$ we still associate a block column M_q such that $\mathcal{C}(M_q) = \mathcal{C}(\Sigma_q^{(c)})$.

The *admissible block* $N_q^{(c)}$, playing the role similar to that of a neutered block column of Definition 4.4, complements the tridiagonal block $\Sigma_q^{(c)}$ in its block column. The block $N_q^{(c)}$ is filled with zeros in the case of the matrix M of (5.1) for every $q, q = 0, \dots, 7$, but not so in the case of general 8×8 block matrix embedding the matrix M of (5.1).

Here are some sample unions of the tridiagonal blocks of the matrix M of (5.1), $\Sigma_{0,1,\dots,7}^{(c)} = M$,

$$\Sigma_{0,1,2,3}^{(c)} = \begin{pmatrix} B_7 & O & O & O \\ \Sigma_0 & B_0 & O & O \\ A_1 & \Sigma_1 & B_1 & O \\ O & A_2 & \Sigma_2 & B_2 \\ O & O & A_3 & \Sigma_3 \\ O & O & O & A_4 \end{pmatrix}, \quad \Sigma_{0,1}^{(c)} = \begin{pmatrix} B_7 & O \\ \Sigma_0 & B_0 \\ A_1 & \Sigma_1 \\ O & A_2 \end{pmatrix}, \quad \text{and} \quad \Sigma_{2,3}^{(c)} = \begin{pmatrix} B_1 & O \\ \Sigma_2 & B_2 \\ A_3 & \Sigma_3 \\ O & A_4 \end{pmatrix}.$$

In Figure 5 the admissible blocks are light gray or white; two adjacent blocks of each black diagonal block are darker gray; the triples of these black and gray blocks form the tridiagonal blocks. The neutered block columns are either white or gray.

Let us generalize this demonstration (see Figure 5). Assume a block matrix M with k diagonal blocks Σ_q , of sizes $m_q^{(c)} \times n_q$, for $q = 0, \dots, k - 1$, and glue together its lower and upper block boundaries. Then each diagonal block, including the two extremal blocks Σ_0 and Σ_{k-1} , has exactly two *adjacent blocks* in its block column: they are given by the pair of the subdiagonal and superdiagonal blocks. Define

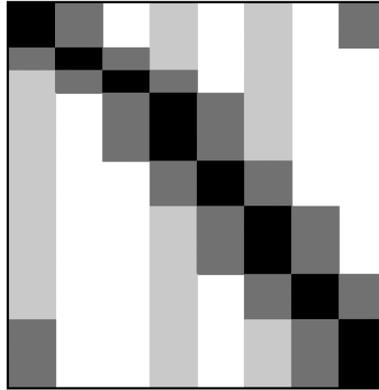


FIGURE 5

the *tridiagonal blocks* $\Sigma_0^{(c)}, \dots, \Sigma_{k-1}^{(c)}$ of sizes $m_q^{(c)} \times n_q$ by combining such triples of blocks where $m_q^{(c)} = m_{q-1 \bmod k} + m_q + m_{q+1 \bmod k}$, $q = 0, \dots, k - 1$. Write $m^{(c)} = \sum_{q=0}^{k-1} m_q^{(c)}$ and notice that $m^{(c)} = 3m$ because the number of rows in each of the three block diagonals sums to m . Therefore $s^{(c)} = \sum_{q=0}^{k-1} m_q^{(c)} n_q \leq m^{(c)} n_+ \leq 3mn_+$.

The complements of the tridiagonal blocks in their block columns are also blocks, called *admissible* (cf. [B10]). We call the matrix itself an *extended HSS matrix*, and we extend accordingly our definitions of the unions of blocks, recursive and balanced merging, ρ -neutered, balanced ρ -HSS, ρ -HSS matrices, as well as (ξ, ρ) -neutered, balanced (ξ, ρ) -HSS, and (ξ, ρ) -HSS matrices (cf. Definitions 4.4, 4.5, and 4.6). Can we extend Theorem 4.6 and Corollary 4.2 to the case of extended balanced ρ -HSS matrices M where we replace the integer parameters m and s by $m^{(c)} = 3m$ and $s^{(c)} \leq m^{(c)} n_+ = 3mn_+$, respectively? The extension of part (i) of Theorem 4.6 is immediate, but in order to extend the algorithms supporting its part (ii), we must impose some restriction on the input matrix M .

Definition 5.1. An extended balanced ρ -HSS matrix is *hierarchically regular* if all its diagonal blocks at the second factorization stage of the associated balanced merging process have full rank. This matrix is *hierarchically well conditioned* if these blocks are also well conditioned.

Theorem 5.1. *Suppose that the matrix M in Theorem 4.6 is replaced by an extended $m \times n$ balanced ρ -HSS matrix $M^{(c)}$ and also suppose that the integer parameters m and s in bounds (4.2) on $\alpha(M)$ and (4.3) on $\beta(M)$ are replaced by $m^{(c)} = 3m$ and $s^{(c)} \leq 3mn_+$, respectively. Then bound (4.2) still holds, and bound (4.3) holds if $m = n$ and if the matrix M is hierarchically regular and hierarchically well conditioned.*

Proof. Revisit the proof of the Theorem 4.6, by replacing the integer parameters m and $s^{(j)}$ according to the assumptions of Theorem 5.1, and verify that the proof still remains valid (use the assumption that the matrix M is hierarchically regular and hierarchically well conditioned in order to extend bound (4.3)). □

Corollary 5.1. *Under the assumptions of Theorem 5.1 suppose that $k\rho = O(n)$ and $n_+ + \rho = O(\log(n))$. Then $\alpha(M) = O((m + n) \log^2(n))$ and $\beta(M) = O(n \log^3(n))$.*

Part III. Computations with CV matrices and extensions

6. APPROXIMATION OF THE CV AND CV^T MATRICES BY HSS MATRICES AND ALGORITHMIC IMPLICATIONS

Our next goal is approximation of CV by HSS matrices, which will imply fast approximate solution of Problems 1–4 because in Part I we reduced them to computations with CV matrices of (1.1), and in Part II we described fast computations with HSS matrices.

