Improved Parallel Computations with Toeplitz-like and Hankel-like Matrices

Dario Bini* Victor Pan†

TR-92-052

September 1992

Abstract

The known parallel 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 evaluations (such as computing 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, FFT's) 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 and in addition show some more minor innovations, such as an improvement of the transition to the solution to a Toeplitz linear system $T\mathbf{x} = \mathbf{b}$ from two computed columns of T^{-1}.

---

*Department of Mathematics, University of Pisa. Dario Bini w as supported by NSF Grants CCR 8805782 and CCR 9020690, and by MPI (40% funds).

†Lehman College, CUNY, and ICSI. Victor Pan was supported by NSF grants CCR 8805782, CCR 9020690 and by PSC CUNY Awards #661340, #668541, #669210 and #662478.

Key words: Toeplitz matrices, Hankel matrices, parallel algorithms, displacement operator, displacement rank, Krylov sequences, polynomial gcd, linear recurrence, least-squares solution.

1991 Math Subject Classification: 65F05, 65Y05, 68Q40, 68Q25, 15A09

1. Introduction.

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

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 [P89], [P92]; and for rapid refinement of an already rather close initial approximation to the solution to a Toeplitz or Toeplitz-like linear system [P]).

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 $\mathbf{v}, T\mathbf{v}, \ldots, T^K\mathbf{v}$; of the traces of T^i, $i = 1, 2, \ldots, K$, $K = O(n)$; 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 null space of T, of the minimum span of a linear recurrence sequence (Berlekamp-Massey problem), of polynomial gcd and lcm, Padé approximation and extended Euclidean scheme for polynomials ([BGP], [KP], [P90b]). 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 [P90a].

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 1.1 (listed for some fundamental problems of parallel computations). We refer the reader to [Q], [JJ] and [L] on verification of these simple cost bounds under some realistic models of parallel computing.

Table 1.1

	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 FFT's)	$O(\log n)$	$O(n)$
multiplication of two bivariate polynomials of degrees m and n in the two variables (by means of 2-dimensional FFT's)	$O(\log mn)$	$O(mn)$

The fundamental complexity bounds of Table 1.1 are immediately extended to many other computations. In particular, 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 is reduced to polynomial multiplication (convolution), (see appendix B).

We use the asymptotic complexity estimates presented in the form $O(t,p)$, which amounts to $O(t)$ asymptotic bound on arithmetic time (the number of arithmetic parallel steps) used in the algorithms 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 [Pe], 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 [KR], [P90b], according to which the number of processors required for the implementation of a parallel algorithm can be decreased by the factor of s, $1 \le s \le p$, 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, and in particular, for $s = p$, the time bound $O(pt, 1)$, which measures the total potential work $O(pt)$ of the parallel algorithm if it were implemented sequentally, on a single processor. A 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 $O(\log n, n/\log n)$ ([Q], [L]) 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, $1 \le s \le n^2/\log n$, due to Brent's principle. Moreover, the general techniques of supereffective slowdown of parallel computations [PP] 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, $0 < a \leq 1$. In particular, for $a = 1/2$, this turns into the bounds $O(n^{1/2}\log^2 n, n/\log n)$.

Note that the total work (sequential time) of our fast parallel algorithm is $O(n^2 \log n)$, even for $a = 1$, which is close to the running time of the sequential algorithm of Levinson-Durbin (widely used for solving Toeplitz linear systems), and which is the best sequential time bound known for computing the characterstic polynomial of a Toeplitz or a Toeplitz-like matrix.

In appendix B, we supply simple but practially promising (for both sequential and parallel implementations) improvements of the known algorithms 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}. Some of the improvements can be extended to the Toeplitz-like case. We also include some little known techniques for simple transition from the traces of the powers of any general matrix T to its characterstic 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 [KKM], [CKL-A]. In appendix C we display some correlations between such operators and the two classical displacement operators of [KKM], [CKL-A]. Some further details can be found in our original technical reports [P90b] (on the algorithms for the computations with Toeplitz and Toeplitz-like matrices) and [B83] (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 $\{T^i\mathbf{v}, i = 0, 1, \ldots\}$ (for any fixed vector $\mathbf{v}$), of the sequence $\{\text{trace}(T^i), i = 0, 1, \ldots\}$ and of the solution and a least-squares solution to a Toeplitz linear system.

