Fast Cauchy-like and Singular Toeplitz-like Matrix Computations

Victor Y. Pan   Ailong Zheng

Ph.D Program in Mathematics
Graduate School and University Center
City University of New York
33 West 42nd Street, New York, NY 10036, USA
email address: azheng@email.gc.cuny.edu

Department of Mathematics and Computer Science
Lehman College, City University of New York
Bronx, NY 10468, USA
email address: vpan@lcvax.lehman.cuny.edu

January 13, 1999

Summary. An effective algorithm of Bitmead and Anderson, 1980, and Morf, 1980, computes the solution $\vec{x} = T^{-1}\vec{b}$ to a nonsingular Toeplitz or Toeplitz-like linear system $T\vec{x} = \vec{b}$, a short displacement generator for the inverse T^{-1} of T, and det T. We extend this algorithm to the similar computations with $n \times n$ Cauchy (generalized Hilbert) and Cauchy-like matrices. Recursive triangular factorization of such a matrix can be computed by our algorithm at the cost of executing $O(nr^2 \log^3 n)$ arithmetic operations, where r is the scaling rank of the input Cauchy-like matrix C ($r = 1$ if C is a Cauchy matrix). Consequently, the same cost bound applies to the computation of the determinant of C, a short scaling generator of C^{-1}, and the solution to a nonsingular linear system of n equations with such a matrix C. Our algorithm does not use the reduction to Toeplitz-like computations. We avoid singularity in this algorithm and run it in an arbitrary field by using randomization. We also ameliorate slightly Kaltofen's solver of Toeplitz-like singular linear systems in an arbitrary field.

1 Introduction

The entries of $n \times n$ dense structured matrices and of their inverses have simple expressions via $O(n)$ parameters, and their structure usually enables dramatic acceleration of the computations with such matrices. For example, for an $n \times n$ Toeplitz matrix $T = [t_{i-j}]$, its product by a vector can be computed, by using FFT, in

$$T_{Mv}(n) = O(n \ \log \ n) \tag{1.1}$$

arithmetic operations (hereafter, we refer to arithmetic operations as to *ops*). Furthermore, the well-known algorithm of [M74], [M80], [BA80] (hereafter, we refer to it as to the *MBA algorithm*) rapidly computes the recursive triangular factorization of T, as well as T^{-1}, det T, and the solution $\vec{x} = T^{-1}\vec{b}$ to a linear system $T\vec{x} = \vec{b}$ or its least-squares (normal equations) solution $(T^H T)^{-1} T^H \vec{b}$, for T^H denoting the Hermitian transpose of T. Namely, the algorithm uses a total of

$$T_{RF}(n) = O(T_{Mv}(n) \log \ n) = O(n \ \log^2 n) \tag{1.2}$$

ops. Furthermore, the algorithm and the latter complexity bound apply to the wider class of Toeplitz-like matrices, having structure of Toeplitz type, which is formally defined in terms of the associated *displacement operators* [KKM]. It became a natural technical challenge to extend such an algorithm for the cited Toeplitz-like matrix computations to other classes of dense structured matrices, in particular, to *Cauchy-like matrices.* Such matrices appear, for example, in applications to rational interpolation [D74] and rational matrix (tangential) interpolation [GO94], conformal mapping [T86], and numerical solution of integral equations [Rok85], [Re90] and have special structure naturally defined in terms of the associated scaling operators. Due to the difference from Toeplitz-like structure, the Cauchy-like extension of the MBA algorithm is not straightforward. In particular, treatment of scaling operators (versus displacement operators) requires distinct techniques. Furthermore, additional techniques (not available in [M74], [M80] and [BA80]) are needed to ensure nonsingularity of the auxiliary block matrices that ought to be inverted, particularly, in the case of computations in finite fields. Nevertheless, we elaborate such an extension in our present paper, thus meeting the cited technical challenge. We use

$$C_{RF}(n) = O(C_{Mv}(n) \log\ n) \tag{1.3}$$

ops, where $C_{Mv}(n)$ denotes the complexity of multiplication of an $n \times n$ Cauchy matrix by a vector, well known as Trummer's problem. With application of FFT, Trummer's problem can be solved in

$$C_{Mv}(n) = O(n \log^2 n) \tag{1.4}$$

ops [Ger], [GO94a], [GO92], which is off by factor log n from the bound (1.1). Consequently, we yield a Cauchy-like bound $C_{RF}(n) = O(n \log^3 n)$, which is slightly inferior to the Toeplitz-like bound (1.2), leaving some room for further improvement.

If we only wish to extend the asymptotic bound $O(n \log^2 n)$ of (1.2) to the Cauchy and Cauchy-like cases, then it suffices to apply either the algorithm of [Gast] in the Cauchy case or (in the general case) the techniques of [P89/90] for the reduction (at the cost of $O(n \log^2 n)$) from Cauchy-like to Toeplitz-like computations. In fact, the techniques of [P89/90] can be also applied for the transition from Toeplitz-like to Cauchy-like computations, and here they can be improved [H], [GKO, Sec.3] to yield the transition by means of very simple FFT based algorithms running in $O(n \log\ n)$ time. Such an easy transition to the Cauchy-like case implies fundamental role of Cauchy-like computations. Based on this transition, practically effective Toeplitz and Toeplitz-like solvers have been devised in [GKO], which became an additional motivation for the study of Cauchy-like computations.

The transition from Cauchy-like to Toeplitz-lke computations, however, has not been refined since [P89/90]; it costs order of $n \log^2 n$ ops and involves the solution of a Vandermonde linear system, which generally leads to some additional numerical problems. (In spite of several advanced Vandermonde solvers available [BP70], [H88], [H90], [GO96]. Vandermonde linear systems are well-known for being *ill-conditioned* [Ga75], [Ga90].) Our algorithm may also lead to numerical difficulties at the stage of the solution of Trummer's problem, but here one has an option of applying Rokhlin's approximation algorithm of [Rok85] (also cf. [BP], pages 261-262), or its recent major improvement in [PACLS] which substantially improves the known exact solutions of Trummer's problem for a large class of inputs, both numerically and in terms of the ops count.

On the other hand, in some applications of Cauchy matrices (in particular to Goppa codes), the computations are performed over the finite fields, where no numerical problems arise, and numerical problems do not arise in sections 5, 6, and 7 of our paper, where we elaborate extension of our algorithm to computations in finite fields or any field. Here we apply some special techniques for avoiding singularity of auxiliary blocks, in particular, randomization techniques, which replace symmetrization, applied in the real case. We cannot generally apply FFT over any field, so in sections $5-7$ we rely on polynomial multiplication as our basic operation, instead of FFT. The bounds (1.1) and (1.4) turn into the bounds

$$T_{Mv}(n) = O(P_M(n)), \tag{1.5}$$

$$C_{Mv}(n) = O(P_M(n) \log\ n) \tag{1.6}$$

provided that $P_M(n)$ ops suffice for computing the product of two polynomials of degree at most n (or computing a polynomial product modulo x^{2n+1}). We have

$$P_M(n) = O(n \log\ n) \tag{1.7}$$

if the ground field of constants supports FFT and

$$P_M(n) = O((n \log\ n) \log\log\ n), \tag{1.8}$$

over any field of constants [CK91]. Thus, allowing computations over any field increases our cost estimates only by factor $O(\log\log\ n)$.

Like MBA, our algorithms include the divide-and-conquer techniques; conceptually, we are closer to MBA than to the latter cited algorithms since we do not use operations with the associated polynomial (e.g. such as their interpolation) but directly partition the input matrix and exploit its structure expressed in terms of the associated linear operators. An interesting topic for further exploration would be to establish any correlations of our matrix computations to operations with the associated polynomials.

We organize our paper as follows. In sections 2 and 4, we recall some definitions and auxiliary facts. In section 3, we present and analyze our main algorithm. In section 5, we show how to apply symmetrization and randomization in order to avoid singularities when we perform this algorithm in the complex field and arbitrary fields, respectively. In sections 6 and 7, we consider the extension of our Cauchy-like solvers to the singular case as well as the related problem of the design of Toeplitz-like singular solvers, where we ameliorate slightly the algorithm of [K94]. In Appendix A, we recall some algorithms for Trummer's problem and for solving Cauchy linear systems of equations.

2 Some Definitions and Basic Facts

Definition 2.1. *Hereafter we will write* $D(\vec{v}) = diag(v_0, v_1, \cdots, v_{n-1})$ *for* $\vec{v} = (v_0, v_1, \cdots, v_{n-1})^T$. W^T *will denote the transpose of a matrix or a vector* W.

