Nearly optimal computations with structured matrices

Victor Y. Pan
Mathematics and Computer Science Department
Lehman College, CUNY, Bronx, NY 10468
internet: vpan@alpha.lehman.cuny.edu
(Supported by NSF Grant CCR 9732206 and PSC CUNY Award 669363)

Abstract
We propose a nearly optimal algorithm that uses $2n-2$ random parameters, $O(n)$ memory space and $O((n\log^2 n)\log\log n)$ operations in a fixed arbitrary field in order to compute the rank and a basis for the null space of a structured $n \times n$ matrix X represented with $O(n)$ parameters of its short generator, as well as to solve a linear system $X\mathbf{y} = \mathbf{b}$ or to determine its inconsistency. If rank $X = n$, the algorithm also computes det X and a short generator of X^{-1}. The cost bounds cover correctness verification for the output but not the cost of the generation of random parameters. The algorithm gives a unified treatment of various classes of structured matrices including ones of Toeplitz, Hankel, Vandermonde and Cauchy types.

1 Introduction

1.1 Computations with structured matrices

Computations with structured matrices is a classical subject with long history and is a major subject of computer algebra and applied linear algebra (cf. [4], [11], [17]–[19]). Such matrices are omnipresent in signal and image processing, sciences and engineering. The next table displays some best known structured matrices.

Table 1

Toeplitz matrices: $T = (t_{i-j})_{i,j=0}^{n-1}$	Hankel matrices: $H = (h_{i+j})_{i,j=0}^{n-1}$
Vandermonde matrices: $V = (t_i^j)_{i,j=0}^{n-1}$	Cauchy matrices: $C = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$

The most celebrated are *Toeplitz* and *Hankel* matrices, but *Vandermonde* and *Cauchy* matrices are well recognized too, in particular for their applications to celestial mechanics and algebraic decoding and their correlation to polynomial and rational interpolation and multipoint evaluation (see [3], [30], [33], [29], [23]-[25], and bibliography therein). The four more general classes of *Toeplitz-like*, *Hankel-like*, *Vandermonde-like* and *Cauchy-like matrices* cover also many other important classes of structured matrices (such as Bezoutians, Loewner, Pick, Sylvester and subresultant matrices) and can be defined in terms of the associated linear operators of displacement and scaling. Equivalently, such an $n \times n$ matrix X can be expressed as a bilinear or trilinear combination of a few diagonal, Toeplitz, Hankel, Vandermonde and/or Cauchy matrices and represented by a pair of $n \times l$ matrices for $l = O(1)$, called a *generator* of X of length l. One may operate with $O(n)$ entries of the generator instead of n^2 entries of the matrix [14], [5], [26], [3], [10]. This opens door to dramatic saving of memory space and computer time though may require nontrivial techniques to control the size of the generators in the process of the computation.

1.2 Our results

In this paper we will present a unified, working in any field and nearly optimal algorithm for all the cited matrix classes, which also allows its immediate extension to various other classes of structured matrices X. The algorithm computes the rank, a basis for the null space of an $n \times n$ structured matrix X, and a solution to a linear system $X\mathbf{y} = \mathbf{b}$ (or reports its inconsistency). If X is nonsingular, the algorithm also outputs det X and a short generator for X^{-1}. The algorithm uses $2n-2$ random parameters, $O(n)$ memory words, and $O((n\log^2 n)\log\log n)$ field operations over any field, which decreases to $O(n\log^2 n)$ if the field supports FFT. That is, it uses from $O(\log^2 n)$ to $O((\log^2 n)\log\log n)$ time per input parameter (see Corollary 6.1). The bounds cover the cost of the computation of the output and of testing its correctness but not the cost of the generation of random parameters. Hereafter we refer to the field operations as *ops*.

Our results rely on exploiting matrix structure. For general $n \times n$ matrices the known best algorithms use from $Cn^{2.38}$ (for an immense constant C) to order of $n^{\log_2 7} = n^{2.80\ldots}$ or n^3 ops [3], sect. 2.2; [9], [20].

1.3 Comparison with the known complexity results and our technical improvements So far, the nearly linear complexity estimates stated above were known only in the Toeplitz-like case [21], [2], [13] and Cauchy-like case [32]. The algorithm of [21], [2] was devised only (for strongly nonsingular matrices) over the real or complex fields. The Toeplitz-like algorithm of [13] is as fast as ours but allows failure with double probability versus ours and uses order of $n \log n$ random parameters. In the Cauchy-like case, restricted both to the computations in the field of complex numbers and to having Hermitian positive definite input matrices (such a pair of restrictions excludes singularity of the input and any auxiliary matrix), the recent algorithms of [23] use order of $\log^3 n$ ops per input parameter. [32] improves this bound by logarithmic factor by reducing the Cauchy-like case to the Toeplitz-like case, though in the latter paper some important parts of the algorithms (e.g., projection of the generators of a matrix into the generators of its blocks, reduction of the length of a generator, and/or the treatment of singularities) were not fully elaborated. For Vandermonde-like matrices, the very recent record of order of n ops per input parameter is due to [25], where it was also shown that the computation of a vector from the null space of a Vandermonde-like matrix is the bottleneck of several highly important problems of algebraic and algebraic-geometric decoding. Another important application of Vandermonde-like computations was shown in [24], where our present acceleration of the solution of a Vandermonde-like linear system of n equations from order of n^2 to $n \log^2 n$ was translated into a similar acceleration of rational interpolation. Besides achieving improved and nearly optimal computational cost estimates, our algorithm has an advantage of treating all the cited classes of matrices in a unified way and allows its immediate extension to various other structured matrices, e.g., to the important classes of Toeplitz-like+Hankel-like and Chebyshev-Vandermonde-like matrices (cf. Remarks 5.1 and 6.2).

As a goal, such an extension, assuming computations with positive definite input matrices in the complex field and with no singularities involved was stated in [23], where it was proposed to pursue the goal based on similar computations for rational interpolation. Although in [23] this approach has lead to highly important interpolation applications, in our present work we prefer a more direct matrix approach, which enables simultaneously acceleration by factor $\log n$, simplification and clarification versus [23] as well as achieving generality and universality of the resulting algorithm.

Technically, we elaborated a nearly optimal algorithm for computations with structured matrices with many more details and much greater generality than in any of the previous works. In particular, this covers the extension of the structure of the input matrices (in terms of the associated operators and short generators) in the process of their multiplication, inversion, projection into their block submatrices, and computation of the Schur complements, as well as randomization (as a means of treatment of singularities) and transformations (mappings) among various classes of structured matrices (as a means of improving the known algorithms). In some cases, our extensions and combinations of some known patterns revealed their additional power. In particular this is the case for the transformation and randomization techniques of [26] and [13], whose extension in this paper enabled an improvement of the time complexity estimate of [23], from order of $\log^3 n$ per point to roughly $O(\log^2 n)$, in addition to removing the strong restriction on the input required in [23]. Our version of the latter techniques as well as our techniques of manipulation with short generators of structured matrices, including the generator compression techniques, are of independent general interest for computations with structured matrices, and our elaboration of a unified nearly optimal algorithm should be practically valuable.

Due to the restriction on the size of this paper, we compress our presentation (see more details in [28]).

2 A divide-and-conquer algorithm for general matrices

Hereafter, $\mathbf{F}$ is a fixed field. I is the identity matrix, W^T is the transpose of a matrix W. $W^{(k)}$ is the $k \times k$ *leading principal* (northwestern) *submatrix* of W. A matrix W of rank ρ has *generic rank profile* if det $W^{(k)} \neq 0$ for $k = 1, \ldots, \rho$. Such a matrix is *strongly nonsingular* if it is nonsingular. By applying block Gauss-Jordan elimination to the 2×2 nonsingular block matrix

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}, \quad \det X_{11} \neq 0, \tag{2.1}$$

we factorize X and X^{-1} as follows:

$$\begin{pmatrix} I & 0 \\ X_{21}X_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} X_{11} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & X_{11}^{-1}X_{12} \\ 0 & I \end{pmatrix}, \tag{2.2}$$

$$\begin{pmatrix} I & -X_{11}^{-1}X_{12} \\ 0 & I \end{pmatrix} \begin{pmatrix} X_{11}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -X_{21}X_{11}^{-1} & I \end{pmatrix}. \tag{2.3}$$

$$S = S(X_{11}, X) = X_{22} - X_{21}X_{11}^{-1}X_{12}$$

is called the *Schur complement of* X_{11} *in* X.