Hereafter $\lceil x \rceil$ and $\lfloor x \rfloor$ 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, $h = 0, 1, \ldots, n-1$; 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 the 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,

$$X_{i+1} = 2X_i - X_i T X_i \ , \qquad i = 0, 1, \ldots \ , \tag{2.1}$$

but we will apply it to the *parametrized* matrix

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

such that

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

thus outputing the powers of T.

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

$$S_{i+1} = 2S_i - S_i T(\lambda) S_i = (2I - S_i T(\lambda)\,) S_i \ , \qquad i = 0, \ldots, k-1 \ . \tag{2.2}$$

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 $I, T, \ldots, T^{K-1}$.

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

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

$$\widetilde{S}_i = S_i \bmod \lambda^{2^i} = I + \lambda T + \lambda^2 T^2 + \cdots + \lambda^{2^i - 1} T^{2^i - 1} = (T(\lambda)\,)^{-1} \bmod \lambda^{2^i} \ . \tag{2.4}$$

Let us estimate the computational cost of this algorithm assuming that T is a Toeplitz matrix (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 $\widetilde{S}_i = S_i \bmod \lambda^{2^i} = T(\lambda)^{-1} \bmod \lambda^{2^i}$ via its first and last columns, by applying the Gohberg-Semencul formula for the inverse of a Toeplitz matrix (see [GS], [FMKL], [T]):

$$\widetilde{S}_i = \frac{1}{u_0^{(i)}} \left(L(\mathbf{u}^{(i)}) L^T (J \mathbf{v}^{(i)}) - L(Z \mathbf{v}^{(i)}) L^T (ZJ \mathbf{u}^{(i)}) \right) \bmod \lambda^{2^i} \ , \tag{2.5}$$

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

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:

$$\mathbf{u}^{(i+1)} = 2\mathbf{u}^{(i)} - \widetilde{S}_i T(\lambda) \mathbf{u}^{(i)} \ ,$$

$$\mathbf{v}^{(i+1)} = 2\mathbf{v}^{(i)} - \widetilde{S}_i T(\lambda) \mathbf{v}^{(i)} \ ,$$

and algorithm 2.1 outputs the pair of vector polynomials $\widetilde{S}_k \mathbf{e}^{(0)}$ and $\widetilde{S}_k \mathbf{e}^{(n-1)}$, that is, the first and the last columns of the matrix polynomial $\widetilde{S}_k = S_k \bmod \lambda^K$. Due to (2.5), this computation is reduced to 10 multiplications of $n \times n$ Toeplitz matrix polynomials modulo λ^{2^i} by vectors. 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 2^i in their two variables and performed at the parallel cost $O(i+\log n, n2^i)$ by using 2-dimensional FFT's. The overall parallel cost of steps $i=0,\ldots,k-1$ 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 $i=0,1,\ldots,k-1-\lceil\log k\rceil$, which suggests the bound

$$O(\log(Kn)\log K, nK/\log K) \tag{2.6}$$

if we exclude the last $\lceil\log k\rceil = \lceil\log\log K\rceil$ steps. Instead of their exclusion, 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 $\widetilde{S}_k\mathbf{e}^{(0)}$ and $\widetilde{S}_k\mathbf{e}^{(n-1)}$, we may immediately recover the vector polynomial

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

defining the Krylov sequence

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

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

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

This gives us trace(T^i), $i=0,1,\ldots,K-1$, and the result has various further applications.

In particular, having computed trace(T^i) for $i=0,1,\ldots,n$, 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 of further applications, 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 $c_0=c_1=\cdots=c_{n-r-1}=0$, $c_{n-r}\neq 0$ (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 ([P90]):

$$T^+ = (1/c_{n-r}) \sum_{i=n-r+1}^{n-1} ((c_{n+1-r}/c_{n-r})c_i - c_{i+1}) T^{i-n+r} + (c_{n+1-r}/c_{n-r}) T^r \ . \tag{2.7}$$

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

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

Remark 2.2. The results of this section can be applied over any field of constants supporting 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), $i = 0, 1, \ldots, n$, to $c_0, \ldots, c_{n-1}$ shown in appendix A requires divisions by $2, 3, \ldots, n$ and thus cannot be performed in the fields of characteristic p for $1 < p \leq n$. Alternate techniques of [KPa] use randomization to ensure such a transition over any field at the computation cost

$$O\left(\log^2 n\, d(n,p),\, n^2 \log\log n \,/\, d(n,p) \log n)\right)$$

where p is the characteristic of the field of constants, $d(n,p) = \lceil \log n / \log p \rceil$ if $p > 0$, $d(n,p) = 1$ 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 [B83] 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 [KKM], [CKL-A].