Definition 2.2 [GO94a]. *For a fixed field* $\mathbf{F}$ *(say,* $\mathbf{F} = \mathbf{C}$*, the field of complex numbers) and for a pair of n-dimensional vectors* $\vec{q} = (q_i)$ *and* $\vec{t} = (t_j)$*,* $q_i \neq t_j$*,* $i, j = 0, \cdots, n-1$*, an* $n \times n$ *matrix* $A \in \mathbf{F}^{n\times n}$ *is called a Cauchy-like matrix if*

$$F_{[D(\vec{q}),D(\vec{t})]}(A) = D(\vec{q})A - AD(\vec{t}) = GH^T, \tag{2.1}$$

G*,* $H \in \mathbf{F}^{n\times r}$*, and the integer r is bounded by a constant independent of n or, more generally, if according to a certain fixed measure, r is small relatively to n. Furthermore, the pair of matrices* (G, H^T) *of (2.1) is called a* $[D(\vec{q}), D(\vec{t})]$*-generator (or a scaling generator) of a length (at most) r for A and is hereafter denoted* $s.g._r(A)$*. The minimum r allowing the above representation (2.1) is equal to rank* $F_{[D(\vec{q}),D(\vec{t})]}(A)$ *and is called the* $[D(\vec{q}), D(\vec{t})]$ *-rank (or the scaling rank) of A.*

We will next recall some known properties of Cauchy-like matrices.

Lemma 2.1 [GO94a]. *Let* A, $\vec{q}$, $\vec{t}$, $G=[\vec{g}_1,\cdots,\vec{g}_r]=(\vec{u_i}^T)_{i=0}^{n-1}\in\mathbf{F}^{n\times r}$, $H=[\vec{h}_1,\cdots,\vec{h}_r]=(\vec{v_j}^T)_{j=0}^{n-1}\in\mathbf{F}^{n\times r}$ *be as in Definition 2.2, such that (2.1) holds. Then*

$$A=\sum_{m=1}^{r} diag(\vec{g}_m)C(\vec{q},\vec{t})diag(\vec{h}_m)=\left(\frac{\vec{u_i}^T\vec{v_j}}{q_i-t_j}\right)_{i,j=0}^{n-1}, \tag{2.2}$$

where $C(\vec{q},\vec{t})$ *is a Cauchy matrix.*

It follows from (2.2) that (2.1) is satisfied by matrices A of the form $(\frac{\vec{u_i}^T\vec{v_j}}{q_i-t_j})_{i,j=0}^{n-1}$, where $\vec{u_i}$ and $\vec{v_j}$ are r-dimentional vectors for $i,j=0,1,\cdots,n-1$. A *Cauchy (generalized Hilbert) matrix* $(\frac{1}{q_i-t_j})_{i,j=0}^{n-1}$ and a Loewner matrix $(\frac{r_i-s_j}{q_i-t_j})_{i,j=0}^{n-1}$ are two important special cases of Cauchy-like matrices; they have $[D(\vec{q}),D(\vec{t})]$*-ranks* (at most) 1 and 2, respectively.

Lemma 2.2 (see appendix A or [Ger]). *Given an* $n\times n$ *Cauchy matrix* A *and an* n*-dimensional vector* $\vec{v}$*, the product* $A\vec{v}$ *can be computed in* $O(n\log^2 n)$ *ops. Consequently, if* A *is an* $n\times n$ *Cauchy-like matrix given with an* $s.g._r(A)$*, then the product* $A\vec{v}$ *can be computed in* $O(nr\log^2 n)$ *ops.*

Lemma 2.3. *Let* $A_i\in\mathbf{F}^{n\times n}$, $i=1,2$, *denote two Cauchy-like matrices, such that* $F_{[D(\vec{q_i}),D(\vec{q}_{i+1})]}(A_i)=G_iH_i^T$, G_i, $H_i\in\mathbf{F}^{n\times r_i}$, $i=1,2$; $\vec{q_j}\in\mathbf{F}^{n\times 1}$, *j=1,2,3, and all components of the vector* $\vec{q_1}$ *are distinct from all components of the vector* $\vec{q_3}$*. Then the matrix* $A=A_1A_2$ *is a Cauchy-like matrix with* $F_{[D(\vec{q_1}),D(\vec{q_3})]}(A)=GH^T$, $G=[G_1,A_1G_2]$, $H=[A_2^TH_1,H_2]$, G, $H\in\mathbf{F}^{n\times r}$, $r=r_1+r_2$*. Furthermore,* $O(nr_1r_2\log^2 n)$ *ops suffice to compute its scaling generator of a length at most* r*.*

Proof. Observe that

$$F_{[D(\vec{q_1}),D(\vec{q_3})]}(A_1A_2)=G_1H_1^TA_2+A_1G_2H_2^T=GH^T,$$

$G=[G_1,A_1G_2]$, $H=[A_2^TH_1,H_2]$. To deduce the desired complexity bound of $O(nr_1r_2\log^2 n)$ ops for obtaining G and H, apply Lemmas 2.1 and 2.2. □

Remark 2.1. *Lemma 2.3 can be easily extended to the computation of the product of* k *Cauchy-like matrices for any fixed integer* $k\geq 2$*.*

Lemma 2.4 [H]. *Let* A *denote an* $n\times n$ *nonsingular Cauchy-like matrix with* $F_{[D(\vec{q}),D(\vec{t})]}(A)=GH^T$, $G=[\vec{g}_1,\cdots,\vec{g}_r]\in\mathbf{F}^{n\times r}$, $H=[\vec{h}_1,\cdots,\vec{h}_r]\in\mathbf{F}^{n\times r}$*. Then* A^{-1} *is also a Cauchy-like matrix such that* $F_{[D(\vec{t}),D(\vec{q})]}(A^{-1})=-UV^T$*, where the matrices* $U=[\vec{u}_1,\cdots,\vec{u}_r]\in\mathbf{F}^{n\times r}$, $V=[\vec{v}_1,\cdots,\vec{v}_r]\in\mathbf{F}^{n\times r}$ *satisfy* $AU=G$, $V^TA=H^T$*.*

Corollary 2.1. *Under the assumptions of Lemma 2.4, we have rank* $F_{[D(\vec{t}),D(\vec{q})]}(A^{-1})\leq r$*.*

Lemma 2.5. *Let a Cauchy-like matrix* A *satisfy (2.1) and let* $B_{I,J}$ *be a* $k\times d$ *submatrix of* A, $1\leq k,d\leq n$*. Then* $B_{I,J}$ *is a Cauchy-like matrix with a* $[D(\vec{q_I}),D(\vec{t_J})]$*-generator of a length at most* r*, where* $I=[i_1,\cdots,i_k]$, $J=[j_1,\cdots,j_d]$, $D(\vec{q_I})=diag(q_{i_1},\cdots,q_{i_k})$, $D(\vec{t_J})=diag(t_{j_1},\cdots,t_{j_d})$*.*

Proof. Deduce from (2.1) that

$$B_{I,J}=\left(\frac{\vec{u_i}^T\vec{v_j}}{q_i-t_j}\right)_{i=i_1,j=j_1}^{i_k,\ j_d}$$

and recall (2.2). $\square$

Lemma 2.6. *Let A and B denote a pair of $n\times n$ Cauchy-like matrices. Let A satisfy (2.2) and let*

$$F_{[D(\vec{q}),D(\vec{t})]}(B) = D(\vec{q})B - BD(\vec{t}) = XW^T,$$

$$B = \left(\frac{\vec{x}_i^{\,T}\vec{w}_j}{q_i - t_j}\right)_{i,j=0}^{n-1},$$

where $X^T = [\vec{x}_0,\cdots,\vec{x}_{n-1}] \in \mathbf{F}^{r_1\times n}$, $W^T = [\vec{w}_0,\cdots,\vec{w}_{n-1}] \in \mathbf{F}^{r_1\times n}$. Then the matrices $A+B$ and $A-B$ are Cauchy-like matrices associated with a $[D(\vec{q}),D(\vec{t})]$-generator of length at most $r+r_1$.

Proof. We have

$$A - B = \left(\frac{\vec{z}_i^{\,T}\vec{y}_j}{q_i - t_j}\right)_{i,j=0}^{n-1},$$

where $\vec{z}_i^{\,T} = (\vec{u}_i^{\,T},\ \ \vec{x}_i^{\,T})$, $\vec{y}_j^{\,T} = (\vec{v}_j^{\,T},\ \ -\vec{w}_j^{\,T})$, that is, $\vec{z}_i, \vec{y}_j \in \mathbf{F}^{(r+r_1)\times 1}$, which proves the lemma for the matrix $A-B$ (the proof for the matrix $A+B$ is similar). $\square$

Lemma 2.7 (cf. [C841]). *An $n\times n$ Cauchy matrix $C(\vec{q},\vec{t})$ is well-defined and nonsingular if and only if all the $2n$ components of the vectors $\vec{q}$ and $\vec{t}$ are distinct. Furthermore, every square submatrix of a nonsingular Cauchy matrix is nonsingular.*

3 Recursive factorization of a strongly nonsingular matrix

Definition 3.1. *A matrix W is strongly nonsingular if all its leading principal submatrices are nonsingular.*

Hereafter I (or I_s) denotes the $s\times s$ identity matrix, O denotes a null matrix of appropriate size.