6.1. Small-rank approximation of certain Cauchy matrices.

Definition 6.1 (See [CGS07, p. 1254]). For a *separation bound* $\theta < 1$ and a complex *separation center* c , a pair of complex points s and t is (θ, c) -separated if $|\frac{t-c}{s-c}| \leq \theta$. A pair of sets of complex numbers \mathcal{S} and \mathcal{T} is (θ, c) -separated if every pair of points $s \in \mathcal{S}$ and $t \in \mathcal{T}$ is (θ, c) -separated.

Lemma 6.1 (See [R85] and [CGS07, equation (2.8)] or [Pb]). *Suppose a pair of complex points s and t is (θ, c) -separated for $0 \leq \theta < 1$ and a complex center c . Fix a positive integer ρ and write $q = \frac{t-c}{s-c}$ and $|q| \leq \theta$. Then $\frac{1}{s-t} = \frac{1}{s-c} \sum_{h=0}^{\rho-1} \frac{(t-c)^h}{(s-c)^h} + \frac{q^\rho}{s-c}$ for $|q_\rho| = \frac{|q|^\rho}{1-|q|} \leq \frac{\theta^\rho}{1-\theta}$ and a positive integer ρ .*

Corollary 6.1 (Cf. [CGS07, Section 2.2], [B10], or [Pb]). *Suppose that two sets of $2n$ distinct complex numbers $\mathcal{S} = \{s_0, \dots, s_{m-1}\}$ and $\mathcal{T} = \{t_0, \dots, t_{n-1}\}$ are (θ, c) -separated from one another for $0 < \theta < 1$ and a global complex center c . Define the Cauchy matrix $C = (\frac{1}{s_i - t_j})_{i,j=0}^{m-1, n-1}$ and let $\delta = \delta_{c, \mathcal{S}} = \min_{i=0}^{m-1} |s_i - c|$ denote the distance from the center c to the set \mathcal{S} . Fix a positive integer ρ and define the $m \times \rho$ matrix $F = (1/(s_i - c)^{\nu+1})_{i,\nu=0}^{m-1, \rho-1}$ and the $n \times \rho$ matrix $G = ((t_j - c)^\nu)_{j,\nu=0}^{n-1, \rho-1}$. (We can compute these matrices by using $(m + n)\rho + m$ arithmetic operations.) Then*

$$(6.1) \quad C = FG^T + E, \quad |E| \leq \frac{\theta^\rho}{(1 - \theta)\delta}.$$

6.2. **Block partition of a Cauchy matrix.** Generally neither CV matrix of equation (1.1) nor its blocks of a large size have global separation centers. So, instead of the approximation of a CV matrix by a low-rank matrix, we seek its approximation by an extended balanced ρ -HSS matrix for a bounded integer ρ . At first we fix a reasonably large integer k and then partition the complex plane into k congruent sectors sharing the origin 0. The following definition induces a uniform k -partition of the knot sets \mathcal{S} and \mathcal{T} and thus induces a block partition of the associated Cauchy matrix. In the next subsection we specialize these partitions to the case of a CV matrix.

Definition 6.2 (See Figure 6). $\mathcal{A}(\phi, \phi') = \{z = \exp(\psi\sqrt{-1}) : 0 \leq \phi \leq \psi < \phi' < 2\pi\}$ is the *semi-open arc* of the unit circle $\{z : |z| = 1\}$ with length $\phi' - \phi$ and endpoints $\tau = \exp(\phi\sqrt{-1})$ and $\tau' = \exp(\phi'\sqrt{-1})$. $\Gamma(\phi, \phi') = \{z = r \exp(\psi\sqrt{-1}) : r \geq 0, 0 \leq \phi \leq \psi < \phi' < 2\pi\}$ is the *semi-open sector*. $\overline{\Gamma}(\phi, \phi')$ is its exterior.

can be nonempty, and so all the other block columns are occupied entirely by admissible blocks and hence have ranks at most ρ .

6.3. Separation of the tridiagonal and admissible blocks of a CV matrix.
 The following lemma can be readily verified (cf. Figure 6).

Lemma 6.2. $0 \leq \chi \leq \phi \leq \eta < \phi' < \chi' \leq \pi/2$ and write $\tau = \exp(\phi\sqrt{-1})$, $c = \exp(\eta\sqrt{-1})$, and $\tau' = \exp(\phi'\sqrt{-1})$. Then $|c - \tau| = 2 \sin(\frac{\eta-\phi}{2})$ and the distance from the point c to the sector $\overline{\Gamma(\chi, \chi')}$ is equal to $\sin(\psi)$, for $\psi = \min\{\eta - \chi, \chi' - \eta\}$.

Next we specialize the block partition of the previous subsection to the case of a CV matrix $C_{s,f}$ of (1.1) for a fixed complex f such that $|f| = 1$. In this case $t_j = f\omega_k^j$ for $\omega_k = \exp(2\pi\sqrt{-1}/k)$, $j = 0, \dots, n - 1$, and every arc \mathcal{A}_q contains $\lceil n/k \rceil$ or $\lfloor n/k \rfloor$ knots t_j .

In Figure 7, $\psi = \phi_1 + \frac{\phi_0}{2}$.