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 $\mathbf{F}$:

$$F_V(A) = AV - VA \ . \tag{3.1}$$

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

Observe that 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 $\mathcal{N}(F_V)$ of F_V and of a matrix R belonging to the range of F_V, i.e. $\mathcal{R}(F_V) = \{F_V(A)\colon A \in \mathbf{F}^{n\times n}\}$.

In this section we determine some choices for the matrix V, which define operators F_V particularly effective in the study of Toeplitz-like, Hankel-like and the sums of Toeplitz-like and Hankel-like matrices. (Such a natural extension of the classes of Toeplitz, Hankel and the sums of Toeplitz and 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:

– the null-space $\mathcal{N}(F_V)$ must be made up by "computationally easy" matrices;
– the range $\mathcal{R}(F_V)$ must be made up by 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 with 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

$$F_V(AB) = AF_V(B) + F_V(A)B \ , \tag{3.2}$$

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

Moreover, if A is nonsingular, then

$$F(A^{-1}) = -A^{-1}F(A)A^{-1} \ . \tag{3.4}$$

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 column-wise, i.e. $\mathbf{a} = (a_{11}, a_{21}, \ldots, a_{n1}, a_{12}, a_{22}, \ldots, a_{n2}, \ldots, a_{nn})^T$, where $a_{ij} = (A)_{ij}$. This way equation (3.1) can be rewritten in the following form:

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

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 made up by the algebra of lower triangular Toeplitz matrices. The null space of F^- is made up by 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,*

$$F^+(A) = \mathbf{e}^{(0)}\mathbf{e}^{(0)T}AZ - ZA\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T} = \mathbf{e}^{(0)}(JZA\mathbf{e}^{(n-1)})^T - ZA\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T} \ ,$$

$$F^-(A) = \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}AZ^T - Z^TA\mathbf{e}^{(0)}\mathbf{e}^{(0)T} = \mathbf{e}^{(n-1)}(JZ^TA\mathbf{e}^{(0)})^T - Z^TA\mathbf{e}^{(0)}\mathbf{e}^{(0)T} \ .$$

(c) *Let $G = [\mathbf{g}_1, \ldots, \mathbf{g}_d]$, $H = [\mathbf{h}_1, \ldots, \mathbf{h}_d] \in \mathbf{F}^{n\times d}$. 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 $\sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} Z^{n-j-1}\mathbf{g}_i = 0$, where $\mathbf{h}_i = [h_j^{(i)}]$ and any 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 $\sum_{i=1}^d \sum_{j=0}^{n-1} h_j^{(i)} (Z^T)^j \mathbf{g}_i = 0$ and any 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 proofs of parts (a) and (b) are immediate. To prove part (c), we follow [B83] 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 tridiagonal system of linear equations:

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

This immediately implies the vector equation of part (c).

Now we fix the first column $\mathbf{a}_0$ of A, compute the remaining columns by using back substitution and arrive at the first matrix equation of part (c). To yield the extensions to the other equations of parts (c) and (d), combine the relations $(F^-(A))^T = -F^+(A^T)$, $F^-(A) = JF^+(JAJ)J$ obtained 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 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) or proposition 3.1 and the matrix equation (3.4) together yield the following inversion formula for a nonsingular Toeplitz matrix A:

$$A^{-1} = L(\mathbf{a})L^T(\mathbf{e}^{(0)} - J\mathbf{b}) - L(\mathbf{b})L^T(J\mathbf{a}) \ , \mathbf{a} = A^{-1}\mathbf{e}^{(0)} \ , \qquad \mathbf{b} = A^{-1}ZA\mathbf{e}^{(n-1)} \ . \tag{3.7}$$

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 division.

A matrix A having an F-generator of length d bounded from above by a fixed (small) constant [or formally, using the "O" notation, of length $d = O(1)$ as $n \to \infty$], 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 FFT's and one inverse FFT at the sequential cost of $O(n \log n)$ arithemtic operations and at the parallel cost $O(\log n, n)$.