For an $n\times n$ strongly nonsingular matrix A, we have the following identities:

$$A = \begin{pmatrix} I & O \\ EB^{-1} & I \end{pmatrix}\begin{pmatrix} B & O \\ O & S \end{pmatrix}\begin{pmatrix} I & B^{-1}C \\ O & I \end{pmatrix}, \tag{3.1}$$

$$A^{-1} = \begin{pmatrix} I & -B^{-1}C \\ O & I \end{pmatrix}\begin{pmatrix} B^{-1} & O \\ O & S^{-1} \end{pmatrix}\begin{pmatrix} I & O \\ -EB^{-1} & I \end{pmatrix}, \tag{3.2}$$

where

$$A = \begin{pmatrix} B & C \\ E & J \end{pmatrix}, \qquad S = J - EB^{-1}C, \tag{3.3}$$

B is a $k\times k$ matrix, and S is an $(n-k)\times(n-k)$ matrix, called the *Schur complement of B in A*. Factorization (3.1) represents block Gauss-Jordan elimination applied to the 2×2 block matrix A of (3.3). If the matrix A is strongly nonsingular, then the Schur complement matrix S can be obtained in $n-k$ steps of Caussian elimination.

We have the following simple result ([BP], Exercise 4 of chapter 2, on page 212):

Lemma 3.1. *If A is strongly nonsingular, so are B and S.*

The above observations imply the next lemma.

Lemma 3.2 (cf. [GL] or [BP], Proposition 2.2.3). *Let A be an $n \times n$ strongly nonsingular matrix and let S be defined by* (3.3). *Let A_1 be a leading principal submatrix of S and let S_1 denote the Schur complement of A_1 in S. Then S^{-1} and S_1^{-1} form the respective southeastern blocks of A^{-1}.*

Lemma 3.3. *If (3.1) holds, then det $A =$ (det B)det S.*

Due to Lemma 3.1, we may extend the factorization (3.1) of a strongly nonsingular matrix A to the submatrix B and to its Schur complement S, and we may recursively continue this decomposition process until we complete it by arriving at 1×1 matrices (compare [St69], [M80], [BA80]). In this process, we descend from A to the matrices B, C, E, and S, and then we similarly descend recursively from B and S to their submatrices and Schur complements. At these descending stages, we only identify the matrices involved in the recursion but do *not* compute them. For their actual computation, we recursively proceed *bottom up*, that is, we first invert the 1×1 leading principal submatrix A_1 of A, then use A_1^{-1} to compute the Schur complement S_1 of A_1 in the 2×2 leading principal submatrix A_2 of A, then invert the 1×1 matrix S_1 and the 2×2 matrix A_2, in the latter case, based on its factorization of the form (3.2), where the inverses A_1^{-1} and S_1^{-1} have already been computed. We recursively continue this lifting process until we arrive at A^{-1}. As a by-product, we compute all the matrices defined in the recursive descending process. The entire computation will be called the *CRF (or complete recursive factorization)* of A. Besides the inversion of 1×1 matrices, the CRF only requires matrix multiplications and subtractions.

Hereafter, we will always assume *balanced* CRFs, that is, we will balance the factorization (3.1) in its first step, such that B is the $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$ submatrix of A, and we will maintain the similar balancing property in all the subsequent recursive steps. The balanced CRF has depth at most $d = \lceil \log_2 n \rceil$.

Let us summarize and formalize our description in the form of a recursive algorithm.

Algorithm 3.1. *Recursive triangular factorization and inversion of a strongly nonsingular matrix.*

Input: a strongly nonsingular $n \times n$ matrix A of (2.2).

Output: balanced CRF of A, including the matrix A^{-1}.

Computations:

1. Apply Algorithm 3.1 to the matrix B (replacing A as its input) in order to compute the balanced CRF of B (including B^{-1}).
2. Compute the Schur complement $S = J - EB^{-1}C$.
3. Apply Algorithm 3.1 to the matrix S (replacing A as its input) to compute the balanced CRF of S (including S^{-1}).
4. Compute A^{-1} from (3.2).

Clearly, given A^{-1} and a vector $\vec{b}$, we may immediately compute the vector $\vec{x} = A^{-1}\vec{b}$. If we also seek det A, then it suffices to add the request for computing det B, det S, and det A at stages 1, 3, and 4, respectively.

4 Recursive factorization of a strongly nonsingular Cauchy-like matrix

Hereafter, we will assume for simplicity that $n = 2^d$ is an integer power of 2. We write $\vec{q} = (q_i)_{i=0}^{n-1}$, $\vec{q}^{\,(1)} = (q_i)_{i=0}^{\frac{n}{2}-1}$, $\vec{q}^{\,(2)} = (q_i)_{i=\frac{n}{2}}^{n-1}$, $\vec{t} = (t_i)_{i=0}^{n-1}$, $\vec{t}^{\,(1)} = (t_i)_{i=0}^{\frac{n}{2}-1}$, $\vec{t}^{\,(2)} = (t_i)_{i=\frac{n}{2}}^{n-1}$.

We will start with some auxiliary results.

Lemma 4.1. *Let A be an $n \times n$ strongly nonsingular Cauchy-like matrix with $F_{[D(\vec{q}),D(\vec{t})]}(A) = GH^T$, G, $H \in \mathbf{F}^{n\times r}$. Let A, B, C, E, J, and S satisfy (3.3). Then*

$$rankF_{[D(\vec{t}),\ D(\vec{q})]}(A^{-1}) \le r, \qquad rankF_{[D(\vec{t}^{\,(1)}),\ D(\vec{q}^{\,(1)})]}(B^{-1}) \le r, \tag{4.1}$$

$$rankF_{[D(\vec{q}^{\,(2)}),\ D(\vec{t}^{\,(2)})]}(S) \le r, \tag{4.2}$$

$$rankF_{[D(\vec{t}^{\,(2)}),\ D(\vec{q}^{\,(2)})]}(S^{-1}) \le r, \tag{4.3}$$

$$rankF_{[D(\vec{q}^{\,(1)}),\ D(\vec{t}^{\,(2)})]}(C) \le r, \qquad rankF_{[D(\vec{q}^{\,(2)}),\ D(\vec{t}^{\,(1)})]}(E) \le r. \tag{4.4}$$

Proof. Deduce (4.4) from Lemma 2.5. By applying Corollary 2.1, obtain (4.1). Now, due to Lemmas 3.2 and 2.5, we have (4.3). Then we apply Corollary 2.1 to the matrix $S = (S^{-1})^{-1}$ and obtain (4.2). □

Fact 4.1 (cf. Proposition A.6 of [P92b], [P93a], or [BP], Problem 2.2.11b). *Given an $s.g._{r^*}(A) = (G, H)$ and the scaling rank r of A, $r < r^* \le n$, one can compute an $s.g._r(A)$ by using $O(r^2n)$ ops.*

Now we are ready to present the computational complexity estimates.

Theorem 4.1. *Let A denote an $n \times n$ strongly nonsingular Cauchy-like matrix with its F-generator of a length r for the operator $F = F_{[D(\vec{q}),D(\vec{t})]}$. Then the respective F-generators of all the matrices encountered in the balanced CRF of A (including an $s.g._r(A^{-1})$) can be computed in $O(nr^2 \log^3 n)$ ops and can be stored by using $O(nr \log\ n)$ words of storage space; furthermore, $O(nr^2 \log^3 n)$ ops and $O(nr \log\ n)$ words of storage space suffice in this case in order to compute det A.*

Proof. Let us apply the fast version of Algorithm 3.1 to the matrix A of Theorem 4.1, that is, instead of slower computations with matrices, let us perform faster computations with their short scaling generators. Let $\phi_r(n)$ ops be involved in computing the balanced CRF of A (including the computation of an $s.g._r(A^{-1})$). Furthermore, let $\sigma_r(n)$ ops be used for computing an $s.g._r(S)$ from given $s.g._r(B^{-1})$, $s.g._r(C)$, $s.g._r(-E)$, and $s.g._r(J)$ (cf.(3.3)), and let $\mu_r(n)$ ops be required for computing an $s.g._r(A^{-1})$ from given $s.g._r(B^{-1})$, $s.g._r(C)$, $s.g._r(E)$, and $s.g._r(S^{-1})$ (cf.(3.2)). This is summarized in the table 1.

