

Improved Parallel Computations With Toeplitz-like and Hankel-like Matrices*[†]

Dario Bini

Dipartimento di Matematica

Università di Pisa

56100 Pisa, Italy

and

Victor Pan

Mathematics & Computer Science Department

Lehman College, CUNY

Bronx, New York 10468

Submitted by Gregory S. Ammar

ABSTRACT

The known fast algorithms for computations with general Toeplitz, Hankel, Toeplitz-like, and Hankel-like matrices are inherently sequential. We develop some new techniques in order to devise fast parallel algorithms for such computations, including the evaluation of Krylov sequences for such matrices, traces of their power sums, characteristic polynomials, and generalized inverses. This has further extensions to computing the solution or a least-squares solution to a linear system of equations with such a matrix and to several polynomial computations (such as computing the gcd, lcm, Padé approximation, and extended Euclidean scheme for two polynomials), as well as to computing the minimum span of a linear recurrence sequence. The algorithms can be applied over any field of constants, with the resulting advantages of using modular arithmetic. The algorithms consist of simple computational blocks (mostly reduced to fast Fourier transforms) and have potential practical value. We also develop the techniques for extending all our results to the case of matrices representable as the sums of Toeplitz-like and Hankel-like matrices.

* Supported by NSF grant CCR 9020690, by CUNY University Award 662478, and by 60% and 40% MURST funds.

[†] The results of this paper were presented at the Annual ACM-SIGSAM International Symposium on Symbolic and Algebraic Computations, 1993.

1. INTRODUCTION

Acceleration of computations with Toeplitz, Hankel and other dense structured matrices by means of their parallelization is highly important for both the theory and the practice of computational linear algebra and of its applications to such areas as control theory, signal processing, and PDEs.

A challenge of this subject is that the known fast and superfast algorithms for Toeplitz and Hankel computations are inherently sequential: they either recursively reduce the dimension of the problem by 1 or, for some similar reasons, require at least n parallel steps for the computations with $n \times n$ Toeplitz matrices (although there are faster parallel algorithms, specially devised for computations with well-conditioned Toeplitz matrices [22, 24] and for rapid refinement of an already close initial approximation to the solution to a Toeplitz or Toeplitz-like linear system [20]).

In this paper we will present another parallel algorithm [using the order of $(\log n)^2$ parallel arithmetic steps and the order of $n^2/\log n$ processors] for any Toeplitz or Hankel $n \times n$ input matrix, and moreover, for any $n \times n$ matrix obtained as the sum of a *Toeplitz matrix* and a *Hankel matrix* or even of a *Toeplitz-like matrix* and a *Hankel-like matrix* (these known matrix classes are defined by using associated displacement operators; see Section 3).

We cover parallel computations with such matrices T , including the computation of the Krylov sequences $\mathbf{v}, T\mathbf{v}, \dots, T^K\mathbf{v}$; of the traces of T^i , $i = 1, 2, \dots, K$, $K = O(n)$; of the characteristic polynomial of T ; and of the solution or a least-squares solution to a linear system $T\mathbf{x} = \mathbf{b}$. The results can be further extended to such computational problems as fast parallel evaluation of rank T , of the null space of T , of the minimum span of a linear recurrence sequence (Berlekamp-Massey problem), and of the polynomial gcd and lcm, Padé approximation, and extended Euclidean scheme for polynomials [5, 17, 25]. Many of our algorithms can be immediately extended to computations with other dense structured matrices, such as Hilbert-like and Vandermonde-like matrices, by applying the techniques of [21].

The algorithms work over (or can be extended to) any field of constants, which enables us to take advantage of using the techniques of residue (modular) arithmetic.

Our algorithms satisfy the stated complexity bounds under any model of parallel computing that supports the cost bounds of Table 1 (listed for some fundamental problems of parallel computations). We refer the reader to [7], [29], [13], and [19] on verification of these cost bounds under some realistic models of parallel computing. In Table 1, $\log^* n$ is the minimum integer h such that $\log(\log(\dots(\log n)\dots)) < 1$, where the function \log is h times composed with itself.

TABLE 1

	Parallel arithmetic time	Processors
Summation of n numbers	$O(\log n)$	$O(n/\log n)$
Discrete Fourier transform on n points (by means of FFT)	$O(\log n)$	$O(n)$
Multiplication of two univariate polynomials of degree n (by reducing to 3 FFTs)	$O(\log n)$	$O(n)$
Multiplication of two bivariate polynomials of degrees m and n in the two variables (by means of 2-dimen- sional FFT's)	$O(\log mn)$	$O(mn)$
Reciprocation of a polynomial modulo x^n	$O(\log n \log^* n)$	$O(n/\log^* n)$

The fundamental complexity bounds of Table 1 are immediately extended to many other computations. In particular, the computation of the inner product of two vectors of dimension n is reduced to one parallel multiplication step on n processors and to the summation of n numbers, whereas multiplication of a Toeplitz matrix by a vector can be reduced to polynomial multiplication (convolution); see Appendix B.

We use the asymptotic complexity estimates presented in the form $O(t, p)$, which amounts to asymptotic bounds $O(t)$ on arithmetic time (the number of arithmetic parallel steps) used in an algorithm and, simultaneously, $O(p)$ on the number of processors. In our case, the constants hidden in this "O" notation are quite small: in particular, at most $3 \log n$ arithmetic time steps and $2n$ processors are needed to support FFT on n points [27], and $2\lceil \log n \rceil$ steps and $\lceil n/\log n \rceil$ processors suffice in order to sum n numbers.

We deduced our estimates assuming Brent's modified principle [18, 26], according to which one processor can simulate s processors in time $O(s)$, so

that the number of processors required for the implementation of a parallel algorithm can be decreased by a factor of s , $1 \leq s \leq p$, at the cost of slowing down the algorithm by $O(s)$ times. This means that the $O(t, p)$ bound also implies the $O(st, p/s)$ bound for any $s > 1$, and in particular, for $s = p$ we arrive at the time bound $O(pt, 1)$, which measures the potential work $O(pt)$ of the parallel algorithm if it were implemented sequentially on a single processor. A slightly more intricate application of Brent's principle enables us to simplify some parallel computations, improving the straightforward bounds on the complexity of the summation of n numbers from $O(\log n, n)$ to $O(\log n, n/\log n)$ [29, 19], and similarly for our algorithm of this paper, from $O(\log^2 n, n^2)$ to $O(\log^2 n, n^2/\log n)$ (see Section 2).

The latter bounds also imply the bounds $O(s \log^2 n, n^2/(s \log n))$ for any s , $1 \leq s \leq n^2/\log n$, due to Brent's principle. Moreover, if we only need to solve the linear system and/or to compute the determinant, then the general techniques of supereffective slowdown of parallel computations [26] enable us to implement our algorithms so as to arrive at the bounds $O(n^{1-a} \log^2 n, n^{2a}/\log n)$ for any a , $0 < a \leq 1$, that is, we may make our algorithms run in $O(n^{1-a} \log^2 n)$ time using $O(n^{2a}/\log n)$ processors for any a , $0 < a \leq 1$. For instance, for $a = \frac{1}{2}$, this turns into the bounds $O(n^{1/2} \log^2 n, n/\log n)$.

If we choose to set $a = 1$, then the potential work (representing the sequential time) of our fast parallel algorithm is $O(n^2 \log n)$, which is close to the running time of the sequential Levinson-Durbin algorithm (widely used for solving Toeplitz linear systems), and which is the best sequential time bound known for computing the characteristic polynomial of a Toeplitz or a Toeplitz-like matrix T . The best alternative sequential time bound is apparently $O(n^2 \log^2 n)$. We may obtain such a bound if we apply more than n parallel steps to factorize the matrices $xI - T$ for $n + 1$ distinct values x_i , which will give us the $n + 1$ values of $c_T(x) = \det(xI - T)$, and if we then obtain the coefficients of $c_T(x)$ by means of interpolation.

In Appendix B, we recall some techniques for multiplication of a Toeplitz-like or Toeplitz-like + Hankel-like matrix by a vector and for the recovery of the solution of a nonsingular Toeplitz linear system $T\mathbf{x} = \mathbf{b}$ from two columns of the inverse matrix T^{-1} . We also include in this paper some little-known techniques for a simple transition from the traces of the powers of any general matrix T to its characteristic polynomial (Appendix A) and further to a least-squares solution to the linear system $T\mathbf{x} = \mathbf{b}$ [see Equation (2.7)].