PROPOSITION 2.1. $\det X = (\det X_{11})\det S$.

PROPOSITION 2.2. *[9]. If X is strongly nonsingular, then so are B and S too.*

PROPOSITION 2.3. *[3]. If X and X_{11} are nonsingular, S^{-1} forms the southeastern block of X^{-1}.*

PROPOSITION 2.4. *[3], [9]. Let $S = S(X^{(k)}, X)$ and $S_1 = S(S^{(h)}, S)$ be defined, that is, let $X^{(k)}$ and $S^{(h)}$ be nonsingular. Then $S_1 = S(X^{(k+h)}, X)$.*

Due to Proposition 2.2, we may extend factorization (2.2), (2.3) to X_{11} and S and recursively continue such a *descending process* until we arrive at 1×1 matrices (cf. [35], [1]). In actual computation, we apply *lifting process* that begins with the inversion of the 1×1 matrix $X^{(1)}$ and finally outputs the *CRF* (*complete recursive factorization*) of X. The CRF is *balanced* if the depth of the recursion from $X \in \mathbf{F}^{n \times n}$ equals $\lceil \log n \rceil$.

Algorithm 2.1. *Computation of the CRF and the inversion of a strongly nonsingular matrix.*

Input: a field $\mathbf{F}$ and a strongly nonsingular $n \times n$ matrix X of (2.1), $X \in \mathbf{F}^{n \times n}$.

Output: balanced CRF of X, including X^{-1}.

Computations:

1. Apply Algorithm 2.1 to the matrix X_{11} to compute the balanced CRF of X_{11}, including X_{11}^{-1}. Compute X_{11}^{-1} via a single division if X_{11} is a 1×1 matrix or based on the extension of (2.3) to the factorization of X_{11}^{-1}.
2. Compute $S = X_{22} - X_{21}X_{11}^{-1}X_{12}$.
3. Apply Algorithm 2.1 to the matrix S to compute the balanced CRF of S (including S^{-1}).
4. Compute X^{-1} from (2.3).

By extending Algorithm 2.1, we may compute the solution $\mathbf{y} = X^{-1}\mathbf{b}$ to a linear system $X\mathbf{y} = \mathbf{b}$ and det X (successively compute det X_{11}, det S, and det X at stages 1, 3, and 4 (see Proposition 2.1)).

In the extension to the case where the matrix X can be singular but has generic rank profile, **generalized Algorithm 2.1** counts the number of divisions (that is, the inversions of 1×1 matrices involved). Whenever division by 0 occurs, we have ρ =rank X in the counter, and then continue the computations until we compute the CRF of $X^{(\rho)}$. Let us also compute an $n \times (n - \rho)$ matrix N whose columns N form a basis for the null space of X (see, e.g., [3], p. 110). That is, recall (2.1) with $X_{11} = X^{(\rho)}$, write

$$(2.4)\quad F = \begin{pmatrix} I_\rho & -X_{11}^{-1}X_{12} \\ 0 & I_{n-\rho} \end{pmatrix}, \quad N = F\begin{pmatrix} 0 \\ I_{n-\rho} \end{pmatrix},$$

and verify easily that $XN = 0$ and N has full rank.

Furthermore, the substitution of $\mathbf{y} = F\mathbf{z}$ reduces the solution of a linear system $X\mathbf{y} = \mathbf{b}$ (or the determination of its inconsistency) to the case of the system $(XF)\mathbf{z} = \mathbf{b}$, for which the problem is simple because $XN = 0$ and we already know X_{11}^{-1}.

DEFINITION 2.1. *The* output set *of generalized Algorithm 2.1 consists of the rank ρ of X, a largest nonsingular submatrix of X, its inverse, a basis for the null space of X, a solution $\mathbf{y}$ to the linear system $X\mathbf{y} = \mathbf{b}$ for a given vector $\mathbf{b}$ (or the determination of its inconsistency), and if $\rho = n$, then also X^{-1} and $\det X$.*

3 Structured matrices and their generators (definitions and some basic properties)

DEFINITION 3.1. *(Cf. Table 1.)* $T = (t_{i,j})_{i,j=0}^{n-1} \in \mathbf{F}^{n \times n}$ *is a* Toeplitz matrix *if* $t_{i+1,j+1} = t_{i,j}, i, j = 0, \ldots, n-2$. $H = (h_{i,j})_{i,j=0}^{n-1} \in \mathbf{F}^{n \times n}$ *is a* Hankel matrix *if* $h_{i+1,j-1} = h_{i,j}$, $i = 0, \ldots, n-2$; $j = 1, \ldots, n-1$. $V(\mathbf{t}) = (t_i^j)_{i,j=0}^{n-1} \in \mathbf{F}^{n \times n}$ *is a* Vandermonde matrix , *and* $C(\mathbf{s}, \mathbf{t}) = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1} \in \mathbf{F}^{n \times n}$ *is a* Cauchy matrix, *for any pair of vectors* $\mathbf{s} = (s_i)_{i=0}^{n-1}$, $\mathbf{t} = (t_j)_{j=0}^{n-1}$ *where* $s_i \neq t_j$ *for every pair* (i, j). *(Many authors use the name "Vandermonde matrix" for* $V^T(\mathbf{t})$.*)*

DEFINITION 3.2. $\mathbf{e}_i$ *is the i-th* coordinate vector. $D(\mathbf{v}) = \mathrm{diag}(v_0, \ldots, v_{n-1}) = (v_i\mathbf{e}_i)_{i=0}^{n-1} \in \mathbf{F}^{n \times n}$ *is a* diagonal matrix. $J = (\mathbf{e}_i)_{i=n-1}^{0}$ *is the* reflection matrix. $\mathbf{v}^k = (v_i^k)_{i=0}^{n-1}$, *for* $\mathbf{v} = (v_i)_{i=0}^{n-1}$ *and an integer* k.

DEFINITION 3.3. $C_f = (\mathbf{e}_1, \ldots, \mathbf{e}_{n-1}, f\mathbf{e}_0) \in \mathbf{F}^{n \times n}$ *(for a scalar* f*) is the* unit f-circulant matrix. $(C_f\mathbf{v} = (fv_{n-1}, v_0, \ldots, v_{n-2})$ *for* $\mathbf{v} = (v_i)_{i=0}^{n-1}$.*)* $C_f(\mathbf{v}) = \sum_{i=0}^{n-1} v_i C_f^i$ *is an* f-circulant matrix. $C_1(\mathbf{v})$ *is a* circulant matrix, *and* $C_0(\mathbf{v})$ *is a lower triangular Toeplitz matrix. (In this paper, we only need to use* $f = 0, \pm 1$.*)*

FACT 3.1. $J^2 = I$, $C_0^n = 0$, $C_f^{-1} = C_{1/f}^T$ *for* $f \neq 0$, $JC_fJ = C_f^T$ *for any* f. $JD(\mathbf{v})J = D(J\mathbf{v})$ *for any* $\mathbf{v}$.

FACT 3.2. *TJ and JT are Hankel matrices if T is a Toeplitz matrix, and vice versa, HJ and JH are Toeplitz matrices if H is a Hankel matrix.*

The next definition (cf. also Theorem 3.1) introduces four natural extensions of the four structured matrix classes defined above.