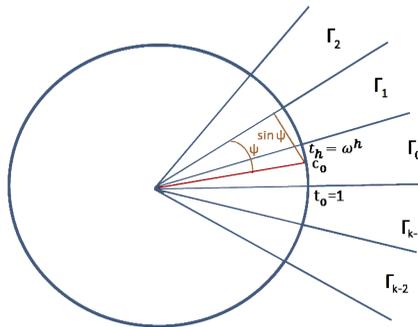


FIGURE 7

Theorem 6.1 (Cf. Figure 7). Assume a uniform k -partition of the knot sets of a CV matrix above for $k \geq 12$. Let Γ'_q denote the union of the sector Γ_q and its two adjacent sectors on both sides, that is, $\Gamma'_q = \Gamma_{q-1 \bmod k} \cup \Gamma_q \cup \Gamma_{q+1 \bmod k}$. Write $\overline{\Gamma'_q}$ to denote the exterior of the sector Γ'_q and write c_q to denote the midpoints of the arcs $\mathcal{A}_q = \mathcal{A}(\phi_q, \phi'_q)$ for $\phi'_q = \phi_{q+1 \bmod k}$ and $q = 0, \dots, k - 1$. Furthermore let $\bar{\delta}_q$ denote the distance from the center c_q to the sector $\overline{\Gamma'_q}$. Then, for every q ,

- (i) $\bar{\delta}_q \geq |\sin(\frac{3\pi}{k})|$ and
- (ii) the arc \mathcal{A}_q and the sector $\overline{\Gamma'_q}$ are (θ, c_q) -separated for $\theta = 2 \sin(\frac{\pi}{2k}) / \sin(\frac{3\pi}{k})$.

Proof. Suppose that $1 \leq q \leq k - 3$. Then $\Gamma'_q = \Gamma(\phi_{q-1}, \phi_{q+2})$. Apply Lemma 6.2, for $\chi = \phi_{q-1}$, $\phi = \phi_q$, $c = c_q$, $\phi' = \phi'_q = \phi_{q+1}$, and $\chi' = \phi_{q+2}$, and obtain the theorem. Similarly prove the theorem in the cases where $q = 0$, $\Gamma'_0 = \Gamma(\phi_{k-1}, \phi_2)$; $q = k - 2$ and $\Gamma'_{k-2} = \Gamma(\phi_{k-3}, \phi_0)$, and $q = k - 1$ and $\Gamma'_{k-1} = \Gamma(\phi_{k-2}, \phi_1)$. \square

Recall that $\sin(y) \approx y$ as $y \approx 0$, and therefore $\theta \approx 1/3$ provided that the integer k is large. Notice that for every q the admissible block $N_q^{(c)}$ is defined by the knots t_j lying on the arc \mathcal{A}_q and the knots s_i lying in the sector $\overline{\Gamma'_q}$, and apply Corollary 6.1. For every q , $q = 0, \dots, k - 1$, write $\delta_q = \min_{s_i \in \overline{\Gamma'_q}} |s_i - c_q|$, then notice that $\delta_q \geq \bar{\delta}_q$, and obtain the following result.

Corollary 6.2. *Assume a sufficiently large integer k , $2k < n$, and let a uniform k -partition of the knot sets \mathcal{S} and \mathcal{T} of an $m \times n$ CV matrix C define k admissible blocks $N_0^{(c)}, \dots, N_{k-1}^{(c)}$. Then all of them have the $|E|$ -ranks at most ρ , that is, C is an extended $(|E|, \rho)$ -neutered matrix, where $|E|$ and ρ satisfy bound (6.1) for $\theta \approx 1/3$ and $\delta = \min_{q=0}^{k-1} |\delta_q| \geq |\sin(\frac{3\pi}{k})|$.*

Our k -uniform partition of the complex plane into k congruent sectors defines a desired partition of CV matrix into (θ, c_q) -separated blocks for $\theta \approx 1/3$ or smaller. Trying to extend our results to the more general class of Cauchy matrices $C_{\mathbf{s}, \mathbf{t}}$ whose knots t_j lie on the unit circle $\{z : |z| = 1\}$, one may consider various other partitions of the complex plane and apply the following extension of Lemma 6.2 and Theorem 6.1.

Lemma 6.3. *Assume the numbers θ, ϕ, ϕ' , and c such that $0 < \theta < 1, 0 \leq \phi < \phi' \leq 2\pi$, and $c = \exp(0.5(\phi' + \phi)\sqrt{-1})$ is the midpoint of the arc $\mathcal{A}(\phi, \phi')$. Write $r = r(\phi, \phi', \theta) = \frac{2}{\theta} \sin(\frac{\phi' - \phi}{4})$. Let $D(c, r) = \{z : |z - c| \leq r\}$ denote the disc on the complex plane with a center c and a radius r and let $\overline{D(c, r)} = \{z : |z - c| > r\}$ denote the exterior of this disc. Then the two sets $\mathcal{A}(\phi, \phi')$ and $\overline{D(c, r)}$ are (θ, c) -separated.*

6.4. Approximation of a CV matrix by a balanced ρ -HSS matrix and the complexity of approximate computations with CV matrices. Let $\delta^{(h)}$ denote the minimum distance from the centers c_q to the knots s_i lying in the admissible blocks after the h th recursive merging. Recall that the angles $2\pi/k$ of the k congruent sectors $\Gamma_0, \dots, \Gamma_{k-1}$ are recursively doubled in every merging. So Lemma 6.2 implies that $\delta^{(h)} \geq \sin(3\pi 2^h/k)$ after the h th merging, $h = 1, \dots, l$. We define the recursive merging by choosing the integers $k = 2^{l+}$ and $l < l_+$. Choose them such that $k/2^l = 2^{l+ - l} \geq 6$. Then $\delta^{(h+1)} > \delta^{(h)} > \delta^{(0)} \geq \delta_- = \sin(\frac{3\pi}{k})$ for all h , and so $\delta_- \approx \frac{3\pi}{k}$ for large integers k . Together with Corollary 6.2 these relationships imply the following result.