Table 1

Input	$s.g._r(A)$	$s.g._r(B^{-1}), s.g._r(C),$ $s.g._r(-E), s.g._r(J)$	$s.g._r(B^{-1}), s.g._r(C),$ $s.g._r(E), s.g._r(S)$
Output	CRF of A	$s.g._r(S)$	$s.g._r(A^{-1})$
ops	$\phi_r(n)$	$\sigma_r(n)$	$\mu_r(n)$

Let $\phi_r(k)$, $\sigma_r(k)$, and $\mu_r(k)$ denote the similar estimates where the input matrix A is replaced by a strongly nonsingular $k \times k$ matrix W given with an $s.g._r(W)$. For simplicity, let n be even. Then, in view of Lemma 4.1, the examination of Algorithm 3.1 gives us the bound

$$\phi_r(n) \leq 2\phi_r(\frac{n}{2}) + \sigma_r(n) + \mu_r(n). \tag{4.5}$$

By expanding (3.2), we deduce that

$$A^{-1} = \begin{pmatrix} B^{-1} + B^{-1}CS^{-1}EB^{-1} & -B^{-1}CS^{-1} \\ -S^{-1}EB^{-1} & S^{-1} \end{pmatrix}. \tag{4.6}$$

Now we apply Lemmas 2.3, 2.5, 2.6, 4.1, and Fact 4.1 and deduce that

$$\sigma_r(n) = O(nr^2 \log^2 n), \quad \mu_r(n) = O(nr^2 \log^2 n). \tag{4.7}$$

Substitute (4.7) into (4.5), recursively extend (4.5), and deduce that

$$\phi_r(n) = O(nr^2 \log^3 n),$$

which gives us the arithmetic time bound of Theorem 4.1. The storage space bound follows similarly when we inspect Algorithm 3.1 applied to the matrix A and apply Lemmas 2.3, 2.5, 2.6, and 4.1. $\square$

Remark 4.1. *The proof of Theorem 4.1 actually gives us the bound $O(r^2 C_{Mv}(n) \log\ n)$ on the number of ops involved in Algorithm 3.1 applied to an $n \times n$ strongly nonsingular matrix A given with an $s.g._r(A)$ (cf. (1.3), (1.4)). A similar argument leads to the bound $O(r^2 T_{Mv}(n) \log\ n)$ for the MBA algorithm applied to an $n \times n$ Toeplitz-like matrix A given with its displacement generator of length at most r.*

5 Ensuring strong nonsingularity of a nonsingular Cauchy-like matrix by means of symmetrization or randomization

We have showed how to compute det A and the balanced CRF of A (including an $s.g._r(A^{-1})$), assuming strong nonsingularity of the matrix A. To extend this solution to the case of any nonsingular matrix A, we will seek a strongly nonsingular $n \times n$ matrix X, such that the matrix AX is strongly nonsingular. Then we may apply our machinery to the matrices X and AX or XA, compute $(AX)^{-1} = X^{-1}A^{-1}$ or $(XA)^{-1} = A^{-1}X^{-1}$, det (AX)=det (XA), and det X, and obtain $A^{-1} = X(AX)^{-1} = (XA)^{-1}X$ and det A = det (AX)/det X. Furthermore, in the case where A is a singular matrix, the same algorithm will involve a division by 0 and thus will show us that det $A = 0$. In fact, we will also extend our algorithm to computing the rank of A, in Remark 5.2.

If our computation is performed in the field of real numbers (or in its subfield), then we could have set $X = A^T$, A^T denoting the transpose of A. Indeed, the matrix $XA = A^TA$ is positive definite and consequently strongly nonsingular provided that A is nonsingular. Moreover, in this case the condition numbers of the submatrix B, of the Schur complement S

of (3.3), and consequently of the similar matrices of smaller sizes computed in all subsequent steps of CRF do not exceed the condition number of A (cf. [GL] or [BP], Fact 2.1.4 and page 237). As a by-product, we immediately arrive at a least-squares (normal equations) solution $(A^TA)^{-1}A^T\vec{b}$ to a Cauchy-like linear system $A\vec{x} = \vec{b}$ for a Cauchy-like $m \times n$ rectangular matrix A having full rank n, $n \leq m$. A problem arises, however, when we apply Lemma 2.1 to the matrices A^TA and AA^T. Indeed, if we replace A in (2.1) by these matrices, then we also ought to replace $F_{[D(\vec{q}),D(\vec{t})]}$ by $F_{[D(\vec{t}),D(\vec{t})]}$ or $F_{[D(\vec{q}),D(\vec{q})]}$, respectively, and the assumption $q_i \neq t_j$ of Definition 2.2 is not extended. To exploit Cauchy-like structure of matrices W associated with the operator $F_{[D(\vec{t}),D(\vec{t})]}$, we may operate with W represented as the product $C^{-1}(\vec{q},\vec{t})Y$, where $Y = C(\vec{q},\vec{t})W$ and $C^{-1}(\vec{q},\vec{t})$ are Cauchy-like matrices. For computations in finite fields symmetrization does not work, but in this case we will solve our problem based on a distinct approach to defining the above matrix X, by using randomization. Namely, having a nonsingular matrix A, we will apply its randomized preconditioning in order to ensure strong nonsingularity of the resulting matrix. For computations in an arbitrary field, we will obtain a desired matrix X by using randomization based on the following simple but fundamental result (note that the estimate involved in this result is sharp and depends only on the total degree but not on the number m of variables):

Lemma 5.1 ([DL], [Schw], [Z]). *Let $p(x) = p(x_1, x_2, \cdots, x_m)$ be a nonzero m-variate polynomial of a total degree d. Let S be a finite set in the domain of the definition of $p(x)$. Let $\vec{x}^{\,*} = (x_1^*, x_2^*, \cdots, x_m^*)$ be a point in S^m, where the random values $x_1^*, \cdots, x_m^*$ are chosen in S, independently of each other and under the uniform probability distribution on S. Then*

$$probability(p(\vec{x}^{\,*}) = 0) \leq \frac{d}{|S|},$$

where $|S| = card(S)$ is the cardinality of S.

Hereafter, we will fix a sufficiently large finite set S from which we will choose all random values that we need. We will choose them from S independently of each other and assuming the uniform probability distribution of S, so as to be able to apply Lemma 5.1. Then application of Lemma 5.1 will ensure (with a high probability) that, for an $n \times n$ nonsingular Cauchy-like matrix A given with its $F_{[D(\vec{q}),D(\vec{t})]}$-generator of a length r and for an $n \times n$ matrix X defined by its $F_{[D(\vec{t}),D(\vec{s})]}$-generator with random entries, the matrix AX is strongly nonsingular. Namely, we will arrive at the following result (see an alternative approach in the next section):

Theorem 5.1. *Let A be an $n \times n$ nonsingular Cauchy-like matrix satisfying the equation (2.2). Let X be a matrix satisfying $X = YC(\vec{q},\vec{s})$,*

$$Y = \sum_{m=1}^{r} diag(\vec{g}_m^*)C(\vec{t},\vec{q})diag(\vec{h}_m^*) \ , \tag{5.1}$$

where $C(\vec{q},\vec{s}) = (\frac{1}{q_i - s_j})_{i,j=0}^{n-1}$ is a fixed nonsingular Cauchy matrix, $\vec{q} \in C^{n\times 1}$, $\vec{t} \in C^{n\times 1}$, $\vec{s} \in C^{n\times 1}$, $\vec{q}$ and $\vec{t}$ are as in Lemma 2.4, $q_i \neq s_j$, $s_i \neq t_j$ for all i and j, $\vec{g}_m^* \in C^{n\times 1}$, $\vec{h}_m^* \in C^{n\times 1}$, $m = 1, \cdots, r$, and the $2nr$ components of the $2r$ latter vectors are random values *from a fixed finite set S. Then, with a probability at least $1-\frac{n(n+1)}{|S|}$, AX is a strongly nonsingular Cauchy-like matrix having an F-rank of at most $2r+1$.*

Proof. First consider matrix Y of (5.1), where the random vectors $\vec{g}_m^*$ and $\vec{h}_m^*$ are replaced by generic vectors whose components are indeterminates. Recall that the $F_{[D(\vec{t}),D(\vec{q})]}$-rank of A^{-1} is at most r, due to Lemma 2.4. Therefore, there exists an assignment of values to the components of the vectors $\vec{g}_m^*$, $\vec{h}_m^*$, for which we have $AY=I$, and then the matrix $AX=C(\vec{q},\vec{s})$ is strongly nonsingular (cf. Lemma 2.7). On the other hand, the determinants of the $k\times k$ leading principal submatrices $(AX)_k$ of AX are polynomials of degrees at most $2k$ in the coordinates of $\vec{g}_m^*$, $\vec{h}_m^*$. Since $AX=C(\vec{q},\vec{s})$ for a particular assignment, these polynomials are not identically 0 if the components are indeterminates. Therefore, by Lemma 5.1, we obtain that

$$probability(det(AX)_k\neq 0, k=1,\cdots,n)\geq\prod_{k=1}^{n}(1-\frac{2k}{|S|})\geq 1-\frac{n(n+1)}{|S|}. \qquad \square$$

By combining Theorem 4.1 with Lemmas 5.1 and 5.2, we obtain the following result:

Corollary 5.1. *Let an $n\times n$ nonsingular Cauchy-like matrix A be given with its F-generator of a length r for the operator $F=F_{[D(\vec{q}),D(\vec{t})]}$. Then an $F_{[D(\vec{t}),D(\vec{q})]}$-generator of a length at most r for A^{-1} can be computed by means of a randomized algorithm using $2nr$ random parameters and $O(nr^2\log^3 n)$ ops and failing with a probability at most $\frac{n(n+1)}{|S|}$.*

Proof. Let us define X as above. By the virtue of Lemma 5.2, with a probability at least $1-\frac{n(n+1)}{|S|}$, the Cauchy-like matrix AX is strongly nonsingular. If it is strongly nonsingular, then by the virtue of Theorem 4.1, we may compute the matrices $(AX)^{-1}$ and $A^{-1}=X(AX)^{-1}$ by using a total of $O(nr^2\log^3 n)$ ops. Due to Lemma 5.1, we also obtain the desired bounds on the number of random parameters used and on the failure probabilities. Finally, we will decrease the length of the computed F-generator of A^{-1} by applying Fact 4.1. $\square$