Besides the cited material of the two appendices, we organize our presentation in the following order. In Section 2 we present our main algorithm and its extensions, in the case of a Toeplitz input matrix T . In Section 4 we extend the results of Section 2 to the case of matrices

representable as the sums of Toeplitz-like and Hankel-like matrices. Such an extension requires developing some special techniques of independent interest, which we present in Section 3. In particular, we study the properties of some displacement operators associated with matrices of the latter class, thus extending the theory of [15, 8]. In Appendix C we display some correlations between such operators and the two classical displacement operators of [15, 8]. Some further details can be found in our original technical reports [25] (on the algorithms for the computations with Toeplitz and Toeplitz-like matrices) and [6] (on the operators associated with the sums of Toeplitz-like and Hankel-like matrices, on their main properties, and on some related results).

2. IMPROVED PARALLEL COMPUTATIONS WITH TOEPLITZ MATRICES

In this section we will show a simple parallel algorithm for computation of the powers of a Toeplitz matrix T , with further extensions to simple parallel computation of the Krylov sequence $\{T^i \mathbf{v}, i = 0, 1, \dots\}$ (for any fixed vector \mathbf{v}), of the sequence $\{\text{trace } T^i, i = 0, 1, \dots\}$, and of the solution and a least-squares solution to a Toeplitz linear system.

Hereafter $\lfloor x \rfloor$ and $\lceil x \rceil$ will denote two integers nearest to a real x and such that $\lfloor x \rfloor \leq x \leq \lceil x \rceil$. $\mathbf{e}^{(h)}$ will denote the h th unit coordinate vector, that is, the h th column of the $n \times n$ identity matrix I , $h = 0, 1, \dots, n-1$; \log will denote logarithm to the base 2; $\mathbf{F}^{m \times n}$ will denote the class of $m \times n$ matrices with their entries in a fixed field \mathbf{F} ; and we will also use the following definitions: $(W)_{ij}$ denotes the (i, j) entry of a matrix W . [For a Toeplitz matrix W , we have $(W)_{ij} = (W)_{i+k, j+k}$ for all integers i, j, k for which $(W)_{ij}$ and $(W)_{i+k, j+k}$ are defined.] Z denotes the $n \times n$ downshift matrix, $(Z)_{i, i-1} = 1$, $(Z)_{ij} = 0$ for all pairs i and $j \neq i-1$. J denotes the $n \times n$ reversion matrix $(J)_{g, n-1-g} = 1$, $(J)_{gh} = 0$ for all pairs g and $h \neq n-1-g$. $L(\mathbf{v})$ denotes the lower triangular Toeplitz matrix with first column \mathbf{v} . W^T and W^H denote the transpose and the Hermitian transpose of a matrix W , respectively.

We recall Newton's iteration for inverting a matrix T ,

$$X_{i+1} = 2X_i - X_i T X_i, \quad i = 0, 1, \dots, \quad (2.1)$$

but we will apply it to the *parametrized* matrix

$$T(\lambda) = I - \lambda T$$

such that

$$T(\lambda)^{-1} = (I - \lambda T)^{-1} = I + \lambda T + (\lambda T)^2 + \cdots = \prod_{i=0}^{\infty} [I + (\lambda T)^{2^i}].$$

ALGORITHM 2.1. For two positive integers k and n and for a given $n \times n$ matrix T , set $S_0 = I$, $T(\lambda) = I - \lambda T$, and apply the parametrized Newton iteration by recursively computing

$$S_{i+1} = 2S_i - S_i T(\lambda) S_i = [2I - S_i T(\lambda)] S_i, \quad i = 0, \dots, k-1. \quad (2.2)$$

Denote $K = 2^k$, and output the entries of the matrix polynomial $S_k \bmod \lambda^K = I + \lambda T + \cdots + (\lambda T)^{K-1}$, that is, the entries of the matrix powers I, T, \dots, T^{K-1} .

The latter equation follows because $I - T(\lambda)S_0 = \lambda T = 0 \bmod \lambda$, $I - T(\lambda)S_{i+1} = [I - T(\lambda)S_i]^2$, $i = 0, 1, \dots$, and therefore,

$$I - T(\lambda)S_i \bmod \lambda^{2^i} = 0, \quad i = 0, 1, \dots, \quad (2.3)$$

$$\begin{aligned} \tilde{S}_i &= S_i \bmod \lambda^{2^i} = I + \lambda T + \lambda^2 T^2 + \cdots + \lambda^{2^i-1} T^{2^i-1} \\ &= [T(\lambda)]^{-1} \bmod \lambda^{2^i}. \end{aligned} \quad (2.4)$$

Let us estimate the computational cost of this algorithm, assuming that T is a Toeplitz matrix and (consequently) $T(\lambda)$ is a Toeplitz matrix polynomial (that is, a Toeplitz matrix filled with polynomials). In this case we may express the matrix polynomial $\tilde{S}_i = S_i \bmod \lambda^{2^i} = T(\lambda)^{-1} \bmod \lambda^{2^i}$ via its first and last columns, by applying the Gohberg-Semencul formula for the inverse of a Toeplitz matrix [12, 9, 31]:

$$\tilde{S}_i = \frac{1}{u_0^{(i)}} [L(\mathbf{u}^{(i)}) L^T(J \mathbf{v}^{(i)}) - L(Z \mathbf{v}^{(i)}) L^T(ZJ \mathbf{u}^{(i)})] \bmod \lambda^{2^i}, \quad (2.5)$$

where $u_0^{(i)} = (\tilde{S}_i)_{00} = 1 \bmod \lambda$, and $\mathbf{u}^{(i)}$ and $\mathbf{v}^{(i)}$ are the first and last columns of \tilde{S}_i , respectively, $\mathbf{u}^{(i)} = \tilde{S}_i \mathbf{e}^{(0)}$, $\mathbf{v}^{(i)} = \tilde{S}_i \mathbf{e}^{(n-1)}$.

In fact, the Gohberg-Semencul formula can be applied to matrices having entries over any ring provided that $u_0^{(i)}$ has its own reciprocal in the ring. In

our case we consider the ring of polynomials in λ modulo λ^{2^i} , where $u_0^{(i)}$ has a reciprocal, since $u_0^{(i)} = 1 \bmod \lambda$.

Therefore, for every i , the iteration (2.2), performed modulo λ^{2^i} , can be reduced to the computation of a pair of vector polynomials:

$$\begin{aligned}\mathbf{u}^{(i+1)} &= 2\mathbf{u}^{(i)} - \tilde{S}_i T(\lambda) \mathbf{u}^{(i)}, \\ \mathbf{v}^{(i+1)} &= 2\mathbf{v}^{(i)} - \tilde{S}_i T(\lambda) \mathbf{v}^{(i)},\end{aligned}$$

and Algorithm 2.1 outputs the pair of vector polynomials $\tilde{S}_k \mathbf{e}^{(0)}$ and $\tilde{S}_k \mathbf{e}^{(n-1)}$, that is, the first and the last columns of the matrix polynomial $\tilde{S}_k = S_k \bmod \lambda^K$. Due to (2.5) and (2.3), this computation can be performed by means of multiplications of $n \times n$ Toeplitz matrix polynomials by vectors. In order to show this, we apply the iteration (2.2) by replacing the matrix S_i with the matrix

$$\hat{S}_i = \left(\frac{1}{u_0^{(i)}} \bmod \lambda^{2^i} \right) \left[L(\mathbf{u}^{(i)}) L^T(J \mathbf{v}^{(i)}) - L(Z \mathbf{v}^{(i)}) L^T(ZJ \mathbf{u}^{(i)}) \right]$$

and, for each i , reducing modulo $\lambda^{2^{i+1}}$ the matrix obtained this way. Therefore, the vectors $\mathbf{u}^{(i)}$ and $\mathbf{v}^{(i)}$ satisfy the relation

$$\begin{aligned}\mathbf{u}^{(i+1)} &= 2\hat{\mathbf{u}}^{(i)} - \hat{S}_i T(\lambda) \hat{\mathbf{u}}^{(i)} \bmod \lambda^{2^{i+1}}, \\ \mathbf{v}^{(i+1)} &= 2\hat{\mathbf{v}}^{(i)} - \hat{S}_i T(\lambda) \hat{\mathbf{v}}^{(i)} \bmod \lambda^{2^{i+1}},\end{aligned}$$

where $\hat{\mathbf{u}}^{(i)}$ and $\hat{\mathbf{v}}^{(i)}$ are the first and last columns of \hat{S}_i , respectively. Observe that replacing the matrix S_i with the matrix \hat{S}_i introduces an error of degree generally greater than $2^i - 1$, which, due to (2.3), does not affect the entries of S_{i+1} of degrees less than 2^{i+1} . On the other hand, working with the matrix \hat{S}_i allows us to reduce the computation of $\mathbf{u}^{(i+1)}$ and $\mathbf{v}^{(i+1)}$ to the computation of the reciprocal of a polynomial modulo λ^{2^i} to 10 multiplications of $n \times n$ Toeplitz matrix polynomials by vectors or vector polynomials, and to $2n$ polynomial multiplications, where the degree of the result does not exceed $2^{i+2} - 3$. Each such a matrix-by-vector multiplication can be reduced to the multiplication of two bivariate polynomials of degrees $n - 1$ or $2n - 2$ and at most 2^{i+1} in their two variables, and performed at the parallel cost $O(i + \log n, n2^i)$ by using 2-dimensional FFTs.