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

$$F^{\pm}(A) = AM - MA \ . \tag{3.8}$$

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 made up by the algebra τ generated by $I, M, M^2, \ldots, M^{n-1}$; the entries of any matrix $U \in \tau$ are defined by the equations*

$$u_{i,j-1} + u_{i,j+1} = u_{i-1,j} + u_{i+1,j} \ ,$$

$$u_{i,j} = 0 \text{ if } i \in \{-1, n\}, \text{ or } j \in \{-1, n\} \ .$$

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

(c) *Let $G = [\mathbf{g}_1, \ldots, \mathbf{g}_i]$, $H = [\mathbf{h}_1, \ldots, \mathbf{h}_d] \in \mathbf{F}^{n \times d}$. For any matrix $A \in F^{n \times n}$, we have $F^{\pm}(A) = GH^T = \sum_{i=1}^{d} \mathbf{g}_i \mathbf{h}_i^T$ if and only if simultaneously $\sum_{i=1}^{d} \sum_{j=0}^{n-1} h_j^{(i)} T_{n-j-1}(M)\mathbf{g}_i + T_n(A\mathbf{e}^{(0)}) = 0$, where $T_j(x)$ denotes the Chebyshev polynomial of degree j, and any of the four following matrix equations holds:*

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

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

(d)

$$F^{\pm}(JAJ) = JF^{\pm}(A)J \ , F^{\pm}(A^T) = -(F^{\pm}(A))^T \ .$$

Proof: The proof is analogous to the proof of proposition 3.1 and is left to the reader. We only observe that for the operator $F^{\pm}$, equation (3.2) takes the following form:

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

and that the matrix M has n distinct eigenvalues.

Proposition 3.2, together with (3.2)–(3.4), allows us to represent the product of Toeplitz and Hankel matrices and the inverse of nonsingular Toeplitz and Hankel matrices in terms of a sum of products of matrices of the class τ and upper triangular Toeplitz matrices. 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 $n \to \infty$], with respect to the operator $F^{\pm}$ of (3.7), is called *representable as a sum of Toeplitz-like and Hankel-like matrices.*

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 uniquely determined by their first column. Moreover, for any matrix $A = (a_{ij}) \in \tau$, the following relations hold ([BC]):

$$A = S^T DS \ , D = \mathrm{diag}(\sigma_0, \ldots, \sigma_{n-1}) \ , \sigma_{i-1} = \frac{\sum_{j=1}^{n} a_{j-1,0} \sin[ij\pi/(n+1)]}{\sin[i\pi/(n+1)]} \ , \qquad i = 1, \ldots, n \ , S = \frac{1}{\sqrt{n+1}} \Big(\mathrm{si}$$

Here the matrix S, associated to the *sine transform* $\mathbf{x} \to S\mathbf{x}$, is symmetric and orthogonal, that is, such that $S = S^T$, $S^T S = I$. Performing the sine transform of a real vector involves by factor 1/2 fewer operations than performing its FFT [PFTV].

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

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

$$A^{-1} = \tau(\mathbf{a}_0)L^T(\mathbf{e}^{(0)} - \mathbf{d}_0) + \tau(\mathbf{c}_0)L^T(\mathbf{b}_0 - \tau(\mathbf{a}_{n-1})L^T(\mathbf{d}_{n-1}) + \tau(\mathbf{c}_{n-1})L^T(\mathbf{b}_{n-1}), \mathbf{a}_i = A^{-1}\mathbf{e}^{(i)} \ , \ \mathbf{b}_i = A^{-T}\mathbf{e}^{(i)}$$

Remark 3.1. Our approach can be effectively applied also with other choices of the matrix V of (3.1). In particular, by choosing $V = C_0$, with C_0 being the unit circulant matrix [with the first row $\mathbf{e}^{(n-1)}$], the reader may deduce useful formulae for the inverse of a Toeplitz matrix, similar to ones of [AG].

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 Toeplitz-like and Hankel-like matrices. We first present an algorithm that computes the traces of matrix powers, uses $O(n^2 \log n)$ ops 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, for an $n \times n$ Toeplitz-like + Hankel-like matrix. As in section 2, the algorithms are based on the technique of the *parametrization of Newton's iteration* and involve FFT's and (for the transition to the characteristic polynomial) divisions by $2, 3, \ldots, n$.

Now we 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 $\mathrm{rank}(F(S_i)) = \mathrm{rank}(F(T))$ 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 $d = \mathrm{rank}(F(T))$ 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 $T, T^2, \ldots, T^{2^i-1}$.

Actually, we seek the traces of $T, T^2, \ldots, T^n$ and the sequence of vectors $T\mathbf{b}, T^2\mathbf{b}, \ldots$, $T^n\mathbf{b}$, rather than all the entries of $T^2, \ldots, T^n$. 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 $T(\lambda)$. 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 yield $F(S_i) = -S_i F(T(\lambda)) S_i \bmod \lambda^{2^i}$. Now, by using (2.2), we obtain

$$F(S_{i+1}) = G_{i+1} H_{i+1}^T \, , G_{i+1} = -S_{i+1} G = -(2S_i - S_i T(\lambda) S_i) G \, , H_{i+1}^T = H^T S_{i+1} = H^T (2S_i - S_i T(\lambda) S_i) \, . \tag{4.1}$$

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 column of S_{i+1}, from the F-generator G_i, H_i^T and the first 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 post-multiplying 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 $F(T) = GH^T$.