Remark 5.1. *If $C(\vec{t},\vec{q})$ is a (strongly) nonsingular Cauchy matrix, then we easily deduce from Lemma 5.1 that the matrix X is strongly nonsingular as well, with a probability at least $1-\frac{n(n+1)}{|S|}$. On the other hand, if X is a strongly nonsingular matrix, then by Theorem 4.1, we may compute det (AX), det X, and then det $A=\frac{det\ (AX)}{det\ X}$ at the randomized cost $O(nr^2\log^3 n)$.*

Remark 5.2. *As we have already mentioned, we may extend computing det A to the case where A is singular. (Failure means that det $A=0$.) Furthermore, the algorithm can be easily extended to the computation of $\rho=\text{rank}A$. Indeed, with a probability at least $\frac{\rho(\rho+1)}{|S|}$, $\rho\times\rho$ is the maximum size of a nonsingular leading principal submatrix of AX for the matrix X of Lemma 5.2, that is, the maximum size of the leading principal solution of AX whose nonzero determinant is computed by the algorithm of section 4 applied to AX. This computation still has the same randomized cost of $O(n\rho^2\log^3 n)$ ops.*

Remark 5.3. *Our algorithms and complexity estimates can be applied in any algebraic field in which a nonsingular $n\times n$ Cauchy-like matrix is defined. The definition of such a matrix requires at least $2n$ distinct components of $\vec{q}$ and $\vec{t}$. To apply Lemma 5.2, we need at least $3n$ distinct components of $\vec{q}$, $\vec{s}$, and $\vec{t}$. The n extra elements can be added by means of algebraic extension of the original field. This entails minor increase of the computational cost (by a constant factor).*

6 Fast Cauchy-like Computations - Singular Case

Studying the solution of a singular Cauchy-like linear system, we will use the next result and definition.

Lemma 6.1 [K94]. *Let A be an $n \times n$ matrix of rank ρ with entries from a fixed field* $\mathbf{F}$ *and suppose that the leading $\rho \times \rho$ principal submatrix A_ρ is nonsingular. Then for a vector $\vec{y}$ with coordinates from the field* $\mathbf{F}$ *the vector*

$$\vec{x} = \begin{pmatrix} {A_\rho}^{-1}\vec{b}\,' \\ \vec{0} \end{pmatrix} - \vec{y}$$

is a solution to $A\vec{x} = \vec{b}$, where the vector $\vec{b}\,'$ consists of the first ρ coordinates of $\vec{b} + A\vec{y}$ and $\vec{0}$ denotes the null vector of dimension $n - \rho$.

Definiton 6.1. *Let A_i be the $i \times i$ leading principal submatrix of A, where $1 \leq i \leq n$. We say that A has* generic rank profile *if A_j is nonsingular for all $1 \leq j \leq rank\ A$.*

The next theorem extends the results known in the Toeplitz-like case (cf. [BP], pp.206, or [KS91]) to the Cauchy-like case and may also be applied as an alternative to Theorem 5.1 in the case where the input Cauchy-like matrix is nonsingular (see Remark 6.1 later on in this section).

Theorem 6.1. *For an $n \times n$ Cauchy-like matrix A of rank ρ represented by an $s.g._r(A)$ and satisfying (2.1) and (2.2), consider the matrix product $\overline{A} = LAM$, where L and M are also Cauchy-like matrices with scaling generators of length* 1. *Assume the following relations:*

$$F_{[D(\vec{s}),\ \ D(\vec{q})]}(L) = YZ^T,$$

$$F_{[D(\vec{t}),\ \ D(\vec{p})]}(M) = XW^T,$$

$$Y^T = [y_1, \cdots, y_n] \in \mathbf{F}^n,\ \ Z^T = [z_1, \cdots, z_n] \in \mathbf{F}^n,$$

$$X^T = [x_1, \cdots, x_n] \in \mathbf{F}^n,\ \ W^T = [w_1, \cdots, w_n] \in \mathbf{F}^n,$$

$$L = \Big(\frac{y_{i+1}z_{j+1}}{s_i - q_j}\Big)_{i,j=0}^{n-1}, \quad M = \Big(\frac{x_{i+1}w_{j+1}}{t_i - p_j}\Big)_{i,j=0}^{n-1},$$

where the entries of the matrices Y, Z, X, W are random and selected independently of each other from a fixed finite subset S of the field $\mathbf{F}$ *assuming the uniform probability distribution on S, where S does not contain* 0. *Let s_i, q_j, p_k be all pairwise different for $i, j, k = 0, \cdots, n-1$. Then*

(1) L and M are strongly nonsingular matrices and

(2) with probability no less than

$$1 - \frac{2\rho(\rho+1)}{|S|}$$

$\overline{A}$ has generic rank profile.

Proof: Part (1) follows from (2.2) and Lemma 2.7 since S does not contain 0. Let us prove part (2). For an $n \times n$ matrix D, denote by $D_{I,J}$ the determinant of the submatrix of D formed by removing from D all rows not contained in the set I and all columns not contained in the set J. First, let Y, Z, X, and W be generic matrices. For $I = [1, 2, \cdots, i]$, $J = [j_1, j_2, \cdots, j_i]$, $K = [k_1, k_2, \cdots, k_i]$, $i = 1, 2, \cdots, \rho$, we have from the Cauchy-Binet formula that

$$\overline{A}_{I,I} = \sum_J \sum_K L_{I,J} A_{J,K} M_{K,I}.$$

Let us prove that

$$\overline{A}_{I,I} \neq 0 \ for \ i = 1,\ 2,\ \cdots,\ \rho\ . \tag{6.1}$$

Observe that, for a fixed pair of $J = [j_1, j_2, \cdots, j_i]$ and $K = [k_1, k_2, \cdots, k_i]$, the determinant $L_{I,J}$ has the unique term

$$a y_1 y_2 \cdots y_i z_{j_1} \cdots z_{j_i},$$

where $a \neq 0$ is a constant. Likewise, $M_{K,I}$ has the unique term

$$b x_{k_1} \cdots x_{k_i} w_1 \cdots w_i,$$

where $b \neq 0$ is a constant. Therefore, $\overline{A}_{I,I} \neq 0$ provided that there exists a pair J, K such that $A_{J,K} \neq 0$. This is true for all $i \leq \rho$, since A has rank ρ, and we arrive at (6.1).

Now we are ready to deduce part (2) of Theorem 6.1. Indeed, $\overline{A}_{I,I}$ is a polynomial of degree at most $4i$ in the coordinates of the variables y_m, z_m, x_m, w_m. Therefore, under the random choice of the values of these variables specified in Theorem 6.1, we apply Lemma 5.1 and obtain that

$$probability(\overline{A}_{I,I} \neq 0, i = 1, \cdots, \rho) \geq \prod_{i=1}^{\rho} (1 - \frac{4i}{|S|}) \geq 1 - \frac{2\rho(\rho+1)}{|S|}.$$

This proves part (2) of Theorem 6.1. □

Remark 6.1. *If the input Cauchy-like matrix is nonsingular, we may apply Theorem 6.1 as an alternative to Theorem 5.1. The application of Theorem 6.1 rather than Theorem 5.1, requires by factor $2/r$ fewer random parameters ($4n$ versus $2nr$), involves scaling generators of roughly half length ($r + 2$ versus $2r + 1$), but doubles the probability of errors ($\frac{2n(n+1)}{|S|}$ versus $\frac{n(n+1)}{|S|}$).*

To prove Theorem 6.1, we devised an algorithm that for an $n \times n$ Cauchy-like matrix A of rank ρ given with an $s.g._r(A)$, computes a random pair $s.g._1(L)$ and $s.g._1(M)$, where L and M are $n \times n$ Cauchy-like matrices having scaling rank 1 and such that, with a probability no less than $1 - \frac{2\rho(\rho+1)}{|S|}$, the matrix $\overline{A} = LAM$ has generic rank profile. Furthermore, by using Lemma 2.3, we compute $s.g._{r+2}(\overline{A})$ at the cost of performing at most $O(r^2 n \log^2 n)$ ops.

Now, we assume that we have been already given $s.g._1(L)$, $s.g._1(M)$, and $s.g._{r+2}(\overline{A})$ for a pair of nonsingular matrices L and M and an $n \times n$ matrix $\overline{A} = LAM$ having generic rank profile and propose the following algorithm.

Algorithm 6.1. *Computing the largest nonsingular leading principal inverse.*
Input: *vectors $\vec{q} = (q_i)_{i=0}^{n-1}$, $\vec{t} = (t_j)_{j=0}^{n-1}$, $q_i \neq t_j$, $i, j = 0, 1, \cdots, n-1$, and $\vec{g_1}, \cdots, \vec{g_{r+2}}, \vec{h_1}, \cdots, \vec{h_{r+2}}$ such that the Cauchy-like matrix*

$$\overline{A} = \sum_{m=1}^{r+2} diag(\vec{g}_m) C(\vec{q}, \vec{t}) diag(\vec{h}_m)$$

has generic rank profile.