DEFINITION 3.4. *Given two scalars e and f, $ef \neq 1$, two vectors $\mathbf{s}$, $\mathbf{t} \in \mathbf{F}^{n \times 1}$, with $ft_i^{n+1} \neq t_i$, $s_i \neq t_j$ for all i, j, and a pair of $n \times \ell$ matrices $G = (\mathbf{g}_i)_{i=1}^{\ell}$, $M = (\mathbf{m}_i)_{i=1}^{\ell} \in \mathbf{F}^{n \times \ell}$, we write $T_f = T_f(G, M)$,*

$$T_f = (fC_f(T_f\mathbf{e}_0) + \frac{1}{1-ef}\sum_{k=1}^{\ell} C_f(\mathbf{g}_k)C_e^T(\mathbf{m}_k))C_{1/f}^T$$

(3.1)

for $f \neq 0$,

$$(3.2) \quad T_0 = C_0(T_0\mathbf{e}_0) + \sum_{k=1}^{\ell} C_0(\mathbf{g}_k)C_0^T(C_0\mathbf{m}_k),$$

$$V_f(\mathbf{t}, G, M) = D((\mathbf{t} - f\mathbf{t}^{n+1})^{-1})\sum_{k=1}^{\ell} D(\mathbf{g}_k)V(\mathbf{t})C_f^T(\mathbf{m}_k),$$

(3.3)

$$(3.4) \quad C(\mathbf{s}, \mathbf{t}, G, M) = \sum_{k=1}^{\ell} D(\mathbf{g}_k)C(\mathbf{s}, \mathbf{t})D(\mathbf{m}_k).$$

The matrix pair (G, M) *is called a* (K, L)-generator *(or just* a generator) *of length* ℓ *for a matrix* X *where* $K = -L = C_f$ *for* $X = T_f$ *and for any scalar* f; $K = -C_f^T$, $L = D^{-1}(\mathbf{t})$ *for* $X = V_f(\mathbf{t}, G, M)$, *and* $K = -D(\mathbf{t}), L = D(\mathbf{s})$ *for* $X = C(\mathbf{s}, \mathbf{t}, G, M)$. *For a fixed* (K, L), *the minimum* ℓ *in all* (K, L)*-generators of* X *is called the* (K, L)-rank *or a* generator rank *of* X *and is denoted by* $r_{K,L}(X)$. *The operator* $X \to XK + LX$ *is called a* basic operator *for* X, *and the pair* (K, L) *is called a* basic matrix pair *for* X.

Clearly, the matrices $T, H, V(\mathbf{t})$, and $C(\mathbf{s}, \mathbf{t})$ have short (K, L)-generators: $r_{C_f,-C_f}(T) \leq 2, r_{C_f,-C_f^T}(JT) \leq 2, r_{C_f^T,-C_f}(TJ) \leq 2, r_{-C_f^T,D^{-1}(\mathbf{t})}(V(\mathbf{t})) \leq 1$ for all f, and $r_{-D(\mathbf{t}),D(\mathbf{s})}(C(\mathbf{s}, \mathbf{t})) \leq 1$. If $\ell = O(1)$, we call the matrices T_f of (3.1), (3.2) *Toeplitz-like*, JT_f and T_fJ *Hankel-like*, $V_f(\mathbf{t}, G, H)$ of (3.3) *Vandermonde-like*, for all scalars f, and $C(\mathbf{s}, \mathbf{t}, G, H)$ of (3.4) *Cauchy-like* (cf. Tables 2-4).

For a fixed 4-tuple (X, K, L, l), (3.1)-(3.4) give nonunique compressed representations of matrix X via the $2ln$ entries of its (K, L)-generator (G, M) (and in the case of (3.1), (3.2), in addition, the n entries of the first column of X). The number of terms, ℓ, in the summations of (3.1)-(3.4) may exceed $r_{K,L}(X)$, but this can be repaired by applying the *generator compression techniques*:

FACT 3.3. *(See Proposition A.6 of [27] or Problem 2.2.11b of [3].) Let* X *stand for* $T_f, V_f(\mathbf{t}, G, M)$ *or* $C(\mathbf{s}, \mathbf{t}, G, M)$ *of (3.1)-(3.4). Let a* (K, L)*-generator* (G, M) *of a length* l *for a matrix* X *and the* (K, L)*-rank of* X, $r = r_{K,L}(X)$, *be given as an input, together with* f *and* $T_f\mathbf{e}_0$, f *and* $\mathbf{t}$, *or* $\mathbf{s}$ *and* $\mathbf{t}$, *respectively. Then it is sufficient to use* $O(\ell^2 n)$ *ops in order to compute a* (K, L)*-generator of length* r *for* X.

The next result enables equivalent definition of matrix structure in terms of the associated operators of scaling and displacement [14], [5], [3], [18].

THEOREM 3.1. *A matrix* X *satisfies Sylvester's matrix equation*

$$(3.5) \qquad XK + LX = GM^T$$

if (G, M) *is a* (K, L)*-generator of* X *for the triples* (X, K, L) *defined above, that is, if*

a) $T_fC_f - C_fT_f = GM^T$ *under (3.1), (3.2),*

b) $(JT_f)C_f - C_f^T(JT_f) = (JG)M^T$, $(T_fJ)C_f^T - C_f(T_fJ) = GM^TJ$, *under (3.1), (3.2),*

c) $D^{-1}(\mathbf{t})V_f(\mathbf{t}, G, M) - V_f(\mathbf{t}, G, M)C_f^T = GM^T$ *under (3.3), and*

d) $D(\mathbf{s})C(\mathbf{s}, \mathbf{t}, G, M) - C(\mathbf{s}, \mathbf{t}, G, M)D(\mathbf{t}) = GM^T$ *under (3.4).*

Furthermore, $r_{K,L}(X) = \text{rank } (XK + LX)$ *in all cases.*

Proof. Parts a), c), and d) of Theorem 3.1 follow from Theorems 1.1, 2.1, and 3.1 of [10] and Theorem 2.11.3a) of [3]. Part b) follows from part a) and Fact 3.1.

REMARK 3.1. *[3], [10]. There exists a unique solution to Sylvester's matrix equation (3.5) for any fixed generator* (G, M) *and each basic pair* (K, L) *of (3.3), (3.4). In the Toeplitz/Hankel-like case of (3.1), (3.2), there exists a unique solution if and only if* $\sum_{k=1}^{l} C_f(\mathbf{g}_k)C_{1/f}(\mathbf{m}_k)^T = 0$ *for* $f \neq 0$ *or* $\sum_{k=1}^{l} C_f^T(\mathbf{g}_k)\mathbf{m}_k = \sum_{k=1}^{l} C_f(J\mathbf{m}_k)\mathbf{g}_k = \mathbf{0}$ *for* $f = 0$.

REMARK 3.2. *Because* $\text{rank}(C_f - C_e) = 1$ *for* $e \neq f$, *we have close correlation between the classes* $\{T_f(G, M)\}$ *and* $\{T_e(G, M)\}$, *as well as between the classes* $\{V_f(\mathbf{t}, G, M)\}$ *and* $\{V_e(\mathbf{t}, G, M)\}$, *and we may easily define a* $(C_e^T, -C_f)$*-generator for* T_f, *a* $(C_e, -C_f^T)$*-generator for* JT_f, *and a* $(CT_e, -C_f)$*-generator for* T_fJ.

Matrix equation (3.5) can be varied by the transposition of the matrices on its both sides, by its multiplication by a scalar, e.g., by -1, and/or (cf. [8]) *by any similarity transformation* of the matrices K and L:

$$(3.6) \qquad X^TL^T + K^TX^T = HG^T,$$

$$(3.7) \qquad \hat{X}\hat{K} + \hat{L}\hat{X} = \hat{G}\hat{H}^T,$$

where $\hat{X} = UXW$, $\hat{K} = W^{-1}KW$, $\hat{L} = ULU^{-1}$, $\hat{G} = UG$, $\hat{H} = H^TW$, U and W are nonsingular matrices. In particular, for $U = W = J$, we obtain

$$(3.8) \quad (JXJ)(JKJ) + (JLJ)(JXJ) = JGH^TJ.$$

We may immediately extend the representations (3.1)-(3.4) of a matrix X to the matrices X^T, $\hat{X}$ and $\hat{X}^T$.