Theorem 6.2. *The CV matrix C of Corollary 6.2 as well as its transpose CV^T matrix C^T are two extended balanced (ξ, ρ) -HSS matrices where the values ξ and ρ are linked by bound (6.1) for $|E| = \xi, \theta = 2\sin(\frac{\pi}{2k})/\sin(\frac{3\pi}{k})$, and $\delta = \delta_h \geq \delta_- = \sin(\frac{3\pi}{k})$, so that $\theta \approx 1/3$ and $\delta_- \approx \frac{3\pi}{k}$, for large integers k .*

Combine Corollary 5.1 with this theorem applied for $k = 2^{l+}$ of order $n/\log(n)$, for ρ and $\log(1/\xi)$ of order $\log(n)$, and for $l < l_+$ such that $l_+ - l \geq 6$ (verify that in this case the assumptions of the corollary are satisfied), and obtain the following complexity estimates for CV matrices C and CV^T matrices C^T .

Theorem 6.3. *Assume an $m \times n$ CV matrix C and a positive ξ such that $\log(1/\xi) = O(\log(n))$. Then $\alpha_\xi(C) = O((m + n) \log^2(n))$. If in addition $m = n$ and if the matrix C is ξ -approximated by a hierarchically regular extended balanced ρ -HSS matrix, then $\beta_\xi(C) = O(n \log^3(n))$. The same bounds hold for the CV^T matrix C^T replacing C .*

7. EXTENSIONS AND IMPLEMENTATION

7.1. Computations with matrices having displacement structure, polynomials, and rational functions. By combining the algebraic techniques of transformation of matrix structure of [P90] with the FMM/HSS techniques, [P15, Section

9] extends the complexity bounds of Theorems 6.3 and 7.1 to generalized Cauchy matrices $M = (f(s_i - t_j))_{i,j=0}^{m-1,n-1}$ for various functions $f(z)$ such as z^{-p} for a positive integer p , $\ln z$, and $\tan z$, to $n \times n$ structured matrices M having the displacement structures of Toeplitz, Hankel, Cauchy and Vandermonde types (cf. also [Pb]), and in particular to Cauchy matrices $M = C_{\mathbf{s},\mathbf{t}}$ having arbitrary sets of knots \mathcal{S} and \mathcal{T} . In the latter case the approximation error bound ξ increases by a factor bounded from above by the condition number $\kappa(M) = \|M\| \|M^+\|$, and the results are readily extended to Problems 3 and 4 of multipoint rational evaluation and interpolation. Next we specify the simpler extension to computations with a Vandermonde matrix, its transpose, and polynomials.

Theorem 7.1. *For a positive ξ and a vector $\mathbf{s} = (s_i)_{i=0}^{m-1}$, write $V = V_{\mathbf{s}}$ and $s_+ = \max_{i=0}^{m-1} |s_i|$.*

(i) *Then $\alpha_{\xi}(V) + \alpha_{\xi}(V^T) = O((m+n)\rho \log^2(n))$ provided that s_+ is bounded from above by a constant.*

(ii) *Suppose that, for $m = n$ and some complex f , $|f| = 1$, the CV matrix $C_{\mathbf{s},f}$ has been ξ -approximated by a hierarchically nonsingular extended balanced (ξ, ρ) -HSS matrix. Then $\beta_{\xi}(V) + \beta_{\xi}(V^T) = O(n\rho^3 \log(n))$.*

(iii) *One can extend the above bounds on $\alpha_{\xi}(V)$ and $\beta_{\xi}(V)$ to the solution of Problems 1 and 2 of Section 3.*

Proof. Without loss of generality we can assume that $m = n$. Combine Theorem 6.3, equations (3.2), (3.4) and their transposes. The matrices $\text{diag}(\omega^j)_{j=0}^{n-1}$, Ω/\sqrt{n} , Ω^H/\sqrt{n} , and $\text{diag}(f^j)_{j=0}^{n-1}$ are unitary, and so multiplication by them and by their inverses makes no impact on the output error norms. Multiplication by the matrix $\text{diag}(s_i^n - f^n)_{i=0}^{n-1}$ can increase the value ξ by at most a factor of $1 + s_+^n \leq 1 + |V_{\mathbf{s}}|s_+$, while multiplication by the inverse of this matrix increases ξ by a factor of $\Delta = 1/\max_f: |f|=1 \min_{i=0}^{n-1} |s_i^n - f^n|$, which is at most $2n$ for a proper choice of the value f such that $|f| = 1$. Then the increase by a factor of Δ would make no impact on the asymptotic bounds of Theorem 7.1, and so we complete the proof of parts (i) and (ii). The equations of Problem 1 extend the proof to part (iii). \square

7.2. Simplified implementation. One can implement our algorithms by computing the centers c_q and the admissible blocks \widehat{N}_q of bounded ranks in the merging process, but can avoid a large part of the computations by following the recipe of the papers [CGS07], [X12], [XXG12], and [XXCB14]. The idea is to bypass the computation of the centers c_q and immediately compute HSS generators for the admissible blocks \widehat{N}_q , defined by HSS trees. The length (size) of the generators at every merging stage (represented by a fixed level of the tree) can be chosen equal to the available upper bound on the numerical ranks of these blocks or can be adapted empirically. See [PLSZa, Section 10.1] for a recent acceleration of this stage.