Output: the traces of the matrices $T, T^2, \ldots, T^n$.

Computation:

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

$$G_{i+1} = -(2S_i - S_i T(\lambda) S_i) G \bmod \lambda^{2^{i+1}} \, , H_{i+1}^T = H^T (2S_i - S_i T(\lambda) S_i) \bmod \lambda^{2^{i+1}} \, , \qquad i = 0, \ldots, h-1,$$

where $\mathbf{s}_0 = \mathbf{e}^{(0)}$ for the operators F^+, $F^\pm$; $\mathbf{s}_0 = \mathbf{e}^{(n-1)}$ for the operator F^-, $S_{i+1} = 2S_i - S_i T(\lambda) S_i$. 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 $G_h H_h^T = F(I + \lambda T + \cdots + \lambda^n T^n) \bmod \lambda^{n+1}$.

2. By using propositions 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 power sums of the eigenvalues of T.

The most expensive stage of the computation by the above algorithm is the performance of 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 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 $\mathbf{b}, T\mathbf{b}, T^2\mathbf{b}, \ldots, T^{n-1}\mathbf{b}$, for a given vector $\mathbf{b}$ and for a given Toeplitz-like and Hankel-like matrix T and to compute the solution $T^{-1}\mathbf{b}$ or a least-squares solution $T^{+}\mathbf{b}$ to a linear system of equations at the cost $O(n^2 \log n)$ ops 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 mod λ^{n+1} given by propositions 3.1 and 3.2. Thus, we may compute the product S_h mod λ^{n+1} by performing multiplications of matrices and vectors by means of FFT's, and/or sine transforms. The result of this computation is the vector polynomial $\mathbf{b} + \lambda T\mathbf{b} + (\lambda T)^2\mathbf{b} + \cdots + (\lambda T)^{n-1}\mathbf{b}$, which gives us the Krylov sequence.

In order to compute the vector $\mathbf{x} = T^{-1}\mathbf{b}$, we apply the modification of algorithm 4.1 to compute the coefficients c_i, $i = 0, \ldots, n$ 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 of the inverse matrix together with its first (or last) column. Even for this extension 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 $\begin{bmatrix} O & T^H \\ T & O \end{bmatrix}$.

Appendix A. A fast transition from the power sum of polynomial zeros to the polynomial coefficients.

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

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

and let

$$t_k = \sum_{j=1}^{n} x_j^k , \qquad k = 0, 1, \ldots .$$

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

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

$$t_{n+k} + \sum_{i=1}^{n} c_{n-i}t_{n+k-i} = 0 , \qquad k = 1, 2, \ldots, m-n ,$$

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 converse computation, however, a simpler algorithm is available, due to [S] (see also [P90]). 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

$$(\ln(1+u(x)))' = u'(x)/(1+u(x)\,) = -\sum_{j=1}^{k} t_j x^{j-1} \bmod x^k \ , \quad k \le n+1 \ . \tag{A.1}$$

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 $r = 1, 2, 3, \ldots$. Start with the equation

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

where

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

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

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

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

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

Since we know the coefficients of $u_r(x)$ and the values of t_j for $j \le 2r + 1$, 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), $k = 0, 1, \ldots$. 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 $i = 0, 1, \ldots, n$.

Appendix B. A simplification of multiplications of Toeplitz and Toeplitz-like matrices by vectors.

Consider the vector equation,

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

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 (DFT's) on $m \geq 2n - 1$ points, performed by means of FFT's.

Let $\mathbf{i}$ be the imaginary unit and $\omega_m = \cos \frac{2\pi}{m} + \mathbf{i} \sin \frac{2\pi}{m}$ 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 $\mu_i = q(\omega_m^i)$, $\quad i = 0, 1, \ldots, m-1$, by means of an FFT on m points;

2– Compute $\nu_i = r(\omega_m^i)$, $\quad i = 0, 1, \ldots, m-1$, by means of an FFT on m points;

3– Compute $\eta_i = \mu_i \nu_i$, $\quad i = 0, 1, \ldots, m-1$;

4– Compute $p_i = \frac{1}{m} \sum_{j=0}^{m-1} \omega_m^{-ij} \eta_j$, $\quad i = 0, 1, \ldots, m-1$, 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 FFT's 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 $\mathbf{w}$.