Tables 2-4 summarize some particular classes of structured matrices (cf. Theorem 3.1, equations (3.6) and (3.8) and Fact 3.2). Some other variations and extensions can be found in [3], pp. 187-188, and [?].

Table 2, *Toeplitz/Hankel-like matrices*

X	T_f	T_f^T	JT_f	T_fJ
K	C_f	$-C_f^T$	C_f	C_f^T
L	$-C_f$	C_f^T	$-C_f^T$	$-C_f$

Table 3, *Vandermonde-like matrices* $V_f = V_f(\mathbf{t}, G, M)$, V_f^T, JV_fJ, *and* JV_f^TJ

X	V_f	V_f^T	JV_fJ	JV_f^TJ
K	$-C_f^T$	$D^{-1}(\mathbf{t})$	$-C_f$	$D^{-1}(J\mathbf{t})$
L	$D^{-1}(\mathbf{t})$	$-C_f$	$D^{-1}(J\mathbf{t})$	$-C_f^T$

Table 4, *Cauchy-like matrices*

X	$C(\mathbf{s},\mathbf{t},G,H)$	$C^T(\mathbf{s},\mathbf{t},G,H)$
K	$-D(\mathbf{t})$	$D(\mathbf{s})$
L	$D(\mathbf{s})$	$-D(\mathbf{t})$

4 Operations with matrices represented by their short (K, L)-generators

We will accelerate the computations with structured (versus general) matrices by relying on their compact representation via their short (K, L)-generators and on the following well-known estimates.

THEOREM 4.1. *Over a field* $\mathbf{F}$, $n \times n$ *f-circulant, Toeplitz, Hankel, Vandermonde, transposed Vandermonde, and Cauchy matrices can be multiplied by a vector by using* $C_{f,v}(n) = O(n \log n)$, $T_v(n) = H_v(n) = O(n \log n)$, $V_v(n) = V_v^T(n) = O(n \log^2 n)$, *and* $C_v(n) = O(n \log^2 n)$ *ops, respectively, if* $\mathbf{F}$ *supports discrete Fourier transform at* $2^{\lceil \log_2 n \rceil}$ *points; with an extra factor* $O(\log \log n)$, *all bounds apply over any field* $\mathbf{F}$.

Proof. $V_v^T(n) = V_v(n)$ by Tellegen's theorem (see [31]). For other estimates, see, e.g., [3], [10].

COROLLARY 4.1. *The matrices* T_f, T_0, $V_f(\mathbf{t}, G, M)$ *and* $C(\mathbf{s}, \mathbf{t}, G, M)$ *of (3.1)-(3.4) can be multiplied by a vector by using* $lC_{e,v}(n)+(l+1)C_{f,v}(n)+(l+2)n+1$, $(2l+1)C_{0,v}+nl$, $(C_{f,v}(n)+V_v(n)+2n)l$, *and* $C_v(n)l+(3l-1)n$ *ops, respectively.*

Applying Algorithm 2.1 to stuctured matrices, we will rely on the next five propositions, which specify basic matrix pairs and short generators for linear combinations, products, inverses, blocks, and Schur complements of matrices. Proposition 4.2 also enables mappings among various classes of structured matrices.

PROPOSITION 4.1. *Let* $XK + LX = G_XM_X^T$, $YK + LY = G_YM_Y^T$. *Then* $(X + aY)K + L(X + aY) = (G_X, aG_Y)) \begin{pmatrix} M_X \\ M_Y^T \end{pmatrix}$, *for a scalar a.*

PROPOSITION 4.2. *(See [26] and Table 5.) Let* $-XW + QX = G_XM_X^T, YU + WY = G_YM_Y^T$. *Then* $(XY)U + Q(XY) = (G_X, XG_Y) \begin{pmatrix} M_X^TY \\ M_Y^T \end{pmatrix}$.

Proof. $(XY)U + Q(XY) = X(YU + WY) + (-XW + QX)Y$.

Table 5, *Basic matrix pair for the matrix product*

	X	Y	XY
K	$-W$	U	U
L	Q	W	Q

PROPOSITION 4.3. *Let* $XK + LX = G_XM_X^T$ *and let* $\det X \neq 0$ *in* $\mathbf{F}$. *Then* $X^{-1}L + KX^{-1} = (X^{-1}G_X)(M_X^TX^{-1})$.

Proof. Pre- and post-multiply the matrix equation $XK + LX = G_XM_X^T$ by X^{-1}.

PROPOSITION 4.4. *Let* $n = 2m$ *be even. Let us write* $K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$, $L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix}$, $X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$, $C_f = \begin{pmatrix} C_0^{(m)} & fU \\ U & C_0^{(m)} \end{pmatrix}$, $G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}$, $H = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}$, $D(\mathbf{v}) = \begin{pmatrix} D_1(\mathbf{v}) & 0 \\ 0 & D_2(\mathbf{v}) \end{pmatrix}$, *where* $K_{i,j}$, $L_{i,j}$, $X_{i,j}$, *and* $D_i(\mathbf{v})$, *for* $i, j = 1, 2$, *as well as* $C_0^{(m)}$ *and* U *are* $m \times m$ *matrices;* G_i, $M_i \in \mathbf{F}^{m \times l}$ *for* $i = 1, 2$; $U = \mathbf{e}_0^{(m)}(\mathbf{e}_{m-1}^{(m)})^T$ *is a rank-1 matrix;* $\mathbf{e}_0^{(m)}$, $\mathbf{e}_{m-1}^{(m)}$ *are unit coordinate vectors. Let (3.1) hold. Then* $X_{ij}\overline{K}_{jj} + \overline{L}_{ii}X_{ij} = G_iM_j^T + R_{i,j}$, $i, j = 1, 2$, *where*

a) $R_{11} = (fX_{11} - X_{12})U + fU(X_{21} - X_{11})$, $R_{12} = f(X_{12} - X_{11})U + fU(X_{22} - X_{12})$, $R_{21} = (fX_{21} - X_{22})U + U(X_{11} - fX_{21})$, $R_{22} = f(X_{22} - X_{21})U + U(X_{12} - fX_{22})$, *if* $X = T_f$, $K = C_f$, $L = -C_f$, $\overline{K}_{jj} = K_{jj} + fU = -\overline{L}_{ii} = -L_{ii} - fU = C_f^{(m)}$,

b) $R_{11} = f(X_{12} - X_{11})U^T$, $R_{12} = (X_{11} - fX_{12})U^T$, $R_{21} = f(X_{22} - X_{21})U^T$, $R_{22} = (X_{21} - fX_{22})U^T$, *if* $X = V_f(\mathbf{t}, G, H)$, $K = -C_f^T$, $L = D^{-1}(\mathbf{t})$, $\overline{K}_{jj} = K_{jj} - fU^T = -(C_f^{(m)})^T$, $\overline{L}_{ii} = L_{ii} = D_i(\mathbf{t})$,

c) $R_{11} = R_{12} = R_{21} = R_{22} = 0$, *if* $X = C(\mathbf{s}, \mathbf{t}, G, H)$, $K = -D(\mathbf{t})$, $L = D(\mathbf{s})$, $\overline{K}_{jj} = K_{jj} = -D_j(\mathbf{t})$, $\overline{L}_{ii} = L_{ii} = D_i(\mathbf{s})$.