Part IV. Numerical tests and conclusions

8. NUMERICAL EXPERIMENTS

Numerical experiments have been performed under our supervision in the Graduate Center of the City University of New York by Franklin Lee and Aron Wolinetz (Section 8.1) and by Liang Zhao (Section 8.2). All computations have been performed with the IEEE standard double precision. The codes are available upon request.

8.1. Experimental computation of numerical ranks of the admissible blocks of CV matrices. The test programs were written in Python 3.3.3, using the Numpy 1.7.1, Scipy 0.12.1, and Sympy 0.7.3 libraries. The tests were run on Windows 7 64-bit SP1 on a Toshiba Satellite L515-S4925 with a Pentium Dual-Core T4300 @ 2.10GHz x2 processor. Random numbers were generated uniformly with the language's Mersenne twister over the range $\{x : 0 \leq x < 1\}$ and extended to the ranges $\{y : a \leq y < b\}$ for $y = a + (b - a)x$.

For $n = 1024, 2048, 4096$ we computed the vectors $(\omega_j)_{i=0}^{n-1}$ of the n th roots of unity, and for every pair of n and h , $h = 0, 1, 4$, we generated 100,000 instances of complex numbers s_0, \dots, s_{n-1} , thus defining $n \times n$ CV matrices $C_{s,1} = (\frac{1}{s_i - \omega^j})_{i,j=0}^{n-1}$.

We generated the knots $s_i = |s_i| \exp(\phi_i \sqrt{-1})$ as follows. At first we generated the angles $\bar{\phi}_i$ over the range $0 \leq \bar{\phi}_i < 2\pi$ and the values $|s_i|$ over the range $[1 - 1/2^h, 1 + 1/2^h)$ for $h = 0, 1, 4$ and $i = 0, \dots, n - 1$, in all cases independently for all i and t . Then for every vector $(\bar{\phi}_i)_{i=0}^{k-1}$ we computed the permutation matrix P defining the vector $(\phi_i)_{i=0}^{n-1} = P(\bar{\phi}_i)_{i=0}^{k-1}$ with the coordinates $\phi_0, \dots, \phi_{n-1}$ in nondecreasing order. For every pair of vectors $(|s_i|)_{i=0}^{n-1}$ and $(\phi_i)_{i=0}^{n-1}$ we defined the vector

$$(s_i)_{i=0}^{n-1} = (|s_i| \exp(\phi_i \sqrt{-1}))_{i=0}^{n-1}$$

and the CV matrix

$$C = \left(\frac{1}{s_i - \omega^j} \right)_{i,j=0}^{n-1}.$$

Then we fixed the integers $k = 4, 32, 512, 2048$, skipped integer pairs (k, n) where $k < 2$ or $n/k < 2$, and defined tridiagonal and admissible blocks by following the recipes of Section 6.

Finally we fixed the tolerances $\xi = 10^{-q}$ for $q = 2, 3, 4$ and computed the ξ -ranks of nonempty admissible blocks $N_q^{(c)}$ by applying the rank function `numpy.linalg.matrix_rank(X, tol)`.

Tables 8.1–8.3 show the average computed values of the ξ -ranks in these tests. They vary rather little, remaining consistently small, when we changed the parameters h , k , and ξ , and they grew very slowly when we doubled the matrix dimension n .

We also computed the average norms of the admissible blocks. They ranged between 100 and 1000.

TABLE 8.1. The ξ -ranks of the admissible blocks for $h = 0$

ξ	n	$k = 4$	$k = 32$	$k = 512$
0.01	1024	5.0	5.0	2.0
0.01	2048	5.0	5.0	3.0
0.01	4096	5.0	5.0	3.8
0.001	1024	6.0	6.0	2.0
0.001	2048	6.0	6.0	3.8
0.001	4096	6.0	6.3	4.3
0.0001	1024	7.0	7.0	2.0
0.0001	2048	7.0	7.0	4.0
0.0001	4096	7.0	7.8	5.0

TABLE 8.2. The ξ -ranks of the admissible blocks for $h = 1$

ξ	n	$k = 4$	$k = 32$	$k = 512$
0.01	1024	4.0	5.0	2.0
0.01	2048	4.0	5.0	3.4
0.01	4096	5.0	5.8	4.0
0.001	1024	5.0	6.0	2.0
0.001	2048	5.0	6.0	4.0
0.001	4096	6.0	7.0	4.8
0.0001	1024	6.0	7.0	2.0
0.0001	2048	6.0	7.0	4.0
0.0001	4096	6.0	8.0	5.4

TABLE 8.3. The ξ -ranks of the admissible blocks for $h = 4$

ξ	n	$k = 4$	$k = 32$	$k = 512$
0.01	1024	4.0	5.0	2.0
0.01	2048	4.0	5.0	4.0
0.01	4096	4.0	5.0	5.0
0.001	1024	4.0	6.0	2.0
0.001	2048	5.0	6.0	4.0
0.001	4096	5.0	6.0	5.9
0.0001	1024	5.0	7.0	2.0
0.0001	2048	5.0	7.0	4.0
0.0001	4096	6.0	7.0	6.6

8.2. Multipoint numerical evaluation of polynomials. We tested numerical behavior of our algorithms for approximate evaluation of real and complex Gaussian random polynomials $p(x)$ of degree $n - 1$, for $n = 64, 128, 256, 512, 1024, 2048, 4096$, and generated the knots of the evaluation lying in the unit disc $\{z : |z| \leq 1\}$.

We performed the tests on a Dell server running Windows system and using MATLAB R2014a with double precision. We applied the MATLAB function "randn()" in order to generate the real polynomial coefficients and the real and imaginary parts separately for the complex coefficients.

The knots of the evaluation, $s_i = r \times \exp(2\pi\theta\sqrt{-1})$, depended on two parameters r and θ . In all tests we defined the values θ by applying the uniform random number generator "rand()" to the line interval $[0, 1)$, and we generated the absolute values r in two ways.