Output: *An integer* $\rho \le n$ *and vectors* $\vec{u_1}, \cdots, \vec{u_{\overline{r}}}$, $\vec{v_1}, \cdots, \vec{v_{\overline{r}}}$, $\vec{u_m}, \vec{v_m} \in C^{n\times 1}$, $m = 1, 2, \cdots, \overline{r}$, $\overline{r} \le r+2$, *such that* $\rho = rank(\overline{A})$ *and*

$$\overline{A}_\rho^{\ -1} = \sum_{m=1}^{\overline{r}} diag(\vec{u}_m) C(\vec{t}, \vec{q}) diag(\vec{v}_m).$$

1. Represent $\overline{A}$ as $\overline{A} = \begin{pmatrix} \overline{B} & \overline{C} \\ \overline{E} & \overline{J} \end{pmatrix}$, cf. (3.3), where $k = \lceil \frac{n}{2} \rceil$, and the $k \times k$ submatrix $\overline{B}$ of $\overline{A}$ is singular if and only if $k > \rho$ (since $\overline{A}$ has generic rank profile). Apply Algorithm 6.1 recursively to the input matrix $\overline{B}$ replacing $\overline{A}$. (Note that we are given an $s.g._r(\overline{B})$.) If $\rho < k$, the output of this stage is the desired output of the algorithm. Otherwise, the matrix $\overline{B}$ is nonsingular, and then we obtain $s.g._r(\overline{B}^{-1})$.

2. Apply Algorithm 3.1 to compute an $s.g._r(\overline{S})$ for the matrix $\overline{S} = \overline{J} - \overline{E}\ \overline{B}^{-1}\overline{C}$.

3. Apply the algorithm recursively to the Cauchy-like input matrix $\overline{S}$, replacing $\overline{A}$. Output $\rho = \text{rank}(\overline{A}) = k + \text{rank}(\overline{S})$.

4. By using the definitions and the results of section 2, compute an $s.g._{2r+4}(\overline{A}_\rho^{-1})$ (see further comments below).

5. Apply Fact 4.1, to compute and output $s.g._{r+2}(\overline{A}_\rho^{-1})$.

Let us specify stage 4. Consider the $\rho\times\rho$ leading principal submatrix, $\overline{A}_\rho = \begin{pmatrix} \overline{B} & \overline{G} \\ \overline{D} & \overline{R} \end{pmatrix}$,

$$\overline{G},\ \overline{D}^T \in C^{k\times(\rho-k)},\ \overline{R} \in C^{(\rho-k)\times(\rho-k)}.$$

Write $\hat{S} = \overline{R} - \overline{D}\ \overline{B}^{-1}\overline{G}$. Note that at the preceding stages we have computed $s.g._{r+2}(\overline{G})$, $s.g._{r+2}(\overline{D})$, $s.g._{r+2}(\overline{B}^{-1})$, $s.g._{r+2}(\overline{DB}^{-1})$, $s.g._{r+2}(\overline{B}^{-1}\overline{G})$, and $s.g._{r+2}(\hat{S}^{-1})$ (cf. Theorem 4.1). Represent $\overline{A}_\rho^{\ -1}$ as follows:

$$\overline{A}_\rho^{\ -1} = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & \overline{S}^{-1} \end{pmatrix},$$

where $B_{1,2} = -\overline{B}^{-1}\overline{G}\ \overline{S}^{-1}$, $B_{2,1} = -\overline{S}^{-1}\overline{D}\ \overline{B}^{-1}$, $B_{1,1} = \overline{B}^{-1} - B_{1,2}\overline{D}\ \overline{B}^{-1}$ (cf. (4.6)). Due to Lemma 2.5 and Corollary 2.1, the matrices $B_{1,1}$, $B_{1,2}$, $B_{2,1}$, and $\overline{S}^{-1}$ have scaling rank at most $r+2$, and we may apply Fact 4.1, Algorithm 3.1, and the results of section 2 in order to compute the respective short scaling generators of these matrices. Let us specify the operators defining these generators. Write $\vec{q}^{\ (1)} = (q_i)_{i=0}^{k-1}$, $\vec{q}^{\ (2)} = (q_i)_{i=k}^{\rho-1}$, $\vec{t}^{\ (1)} = (t_i)_{i=0}^{k-1}$, $\vec{t}^{\ (2)} = (t_i)_{i=0}^{\rho-1}$, $\vec{q}^{\ (0)} = \begin{pmatrix} \vec{q}^{\ (1)} \\ \vec{q}^{\ (2)} \end{pmatrix}$ and $\vec{t}^{\ (0)} = \begin{pmatrix} \vec{t}^{\ (1)} \\ \vec{t}^{\ (2)} \end{pmatrix}$. Now obtain that

$$F_{[D(\vec{t}^{(0)}),\ D(\vec{q}^{(0)})]}(\overline{A}_\rho^{\ -1}) = \begin{pmatrix} diag(\vec{t}^{(1)}) & O \\ O & diag(\vec{t}^{(2)}) \end{pmatrix} \overline{A}_\rho^{\ -1} - \overline{A}_\rho^{\ -1} \begin{pmatrix} diag(\vec{q}^{(1)}) & O \\ O & diag(\vec{q}^{(2)}) \end{pmatrix}$$

$$= \begin{pmatrix} F_{[D(\vec{t}^{\ (1)}),\ D(\vec{q}^{\ (1)})]}(B_{1,1}) & F_{[D(\vec{t}^{\ (1)}),\ D(\vec{q}^{\ (2)})]}(B_{1,2}) \\ F_{[D(\vec{t}^{\ (2)}),\ D(\vec{q}^{\ (1)})]}(B_{2,1}) & F_{[D(\vec{t}^{\ (2)}),\ D(\vec{q}^{\ (2)})]}(\hat{S}^{-1}) \end{pmatrix},$$

which gives us an $s.g._{2r+4}(\overline{A}_\rho^{-1})$. □

To solve a singular Cauchy-like linear system $A\overline{x} = \overline{b}$, first compute a vector $\vec{y}$ that satisfies $LAM\vec{y} = L\vec{b}$ and then recover the vector $\vec{x} = M\vec{y}$ that satisfies $A\vec{x} = \vec{b}$. Since L and M are nonsingular, rank A = rank (LAM). By using $O(rn \log\ n)$ ops we may verify if $A\vec{x} = \vec{b}$.

7 Extension to Solving Singular Toeplitz-like Linear Systems

If we need to solve a singular Toeplitz-like linear system, which is a major operation in signal processing and computing Padé approximation, we may reduce the problem to Cauchy-like linear system [H], [GKO] and then apply our algorithm of section 6. It is more effective, however, to extend the techniques of section 6 directly to solving singular Toeplitz-like system. We will next do this, by following and then slightly improving the known best randomized algorithm of [K94]. (We will use fewer ops and random parameters and due to the incorporation of Lemma 7.4 below, which is a result from [P92b].)

Definition 7.1 (cf. e.g. [BP], Definition 11.1). *For an $n \times n$ matrix T, define the two displacement generators,*

$$F_{-}(T) = T - \tilde{Z}^T T \tilde{Z}, \quad F_{+}(T) = T - \tilde{Z} T \tilde{Z}^T, \tag{7.1}$$

where

$$\tilde{Z} = \begin{pmatrix} 0 & & & & 0 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & & \\ 0 & & & 1 & 0 \end{pmatrix}$$

is a down shift $n \times n$ matrix. If for $F = F_{+}$ or $F = F_{-}$, we have

$$F(T) = G^* H^{*T}, \tag{7.2}$$

where $G^*, H^* \in \mathbf{F}^{n \times r}$ for a fixed field $\mathbf{F}$ (say, for $\mathbf{F} = \mathbf{C}$), then the pair of matrices (G^*, H^*) is called an F-generator or a displacement generator *of T of length r and will be denoted $d.g._r(T)$. The minimum r allowing the above representation (7.2) is called the F-rank of T. T is called a Toeplitz-like matrix if r is small relative to n.*

Next, we will recall some known properties of Toeplitz-like matrices.

Lemma 7.1 [BA80]. *For any $n \times n$ matrix A,*

$$rank F_{-}(A) - 2 \leq rank F_{+}(A) \leq rank F_{-}(A) + 2.$$

Furthermore, given a $d.g._r(T)$ under $F = F_{+}$ (resp. $F = F_{-}$), it suffices to use $O(rT_{Mv}(n))$ ops (for $T_{Mv}(n)$ of (1.5), (1.7), (1.8)) in order to compute a $d.g._{r+2}(T)$ under $F = F_{-}$ (resp. $F = F_{+}$).