PROPOSITION 4.5. *Under the assumptions of Proposition 4.4, let X_{11} be a nonsingular matrix. Then the Schur complement S of X_{11} in X satisfies the following matrix equations:* $SK_{22} + L_{22}S = (G_2 - X_{21}X_{11}^{-1}G_1)(M_2^T - M_1^T X_{11}^{-1} X_{12}) + R$,

$$(4.1) \qquad R = SK_{21}X_{11}^{-1}X_{12} + X_{21}X_{11}^{-1}L_{12}S.$$

Proof. Recall Proposition 2.3 and write $\overline{X} = X^{-1} = \begin{pmatrix} \overline{X}_{11} & \overline{X}_{12} \\ \overline{X}_{21} & S^{-1} \end{pmatrix}$. Deduce from Proposition 4.4 that $K\overline{X} + \overline{X}L = \overline{X}GM^T\overline{X}$. Deduce from the latter equation that $K_{22}S^{-1} + S^{-1}L_{22} + K_{21}\overline{X}_{12} + \overline{X}_{21}L_{12} = (\overline{X}_{21}G_1 + S^{-1}G_2)(M_1^T\overline{X}_{12} + M_2^T S^{-1})$ and, consequently,

$SK_{22} + L_{22}S = (S\overline{X}_{21}G_1 + G_2)(M_1^T\overline{X}_{12}S + M_2^T) - SK_{21}\overline{X}_{12}S - S\overline{X}_{21}L_{12}S$. Substitute the equations $\overline{X}_{12}S = -X_{11}X_{12}$ and $S\overline{X}_{21} = -X_{21}X_{11}^{-1}$ implied by (2.3) and obtain Proposition 4.5.

The expressions of Proposition 4.5 for a (K, L)-generator of S are the simplest where $K_{21} = L_{12} = 0$, which implies that $R = 0$ (cf. [23]). This is the case where $K = -L^T = C_0^T$; $K = -C_0^T$, $L = D^{-1}(\mathbf{t})$, and $K = -D(\mathbf{t})$, $L = D(\mathbf{s})$.

5 Divide-and-conquer algorithm for structured matrices

Let the input matrix X of generalized Algorithm 2.1 be given with its short (K, L)-generator for (K, L) from Tables 2-4. Application of Propositions 4.1-4.4 in the associated descending process defines the basic matrix pairs (K, L) and the (K, L)-ranks for all matrices involved in the CRF. (Here and hereafter, we slightly abuse the notation by writing the same letters K, L for the basic matrix pairs of all matrices of the CRF.)

PROPOSITION 5.1. *If the input matrix X of generalized Algorithm 2.1 is given with its (K, L)-generator from Tables 2-4, then in the associated descending process the computed basic matrix pairs (K, L) for all matrices of the resulting CRF stay in the same table.*

Proof. The proof is by recursive application of Propositions 2.3, 2.4, and 4.1-4.5.

In the lifting process of generalized Algorithm 2.1, we compute the $O((m_1 + m)l)$ entries of a short (K, L)-generator (of length $l = r_{K,L}(Y)$) for every $m \times m_1$ matrix Y of the CRF (for K, L defined in the descending process) and, in the Toeplitz/Hankel-like case, also the m or m_1 entries of the first (or last) column or row of Y, respectively; the output fully defines the matrix by (3.1)-(3.8). The computation relies on equations (2.1)-(2.3), Propositions 4.1-4.5, Corollary 4.1, and Fact 3.3 (applied to every matrix Y computed with its (K, L)-generator of length ℓ where $\ell > r_{K,L}(Y)$). We will call such a lifting process *compressed computation* of the matrices. Similarly, we represent the matrices F and N of (2.4) by a short generator of the matrix $-X_{11}^{-1}X_{12}$. Summarizing and taking into account Proposition 5.1, we obtain the following estimates:

THEOREM 5.1. *Let an $n \times n$ input matrix X of generalized Algorithm 2.1 have generic rank profile and have (K, L)-rank r for (K, L) of Theorem 3.1. Let X be given with its (K, L)-generator of length ℓ, $r \le \ell \le n$. Let $C_{f,v}(n)$, $T_v(n)$, $V_v(n)$ and $C_v(n)$ be defined as in Theorem 4.1. Then compressed computation of an output set of the algorithm requires $O(n\ell)$ words of memory and*

$$(5.1) \qquad F_{K,L}(n) = O(\ell^2 n + M_{v,K,L}(n) r^2 \log n)$$

ops where

$$(5.2) \qquad M_{v,K,L}(n) = C_{f,v}(n)$$

if $K = -L = C_f$ or $K = C_f^T$, $L = -C_f$, for any f,

$$(5.3) \qquad M_{v,K,L}(n) = V_v(n) + C_{f,v}(n)$$

if $K = -C_f^T$, $L = D^{-1}(\mathbf{t})$ for any f,

$$(5.4) \qquad M_{v,K,L}(n) = C_v(n),$$

if $K = -D(\mathbf{t})$, $\quad L = D(\mathbf{s})$.

REMARK 5.1. *Based on equations (3.6), (3.8), one may immediately extend the estimates of Theorem 5.1 to all basic matrix pairs (K, L) from Tables 2-4, and similarly for Corollary 6.1 of the next section. We also have an immediate extension to the basic matrix pairs $(C_0 + C_0^T, -C_0 - C_0^T)$ and $(C_0 - C_0^T, C_0^T - C_0)$, which define the class of Toeplitz-like + Hankel-like matrices (see [3], pp. 185-188).*

6 Transformations among structured matrices and acceleration of Vandermonde-like and Cauchy-like computations

For the Vandermonde-like and Cauchy-like input matrices X, we may apply Proposition 4.2 and Table 5 to improve the asymptotic cost estimates of (5.1), (5.3), (5.4) to yield the Toeplitz/Hankel-like level of (5.1), (5.2). Indeed, let $Y = V_f(\mathbf{t}, G, M)$ (cf. (3.4)) and write $U = -C_f^T$, $W = D^{-1}(\mathbf{t})$, $X = -V^T(\mathbf{t})$, $Q = C_f$, $YU + WY = G_Y M_Y^T$; $G_Y, M_Y \in \mathbf{F}^{n \times l}$. Then we have $QX - XW = V^T(\mathbf{t})D^{-1}(\mathbf{t}) - C_f V^T(\mathbf{t}) = \mathbf{e}_0(\mathbf{t}^{-1} - f\mathbf{t}^{n-1})^T$, and with Proposition 4.2 yield $-(XY)C_f^T + C_f(XY) = (XY)U + Q(XY) = (XG_Y)M_Y^T + \mathbf{e}_0((\mathbf{t}^{-1} - f\mathbf{t}^{n-1})^T Y)$. That is, we obtain a $(-C_f^T, C_f)$-generator of length $\ell + 1$ for the matrix $-V_f^T(\mathbf{t})Y$. By (3.3) and Corollary 4.1, this computation requires $O((\ell n \log^2 n) \log\log n)$ ops.

Similarly, we compute a $(C_f^T, -C_f)$-generator of length $l+2$ for the matrix $V^T(\mathbf{s}^{-1})YV(\mathbf{t}^{-1})$ provided that a matrix $Y = C(\mathbf{s}, \mathbf{t}, G, M)$ of (3.4) is given with its $(-D(\mathbf{t}), D(\mathbf{s}))$-generator of length l. The computation uses $(V_v^T(n) + V_v(n) + O(n))l = O((nl\log^2 n)\log\log n)$ ops. By Theorem 5.1, we may compute the output set of generalized Algorithm 2.1 applied to the two Hankel-like input matrices $X = -V_f^T(\mathbf{t})V_f(\mathbf{t}, G, M)$ and $X = V^T(\mathbf{s}^{-1})C(\mathbf{s}, \mathbf{t}, G, M)V(\mathbf{t}^{-1})$ at the cost (5.1), (5.2). Then we may obtain the partial output sets (cf. Definition 2.1) of the same algorithm for the matrices $V_f(\mathbf{t}, G, M)$ and $C(\mathbf{s}, \mathbf{t}, G, M)$ at the cost (5.1), (5.2).