In one series of our tests we set the absolute value r to 1, thus placing the knots s_i onto the unit circle $\{x : |x| = 1\}$, and then we displayed the test results in Tables 8.4 and 8.6.

In another series of our tests we generated the absolute value r at random by applying the uniform random number generator "rand()" to the line interval $[0, 1]$, and then we displayed the test results in Tables 8.5 and 8.7. The latter tests cover polynomial evaluation at the knots lying in the unit disc $\{z : |z| \leq 1\}$, but can be extended to the evaluation outside it, by shifting from a polynomial $p(x)$ of degree n to the reverse polynomial $x^n p(1/x)$.

TABLE 8.4. Evaluation of real Gaussian polynomials on the unit circle

Degree	Max. Rank	Error
32	13	6.60×10^{-07}
64	11	8.05×10^{-08}
128	12	5.88×10^{-07}
256	12	4.01×10^{-07}
512	12	2.27×10^{-07}
1024	12	5.77×10^{-08}
2048	13	1.38×10^{-06}
4096	13	2.99×10^{-05}

TABLE 8.5. Evaluation of real Gaussian polynomial in the unit disk

Degree	Max. Rank	Error
32	18	1.90×10^{-06}
64	13	1.47×10^{-06}
128	13	1.13×10^{-06}
256	12	9.09×10^{-07}
512	13	7.05×10^{-07}
1024	12	5.49×10^{-07}
2048	13	4.67×10^{-07}
4096	13	3.80×10^{-07}

TABLE 8.6. Evaluation of complex Gaussian polynomials on the unit circle

Degree	Max. Rank	Error
32	12	5.68×10^{-08}
64	11	5.05×10^{-07}
128	12	1.41×10^{-07}
256	11	1.42×10^{-07}
512	12	2.73×10^{-07}
1024	12	5.34×10^{-08}
2048	13	5.18×10^{-06}
4096	13	1.62×10^{-04}

In all tables the columns “Max. Rank” represent the maximum ξ -ranks of the off-tridiagonal blocks in the computation, for $\xi = 10^{-5}$. The columns “Error” represent the absolute difference of our computed values of the polynomials and the output of the MATLAB function ”polyval()” for the same inputs.

All tests have been repeated 100 times for each n and the average results have been displayed.

According to the test results, the computed maximum numerical rank was consistently low, implying that our algorithm ran fast, even though it still produced quite accurate output values.

TABLE 8.7. Evaluation of complex Gaussian polynomial in the unit disk

Degree	Max. Rank	Error
32	18	1.77×10^{-06}
64	13	1.39×10^{-06}
128	13	1.16×10^{-06}
256	12	8.71×10^{-07}
512	12	6.97×10^{-07}
1024	12	5.40×10^{-07}
2048	13	4.73×10^{-07}
4096	13	3.86×10^{-07}

TABLE 8.8. Polynomial evaluation by using the algorithm of [MB72]

Degree	Real Gaussian		Complex Gaussian	
	mean	std	mean	std
16	5.19×10^{-09}	1.21×10^{-08}	8.91×10^{-11}	6.50×10^{-11}
32	4.54×10^{-02}	6.72×10^{-02}	1.66×10^{-03}	8.86×10^{-04}
64	$9.47 \times 10^{+21}$	$2.99 \times 10^{+22}$	$2.96 \times 10^{+11}$	$1.22 \times 10^{+11}$
128	$2.87 \times 10^{+53}$	$7.21 \times 10^{+53}$	$2.12 \times 10^{+164}$	<i>Inf</i>

For comparison, Table 8.8 displays the mean values and standard deviations of the output errors observed in our test of the polynomial evaluation algorithm of [MB72] applied to the same inputs and also with the IEEE standard double precision. According to these results, the algorithm has consistently performed with much inferior output accuracy for polynomials of degree 32 and higher.

9. CONCLUSIONS

The papers [MRT05], [CGS07], [XXG12], and [XXCB14] combine the FMM/HSS techniques with the transformation of matrix structures (traced back to [P90]) in order to devise fast algorithms that approximate the solution of Toeplitz, Hankel, Toeplitz-like, and Hankel-like linear systems of equations by using nearly linear number of arithmetic operations performed with bounded precision. We yielded similar results (that is, used nearly linear number of arithmetic operations performed with bounded precision) for multiplication of Vandermonde and Cauchy matrices by a vector, the solution of linear systems of equations with these matrices, and polynomial multipoint evaluation and interpolation. This can be compared with quadratic arithmetic time of the known algorithms. The more involved techniques of 2D FMM should help to decrease our upper bounds $\alpha(M)$ by a logarithmic factor (cf. [B10, Section 3.6]).

Our Section 7.1 and the papers [P15] and [Pa16] cover some extensions of our techniques and results to computations with other structured matrices and rational functions. Our study also promises a natural extension to the important class of polynomial Vandermonde matrices, $V_{\mathbf{P},\mathbf{s}} = (p_j(x_i))_{i,j=0}^{m-1,n-1}$, where $\mathbf{P} = (p_j(x))_{j=0}^{n-1}$ is any basis in the space of polynomials of degree less than n . This extension

should exploit the following generalization of our equation (3.1), which reproduces [P01, equation (3.6.8)],

$$C_{\mathbf{s},\mathbf{t}} = \text{diag}(l(s_i)^{-1})_{i=0}^{m-1} V_{\mathbf{P},\mathbf{s}} V_{\mathbf{P},\mathbf{t}}^{-1} \text{diag}(l'(t_j))_{j=0}^{n-1}, \quad l(x) = \prod_{j=0}^{n-1} (x - t_j).$$

For a natural further direction, we plan to recast our algorithms into the form of algorithms for computations with H and H2 matrices. This will enable us to apply the efficient subroutines available in the HLib library developed at the Max Planck Institute for Mathematics in the Sciences by L. Grasedyck and S. Börm, www.hlib.org, and in the H2Lib, <http://www.h2lib.org/>, <https://github.com/H2Lib/H2Lib>.