Lemma 7.2 [KKM]. *Let $F_{-}, F_{+}, T, G^*, H^*$, and r be as in (7.1) and (7.2). Then $F(T) = G^* H^{*T} = \sum_{i=1}^{r} \vec{g}_i^* (\vec{h}_i^*)^T$ if and only if*

$$T = \sum_{i=1}^{r} L^T(\vec{g}_i^*) L(\vec{h}_i^*) \ for \ F = F_{-}, \ T = \sum_{i=1}^{r} L(\vec{g}_i^*) L^T(\vec{h}_i^*) \ for \ F = F_{+}, \tag{7.2}$$

where $G^* = [\vec{g}_1^*, \cdots, \vec{g}_r^*]$, $H^* = [\vec{h}_1^*, \cdots, \vec{h}_r^*]$, and $L(\vec{v})$ is a lower triangular Toeplitz matrix with the first column $\vec{v}$.

Lemma 7.3 (cf. e.g. [BP], Corollary 12.1). *Let T_1 and T_2 be two Toeplitz-like matrices, given with their F-generators of lengths r_1 and r_2, respectively, for $F = F_+$ or $F = F_-$. Then an F-generator of length at most $r_1 + r_2 + 1$ for the matrix T_1T_2 can be computed by using $O(r_1r_2)$ polynomial multiplications modulo $x^{O(n)}$ and $O(r_1 + r_2)$ summations of $O(r_1 + r_2)$ vectors of dimension n, at the overall cost of $O((r_1+r_2)^2 P_{M,\mathbf{F}}(n))$ ops, where $P_{M,\mathbf{F}}(n)$ denotes the cost of polynomial multiplication modulo $x^{O(n)}$ in $\mathbf{F}$ (cf. (1.7), (1.8)). Furthermore, a $d.g._r(UTL)$ for a given $d.g._r(T)$ and a given pair of lower triangular Toeplitz matrices L and U can be computed at the cost $2r^2P_{M,\mathbf{F}}(n)$, provided that $F = F_-$.*

Lemma 7.4 (cf. Proposition A.6 of [P92b], [P93a], or [BP], Problem 2.2.11b). *Given an $d.g._{r^*}(A) = (G, H)$ and the scaling rank r of A, $r < r^* \le n$, one can compute $d.g._r(A)$ by using $O(r^2n)$ ops.*

Lemma 7.5 [KKM]. *Let T be a nonsingular Toplitz-like matrix. Then rank $F_+(T^{-1}) =$* rank *$F_-(T)$.*

Lemma 7.6 (cf. [M80], [BA80], [BP]). *Let T be an $n \times n$ strongly nonsingular Toeplitz-like matrix such that*

$$T = \begin{pmatrix} B & C \\ E & J \end{pmatrix}, \qquad S = J - EB^{-1}C,$$

B is a $k \times k$ matrix, and S is the $(n-k) \times (n-k)$ Schur complement of *B* in *T (cf. (3.3)). Let $r = rankF_+(T)$. Then*

$$rank\ F_-(S^{-1}) = rank\ F_+(S) \le r,$$

$$rank\ F_-(B^{-1}) = rank\ F_+(B) \le r,$$

$$rank\ F_+(S^{-1}) = rank\ F_-(S) \le r + 2,$$

$$rank\ F_+(B^{-1}) = rank\ F_-(B) \le r + 2.$$

Proof. The lemma follows from Lemmas 3.2, 7.1, and 7.5. □

Theorem 7.1 (cf. [K94]). *For an $n \times n$ Toeplitz-like matrix T of rank ρ, represented by its $s.g._r(T)$ satisfying (7.1) and (7.2), consider the matrix product $\tilde{T} = UTL$, where U^T and L are two unit lower triangular Toeplitz matrices whose $2n - 2$ elements are randomly and independently of each other selected from a subset S of a fixed field containing the entries of T, under the uniform probability distribution on S. Then $\tilde{T}$ has generic rank profile with probability no less than*

$$1 - \frac{\rho(\rho + 1)}{|S|}.$$

Due to Lemma 7.3, we may compute $d.g._r(\tilde{T})$ at the cost of performing at most $2r^2P_{M,\mathbf{F}}(n)$ ops.

Now, we assume that we have been already given a $d.g._r(\tilde{T})$ for an $n \times n$ matrix $\tilde{T}$ having generic rank profile. We propose the following algorithm.

Algorithm 7.1. *Computing the largest nonsingular leading principal inverse.*

Input: *a field $\mathbf{F}$ and vectors $\vec{\tilde{g}}_1, \cdots, \vec{\tilde{g}}_r, \vec{\tilde{h}}_1, \cdots, \vec{\tilde{h}}_r$ with components from $\mathbf{F}$ such that the Toeplitz-like matrix*

$$\tilde{T} = \sum_{i=1}^{r} L^T(\vec{\tilde{g}}_i)L(\vec{\tilde{h}}_i)$$

has generic rank profile.

Output: *An integer* $\rho \leq n$ *and vectors* $\vec{\tilde{u}}_1, \cdots, \vec{\tilde{u}}_r$, $\vec{\tilde{v}}_1, \cdots, \vec{\tilde{v}}_r$, $\vec{\tilde{u}}_m, \vec{\tilde{v}}_m \in \mathbf{F}^{n\times 1}$, $m = 1, 2, \cdots, r$, *such that* $\rho = rank(\tilde{T})$ *and*

$$\tilde{T}_\rho^{-1} = \sum_{m=1}^{r} L(\vec{\tilde{u}}_m) L^T(\vec{\tilde{v}}_m).$$

1. Represent $\tilde{T}$ as $\tilde{T} = \begin{pmatrix} \tilde{B} & \tilde{C} \\ \tilde{E} & \tilde{J} \end{pmatrix}$, as in (3.3), for $k = \lceil \frac{n}{2} \rceil$, where the $k \times k$ submatrix $\tilde{B}$ of $\tilde{T}$ is singular if and only if $k > \rho$ (since $\tilde{T}$ has generic rank profile). Apply Algorithm 7.1 recursively to the input matrix $\tilde{B}$ replacing $\tilde{T}$. (Note that the first k components of the given vectors $\vec{\tilde{g}}_i$ and $\vec{\tilde{h}}_i$ define a $d.g._r(\tilde{B})$.) If $\rho \geq k$, the output of this stage is the desired output of the algorithm. Otherwise, the matrix $\tilde{B}$ is nonsingular, and then we obtain a $d.g._{r+2}(\tilde{B}^{-1})$ for $F = F_-$ and a $d.g._r(\tilde{B}^{-1})$ for $F = F_+$.

2. Apply Lemma 7.3 to compute a $d.g._r(\tilde{S})$ for the matrix $\tilde{S} = \tilde{J} - \tilde{E}\ \tilde{B}^{-1}\tilde{C}$ and for $F = F_+$.

3. Apply the algorithm recursively to the Toeplitz-like input matrix $\tilde{S}$, replacing $\tilde{T}$. Output $\rho = \text{rank}(\tilde{T}) = k + \text{rank}(\tilde{S})$.

4. By using the Definition 7.1 and Lemmas 7.1-7.6, compute $s.g._r(\tilde{T}_\rho^{-1})$ for $F = F_+$ (see some further comments below).

Let us specify stage 4. Consider the $\rho \times \rho$ leading principal submatrix, $\tilde{T}_\rho = \begin{pmatrix} \tilde{B} & \tilde{G} \\ \tilde{D} & \tilde{R} \end{pmatrix}$,

$$\tilde{G},\ \tilde{D}^T \in C^{k\times(\rho-k)},\ \tilde{R} \in C^{(\rho-k)\times(\rho-k)}.$$

Write $\breve{S} = \tilde{R} - \tilde{D}\ \tilde{B}^{-1}\tilde{G}$. Note that at the preceding stages we have computed $d.g._r(\tilde{G})$ and $d.g._r(\tilde{D})$ for $F = F_-$, $d.g._r(\tilde{B}^{-1})$, $d.g._{2r+1}(-\tilde{B}^{-1}\tilde{G})$, $d.g._{2r+1}(-\tilde{D}\tilde{B}^{-1})$, and $d.g._r(\breve{S}^{-1})$ for $F = F_+$. We obtain the following block representation:

$$\tilde{T}_\rho^{-1} = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & \breve{S}^{-1} \end{pmatrix},$$

where $M_{1,2} = -\tilde{B}^{-1}\tilde{G}\ \breve{S}^{-1}$, $M_{2,1} = -\breve{S}^{-1}\tilde{D}\ \tilde{B}^{-1}$, $M_{1,1} = \tilde{B}^{-1} - M_{1,2}\tilde{D}\ \tilde{B}^{-1}$. By applying Lemmas 7.1-7.6, we compute $d.g._r(\tilde{T}_\rho^{-1})$ for $F = F_+$. $\Box$

Appendix A. Computations with Cauchy Matrices.