COROLLARY 6.1. *Under the assumptions of Theorem 5.1, generalized Algorithm 2.1 applied to yield the compressed computation of the output set of the matrix X uses $O(n\ell)$ memory space and $F_{K,L}(n)$ ops for $F_{K,L}(n) = O(nl^2 + (nr^2\log^2 n)\log\log n)$ of (5.1), (5.2), over any field of constants. For $\ell = O(1)$, this turns into the space bound $O(1)$ and the time bound $O((\log^2 n)\log\log n)$ per an input parameter.*

The above transformations among various classes of structured matrices use *Vandermonde multipliers*, as this was proposed in [26]. In the special case where $\mathbf{s}$ and/or $\mathbf{t}$ are the (scaled) vectors of roots of 1, the Vandermonde multipliers turn into the matrices of (scaled) discrete Fourier transforms. Let us simplify the transformations in this special case [28], [8].

DEFINITION 6.1. *Write $\mathbf{e} = (g^i)_{i=0}^{n-1}$, $\mathbf{f} = (h^i)_{i=0}^{n-1}$ provided that $g^n = e, h^n = f$; $\mathbf{w} = (\omega)_{i=0}^{n-1}$ where ω is a primitive n-th root of 1, $\omega^n = 1$, $\omega^s \neq 1$ for $s = 1, \ldots, n-1$; $\Omega = (\omega^{ij})_{i,j=0}^{n-1}$. ($\Omega$ denotes the matrix of the n-point discrete Fourier transform, DFT. In the complex field $\mathbf{C}$, we may choose $\omega = exp(2\pi\sqrt{-1}/n)$, and we have $\Omega^H\Omega = I$.)*

PROPOSITION 6.1. *[6]. Let $f \neq 0$, $h^n = f$. Then*

$$C_f = U_f^{-1}D(g\mathbf{w})U_f \tag{6.1}$$

where

$$U_f = \Omega D(\mathbf{f}). \tag{6.2}$$

PROPOSITION 6.2. *Let $J\mathbf{t} = h\mathbf{w}$, $f = h^n \neq 0$, $X = JV_f(\mathbf{t}, G, H)J$,*

$$XC_f - D^{-1}(J\mathbf{t})X = G_XH_X^T \tag{6.3}$$

(cf. Table 3). Then we have (cf. Theorem 3.1a) that $T_fC_f - C_fT_f = GH^T$,

$$G = U_f^{-1}G_X, H = H_X, T_f = U_f^{-1}X, \tag{6.4}$$

for U_f of (6.2).

Proof. Pre-multiply (6.3) by U_f^{-1} and substitute equations (6.1), (6.2), and (6.4).

PROPOSITION 6.3. *[26], [12], [8]. Let $\mathbf{s} = g\mathbf{w}, \mathbf{t} = h\mathbf{w}$, $e = g^n \neq 0$, $f = h^n \neq 0$, $Y = C(\mathbf{s}, \mathbf{t}, G, H)$. Define U_e and U_f by (6.2). Let*

$$YD(\mathbf{t}) - D(\mathbf{s})Y = G_YH_Y^T \tag{6.5}$$

(cf. Table 4). Then we have (cf. Theorem 3.1a) that $\tilde{T}C_e - C_f\tilde{T} = GH^T$ where

$$G = U_f^{-1}G_Y, H^T = H_Y^TU_e, \tilde{T} = U_f^{-1}YU_e. \tag{6.6}$$

Proof. Pre-multiply (6.5) by U_f^{-1}, post-multiply by U_e, and substitute equations (6.1), (6.2), (6.6), and $U_e = \Omega D(\mathbf{e})$, which is a variation of (6.2).

REMARK 6.1. *The equation $\tilde{T}C_e - C_f\tilde{T} = GH^T$ defines a $(C_e, -C_f)$-generator (G, H) for $\tilde{T}$. Equivalently, we have $\tilde{T}C_f - C_f\tilde{T} = \tilde{G}\tilde{H}^T$ where $\tilde{G} = (G, \tilde{T}\mathbf{e}^{(0)}), \tilde{H} = (H, (f-e)\mathbf{e}^{n-1})$. By Theorem 3.1a) for $\tilde{T}, \tilde{G}, \tilde{H}$ replacing T_f, G, H, respectively, we conclude that $(\tilde{G}, \tilde{H})$ is a $(C_f, -C_f)$-generator for $\tilde{T}$, whose length increases at most by 1 versus the $(C_e, -C_f)$-generator (G, H).*

By extending Propositions 6.2 and 6.3, we may transform the (K, L)-generators of all the Vandermonde-like and Cauchy-like matrices associated with basic matrix pairs (K, L) of Tables 3 and 4 (at the cost of 1 or 2 diagonal scalings and performing 1 or 2 DFTs) into generators of the same length for Toeplitz/ Hankel-like matrices provided that K and L are of the form $\pm C_f, \pm C_f^T$ and/or $\pm D(g\mathbf{w})$ for two scalars, g and $f = g^n \neq 0$. The first (or last) column or row of every resulting matrix can be also computed easily. (All the above computations are further simplified slightly in the case where $g = f = 1$, $D(t) = I$.)

REMARK 6.2. *We may post-multiply (6.3) by U_f^{-1} (cf. (6.2)) and then substitute (6.1) to map the Vandermonde-like matrix X of (6.3) into the Cauchy-like matrix XU_f^{-1} satisfying the matrix equation $(XU_f^{-1})D(g\mathbf{w}) - D^{-1}(J\mathbf{t})(XU_f^{-1}) = G_X(H_X^TU_f^{-1})$. Similarly, we may map every matrix pair of Table 3 into one of Table 4 and also any Toeplitz/Hankel-like matrix into a closely related matrix from either of Tables 3 and 4. [8] uses such maps to improve Toeplitz-like matrix. computations with pivoting. Similarly, one may apply Discrete Consine Transforms and diagonal scaling to transform a real Chebyshev-Vandermonde-like matrix into a Cauchy-like matrix (see the definitions and specific details in [15]), and this enables immediate extension of our results to the important class of Chebyshev-Vandermonde-like matrices.*

7 Transformations of general matrix into one with generic rank profile and an extended randomized algorithm

THEOREM 7.1. *[7] (cf. also [34], [36]). Let $p(\mathbf{x}) = p(x_1, x_2, \cdots, x_m)$ be a nonzero m-variate polynomial of total degree d. Let* $\mathbf{S}$ *be a finite set of cardinality* $|\mathbf{S}|$. *Let the values $x_1^*, \cdots, x_m^*$ be sampled randomly from* $\mathbf{S}$, *that is, independently of each other under the uniform probability distribution on* $\mathbf{S}$. *Let $\mathbf{x}^* = (x_1^*, x_2^*, \cdots, x_m^*)$. Then probability$(p(\mathbf{x}^*) = 0) \le d/|\mathbf{S}|$.*

LEMMA 7.1. *[22], [4]. An $n \times n$ Cauchy matrix $C(\mathbf{s}, \mathbf{t})$ is nonsingular (together with every its square submatrix) if and only if all the 2n components of the vectors* $\mathbf{s}$ *and* $\mathbf{t}$ *are distinct.*

Next, we will generalize [16] to define structured matrices (*preconditioners*) A and B such that the matrix

$$\tilde{X} = AXB \tag{7.1}$$

has generic rank profile.