ACKNOWLEDGEMENTS

Our research has been supported by the NSF Grants CCF 1116736 and CCF-1563942 and PSC CUNY Awards 67699-00 45 and 68862-00 46. We also greatly appreciate the reviewers' thoughtful and helpful comments.

REFERENCES

- [B09] S. Börm, *Construction of data-sparse \mathcal{H}^2 -matrices by hierarchical compression*, SIAM J. Sci. Comput. **31** (2009), no. 3, 1820–1839, DOI 10.1137/080720693. MR2491547
- [B10] S. Börm, *Efficient Numerical Methods for Non-Local Operators: \mathcal{H}^2 -Matrix Compression, Algorithms and Analysis*, EMS Tracts in Mathematics, vol. 14, European Mathematical Society (EMS), Zürich, 2010. MR2767920
- [BEGO08] T. Bella, Y. Eidelman, I. Gohberg, and V. Olshevsky, *Computations with quasiseparable polynomials and matrices*, Theoret. Comput. Sci. **409** (2008), no. 2, 158–179, DOI 10.1016/j.tcs.2008.09.008. MR2474333
- [BF00] D. A. Bini and G. Fiorentino, *Design, analysis, and implementation of a multiprecision polynomial rootfinder*, Numer. Algorithms **23** (2000), no. 2-3, 127–173, DOI 10.1023/A:1019199917103. MR1772050
- [BGH03] S. Börm, L. Grasedyck, and W. Hackbusch, *Introduction to hierarchical matrices with applications*, Engineering Analysis with Boundary Elements **27** (2003), no. 5, 405–422.
- [BH02] W. Hackbusch and S. Börm, *Data-sparse approximation by adaptive \mathcal{H}^2 -matrices*, Computing **69** (2002), no. 1, 1–35, DOI 10.1007/s00607-002-1450-4. MR1954142
- [BM75] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier Publishing Co., Inc., New York-London-Amsterdam, 1975. Elsevier Computer Science Library; Theory of Computation Series, No. 1. MR0468309
- [CDG06] S. Chandrasekaran, P. Dewilde, M. Gu, W. Lyons, and T. Pals, *A fast solver for HSS representations via sparse matrices*, SIAM J. Matrix Anal. Appl. **29** (2006/07), no. 1, 67–81, DOI 10.1137/050639028. MR2288014
- [CGS07] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, *A superfast algorithm for Toeplitz systems of linear equations*, SIAM J. Matrix Anal. Appl. **29** (2007), no. 4, 1247–1266, DOI 10.1137/040617200. MR2369294
- [DGR96] A. Dutt, M. Gu, and V. Rokhlin, *Fast algorithms for polynomial interpolation, integration, and differentiation*, SIAM J. Numer. Anal. **33** (1996), no. 5, 1689–1711, DOI 10.1137/0733082. MR1411845
- [DV98] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, MA, 1998. MR1641480
- [EG02] Y. Eidelman and I. Gohberg, *A modification of the Dewilde-van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl. **343/344** (2002), 419–450, DOI 10.1016/S0024-3795(01)00363-9. Special issue on structured and infinite systems of linear equations. MR1878953

- [EGH13] Y. Eidelman, I. Gohberg, and I. Haimovici, *Separable Type Representations of Matrices and Fast Algorithms. Vol. 1: Basics. Completion Problems. Multiplication and Inversion Algorithms*, Operator Theory: Advances and Applications, vol. 234, Birkhäuser/Springer, Basel, 2014. MR3137083
- [G88] A. Gerasoulis, *A fast algorithm for the multiplication of generalized Hilbert matrices with vectors*, Math. Comp. **50** (1988), no. 181, 179–188, DOI 10.2307/2007921. MR917825
- [GGS87] A. Gerasoulis, M. D. Grigoriadis, and L. Sun, *A fast algorithm for Trummer’s problem*, SIAM J. Sci. Statist. Comput. **8** (1987), no. 1, S135–S138, DOI 10.1137/0908017. MR873954
- [GH03] L. Grasedyck and W. Hackbusch, *Construction and arithmetics of \mathcal{H} -matrices*, Computing **70** (2003), no. 4, 295–334, DOI 10.1007/s00607-003-0019-1. MR2011419
- [GKO95] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp. **64** (1995), no. 212, 1557–1576, DOI 10.2307/2153371. MR1312096
- [GL13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013. MR3024913
- [GOS08] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *How to find a good submatrix*, Report 08-10, ICM HKBU, Kowloon Tong, Hong Kong, 2008.
- [GT01] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, Structured matrices in mathematics, computer science, and engineering, I (Boulder, CO, 1999), Contemp. Math., vol. 280, Amer. Math. Soc., Providence, RI, 2001, pp. 47–51, DOI 10.1090/conm/280/4620. MR1850400
- [H72] E. Horowitz, *A fast method for interpolation using preconditioning*, Information Processing Lett. **1** (1971/72), 157–163. MR0315864
- [H99] W. Hackbusch, *A sparse matrix arithmetic based on \mathcal{H} -matrices. I. Introduction to \mathcal{H} -matrices*, Computing **62** (1999), no. 2, 89–108, DOI 10.1007/s006070050015. MR1694265
- [HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev. **53** (2011), no. 2, 217–288, DOI 10.1137/090771806. MR2806637
- [K68] D. E. Knuth, *The Art of Computer Programming. Vol. 1: Fundamental Algorithms*, Addison-Wesley, Reading, MA, 1997. Third edition [of MR0286317]. MR3077152
- [LWMRT] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert, *Randomized algorithms for the low-rank approximation of matrices*, Proc. Natl. Acad. Sci. USA **104** (2007), no. 51, 20167–20172, DOI 10.1073/pnas.0709640104. MR2366406
- [M11] M. W. Mahoney, *Randomized algorithms for matrices and data*, Foundations and Trends in Machine Learning, NOW Publishers, **3**, **2**, 2011. (Abridged version in: Advances in Machine Learning and Data Mining for Astronomy, edited by M. J. Way, et al., pp. 647–672, 2012.)
- [M11a] P. G. Martinsson, *A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix*, SIAM J. Matrix Anal. Appl. **32** (2011), no. 4, 1251–1274, DOI 10.1137/100786617. MR2854612
- [MB72] R. Moenck, A. Borodin, *Fast Modular Transform via Division*, Proc. 13th Annual Symposium on Switching and Automata Theory, 90–96, IEEE Comp. Society Press, Washington, DC, 1972.
- [MRT05] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl. **50** (2005), no. 5–6, 741–752, DOI 10.1016/j.camwa.2005.03.011. MR2165636
- [P90] V. Pan, *On computations with dense structured matrices*, Math. Comp. **55** (1990), no. 191, 179–190, DOI 10.2307/2008798. MR1023051
- [P95] V. Y. Pan, *An algebraic approach to approximate evaluation of a polynomial on a set of real points*, Adv. Comput. Math. **3** (1995), no. 1–2, 41–58, DOI 10.1007/BF02431995. MR1314901
- [P01] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser Boston, Inc., Boston, MA; Springer-Verlag, New York, 2001. MR1843842