Let us show a simpler way in the case of a Toeplitz matrix A. Let $d = k - n + 1 > 0$. Then we may delete the first $n-1$ and the last $n-1$ equations of (B.1) and arrive at the product of a general $d \times n$ Toeplitz matrix by a vector, which is thus reduced to multiplication of two polynomials of degrees $k - 1 = d + n - 2$ and $n - 1$, respectively. This in turn can be reduced to three FFTs on $m \geq k + n - 1 = 2n + d - 2$ points.

If we need to compute $A^{-1}\mathbf{w}$, (for a Toeplitz matrix A), we may combine (3.7) with (B.1), provided that the vectors $\mathbf{a}$ and $\mathbf{b}$ of (3.7) are available.

An easy (but apparently novel and practically promising) simplification of this computation can be obtained by observing that the multiplication of two polynomials $q(x)$ and $r(x)$ of degrees at most $n - 1$ yields both of the products $L(\mathbf{q})\mathbf{r}$ [given by the first n equations of (B.1) with $k = n$] and

$$L^T(J\mathbf{q})\mathbf{r} = JL(\mathbf{q})J\mathbf{r}$$

[given by the last n equations of (B.1)]. This saves for us two FFT's in the computation of the product $A^{-1}\mathbf{w}$, if we are given the vectors $\mathbf{a}$ and $\mathbf{b}$ of (3.7) and $\mathbf{w}$. Such a computation may now be performed at the overall cost of 8 FFT's.

Similar consideration and the same conclusion apply to the Gohberg-Semencul formula

$$A^{-1} = \frac{1}{u_0}(L(\mathbf{u})L^T(J\mathbf{v}) - L(Z\mathbf{v})L^T(ZJ\mathbf{u}))\ ,$$

where

$$\mathbf{u} = A^{-1}\mathbf{e}^{(0)}, \mathbf{v} = A^{-1}\mathbf{e}^{(n-1)}\ .$$

Moreover, if A is also symmetric, it is easy to check that $\mathbf{v} = J\mathbf{u}$, so that

$$A^{-1} = \frac{1}{u_0}(L(\mathbf{u})L^T(\mathbf{u}) - L(ZJ\mathbf{u})L^T(ZJ\mathbf{u}))\ .$$

Since DFT of any vector $\mathbf{t}$ also gives us DFT of the vector $J\mathbf{t}$, we may now compute the product $A^{-1}\mathbf{w}$, (for given vectors $\mathbf{w}$ and $\mathbf{u}$), in 7 FFT's.

In the case of a Hermitian and positive definite Toeplitz matrix A, the known best way for the recovery of A^{-1} $\mathbf{w}$ from $A^{-1}\mathbf{e}^{(n-1)}$ is shown in [AG] (also compare our Remark 3.1). This approach is reduced to the computation of two vectors, each taking the form

$$L(\mathbf{u})C\mathbf{w}\ , \qquad (B.2)$$

where we are given a circulant matrix C and two vectors $\mathbf{u}$ and $\mathbf{w}$. The vector $C\mathbf{w}$ can be computed via three DFT's on s points for any $s \geq n$, and then the computation of $L(\mathbf{u})\mathbf{v}_*$ can be reduced to three DFT's on $2s$ points. We will show how to replace one of the latter DFT's by a DFT on s points. For simplicity, assume that $s = n$ and denote $m = 2n$, so that $C\mathbf{w}$ is the DFT of some n-dimensional vector $\mathbf{v}=(v_i)$, and we need to compute the m point DFT of this vector extended by the vector of n zeros:

$$C\mathbf{w} = (v_k^*), v_k^* = \frac{1}{n}\sum_{j=0}^{n-1}\sum_{i=0}^{n-1} v_i\omega_n^{-ij}\omega_m^{jk},\ k = 0, 1, \ldots, m-1\ .$$

We now obtain that $\omega_n = \omega_m^2$, so that

$$nw_k^* = \sum_{i=0}^{n-1} c_i v_i,\ c_i = \sum_{j=0}^{n-1}\omega_n^{(k-2i)j}\ ,$$

for all k. Therefore, $c_k = n,\ c_i = 0$ unless $i = k$; $w_k^* = v_{k/2}$ for even k, and it now remains to compute

$$w_k^* = \sum_{j=0}^{n-1}(v_j^*\omega_{2n}^j)\omega_n^{js},$$

for odd $k, k = 2s+1; s = 0, 1, \ldots, n-1$. This essentially amounts to DFT on n (rather than $2n$) points, which gives us the desired improvement of the computation of the expression (B.2).