THEOREM 7.2. *Let $X \in \mathbf{F}^{n \times n}$ and $\mathbf{p}, \mathbf{q}, \mathbf{u}, \mathbf{v}$, $\mathbf{y} = (y_j)$, $\mathbf{z} = (z_j) \in \mathbf{F}^{n \times 1}$, where each pair $\mathbf{p}, \mathbf{q}$ and $\mathbf{u}, \mathbf{v}$ is filled with 2n distinct scalars, $y_0 = z_0 = 1$, and y_j, z_j are indeterminates for $j > 0$. Let $\mathbf{1} = (1)_{i=0}^{n-1}$. Then the matrix $\tilde{X}$ of (7.1) has generic rank profile if $(A, B) = (A_\alpha, B_\beta)$, $\{\alpha, \beta\} \in \{1, 2\}$ where $A_1 = C(\mathbf{p}, \mathbf{q}, \mathbf{1}, \mathbf{y})$, $B_1 = C(\mathbf{u}, \mathbf{v}, \mathbf{z}, \mathbf{1})$, $A_2 = C_0^T(\mathbf{y})$, $B_2 = C_0(\mathbf{z})$,*

Proof. For an $n \times n$ matrix M, denote by $M_{I,J}$ the determinant of the submatrix of M formed by all entries lying simultaneously in the rows indexed by the set I and in the columns indexed by the set J. Let $\rho = \text{rank } X$. For the sets $I = \{1, 2, \ldots, i\}$, $J = \{j_1, j_2, \ldots, j_i\}$, $K = \{k_1, k_2, \ldots, k_i\}$, $i = 1, 2, \ldots, \rho$, we have from the Cauchy-Binet formula that

$$\tilde{X}_{I,I} = \sum\nolimits_J \sum\nolimits_K A_{I,J} X_{J,K} B_{K,I},$$

where the summation is in all sets J and K, each made of i distinct indices. Let us prove that

$$\tilde{X}_{I,I} \ne 0 \text{ for } i = 1,\ 2,\ \ldots,\ \rho\,. \tag{7.2}$$

First let $A = A_1$, $B = B_1$. For a fixed pair of $J = [j_1, j_2, \ldots, j_i]$ and $K = [k_1, k_2, \ldots, k_i]$, we have $A_{I,J} = a y_{j_1} \cdots y_{j_i}$ where the monomial $y_{j_1} y_{j_2} \cdots y_{j_i}$ uniquely identifies the set J, and by Lemma 7.1, $a = (C(\mathbf{p}, \mathbf{q}))_{I,J}, a \ne 0$. Likewise, $B_{K,I} = b z_{k_1} \cdots z_{k_i}$ where the monomial $z_{k_1} \cdots z_{k_i}$ identifies the set K and $b = (C(\mathbf{s}, \mathbf{t}))_{K,I} \ne 0$. Therefore, for distinct pairs (J, K), the terms $A_{I,J}\ X_{J,K} B_{K,I}$ cannot cancel each other. Consequently, $\tilde{X}_{I,I} \ne 0$ provided that there exists a pair (J, K) such that $X_{J,K} \ne 0$. This is true for all $i \le \rho$ since X has rank ρ, and we arrive at (7.2).

If $A = A_2$, $B = B_2$, then we define lexicographic order for the variables: $1 < y_1 < y_2 < \ldots < y_{n-1}$ and $1 < z_1 < z_2 < \ldots < z_{n-1}$, and observe that for each set J the determinant $A_{I,J}$ is uniquely defined by its unique lowest order monomial. Furthermore, this monomial does not appear in other determinants $A_{I,J}$ for the same I. Similar property holds for $B_{K,I}$ for all sets K, and again (7.2) follows. The same arguments apply to all other pairs $A = A_\alpha$, $B = B_\beta$ where $\{\alpha, \beta\} \in \{1, 2\}$.

COROLLARY 7.1. *Under the assumptions of Theorem 7.2, let the values of the $2n - 2$ variables, y_j and z_j, $j = 1, \ldots, n-1$, be randomly sampled from a fixed finite set* $\mathbf{S}$ *of cardinality* $|\mathbf{S}|$. *Then the matrix $\tilde{X} = AXB$ of rank ρ has generic rank profile with a probability at least $1 - (\rho + 1)\rho/|\mathbf{S}|$.*

Proof. det $\tilde{X}^{(k)}$ is a polynomial in $y_1, z_1, \ldots, y_{n-1}, z_{n-1}$ of degree at most $2k$; for $k \le \rho$ it does not vanish identically, by Theorem 7.2. Now, it follows from Theorem 7.1 that det $\tilde{X}^{(k)}$ may vanish with a probability at most $2k/|\mathbf{S}|$ under the random sampling of the variables. Therefore, the probability that for $k \le \rho$ neither of det $\tilde{X}^{(k)}$ vanishes under the random sampling is at least $\prod_{k=1}^{\rho}(1 - 2k/|\mathbf{S}|) > 1 - (\rho + 1)\rho/|\mathbf{S}|$.

Corollary 7.1 can be combined with generalized Algorithm 2.1 as follows:

Algorithm 7.1. *Randomized computation of the output set for a general matrix.*

Input: a positive ϵ, a field $\mathbf{F}$, a pair $(\alpha, \beta), \{\alpha, \beta\} \in \{1, 2\}$, an $n \times n$ matrix $X \in \mathbf{F}^{n \times n}$, and a vector $\mathbf{b} \in \mathbf{F}^{n \times 1}$.

Output: FAILURE (with a probability at most ϵ) or the output set of generalized Algorithm 2.1 applied to the matrix X.

Computations.

1. Fix a finite set $\mathbf{S}$ of nonzero elements from $\mathbf{F}$ or from its algebraic extension where $2|\mathbf{S}| > (n + 1)n/\epsilon$, and randomly sample from $\mathbf{S}$ the $2n - 2$ elements y_i, z_i, $i = 1, \ldots, n-1$, defining two matrices, $A = A_\alpha$ and $B = B_\beta$ of Theorem 7.2.

2. Compute the matrix $\tilde{X} = AXB$.

3. Apply generalized Algorithm 2.1 to the matrix $\tilde{X}$, which in particular outputs $\rho = \text{rank } \tilde{X}$.

4. Compute the matrices F and N of (2.4) for X replaced by $\tilde{X}$. Verify whether the matrix $\tilde{X}N$ (formed by the $n - \rho$ last columns of the matrix $\tilde{X}F$) is a null matrix. If "not", output FAILURE,

which indicates that the randomization failed to insure the generic rank profile property for X.

5. Otherwise compute the matrix BN whose columns form a basis for the null space of X.

6. If the linear system $\tilde{X}\mathbf{w} = A\mathbf{b}$ has no solution $\mathbf{w}$, then also the system $X\mathbf{y} = \mathbf{b}$ has no solution $\mathbf{y}$. In this case output INCONSISTENT. Otherwise compute the solutions $\mathbf{w}$ and $\mathbf{y} = B\mathbf{w}$.

7. If $\rho = n$, compute $X^{-1} = B\tilde{X}^{-1}A$ and $\det X = (\det \tilde{X})/((\det A)\det B)$.

Correctness of Algorithm 7.1 is easily verified based on (2.4) and (7.1).

The computational cost (in the case of general matrix X) is clearly dominated by the cost of the application of generalized Algorithm 2.1 (we ignore the cost of generation of random parameters).

REMARK 7.1. *In ν applications of the algorithm, the probability of outputing ν FAILUREs is at most $((\rho+1)\rho/|\mathbf{S}|)^{\nu}$.*

REMARK 7.2. *To increase $|\mathbf{S}|$ in a small field $\mathbf{F}$, one may routinely shift to an algebraic extension of $\mathbf{F}$*

REMARK 7.3. *We may choose $A = A_3 = C_f^T(\mathbf{y})$, $B = B_3 = C_f^T(\mathbf{z})$, for an indeterminate f. Then, clearly, Theorem 7.2 is extended.*

REMARK 7.4. *For a nonsingular real or complex matrix X, apply symmetrization instead of randomization.*

For a singular real or complex input matrix X, one may first apply randomization to obtain a matrix $\tilde{X} = AXB$ and then apply symmetrization to the resulting matrix $\tilde{X}$, to improve numerical stability of the subsequent computations.