The overall parallel cost of steps $i = 0, \dots, k - 1$ of the iteration (2.2) is therefore bounded by $O(\log(Kn) \log K, nK)$, $K = 2^k$. Furthermore, $O(n \lceil K/\log K \rceil)$ processors suffice at steps $i = 0, 1, \dots, k - 1 - \lceil \log k \rceil$,

which suggests the bound

$$O(\log(Kn) \log K, nK/\log K) \quad (2.6)$$

if we exclude the last $\lfloor \log k \rfloor = \lfloor \log \log K \rfloor$ steps. Instead of excluding them, we may slow them down, by applying Brent's principle. Then it suffices to use $O(nK/\log K)$ processors, performing each of these steps in $O(\log(Kn) \log K/\log \log K)$ time and all of them in $O(\log(Kn) \log K)$ time. This enables us to bound the overall computational cost of the k iteration steps (2.2) by (2.6). Note that (2.6) turns into $O(\log^2 n, n^2/\log n)$ for $K = O(n)$.

Due to (2.4), (2.5), from the output vector polynomials $\tilde{S}_k \mathbf{e}^{(0)}$ and $\tilde{S}_k \mathbf{e}^{(n-1)}$ we may immediately recover the vector polynomial

$$\tilde{S}_k \mathbf{v} = S_k \mathbf{v} \bmod \lambda^K = \sum_{i=0}^{K-1} (\lambda T)^i \mathbf{v},$$

defining the Krylov sequence

$$\mathbf{v}, T\mathbf{v}, \dots, T^{K-1}\mathbf{v},$$

for any fixed vector \mathbf{v} . At this point, we may apply the techniques of Krylov subspace iteration as a means of fast parallel approximate solution of a Toeplitz linear system.

On the other hand, from the first column $\mathbf{u}^{(k)} = [u_0^{(k)}, \dots, u_{n-1}^{(k)}]^T$ and the last column $\mathbf{v}^{(k)} = [v_0^{(k)}, \dots, v_{n-1}^{(k)}]^T$ of the matrix polynomial $\tilde{S}_k = S_k \bmod \lambda^K$, we may immediately recover [within the cost bounded by (2.6)]

$$\begin{aligned} \text{trace } \tilde{S}_k &= \text{trace } (I - \lambda T)^{-1} \bmod \lambda^K \\ &= v_0^{-1} \sum_{j=0}^{n-1} \left(\sum_{i=0}^j u_i^{(k)} v_i^{(k)} + \sum_{i=0}^{j-1} u_{n-1-i}^{(k)} v_{n-1-i}^{(k)} \right). \end{aligned}$$

This gives us $\text{trace } T^i$, $i = 0, 1, \dots, K-1$, and the result has various further applications.

In particular, having computed $\text{trace } T^i$ for $i = 0, 1, \dots, n$, we may then obtain the coefficients of the characteristic polynomial of T , $c_T(x) = \det(xI - T) = \sum_{i=0}^n c_i x^i$, either from the system of Newton's identities or by

applying a special algorithm (of Appendix A), whose parallel cost is bounded by

$$O(\log^2 n, n/\log n).$$

As one further application, we may compute a least-squares solution $T^+ \mathbf{b}$ to any Hermitian Toeplitz linear system

$$T \mathbf{x} = \mathbf{b},$$

where T^+ denotes the Moore-Penrose generalized inverse of T and where $c_0 = c_1 = \dots = c_{n-r-1} = 0$, $c_{n-r} \neq 0$ (see the end of Section 4 on a further extension to a more general class of linear systems). Then $r = \text{rank } T$, and we may apply the following simple expression [23]:

$$\begin{aligned} T^+ = & \frac{1}{c_{n-r}} \sum_{i=n-r+1}^{n-1} \left(\frac{c_{n+1-r}}{c_{n-r}} c_i - c_{i+1} \right) T^{i-n+r} \\ & + \left(\frac{c_{n+1-r}}{c_{n-r}} \right) T^r. \end{aligned} \quad (2.7)$$

Postmultiplication by \mathbf{b} expresses $T^+ \mathbf{b}$ through the coefficients c_{n-r}, \dots, c_{n-1} and the vectors $T^i \mathbf{b}$, $i = 1, \dots, r$.

REMARK 2.1. Appendix B shows further small improvements in the computation of $\tilde{S}_i \mathbf{e}^{(0)}$ and $\tilde{S}_i \mathbf{e}^{(n-1)}$.

REMARK 2.2. The results of this section can be applied over any field of constants supporting the FFT, except that the least-squares solution is only considered over the fields of characteristic 0 (of course) and that the transition from trace T^i , $i = 0, 1, \dots, n$, to c_0, \dots, c_{n-1} shown in Appendix A requires divisions by $2, 3, \dots, n$ and thus cannot be performed in the fields of characteristic p for $1 < p \leq n$. Alternate techniques of [16] use randomization to ensure such a transition over any field at the computation cost

$$O\left(\log^2 n d(n, p), \frac{n^2 \log \log n}{d(n, p) \log n}\right),$$

where p is the characteristic of the field of constants, $d(n, p) = \lceil \log n / \log p \rceil$ if $p > 0$, and $d(n, p) = 1$ if $p = 0$.

3. OPERATORS OF TOEPLITZ AND HANKEL TYPE

In this section we will introduce some machinery that we will use in the next section in order to extend the algorithms of Section 2 to a more general class of matrices. In particular, we will follow the line of [6] to define this class of matrices in terms of the associated displacement operators. Our study of these classes of matrices and operators extends the theory developed in [15, 8].

Let V be an $n \times n$ matrix, and consider the following operator defined on the linear space $\mathbf{F}^{n \times n}$ of $n \times n$ matrices over the field \mathbf{F} :

$$F_V(A) = AV - VA. \quad (3.1)$$

Observe that the operator (3.1) is linear and singular; indeed, $F_V(V) = 0$. Moreover, its null space $\mathcal{N}(F_V) = \{A \in \mathbf{F}^{n \times n} : F_V(A) = 0\}$ is made up of all the matrices that commute with V . In particular, if V has all its eigenvalues of geometric multiplicity 1, the null space coincides with the matrix algebra generated by V , i.e. with the linear space spanned by $I, V, V^2, \dots, V^{n-1}$.

Recall that, for any linear operator F (in particular, for $F = F_V$) we may uniquely represent any matrix $A \in \mathbf{F}^{n \times n}$ as the sum of a matrix N belonging to the null space $\mathcal{N}(F)$ of F and a matrix R belonging to the range of F , i.e. $\mathcal{R}(F) = \{F(A) : A \in \mathbf{F}^{n \times n}\}$.

In this section we determine some choices for the matrix V , which define operators F_V particularly effective in the study of Toeplitz-like and Hankel-like matrices and the sums of Toeplitz-like and Hankel-like matrices. (Such a natural extension of the classes of Toeplitz, Hankel, and Toeplitz + Hankel matrices will be formally defined later on.) As usual in the extension of the class of Toeplitz matrices, such choices of the matrix V will be dictated by the two main conditions:

- (1) the null space $\mathcal{N}(F_V)$ must be made up of “computationally easy” matrices;
- (2) the range $\mathcal{R}(F_V)$ must be made up of matrices of small rank.

This will allow us to represent Toeplitz-like, Hankel-like, and Toeplitz-like + Hankel-like matrices with small memory space and to deal with them at a low computational cost.

Such an approach is quite general and can be applied to various classes of matrices by devising suitable operators.

Now we will describe some simple general properties of the operator F_V of (3.1).