Our algorithms of sections 3-5 ultimately reduce Cauchy-like computations to multiplications of Cauchy matrices by vectors (Trummer's problem). For the computation of the solution $\vec{x} = C^{-1}(\vec{s}, \vec{t})\vec{f}$ to a nonsingular Cauchy linear system $C(\vec{s}, \vec{t})\vec{x} = \vec{f}$, the reduction is much simplified due to the following formula (cf. e.g. [Gast]):

$$C^{-1}(\vec{s}, \vec{t}) = -diag(\frac{\Gamma_{\vec{s}}(t_i)}{\Gamma_{\vec{t}}'(t_i)})_{i=0}^{n-1} C^T(\vec{s}, \vec{t}) diag(\frac{\Gamma_{\vec{t}}(s_i)}{\Gamma_{\vec{s}}'(s_i)})_{i=0}^{n-1}, \tag{A.1}$$

where $\Gamma_{\vec{x}}(u)$ denote the polynomial

$$\prod_{i=0}^{n-1}(u - x_i) = u^n + \sum_{i=0}^{n-1} r_i u^i,$$

for vector $\vec{x} = [\ x_o, \cdots, x_{n-1}]^T$, and where " $'$ " denotes the derivative. On the other hand, we have the following matrix equation from [GO92]:

$$C(\ \vec{s}, \vec{t}\ ) = diag(1/\Gamma_{\vec{t}}\ (s_i))_{i=0}^{n-1} V(\ \vec{s}\ ) V^{-1}(\ \vec{t}\ ) diag(\Gamma'_{\vec{t}}\ (t_k))_{k=0}^{n-1}, \tag{A.2}$$

where $V(\vec{x})$ is a Vandermonde matrix.

The latter equation reduces the computation of the product $C(\vec{s}, \vec{t}\ )\vec{v}$, for any vector $\vec{v}$, to the computation of the product of the Vandermonde matrix $V(\ \vec{s}\ )$ by a vector and to the solution of a Vandermonde linear system of n equations. These two operations are equivalent to multipoint polynomial evaluation and to polynomial interpolation, respectively. (Note that the computation of the values of the polynomial $\Gamma_{\vec{t}}\ (s_i)$, for $i = 0, \cdots, n-1$ and for a given vector of the coefficients of this polynomial, is also the problem of multipoint polynomial evaluation.) Since the known fast algorithms (cf. [BP]) perform the latter operations, as well as the computation of the coefficients of $\Gamma_{\vec{t}}\ (x)$ for a given vector $\vec{t}$, in $O((n\ \log^2 n)\ \log\ \log\ n)$ ops over any field of constants, we arrive at Lemma 2.2. $\Box$

Furthermore, we immediately obtain from $(A.2)$ that

$$C^{-1}(\ \vec{s}, \vec{t}\ ) = diag(1/\Gamma'_{\vec{t}}\ (t_i))_{i=0}^{n-1} V(\ \vec{t}\ ) V^{-1}(\ \vec{s}\ ) diag(\Gamma_{\vec{t}}\ (s_i))_{i=0}^{n-1}.$$

Based on this formula, we may deduce the following result.

Fact A.1 [Gast]. *A nonsingular Cauchy linear system of n equations can be solved over any field of constants by using $O((n \log^2 n) \log\log\ n)$ ops.*

Alternatively, we may immediately deduce Fact A.1 from the formula $(A.1)$, which has an advantage of reducing the solution of a Cauchy linear system to the multiplication of a Cauchy matrix by a vector (Trummer's problem). (Recall that our algorithms of sections 3-6 show a similar reduction of Cauchy-like computations.)

Remark A.1 The solution of Trummer's problem and, consequently, of a Cauchy linear system was further improved in [PACLS].

References

[BA80] 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.

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

[BP70] A. Björck, V. Pereyra, Solution of Vandermonde Systems of Equations, *Math. Comp.*, **24**, 893-903, 1970.

[C841] A. L. Cauchy, M*é*morie sur les Fonctions Altern*é*es et sur les Somme Altern*é*es, *Exercises d' Analyse et de Phys. Math.*, **II**, 151-159, 1841.

[CK91] D. G. Cantor, E. Kaltofen, On Fast Multiplication of Polynomials over Arbitrary Rings, *Acta Inf.*, **28, 7**, 697-701, 1991.

[CKL89] J. F. Canny, E. Kaltofen, Y. Lakshman, Solving Systems of Non-Linear Polynomial Equations Faster, *Proc. ACM-SIGSAM Int. Symp. on Symb. and Alg. Comp. (ISSAC'89)*, 121-128, ACM Press, New York, 1989.

[D74] W.F. Donoghue, *Monotone Matrix Functions and Analytic Continuation*, Springer, Berlin, 1974.

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

[Ga75] W. Gautschi, Norm Estimates for the Inverses of Vandermonde Matrices, *Numer. Math*, **23**, 337-347, 1975.

[Ga90] W. Gautschi, How (un)stable are Vandermonde systems?, in "Intern. Symp. on Asymptotic and Computational Analysis: Conference in Honor Frank W. J. Olver's 65th Birthday (R. Wong, Ed.), *Lecture Notes in Pure and Applied Math.* **124**, 193-210, Marcel Decker, New York.

[Gast] N. Gastinel, Inversion d'une Matrice Generalisant la Matrice de Hilbert, *Chiffres* , **3**, 149-152, 1960.

[Ger] A. Gerasoulis, A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors, *Math. Comp.*, **50, 181**, 179-188, 1987.

[GKK] I. Gohberg, T. Kailath, I. Koltracht, Efficient Solution of Linear Systems of Equations with Recursive Structure, *Linear Algebra Appl.*, **80**, 81-113, 1986.

[GKKL] I. Gohberg, T. Kailath, I. Koltracht, P.Lancaster, Linear Complexity Parallel Algorithms for Linear Systems of Equations with Recursive Structure, *Linear Algebra Appl.*, **88/89**, 271-315, 1987.

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

[GL] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, Maryland, 1989/1996.

[GO92] I. Gohberg, V. Olshevsky, Fast Algorithms with Preprocessing for Multiplication of Transposed Vandermonde Matrix and Cauchy Matrix by Vectors, manuscript, *School of Math. Sciences, Tel-Aviv University*, Israel, 1992.

[GO94] I. Gohberg, V. Olshevsky, Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems, *Integral Equations and Operator Theory*, **20**, 1, 44-83, 1994.

[GO94a] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra Appl.*, **202**, 163-192, 1994.

[GO96] I. Gohberg, V. Olshevsky, Fast Inversion of Vandermonde and Vandermonde-like Matrices, 205-221, in *Communications, Computation, Control and Signal Processing: A Tribute to Thomas Kailath* (A. Paulraj, V. Roychowdhury, and C. Shaper eds.), Kluwer Academic Publisher, 1996.

[H] 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, 1994.

[H88] N. J. Higham, *Fast solution of Vandermonde-like systems involving orthogonal polynomials*, IMA J. Numer. Anal., **8**, 473-486, 1988.

[H90] N. J. Higham, *Stability analysis of algorithm for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., **11**, 23-41, 1990.

[HR] H. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators, Operator Theory*, **13**, Birkhaeuser, 1984.

[K94] E. Kaltofen, Asymptotically Fast Solution of Toeplitz-like Singular Linear Systems, *Proc. ACM-SIGSAM Intern. Symp. on Symbolic and Alg. Comp.,* 297-304, ACM Press, New York, 1994.

[K95] 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.

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

[KS91] E. Kaltofen, B. D. Saunders, On Wiedemann's Method for Solving Sparse Linear Systems, *Proc. AAECC-5, Lecture Notes in Computer Science,* **536**, 29-38, Springer, Berlin, 1991.

[M74] M. Morf, Fast Algorithms for Multivariable Systems, PhD Thesis, Standford University, Standford, CA, 1974.

[M80] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proc. IEEE Internat. Conf. on ASSP*, 954-959, IEEE Computer Society Press, 1980.

[P89/90] V.Y. Pan, On Computations with Dense Structured Matrices, *Proc. ACM-SIGSAM Intern. Symp. on Symbolic and Alg. Comp.*, 34-42, ACM Press, New York, 1989, and *Math. of Computation.*, **55, 191**, 179-190, 1990.

[P92b] V.Y. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, *Computers and Mathematices (with Applications)*, **24, 3**, 61-75, 1992.

[P93a] V.Y. Pan, Decreasing the Displacement Rank of a Matrix, *SIAM J. Matrix Anal. Appl.*, **14, 1**, 118-121, 1993.

[PACLS] V. Y. Pan, M. Aby Tabanjeh, Z. Q. Chen, E. I. Landowne, A. Sadikou. New Transformations of Cauchy Matrices and Trummer's Problem, *Preliminary Report*, 1998.

[Re90] L. Reichel, A Matrix Problem with Application to Rapid Solution of Integral Equations, *SIAM J. on Scientific and Stat. Computing*, **11**, 263-280, 1990.

[Rok85] F. Rokhlin, Rapid Solution of Integral Equations of Classical Potential Theory, *J. Comput. Physics*, **60**, 187-207, 1985.

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

[St69] V. Strassen, Gaussian Elimination is Not Optimal, *Numer. Math.*, **13**, 354-356, 1969.

[T86] M. Trummer, An Efficient Implementation of a Conformal Mapping Method Using the Szegö Kernel, *SIAM J. on Numerical Analysis*, **23**, 853-872, 1986.

[Z] R.Z. Zippel, Probabilistic Algorithm for Sparse Polynominals, *Proc. EUROSAM 79, Lecture Notes in Computer Science*, **72**, 216-226, Springer, Berlin, 1979.