8 Transformation of a structured matrix into a matrix having generic rank profile

Let us specify the matrices A and B of (7.1) associated with some specific matrices from Tables 2-4. (The extension to all other matrices of these tables will be straightforward.)

a) For $X = T_f$, $X = JT_f$, $X = T_fJ$ (cf. (3.1), (3.2)) and for $X = T_f + JT_e$, we may choose $A = A_\alpha$, $B = B_\beta$, $\{\alpha, \beta\} \in \{1,2,3\}$ to preserve the Toeplitz/Hankel-like structure of X in the transition to $\tilde{X}$ (cf. Proposition 4.2 and Tables 2 and 5).

b) For $X = C(\mathbf{s}, \mathbf{t}, G, M)$ (cf. (3.4) and Tables 4 and 5), we choose $A = A_1$, $B = B_1$, $\mathbf{q} = \mathbf{s}$, $\mathbf{u} = \mathbf{t}$ and arrive at $\tilde{X} = C(\mathbf{p}, \mathbf{v}, \tilde{G}, \tilde{M})$ for any pair $(\mathbf{p}, \mathbf{v})$.

c) For $X = V_f^T(\mathbf{t}, G, M)$ (cf. (3.3) and Tables 3 and 5), we choose $A = A_2$, $B = B_1$, with $\mathbf{p} = \mathbf{t}^{-1}$ and arrive at a matrix $\tilde{X} = V_f(-\mathbf{q}^{-1}, \tilde{G}, \tilde{M})$, for $\tilde{G}, \tilde{M}$ defined by Proposition 4.2 any choice of $\mathbf{q}^{-1}$.

For Vandermonde-like and Cauchy-like computations over the fields $\mathbf{F}$ that support FFT at $n = 2^h$ points, we may achieve the cost level (5.1), (5.2) in the cases b) and c) as a by-product of randomization, by choosing as the vectors $\mathbf{p} = \mathbf{q}^{-1}$ and $\mathbf{v}$ two scaled vectors of the n-th roots of 1 having a total of $2n$ distinct components (cf. Remark 6.2 and [28]).

By Proposition 4.2, the length of the (K, L)-generator for X increases by at most 2 in the above transitions to $\tilde{X}$, so that the estimates of Theorem 5.1 and Corollary 6.1 apply to the randomized computational cost of performing Algorithm 7.1 too.

References

[1] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley, Reading, Massachusetts, 1974.

[2] R.R. Bitmead, B.D.O. Anderson, Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations, *Linear Algebra Appl.*, **34**, 103-116, 1980.

[3] D. Bini, V.Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms,* Birkhäuser, Boston, 1994.

[4] A. L. Cauchy, Mémorie sur les Fonctions Alternées et sur les Somme Alternées, *Exercises d' Analyse et de Phys. Math.*, **II**, 151-159, 1841.

[5] J. Chun, T. Kailath, H. Lev-Ari, Fast Parallel Algorithm for QR-factorization of Structured Matrices, *SIAM J. Sci. Stat. Comput.*, **8**, **6**, 899-913, 1987.

[6] R. E. Cline, R. J. Plemmons, G. Worm, Generalized Inverses of Certain Toeplitz Matrices, *Linear Algebra Appl.*, **8**, 25–33, 1974.

[7] R.A. Demillo, R.J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Process. Letters*, **7**, **4**, 193-195, 1978.

[8] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gausian Elimination with Partial Pivoting for Matrices with Displacement Structure, *Math. of Computation*, **64**, 1557-1576, 1995.

[9] G. H. Golub, C. F. Van Loan, *Matrix Computations,* Johns Hopkins University Press, Baltimore, Maryland, 1996 (third edition).

[10] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra Appl.*, **202**, 163-192, 1994.

[11] A.S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw-Hill, New York, 1970.

[12] G. Heinig, Inversion of Generalized Cauchy Matrices and the Other Classes of Structured Matrices, *Linear Algebra for Signal Processing, IMA Volume in Math. and Its Applications*, **69**, 95-114, Springer, 1995.

[13] E. Kaltofen, Analysis of Coppersmith's Block Wiedemann Algorithm for the Parallel Solution of Sparse Linear Systems, *Mathematics of Computation*, **64, 210**, 777-806, 1995.

[14] T. Kailath, S.Y. Kung, M. Morf, Displacement Ranks of Matrices and Linear Equations, *J. Math. Anal. Appl.*, **68, 2**, 395-407, 1979.

[15] T. Kailath, V. Olshevsky, Displacement Structure Approach to Chebyshev-Vandermonde and Related Matrices, *Integral Equations and Operator Theory*, **22**, 65-92, 1995.

[16] E. Kaltofen, B. D. Saunders, On Wiedemann's Method for Solving Sparse Linear Systems, *Proc. AAECC-5, LNCS*, **536**, 29-38, Springer, Berlin, 1991.

[17] T. Kailath, A. Sayed, Displacement Structure: Theory and Applications, *SIAM Review*, **37**, 297-386, 1995.

[18] T. Kailath, A. Sayed (editors), *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, 1999.

[19] T. Kailath, A. Viera, M. Morf, Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials, *SIAM Rev.*, **20**, 1, 106-119, 1978.

[20] I. Kaporin, A Practical Algorithm given two aster Matrix Multiplication, *Numerical Linear Algebra with Applications*, accepted.

[21] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proc. IEEE Intern. Conf. on ASSP*, 954-959, IEEE Comp. Soc. Press, 1980.

[22] L. Mirsky, *An Introduction to Linear Algebra*, Dover, New York, 1992.

[23] V. Olshevsky, V. Y. Pan, A Unified Superfast Algorithm for Boundary Rational Tangential Interpolation Problem, *Proc. 39th IEEE FOCS*, 192-201, 1998.

[24] V. Olshevsky, V. Y. Pan, Polynomial and Rational Evaluation and Interpolation (with Structured Matrices), *Proc. 26th ICALP, LNCS*, 1644, 585-594, Springer, 1999.

[25] V. Olshevsky, M. A. Shokrollahi, A Displacement Approach to Efficient Decoding of Algebraic-Geometric Codes, *Proc. 31st SOC*, 235-244, 1999.

[26] V.Y. Pan, On Computations with Dense Structured Matrices, *Math. of Computation*, **55, 191**, 179-190, 1990.

[27] V.Y. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, *Computers and Math. (with Applics.)*, **24, 3**, 61–75, 1992.

[28] V.Y. Pan, A Unified Superfast Divide-and-Conquer Algorithm for Structured Matrices, MSRI Preprint 1999-033, *Mathematical Sciences Research Institute*, Berkeley, California (1999).

[29] V. Y. Pan, M. Abutabanjeh, Z. Chen, E. Landowne, A. Sadikou, New Transformations of Cauchy Matrices and Trummer's Problem, *Computers and Math. (with Applics.)*, **35, 12**, 1-5, 1998.

[30] V. Y. Pan, A. Sadikou, E. Landowne, O. Tiga, A New Approach to Fast Polynomial Interpolation and Multipoint Evaluation, *Computers & Math. (with Applications)*, **25, 9**, 25-30, 1993.

[31] P. Penfield Jr., R. Spencer, S. Duinker, *Tellegen's Theorem and Electrical Networks*, MIT Press, Cambridge, Massachusetts, 1970.

[32] V. Y. Pan, A. Zheng, M. Abutabanjeh, Z.Chen, S. Providence, Superfast Computations with Singular Structured Matrices over Abstract Fields, *Proc. 2rd Workshop on Computer Algebra in Scientific Computing (CASC'99)*, (V. G. Ganzha, E. W. Mayr, and E. V. Vorontsov, Editors), 323-338, Springer, Berlin, 1999.

[33] V. Y. Pan, A. Zheng, X. Huang, Y. Yu, Fast Multipoint Polynomial Evaluation and Interpolation via Computations with Structured Matrices, *Annals of Numerical Math.*, **4**, 483–510, January 1997.

[34] J.T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *J. of ACM*, **27, 4**, 701-717, 1980.

[35] V. Strassen, Gaussian Elimination Is Not Optimal, *Numer. Math.*, **13**, 354-356, 1969.

[36] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proc. EUROSAM 79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.