For any $A, B, C \in \mathbf{F}^{n \times n}$ such that $CC^T = I$, we have

$$F_V(AB) = AF_V(B) + F_V(A)B, \quad (3.2)$$

$$(F_V(A))^T = -F_{V^T}(A^T), \quad F_{CVCT}(A) = CF_V(C^TAC)C^T. \quad (3.3)$$

Moreover, if A is nonsingular, then

$$F_V(A^{-1}) = -A^{-1}F_V(A)A^{-1}. \quad (3.4)$$

Let us now represent the linear operator F_V in matrix form by means of tensor product. For this purpose, represent the matrix A as the vector \mathbf{a} obtained by arranging the entries of A columnwise, i.e. $\mathbf{a} = (a_{00}, a_{10}, \dots, a_{n-1,0}, a_{01}, a_{11}, \dots, a_{n-1,1}, \dots, a_{n-1,n-1})^T$, where $a_{ij} = (A)_{ij}$. This way Equation (3.1) can be rewritten in the following form:

$$\mathbf{f} = (V^T \otimes I - I \otimes V)\mathbf{a}, \quad (3.5)$$

where \mathbf{f} is the vector representing the matrix $F_V(A)$, and \otimes denotes the tensor product defined as $A \otimes B = (a_{ij}B)$ for any pair of matrices A, B .

Next define $F^+(A) = F_Z(A) = AZ - ZA$, $F^-(A) = F_{Z^T}(A) = AZ^T - Z^TA$. We have the following result:

PROPOSITION 3.1.

(a) *The null space of F^+ is the algebra of lower triangular Toeplitz matrices. The null space of F^- is the algebra of upper triangular Toeplitz matrices.*

(b) *For any Toeplitz matrix A , the matrices $F^+(A)$ and $F^-(A)$ have rank at most 2; moreover,*

$$\begin{aligned} F^+(A) &= \mathbf{e}^{(0)}\mathbf{e}^{(0)T}AZ - Z\mathbf{A}\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T} \\ &= \mathbf{e}^{(0)}(JZ\mathbf{A}\mathbf{e}^{(n-1)})^T - Z\mathbf{A}\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}, \\ F^-(A) &= \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}AZ^T - Z^T\mathbf{A}\mathbf{e}^{(0)}\mathbf{e}^{(0)T} \\ &= \mathbf{e}^{(n-1)}(JZ^T\mathbf{A}\mathbf{e}^{(0)})^T - Z^T\mathbf{A}\mathbf{e}^{(0)}\mathbf{e}^{(0)T}. \end{aligned}$$

(c) Let $G = [\mathbf{g}_1, \dots, \mathbf{g}_d]$, $H = [\mathbf{h}_1, \dots, \mathbf{h}_d] \in \mathbf{F}^{n \times d}$. For any matrix $A \in \mathbf{F}^{n \times n}$, we have $F^+(A) = GH^T = \sum_{i=1}^d \mathbf{g}_i \mathbf{h}_i^T$ if and only if simultaneously $\sum_{i=1}^d \sum_{j=0}^{n-1} g_j^{(i)} (Z^T)^j \mathbf{h}_i = \sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} Z^{n-j-1} \mathbf{g}_i = \mathbf{0}$, where $\mathbf{h}_i = [h_j^{(i)}]$ and either of the two following matrix equations holds:

$$A = L(A\mathbf{e}^{(0)}) + \sum_{i=1}^d L(\mathbf{g}_i) L^T(Z\mathbf{h}_i),$$

$$A = L(JA^T \mathbf{e}^{(n-1)}) - \sum_{i=1}^d L^T(ZJ\mathbf{g}_i) L(J\mathbf{h}_i).$$

(d) Similarly, for the operator F^- we have $F^-(A) = GH^T = \sum_{i=1}^d \mathbf{g}_i \mathbf{h}_i^T$ if and only if simultaneously $\sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} (Z^T)^j \mathbf{g}_i = \sum_{i=1}^d \sum_{j=0}^{n-1} g_j^{(i)} Z^{n-j-1} \mathbf{h}_i = \mathbf{0}$ and either of the two following matrix equations holds:

$$A = L^T(A^T \mathbf{e}^{(0)}) - \sum_{i=1}^d L(Z\mathbf{g}_i) L^T(\mathbf{h}_i),$$

$$A = L^T(JA\mathbf{e}^{(n-1)}) + \sum_{i=1}^d L^T(J\mathbf{g}_i) L(ZJ\mathbf{h}_i).$$

Proof. The proof of parts (a) and (b) is immediate. To prove part (c), we follow [6] and apply the matrix representation (3.5) of the operator, that is, we rewrite $F^+(A) = \sum_{i=1}^d \mathbf{g}_i \mathbf{h}_i^T$ in matrix form, thus obtaining the block bidiagonal system of linear equations:

$$\begin{pmatrix} -Z & I & & & O \\ & -Z & I & & \\ & & -Z & \ddots & \\ & & & \ddots & I \\ O & & & & -Z \end{pmatrix} \mathbf{a} = \sum_{i=1}^d \mathbf{h}_i \otimes \mathbf{g}_i, \quad (3.6)$$

or equivalently,

$$-Z\mathbf{a}_{k-1} + \mathbf{a}_k = \sum_{i=1}^d h_{k-1}^{(i)} \mathbf{g}_i, \quad k = 1, \dots, n-1, \quad (3.6a)$$

$$-Z\mathbf{a}_{n-1} = \sum_{i=1}^d h_{n-1}^{(i)} \mathbf{g}_i. \quad (3.6b)$$

Now, we fix the first column \mathbf{a}_0 of A , express the remaining columns \mathbf{a}_i , $i = 1, 2, \dots, n-1$, by using the substitution defined by the equations (3.6a), and arrive at the following equations:

$$\mathbf{a}_k = Z^k \mathbf{a}_0 + \sum_{i=1}^d \left(\sum_{j=0}^{k-1} h_j^{(i)} Z^{k-j-1} \mathbf{g}_i \right), \quad k = 1, \dots, n-1.$$

It follows that A is the sum of two matrices: the first one has columns $\mathbf{a}_0, Z\mathbf{a}_0, Z^2\mathbf{a}_0, \dots, Z^{n-1}\mathbf{a}_0$ and therefore coincides with $L(A\mathbf{e}_0)$; the second one is readily verified to be $\sum_{i=1}^d L(\mathbf{g}_i)L^T(Z\mathbf{h}_i)$. Thus we arrive at the first matrix equation of part (c), which is equivalent to the system (3.6a).

Now substitute the above expression of \mathbf{a}_{n-1} in the equation (3.6b), and since $Z^n = 0$, obtain the second vector equation of part (c):

$$\sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} Z^{n-j-1} \mathbf{g}_i = \mathbf{0}.$$

We also immediately verify the first vector equation of part (c), in a slightly distinct but equivalent form:

$$\sum_{i=1}^d \sum_{j=0}^{n-1} g_j^{(i)} (Z^T)^j \mathbf{h}_i = J \sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} Z^{n-j-1} \mathbf{g}_i.$$

This way we prove the “only if” statement of part (c), but we may reverse the argument and thus prove the “if” statement too.

To yield the extensions to the other equations of parts (c) and (d), combine the results we have just proved with the matrix equations $[F^-(A)]^T = -F^+(A^T)$ and $F^-(A) = JF^+(JAJ)J$ [the latter equation follows from (3.3), with $C = J$, since $Z^T = JZJ$, $J = J^T$]. ■

Proposition 3.1 yields formulae for an efficient representation of matrices A such that $d = \text{rank } F(A)$ is small. In particular, the entries of such matrices are uniquely determined by the first column $A\mathbf{e}^{(0)}$ and by the pair of $n \times d$ matrices (G, H) , which we call an *F-generator of length d* for A . In the specific case of Toeplitz matrices and of the operator F^+ , we have $d = 2$, $\mathbf{g}_1 = \mathbf{e}^{(0)}$, $\mathbf{g}_2 = ZA\mathbf{e}^{(n-1)}$, $\mathbf{h}_1 = J\mathbf{g}_2$, $\mathbf{h}_2 = -\mathbf{e}^{(n-1)}$, and Proposition 3.1 implies that $F^+(A) = 0$ [or $F^-(A) = 0$] if A is a lower [respectively, an upper] triangular Toeplitz matrix. Furthermore, it is easy to specify an

F^+ -generator for the inverse of a nonsingular Toeplitz matrix based on the Gohberg-Semencul formula, and similarly if the operator F^- replaces F^+ . On the other hand, the first matrix equation of part (c) of Proposition 3.1 and the matrix equation (3.4) together yield the following inversion formula for a nonsingular Toeplitz matrix A :

$$\begin{aligned} A^{-1} &= L(\mathbf{a})L^T(\mathbf{e}^{(0)} - J\mathbf{b}) - L(\mathbf{b})L^T(J\mathbf{a}), \\ \mathbf{a} &= A^{-1}\mathbf{e}^{(0)}, \quad \mathbf{b} = A^{-1}Z\mathbf{Ae}^{(n-1)}. \end{aligned} \quad (3.7)$$

Similar inversion formulae for Toeplitz matrices can be obtained from the other matrix equations of Proposition 3.1. Note that, unlike (2.5), these formulae do not involve extra divisions [except, of course, for the divisions by $\det A$, implicit in the computation of the vectors \mathbf{a} and \mathbf{b} in (3.7) and indispensable in *any* matrix inversion formula].