The reader may now work out the extensions of the latter technique in order to improve the known methods, in particular, for successive multiplication of two triangular Toeplitz matrices by a vector and, consequently, for computation of the product of a Toeplitz-like matrix by a vector.

Appendix C. Correlations between F^+, F^- and the classical displacement operators.

The classical displacement operators F_+ and F_- of [KKM], [CKL-A], such that

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

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

are related to operators F^+ and F^- of sections 3 and 4 via the following equations, which hold for any $n \times n$ matrix A ([P90b]):

$$F^+(A)Z^T = F_+(A) - A\mathbf{e}^{(0)}\mathbf{e}^{(0)T}, Z^T F^+(A) = \mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T} A - F_-(A), F^-(A)Z = F_-(A) - A\mathbf{e}^{(n-1)}\mathbf{e}^{(n-1)T}, Z$$

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

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

References

[AG] G. S. Ammar, P. Gader, A Variant of the Gohberg-Semencul Formula Involving Circulant Matrices, *SIAM J. on Matrix Analysis and Its Applications*, 12, 3, 534–541, 1991.

[BC] D. Bini, M. Capovani, Spectral and Computational Properties of Band Symmetric Toeplitz Matrices, *Lin. Alg. and its Applics.*, 52/53, 99–126, 1983.

[BGP] D. Bini, L Gemignani, V. Pan, Improved parallel computations with matrices and polynomials, *Proc. 18-th Intern. Symposium on Automata, Languages and Programming, Lectures Notes in Computer Science,* **510**, 520-531, Springer 1991.

[B83] D. Bini, On a Class of Matrices Related to Toeplitz Matrices, Tech. Rep. TR 83-5, Computer Sci. Dept., SUNYA, Albany, NY, 1983.

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

[FMKL] B. Friedlander, M. Morf, T. Kailath, L. Ljung, New Inversion Formulas for Matrices Classified in Terms of their Distances from Toeplitz Matrices, *Lin. Alg. and its Applics.*, 27, 31–60, 1979.

[GS] I. C. Gohberg, A. A. Semencul, On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs, *Math. Issl.*, 2, 201–233 (in Russian), 1972.

[JJ] J. Ja Ja, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.

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

[KPa] E. Kaltofen, V. Pan, Processor Efficient Parallel Solution of Linear Systems II: Positive Characteristic and Singular Cases, *Proc. 33-rd Ann. IEEE Symp. on Foundations of Computer Science,* 1992.

[KP] E. Kaltofen, V. Pan, Processor Efficient Parallel Solution of Linear Systems over an Abstract Field, *Proc. 3-rd Ann. ACM Symp. on Parallel Computers and Architectures,* 180–191, 1991.

[KR] R. Karp, V. Ramachandran, A Survey of Parallel Algorithms for Shared Memory Computers, *Handbook for Theoretical Computer Science,* North-Holland, Amsterdam (J. van Leeuwen, ed.), 869–941, 1990.

[L] H. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*, Morgan Kauffman, San Mateo, CA, 1991.

[P] V. Pan, Concurrent Iterative Algorithm for Toeplitz-like Linear Systems, *IEEE Trans. on Parallel and Distributed Systems,* to appear.

[P90a] V. Pan, On Computations with Dense Structured Matrices, *Math. of Computation,* 55, 191, 179–190, 1990.

[P89] V. Pan, Parallel Inversion of Toeplitz and Block Toeplitz Matrices, *Operator Theory: Advances and Applics.*, 40, 359–389, Birkhauser, Boston, 1989.

[P90] V. Pan, Parallel Least-Squares Solution of General and Toeplitz-like Linear Systems, *Proc. 2-nd Ann. ACM Symp. on Parallel Algorithms and Architectures*, 244–253, 1990.

[P92] V. Pan, Parallel Solution of Toeplitzlike Linear Systems, *J. Complexity,* 8, 1–21, 1992.

[P90b] V. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, Tech. Rep. CUCS-031-90, *Columbia Univ., Computer Sci. Dept.*, NY, 1990.

[PP] V. Pan, F. Preparata, Supereffective Slow-down of Parallel Computations, *Proc. 4-th Ann. ACM Symp. on Parallel Algorithms & Architectures*, 402–409, 1992.