- [P15] V. Y. Pan, *Transformations of matrix structures work again*, *Linear Algebra Appl.* **465** (2015), 107–138, DOI 10.1016/j.laa.2014.09.004. MR3274665
- [Pa16] V. Y. Pan, *How bad are Vandermonde matrices?*, *SIAM J. Matrix Anal. Appl.* **37** (2016), no. 2, 676–694, DOI 10.1137/15M1030170. MR3504989
- [Pb] V. Y. Pan, *Fast approximation algorithms for computations with Cauchy matrices, polynomials, and rational functions*, arXiv:1506.02285 [math.NA] 34 pages, 7 figures, 8 tables, June 7, 2015, revised in April 2016.
- [PLSZa] Victor Y. Pan, Qi Luan, John Svadlenka, and Liang Zhao, *Primitive and cynical low rank approximation, preprocessing and extensions*, arXiv:1611.01391 [math.NA] (47 pages, 7 figures, 5 tables), April 2017.
- [PQY15] V. Y. Pan, G. Qian, and X. Yan, *Random multipliers numerically stabilize Gaussian and block Gaussian elimination: proofs and an extension to low-rank approximation*, *Linear Algebra Appl.* **481** (2015), 202–234, DOI 10.1016/j.laa.2015.04.021. MR3349652
- [PRT92] V. Pan, J. H. Reif, and S. R. Tate, *The Power of Combining the Techniques of Algebraic and Numerical Computing*, *Procs. of FOCS'92*, IEEE Comp. Soc. Press, 1992, pp. 703–713.
- [PSLT93] V. Pan, A. Sadikou, E. Landowne, and O. Tiga, *A new approach to fast polynomial interpolation and multipoint evaluation*, *Comput. Math. Appl.* **25** (1993), no. 9, 25–30, DOI 10.1016/0898-1221(93)90129-J. MR1208387
- [PZHY97] V. Y. Pan, A. Zheng, X. Huang, and Y. Yu, *Fast multipoint polynomial evaluation and interpolation via computations with structured matrices*, *Ann. Numer. Math.* **4** (1997), no. 1-4, 483–510. MR1422699
- [R85] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, *J. Comput. Phys.* **60** (1985), no. 2, 187–207, DOI 10.1016/0021-9991(85)90002-6. MR805870
- [T00] E. E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic-skeleton method*, *Computing* **64** (2000), no. 4, 367–380, DOI 10.1007/s006070070031. MR1783468
- [VVM] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices* (Volumes 1 and 2), The Johns Hopkins University Press, Baltimore, Maryland, 2007/2008.
- [W14] D. P. Woodruff, *Sketching as a tool for numerical linear algebra*, *Found. Trends Theor. Comput. Sci.* **10** (2014), no. 1-2, iv+157. MR3285427
- [X12] J. Xia, *On the complexity of some hierarchical structured matrix algorithms*, *SIAM J. Matrix Anal. Appl.* **33** (2012), no. 2, 388–410, DOI 10.1137/110827788. MR2970212
- [XXCB14] Y. Xi, J. Xia, S. Cauley, and V. Balakrishnan, *Superfast and stable structured solvers for Toeplitz least squares via randomized sampling*, *SIAM J. Matrix Anal. Appl.* **35** (2014), no. 1, 44–72, DOI 10.1137/120895755. MR3152737
- [XXG12] J. Xia, Y. Xi, and M. Gu, *A superfast structured solver for Toeplitz linear systems via randomized sampling*, *SIAM J. Matrix Anal. Appl.* **33** (2012), no. 3, 837–858, DOI 10.1137/110831982. MR3023454

DEPARTMENTS OF MATHEMATICS AND COMPUTER SCIENCE, LEHMAN COLLEGE AND THE GRADUATE CENTER OF THE CITY UNIVERSITY OF NEW YORK, BRONX, NEW YORK 10468

E-mail address: victor.pan@lehman.cuny.edu