A matrix A having an F -generator of length d bounded from above by a fixed (small) constant [or formally, using the asymptotic “ O ” notation, of length $d = O(1)$ as $n \rightarrow \infty$], with respect to one of the operators F^+ or F^- , is called a *Toeplitz-like matrix*.

Proposition 3.1 together with (3.2) and (3.4) enables us to represent the product of Toeplitz-like matrices and the inverse of a nonsingular Toeplitz-like matrix in terms of the sum of products of lower triangular and upper triangular Toeplitz matrices. In particular, we obtain that, if A has an F -generator of length d , then A^{-1} has an F -generator of length d . If A and B have an F -generator of length d_A and d_B , respectively, then AB has an F -generator of length (at most) $d_A + d_B$.

The class of triangular Toeplitz matrices plays an important role in the representation formulae of Proposition 3.1. We recall in particular that the product of a triangular Toeplitz matrix and a vector can be computed by means of two FFTs and one inverse FFT at the sequential cost of $O(n \log n)$ arithmetic operations and at the parallel cost of $O(\log n, n)$.

In order to deal with the sums of Toeplitz and Hankel matrices, we will choose a different matrix V of (3.1). Consider the matrix $M = Z + Z^T$, and define

$$F^\pm(A) = AM - MA. \quad (3.8)$$

We have the following result:

PROPOSITION 3.2. *For the operator F^\pm of (3.8), the following properties hold:*

(a) *The null space of F^\pm is the algebra τ generated by $I, M, M^2, \dots, M^{n-1}$; the entries of any matrix $U = (u_{ij})$, $i, j = 0, 1, \dots, n$*

– 1, of this algebra satisfy the equations

$$\begin{aligned} u_{i,j-1} + u_{i,j+1} &= u_{i-1,j} + u_{i+1,j}, & i, j &= 0, 1, \dots, n-1, \\ u_{i,j} &= 0 & \text{if } i \in \{-1, n\} \text{ or } j \in \{-1, n\}. \end{aligned}$$

(b) Let $T_j(x)$ denote the Chebyshev-like polynomial of degree j defined by the equations

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, \\ T_{j+1}(x) &= xT_j(x) - T_{j-1}(x), & j &= 1, 2, \dots \end{aligned}$$

Let $\mathbf{u}^{(i)}$ be the i th column of $U \in \tau$. Then we have the vector equations

$$\mathbf{u}^{(i)} = T_i(M)\mathbf{u}^{(0)}, \quad i = 0, 1, \dots, n-1$$

(which motivate us to use the notation $U = \tau(U\mathbf{e}^{(0)})$ for any matrix $U \in \tau$); furthermore, the matrix U is uniquely defined by each of the four vectors: by its first row, by its last row, by its first column, and by its last column.

(c) For any Toeplitz or Hankel matrix A , the matrix $F^\pm(A)$ has rank at most 4; moreover, $F^\pm(A) = \mathbf{e}^{(0)}\mathbf{e}^{(0)T}AM - MA\mathbf{e}^{(0)}\mathbf{e}^{(0)T} + \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}AM - MA\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}$.

(d) Let $G = [\mathbf{g}_1, \dots, \mathbf{g}_l]$, $H = [\mathbf{h}_1, \dots, \mathbf{h}_d] \in \mathbb{F}^{n \times d}$. For any matrix $A \in \mathbb{F}^{n \times n}$, we have $F^\pm(A) = GH^T = \sum_{i=1}^d \mathbf{g}_i \mathbf{h}_i^T$ if and only if simultaneously $\sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} T_{n-j-1}(M) \mathbf{g}_i = \mathbf{0}$, and any of the four following matrix equations holds:

$$\begin{aligned} A &= \tau(A\mathbf{e}^{(0)}) + \sum_{i=1}^d \tau(\mathbf{g}_i) L^T(Z\mathbf{h}_i), \\ A &= \tau(JA\mathbf{e}^{(n-1)}) + \sum_{i=1}^d \tau(J\mathbf{g}_i) L^T(ZJ\mathbf{h}_i), \\ A &= \tau(A^T\mathbf{e}^{(0)}) - \sum_{i=1}^d L(Z\mathbf{g}_i) \tau(\mathbf{h}_i), \\ A &= \tau(JA^T\mathbf{e}^{(n-1)}) - \sum_{i=1}^d L(ZJ\mathbf{g}_i) \tau(J\mathbf{h}_i). \end{aligned}$$

Here $U = \tau(\mathbf{u})$ is the $n \times n$ matrix that belongs to the matrix algebra τ and is such that $U\mathbf{e}^{(0)} = \mathbf{u}$ (compare part (b)).

(e) We have

$$F^\pm(JAJ) = JF^\pm(A)J,$$

$$F^\pm(A^T) = -[F^\pm(A)]^T.$$

Proof. The first statement of part (a) holds because the matrix M has n distinct eigenvalues. Rewriting the equation $UM - MU = 0$ componentwise leads to the second statement. Rewriting this equation columnwise yields $\mathbf{u}^{(i+1)} = M\mathbf{u}^{(i)} - \mathbf{u}^{(i-1)}$, $i = 0, 1, \dots, n-2$, where $\mathbf{u}^{(-1)} = \mathbf{0}$, which leads to the first statement of part (b). Its second statement follows similarly. The proofs of parts (c) and (e) are immediate. The proof of part (d) is analogous to the proof of parts (c) and (d) of Proposition 3.1. We observe that, for the operator F^\pm , Equation (3.5) takes the following form:

$$\begin{pmatrix} -M & I & & & O \\ I & -M & I & & \\ & I & \ddots & \ddots & \\ & & \ddots & -M & I \\ O & & & I & -M \end{pmatrix} \mathbf{a} = \sum_{i=1}^d \mathbf{h}_i \otimes \mathbf{g}_i,$$

and recall that the matrix M has n distinct eigenvalues. Moreover, by setting $\mathbf{a}_{-1} = \mathbf{a}_n = \mathbf{0}$, fixing the first column \mathbf{a}_0 of A , and expressing the other columns by means of substitution, we obtain the following recurrence:

$$\mathbf{a}_{k+1} = M\mathbf{a}_k - \mathbf{a}_{k-1} + \sum_{i=1}^d h_k^{(i)} \mathbf{g}_i, \quad k = 0, 1, \dots, n-1,$$

which can be rewritten in terms of the Chebyshev-like polynomials as follows:

$$\mathbf{a}_{k+1} = T_{k+1}(M)\mathbf{a}_0 + \sum_{i=1}^d \sum_{j=0}^k h_j^{(i)} T_{k-j}(M)\mathbf{g}_i, \quad k = 0, 1, \dots, n-1. \quad (3.9)$$

It follows that A is the sum of two matrices: the first one has columns $\mathbf{a}_0, T_1(M)\mathbf{a}_0, T_2(M)\mathbf{a}_0, \dots, T_{n-1}(M)\mathbf{a}_0$ and therefore coincides with $\tau(A\mathbf{e}_0)$;

the second one is readily verified to be $\sum_{i=1}^d \tau(\mathbf{g}_i) L^T(\mathbf{Z}\mathbf{h}_i)$. Thus we arrive at the first matrix equation of part (d). We combine this equation with part (e) to arrive at the other three matrix equations of part (d). Applying (3.9) for $k = n - 1$, $A = M^s$, $s = 0, 1, \dots$, implies that $T_n(M)M^s \mathbf{e}^{(0)} = \mathbf{0}$. Together with the linear independence of the vectors $M^s \mathbf{e}^{(0)}$ for $s = 0, 1, \dots, n - 1$, this implies $T_n(M) = 0$. Now, since $T_n(M) = 0$, setting $k = n - 1$ in (3.9) yields the vector equation in part (d). ■

Proposition 3.2, together with (3.2)–(3.4), allows us to represent the product of two sums of Toeplitz and Hankel matrices and the inverse of a nonsingular sum of Toeplitz and Hankel matrices in terms of the sums of pairwise products of matrices where each first multiplicand is from the algebra τ and each second is an upper triangular Toeplitz matrix. In particular, the inverse of a Hankel + Toeplitz matrix A is defined by the first and the last rows and columns of the matrices A^{-1} , $A^{-1}MA$. A matrix A having an F -generator of length d bounded by a fixed (small) constant [or formally, using the “ O ” notation, of length $d = O(1)$ as $n \rightarrow \infty$], with respect to the operator $F = F^\pm$ of (3.8), is called *representable as a sum of Toeplitz-like and Hankel-like matrices* or simply a *Toeplitz-like + Hankel-like matrix*.