[Pe] M. Pease, An Adaptation of the Fast Fourier Transform for Parallel Procelssing, *J. ACM*, 15, 252–284, 1968.

[PFTV] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge 1986.

[Q] M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers,* McGraw-Hill, New York, 1987.

[S] A. Schönhage, The Fundamental Theorem of Algebra in Terms of Computational Complexity, unpublished manuscript, 1982.

[T] W. F. Trench, A Note on a Toeplitz Inversion Formula, *Linear Alg. and its Applics.*, 129, 55–61, 1990

[B83] D. Bini, On a Class of Matrices Related to Toeplitz Matrices, Tech. Rep. TR 83-5, Computer Sci. Dept., SUNYA, Albany, NY, 1983.

[BC] D. Bini, M. Capovani, Spectral and Computational Properties of Band Symmetric Toeplitz Matrices, *Lin. Alg. and its Applics.*, 52/53, 99–126, 1983.

[BGP] D. Bini, L. Gemignani, V. Pan, Improved Parallel Computations with Matrices and Polynomials, *Proceedings 18th Intern. SYMP¿ on Automata, Llanguages and Programming, Lecture Notes in Computer Science*, 520-531, Springer, 1991.

[GS] I. C. Gohberg, A. A. Semencul, On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs, *Math. Issl.*, 2, 201–233 (in Russian), 1972.

[FMKL] B. Friedlander, M. Morf, T. Kailath, L. Ljung, New Inversion Formulas for Matrices Classified in Terms of their Distances from Toeplitz Matrices, *Lin. Alg. and its Applics.*, 27, 31–60, 1979.

[KP] E. Kaltofen, V. Pan, Processor Efficient Parallel Solution of Linear Systems over an Abstract Field, *Proc. 3-rd Ann. ACM Symp. on Parallel Computers and Architectures,* 180–191, 1991.

[KPa] E.Kaltofen, V.Pan, Processor Efficient Parallel Solution of Linear Systems II: Positive Characteristic and Singular Cases, *Proc. 33-rd Ann. IEEE Symp. on Foundations of Computer Science,* 1992.

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

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

[KR] R. Karp, V. Ramachandran, A Survey of Parallel Algorithms for Shared Memory Computers, *Handbook for Theoretical Computer Science,* North-Holland, Amsterdam (J. van Leeuwen, ed.), 869–941, 1990.

[JJ] J. Ja Ja, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.

[L] H. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*, Morgan Kauffman, San Mateo, CA, 1991.

[Pe] M. Pease, An Adaptation of the Fast Fourier Transform for Parallel Procelssing, *J. ACM*, 15, 252–284, 1968.

[Q] M. J. Quinn, *Designing Efficient Algorithms for Parallel Computers,* McGraw-Hill, New York, 1987.

[P] V. Pan, Concurrent Iterative Algorithm for Toeplitz-like Linear Systems, *IEEE Trans. on Parallel and Distributed Systems,* to appear.

[T] W. F. Trench, A Note on a Toeplitz Inversion Formula, *Linear Alg. and its Applics.*, 129, 55–61, 1990.

[P89] V. Pan, Parallel Inversion of Toeplitz and Block Toeplitz Matrices, *Operator Theory: Advances and Applics.*, 40, 359–389, Birkhauser, Boston, 1989.

[P90] V. Pan, Parallel Least-Squares Solution of General and Toeplitz-like Linear Systems, *Proc. 2-nd Ann. ACM Symp. on Parallel Algorithms and Architectures*, 244–253, 1990.

[P90a] V. Pan, On Computations with Dense Structured Matrices, *Math. of Computation,* 55, 191, 179–190, 1990.

[P90b] V. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, Tech. Rep. CUCS-031-90, *Columbia Univ., Computer Sci. Dept.*, NY, 1990.

[P92] V. Pan, Parallel Solution of Toeplitzlike Linear Systems, *J. Complexity,* 8, 1–21, 1992.

[PP] V. Pan, F. Preparata, Supereffective Slow-down of Parallel Computations, *Proc. 4-th Ann. ACM Symp. on Parallel Algorithms & Architectures*, 402–409, 1992.

[S] A. Schönhage, The Fundamental Theorem of Algebra in Terms of Computational Complexity, unpublished manuscript, 1982.

$F^+(A)Z^T = F_+(A) - Ae^{(0)}e^{(0)T}, Z^T F^+(A) = e^{(n-1)}e^{(n-1)T}A - F_-(A), F^-(A)Z = F_-(A) - Ae^{(n-1)}e^{(n-1)T}$