The matrices of the class τ satisfy interesting computational properties that play an important role in the representation formulae of Proposition 3.2. Such matrices are real symmetric matrices, each uniquely determined by its first column. Moreover, for any matrix $A = (a_{ij}) \in \tau$, the following relations hold [4]:

$$\begin{aligned} A &= S^T D S, \\ D &= \text{diag}(\sigma_0, \dots, \sigma_{n-1}), \\ \sigma_{i-1} &= \frac{\sum_{j=1}^n a_{j-1,0} \sin[ij\pi/(n+1)]}{\sin[i\pi/(n+1)]}, \quad i = 1, \dots, n, \\ S &= \frac{1}{\sqrt{n+1}} \sin \frac{ij\pi}{n+1}, \quad i, j = 1, \dots, n. \end{aligned}$$

Here the matrix S , associated with the *sine transform* $\mathbf{x} \rightarrow S\mathbf{x}$, is symmetric and orthogonal, that is, such that $S = S^T$, $S^T S = I$. Performing the sine transform of a real vector involves about the same number of operations as is required by its FFT [28].

The above formulae enable us to compute the product of a matrix $A \in \tau$ and a vector by means of three sine transforms at the sequential cost of $O(n \log n)$ arithmetic operations and at the parallel cost $O(\log n, n)$.

If A is the sum of a Toeplitz and a Hankel matrix and is nonsingular, then from (3.4) and Proposition 3.2 we obtain that

$$\begin{aligned} A^{-1} &= \tau(\mathbf{a}_0)L^T(\mathbf{e}^{(0)} - \mathbf{d}_0) + \tau(\mathbf{c}_0)L^T(\mathbf{b}_0) - \tau(\mathbf{a}_{n-1})L^T(\mathbf{d}_{n-1}) \\ &\quad + \tau(\mathbf{c}_{n-1})L^T(\mathbf{b}_{n-1}), \end{aligned}$$

$$\mathbf{a}_i = A^{-1}\mathbf{e}^{(i)}, \quad \mathbf{b}_i = A^{-T}\mathbf{e}^{(i)}, \quad \mathbf{c}_i = A^{-1}MA\mathbf{e}^{(i)}, \quad \mathbf{d}_i = (A^{-1}MA)^T\mathbf{e}^{(i)},$$

$$i = 0, n-1.$$

REMARK 3.1. Our approach can be effectively applied also with other choices of the matrix V of (3.1). In particular, by choosing $V = C_0$, the unit circulant matrix (with first row $\mathbf{e}^{(n-1)}$), the reader may deduce useful formulae for the inverse of a Toeplitz matrix, similar to ones of [1]. For instance, if $F_{C_0}(A) = GH^T = \sum_{i=1}^d \mathbf{g}_i \mathbf{h}_i^T$, then it is easy to prove that (compare [6])

$$A = C(\mathbf{a}_0) + \sum_{i=1}^d C(\mathbf{g}_i)L^T(\mathbf{Z}\mathbf{h}_i),$$

where $C(\mathbf{u})$ denotes the circulant matrix having the first column \mathbf{u} .

4. EXTENSION OF ALGORITHMS TO TOEPLITZ-LIKE + HANKEL-LIKE MATRICES.

In this section we apply the operators F^+ , F^- , and F^\pm as the operator F and the matrices of the class τ , to extend the results of Section 2 to the sums of Toeplitz-like and Hankel-like matrices. We first present an algorithm that, for an $n \times n$ Toeplitz-like + Hankel-like input matrix, computes the traces of the first $O(n)$ matrix powers, uses $O(n^2 \log n)$ operations, and has parallel cost $O_A(\log^2 n, n^2/\log n)$. Then we show some modifications, which allow us to compute the coefficients of the characteristic polynomial, the Krylov sequence, the solution or a least-squares solution to a linear system, and a short F -generator for the inverse. As in Section 2, the algorithms are based on the technique of the *parametrization of Newton's iteration* and involve FFTs and (for the transition to the characteristic polynomial) divisions by $2, 3, \dots, n$.

Now, we will apply Newton's iteration (2.2) in its parametrized version with $T(\lambda) = I - \lambda T$, $S_0 = I$, and deduce from (2.4) that $\text{rank } F(S_i) =$

rank $F(T)$ over the ring of polynomials in λ modulo λ^{2^i} , for any operator F^+, F^-, F^\pm . Therefore, in view of Propositions 3.1 and 3.2, computing a generator of S_i of length $d = \text{rank } F(T)$ and the first column of S_i allows us to compute modulo λ^{2^i} all the entries of S_i , that is, all the entries of T, T^2, \dots, T^{2^i-1} .

Actually, we seek the traces of T, T^2, \dots, T^n and the sequence of vectors $T\mathbf{b}, T^2\mathbf{b}, \dots, T^n\mathbf{b}$, rather than all the entries of T^2, \dots, T^n . Due to Propositions 3.1 and 3.2, these computations can be reduced to the evaluation of an F -generator and the first (or last) column of the matrix $T^{-1}(\lambda)$. On the other hand, due to (3.2) and (3.4), we may modify Newton's iteration so as to compute an F -generator of S_{i+1} from an F -generator of S_i (rather than to compute all the entries of S_{i+1}), thus reducing the computational cost per step.

Specifically, let $F(T(\lambda)) = -\lambda F(T) = GH^T$, where G and H are $n \times d$ matrices. Applying (3.4) with $F = F^+, F = F^-, F = F^\pm$, we obtain $F(S_i) = -S_i F(T(\lambda)) S_i \bmod \lambda^{2^i}$. Now, by using (2.2), we obtain

$$\begin{aligned} F(S_{i+1}) &= G_{i+1} H_{i+1}^T \bmod \lambda^{2^{i+1}}, \\ G_{i+1} &= -S_{i+1} G = -[2S_i - S_i T(\lambda) S_i] G, \\ H_{i+1}^T &= H^T S_{i+1} = H^T [2S_i - S_i T(\lambda) S_i]. \end{aligned} \tag{4.1}$$

Thus, in view of Propositions 3.1 and 3.2, the F -generator G_{i+1}, H_{i+1}^T of S_{i+1} can be computed, together with the first (or last) column of S_{i+1} , from the F -generator G_i, H_i^T and the first (or respectively last) column of S_i , by performing a constant number of multiplications of triangular Toeplitz matrices and/or of the class- τ matrices by vectors, that is, by pre- and postmultiplying the matrix polynomial $S_{i+1} = 2S_i - S_i T(\lambda) S_i$ of the expression (4.1) by the d columns and the d rows of the matrices G and H^T , respectively, and by multiplying S_{i+1} by $\mathbf{e}^{(0)}$.

Summarizing and extending all these considerations, we arrive at the following algorithm:

ALGORITHM 4.1.

Input: An F -generator of length d for an $n \times n$ matrix T (having F -rank at most d), where F denotes $F^+, F^-,$ or F^\pm , that is, two $n \times d$ matrices G, H such that $F(T) = GH^T$; the first column of T .

Output: The traces of the matrices T, T^2, \dots, T^n .

Computation:

1. Set $T(\lambda) = I - \lambda T$, $S_0 = I$, and compute

$$G_{i+1} = -[2S_i - S_i T(\lambda) S_i] G_i \bmod \lambda^{2^{i+1}},$$

$$H_{i+1}^T = H_i^T [2S_i - S_i T(\lambda) S_i] \bmod \lambda^{2^{i+1}},$$

$$\mathbf{s}_{i+1} = 2\mathbf{s}_i - S_i T(\lambda) \mathbf{s}_i,$$

$$i = 0, \dots, h-1, \quad h = \lceil \log(n+1) \rceil,$$

where $\mathbf{s}_0 = \mathbf{e}^{(0)}$ for the operators F^+ , F^\pm ; $\mathbf{s}_0 = \mathbf{e}^{(n-1)}$ for the operator F^- ; and $S_{i+1} = 2S_i - S_i T(\lambda) S_i$. Performing matrix multiplications, apply the representations of $T(\lambda)$ and S_i given in Propositions 3.1 and 3.2 and operate with F -generators rather than with the matrices. Note that $G_h H_h^T = F(I + \lambda T + \dots + \lambda^n T^n) \bmod \lambda^{n+1}$.

2. By using Proposition 3.1 or 3.2, recover from $G_h H_h^T$ the diagonal entries of $S_h \bmod \lambda^{n+1}$ and their sum, which is a polynomial in λ of degree at most n , whose coefficients are the traces of the powers of T .

The most expensive stage of the computation by the above algorithm is the six multiplications of matrices (given with their F -generators of length d) by $d+1$ vectors, over the ring of the polynomials in \mathbf{F} modulo $\lambda^{2^{i+1}}$. This stage can be reduced to $O(d^2)$ multiplications of the bivariate polynomials and for $d = O(1)$ performed in $O(n^2 \log n)$ arithmetic operations, or at the computational cost $O_A(\log^2 n, n^2/\log n)$ under the parallel models of computation.

Algorithm 4.1 can be extended in order to compute (at the same asymptotic cost, if the field of constants allows division by $n!$) the coefficients of the characteristic polynomial of the matrix T (see Appendix A).

Another simple modification of Algorithm 4.1 enables us to compute the Krylov sequence $\mathbf{b}, T\mathbf{b}, T^2\mathbf{b}, \dots, T^{n-1}\mathbf{b}$ for a given vector \mathbf{b} and for a given Toeplitz-like + Hankel-like matrix T , and to compute the solution $T^{-1}\mathbf{b}$ or a least-squares solution $T^+\mathbf{b}$ to the linear system of equations $T\mathbf{x} = \mathbf{b}$ at the cost of $O(n^2 \log n)$ operations and at the parallel cost $O_A(\log^2 n, n^2/\log n)$. For this purpose, once a generator modulo λ^n of length d of the matrix S_h has been computed, together with the first (or last) column of S_h , we have a representation of $S_h \bmod \lambda^{n+1}$ given by Propositions 3.1 and 3.2. Thus, we may compute the matrix-by-vector product $S_h \mathbf{b} \bmod \lambda^{n+1}$, by performing

multiplications of matrices and vectors by means of FFTs and/or sine transforms. The result of this computation is the vector polynomial $\mathbf{b} + \lambda T \mathbf{b} + (\lambda T)^2 \mathbf{b} + \dots + (\lambda T)^{n-1} \mathbf{b}$, which gives us the Krylov sequence.

In order to compute the vector $\mathbf{x} = T^{-1} \mathbf{b}$, we may apply the modification of Algorithm 4.1 to compute the coefficients c_i , $i = 0, \dots, n$, of the characteristic polynomial $\det(\lambda I - T)$ and use the Cayley-Hamilton theorem to write $\mathbf{x} = -(1/c_0) \sum_{i=0}^{n-1} c_{i+1} T^i \mathbf{b}$, $c_n = 1$. Analogously, we may compute a generator of length d for the inverse matrix together with its first (or last) column. Even for these extensions of the computations, the upper bound on the asymptotic cost remains unchanged. If T is (in addition) a Hermitian matrix, we may apply (2.7) in order to similarly compute $T^+ \mathbf{b}$. The latter assumption, however, can be relaxed, since for any matrix T , we may obtain T^+ from the generalized inverse of the Hermitian matrix

$$\begin{bmatrix} 0 & T^H \\ T & 0 \end{bmatrix}.$$

APPENDIX A. A FAST TRANSITION FROM THE POWER SUMS OF POLYNOMIAL ZEROS TO THE POLYNOMIAL COEFFICIENTS

Let $c(x)$ denote a monic polynomial,

$$c(x) = \sum_{i=0}^n c_i x^i = \prod_{j=1}^n (x - x_j), \quad c_n = 1,$$

and let

$$t_k = \sum_{j=1}^n x_j^k, \quad k = 0, 1, \dots$$

The power sums t_k and the coefficients c_i are related to each other via the system of Newton's identities:

$$t_k + \sum_{i=1}^{k-1} c_{n-i} t_{k-i} + k c_{n-k} = 0, \quad k = 1, \dots, n,$$

$$t_{n+k} + \sum_{i=1}^n c_{n-i} t_{n+k-i} = 0, \quad k = 1, 2, \dots, m - n.$$

By using these identities, we may compute the power sums t_k if we are given the coefficients c_i , and vice versa. For the inverse computation, however, a simpler algorithm is available, due to [30] (see also [23]). Consider the reverse polynomial

$$\sum_{i=0}^n c_{n-i} x^i = x^n c(x^{-1}) = 1 + u(x) = 1 + \sum_{i=1}^n c_{n-i} x^i = \prod_{j=1}^n (1 - x x_j).$$

Obtain that

$$\{\ln[1 + u(x)]\}' = \frac{u'(x)}{1 + u(x)} = - \sum_{j=1}^k t_j x^{j-1} \bmod x^k, \quad k \leq n + 1. \quad (\text{A.1})$$

Denote $u_r(x) = u(x) \bmod x^{r+1}$. Note that $u_1(x) = c_{n-1}x$, and show the transition from $u_r(x)$ to $u_{2r}(x)$ for $r = 1, 2, 3, \dots$. Start with the equation

$$1 + u_{2r}(x) = [1 + u_r(x)][1 + v_r(x)] \bmod x^{2r+1}, \quad (\text{A.2})$$

where

$$v_r(x) = \sum_{i=r+1}^{2r} v_i x^i. \quad (\text{A.3})$$

All we need is to compute the coefficients v_{r+1}, \dots, v_{2r} provided that the coefficients of $u_r(x)$ are known. Deduce from (A.3) that

$$\frac{v_r'(x)}{1 + u_r(x)} = v_r'(x) \bmod x^{2r+1}.$$

Then deduce from (A.2) and (A.3) that

$$\{\ln[1 + u_{2r}(x)]\}' = \frac{u_{2r}'(x)}{1 + u_{2r}(x)} = \frac{u_r'(x)}{1 + u_r(x)} + \frac{v_r'(x)}{1 + v_r(x)} \bmod x^{2r}.$$

Combine these equations with (A.1) for $k = 2r + 1$, and deduce that

$$\frac{u_r'(x)}{1 + u_r(x)} + v_r'(x) = - \sum_{j=1}^{2r+1} t_j x^{j-1} \bmod x^{2r}. \quad (\text{A.4})$$

Since we know the coefficients of $u_r(x)$ and the values of t_j for $j \leq 2r + 1$, we may now compute the polynomial $[1 + u_r(x)]^{-1} \bmod x^{2r}$, multiply it by $u_r'(x) \bmod x^{2r}$, substitute the result into (A.4) to obtain the

coefficients of $v_r'(x)$, and finally recover the coefficients of $v_r(x)$ [by using (A.3)] and of $u_{2r}(x)$ [by using (A.2)]. The computation is reduced to a sequence of multiplications of polynomials, and it is easy to verify that their overall cost is bounded by

$$O(\log^2 n, n/\log n).$$

In particular, $c(x)$ may represent the characteristic polynomial of a matrix A , and then t_k represents trace A^k , $k = 0, 1, \dots$. In this case the algorithm enables us to recover the characteristic polynomial of a matrix A from the traces of its powers A^i for $i = 0, 1, \dots, n$.

APPENDIX B. MULTIPLICATION OF A TOEPLITZ MATRIX, ITS INVERSE, AND TOEPLITZ-LIKE + HANKEL-LIKE MATRICES BY VECTORS

Consider the vector equation

$$\mathbf{p} = Q\mathbf{r} = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{k+n-2} \end{pmatrix} = \begin{pmatrix} q_0 & & & & & O \\ q_1 & \cdot & & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & & \cdot & \cdot & & \\ \cdot & & & \cdot & \cdot & \\ q_{k-1} & & & & \cdot & q_0 \\ & \cdot & & & & q_1 \\ & & \cdot & & & \cdot \\ & & & \cdot & & \cdot \\ & & & & \cdot & \cdot \\ O & & & & & q_{k-1} \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-1} \end{pmatrix}, \quad (\text{B.1})$$

which represents the product of a special Toeplitz matrix Q by a vector and equivalently represents the product $p(x) = \sum_{i=0}^{k+n-2} p_i x^i$ of two polynomials

$$q(x)r(x) = \left(\sum_{i=0}^{k-1} q_i x^i \right) \left(\sum_{j=0}^{n-1} r_j x^j \right).$$

Multiplication of any $n \times n$ triangular Toeplitz matrix by a vector is represented by the first n equations of (B.1) for $k = n$ and thus can be reduced to multiplication of two polynomials of degrees at most n , which can be further reduced to three discrete Fourier transforms (DFTs) on $m \geq 2n - 1$ points, performed by means of FFTs. Note that we first extend the coefficient vector of each input polynomial by $m - n$ zeros when we compute the DFT on m points.

Let \mathbf{i} be the imaginary unit, and $\omega_m = \cos(2\pi/m) + \mathbf{i} \sin(2\pi/m)$ be an m th root of 1. Given the coefficients of the polynomials $q(x)$ and $r(x)$, the coefficients of the polynomial $p(x) = q(x)r(x)$ can be computed according to the following stages:

1. Compute $\mu_i = q(\omega_m^i)$, $i = 0, 1, \dots, m - 1$, by means of an FFT on m points.
2. Compute $\nu_i = r(\omega_m^i)$, $i = 0, 1, \dots, m - 1$, by means of an FFT on m points.
3. Compute $\eta_i = \mu_i \nu_i$, $i = 0, 1, \dots, m - 1$.
4. Compute $p_i = (1/m) \sum_{j=0}^{m-1} \omega_m^{-ij} \eta_j$, $i = 0, 1, \dots, m - 1$, by means of an FFT on m points.

Given a vector \mathbf{w} and the vectors \mathbf{a} and \mathbf{b} of (3.7), the computation of the product $A^{-1}\mathbf{w}$ can be performed by means of 10 FFTs on $m \geq 2n - 1$ points.

Propositions 3.1 and 3.2 and Algorithm 4.1 immediately enable us to extend this result to multiplication of a Toeplitz-like + Hankel-like matrix A and its inverse by a vector \mathbf{w} .

There are several techniques leading to further improvements by constant factors, in particular, by using circulant and/or factor circulant matrices instead of triangular Toeplitz matrices. (Note that every Toeplitz matrix is a sum of a circulant and a skew-circulant matrix.) We refer the reader to [1, 2] on the techniques for the Hermitian Toeplitz systems, to [10, 11] on the Toeplitz-like case, and to [32] on the Toeplitz-like + Hankel-like case.

APPENDIX C. RELATIONS BETWEEN F^+ , F^- AND THE CLASSICAL DISPLACEMENT OPERATORS

The classical displacement operators F_+ and F_- of [15, 8], such that

$$F_+(A) = A - ZAZ^T,$$

$$F_-(A) = A - Z^T AZ,$$

are related to the operators F^+ and F^- of Sections 3 and 4 via the following equations, which hold for any $n \times n$ matrix A [25]:

$$\begin{aligned}
 F^+(A)Z^T &= F_+(A) - A\mathbf{e}^{(0)}\mathbf{e}^{(0)T}, \\
 Z^TF^+(A) &= \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}A - F_-(A), \\
 F^-(A)Z &= F_-(A) - A\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}, \\
 ZF^-(A) &= \mathbf{e}^{(0)}\mathbf{e}^{(0)T}A - F_+(A), \\
 F_+(A)Z &= F^+(A) + ZA\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}, \\
 Z^TF_+(A) &= \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}AZ^T - F^-(A), \\
 F_-(A)Z^T &= F^-(A) + Z^TA\mathbf{e}^{(0)}\mathbf{e}^{(0)T}, \\
 ZF_-(A) &= \mathbf{e}^{(0)}\mathbf{e}^{(0)T}AZ - F^+(A), \\
 F^-(A) &= -[F^+(A^T)]^T, \\
 F^-(A) &= JF^+(JAJ)J.
 \end{aligned}$$

These equations are immediately verified from the definition of the operators F^+ , F^- , F_+ , and F_- and from the following simple vector equations:

$$Z^TZ = I - \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}, \quad ZZ^T = I - \mathbf{e}^{(0)}\mathbf{e}^{(0)T}.$$

The authors are grateful to the referee and to the editor Dr. Gregory S. Ammar for helpful comments.

REFERENCES

- 1 G. S. Ammar and P. Gader, A variant of the Gohberg-Semencul formula involving circulant matrices, *SIAM J. Matrix Anal. Appl.* 12(3):534–541 (1991).
- 2 G. S. Ammar and P. Gader, New decompositions of the inverse of Toeplitz matrix, in *Signal Processing, Scattering and Operator Theory and Numerical Methods, Proceedings of International Symposium MTNS-89*, Vol. III, Birkhäuser, Boston, 1990, pp. 421–428.
- 3 G. S. Ammar and W. B. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM J. Matrix Anal. Appl.* 9(1):61–76 (1988).

- 4 D. Bini and M. Capovani, Spectral and computational properties of band symmetric Toeplitz matrices, *Linear Algebra Appl.* 52/53:99–126 (1983).
- 5 D. Bini, L. Gemignani, and V. Pan, Improved parallel computations with matrices and polynomials, in *Proceedings of the 18th International Symposium on Automata, Languages and Programming*, Lectures Notes in Comput. Sci. 510, Springer-Verlag, Berlin, 1991, pp. 520–531.
- 6 D. Bini, On a Class of Matrices Related to Toeplitz Matrices, Tech. Rep. TR 83-5, Computer Science Dept., SUNYA, Albany, N.Y., 1983.
- 7 D. Bini and V. Pan, *Numerical and Algebraic Computations with Matrices and Polynomial*, Birkhauser, Boston, Mass., 1993, to appear.
- 8 J. Chun, T. Kailath, and H. Lev-Ari, Fast parallel algorithm for QR-factorization of structured matrices, *SIAM J. Sci. Statist. Comput.* 8(6):899–913 (1987).
- 9 B. Friedlander, M. Morf, T. Kailath, and L. Ljung, New inversion formulas for matrices classified in terms of their distances from Toeplitz matrices, *Linear Algebra Appl.* 27:31–60 (1979).
- 10 I. Gohberg and V. Olshevsky, Circulants, displacements and decompositions of matrices, *Integral Equations Operator Theory* 15:5 (1992).
- 11 I. Gohberg and V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, Report, School of Mathematical Sciences, Tel-Aviv Univ., Ramat Aviv, Israel, 1992.
- 12 I. C. Gohberg and A. A. Semencul, On the inversion of finite Toeplitz matrices and their continuous analogs (in Russian), *Mat. Issl.* 2:201–233 (1972).
- 13 J. Ja Ja, *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, Mass., 1992.
- 14 J. R. Jain, An efficient algorithm for a large Toeplitz set of linear functions, *IEEE Trans. Acoust. Speech Signal Process.* 27:612–615 (1979).
- 15 T. Kailath, S.-Y. Kung, and M. Morf, Displacement rank of matrices and linear equations, *J. Math. Anal. Appl.* 68(2):395–407 (1979).
- 16 E. Kaltofen and V. Pan, Processor efficient parallel solution of linear systems II: Positive characteristic and singular cases, in *Proceedings of the 33rd Annual IEEE Symposium on Foundations of Computer Science*, IEEE Comput. Soc. Press, 1992, pp. 714–723.
- 17 E. Kaltofen and V. Pan, Processor efficient parallel solution of linear systems over an abstract field, in *Proceedings of the 3rd Annual ACM Symposium on Parallel Computers and Architectures*, ACM Press, 1991, pp. 180–191.
- 18 R. Karp and V. Ramachandran, A survey of parallel algorithms for shared memory computers, in *Handbook for Theoretical Computer Science* (J. van Leeuwen, Ed.), North-Holland, Amsterdam, 1990, pp. 869–941.
- 19 H. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*, Morgan Kauffman, San Mateo, Calif., 1991.
- 20 V. Pan, Concurrent iterative algorithm for Toeplitz-like linear systems, *IEEE Trans. Parallel Distrib. Systems*, to appear.
- 21 V. Pan, On computations with dense structured matrices, *Math. Comp.* 55(191):179–190 (1990).

- 22 V. Pan, Parallel inversion of Toeplitz and block Toeplitz matrices, in *Oper. Theory Adv. Appl.* 40, Birkhäuser, Boston, 1989, pp. 359–389.
- 23 V. Pan, Parallel least-squares solution of general and Toeplitz-like linear systems, in *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures*, 1990, pp. 244–253.
- 24 V. Pan, Parallel solution of Toeplitz-like linear systems, *J. Complexity* 8:1–21 (1992).
- 25 V. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, Tech. Rep. CUCS-031-90, Computer Science Dept., Columbia Univ., New York, 1990.
- 26 V. Pan and F. Preparata, Supereffective slow-down of parallel computations, in *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms & Architectures*, ACM Press, 1992, pp. 402–409.
- 27 M. Pease, An adaptation of the fast Fourier transform for parallel processing, *J. Assoc. Comput. Mach.* 15:252–284 (1968).
- 28 W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, Cambridge U.P., Cambridge, 1986.
- 29 M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers*, McGraw-Hill, New York, 1987.
- 30 A. Schönhage, The fundamental theorem of algebra in terms of computational complexity, unpublished manuscript, 1982.
- 31 W. F. Trench, A note on a Toeplitz inversion formula, *Linear Algebra Appl.* 129:55–61 (1990).
- 32 C. Di Fiore and P. Zellini, submitted.

Received 29 July 1992; final manuscript accepted 20 February 1993