The subsequent computations, their analysis and the complexity estimates remain valid (and we may actually decrease the overhead constants by choosing larger h in section 6 such that $1/h = O(1)$), since instead of Fact 4.1, we may use Proposition 3.4 and the following simple result (in order to estimate $\|X - X_k\|_1$ given $\|X - Y_k\|_1$):

Proposition B.1. *Let a, b, x, y, v, ϵ be scalars, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{u}$ and $\mathbf{v}$ be four vectors of the same dimension such that $ab \neq 0$, $\|\mathbf{x}\|_1 = x$, $\|\mathbf{y}\|_1 = y$, $\|\mathbf{v}\|_1 = v$, $|a - b| < \epsilon$, $\|\mathbf{x} - \mathbf{u}\|_1 < \epsilon$, $\|\mathbf{y} - \mathbf{v}\|_1 < \epsilon$. Then*

$$\begin{aligned}&\|\frac{1}{a}L(\mathbf{x})\,L(\mathbf{y}) - \frac{1}{b}L(\mathbf{u})\,L(\mathbf{v})\|_1\\ &\le \epsilon(\frac{xy}{ab} + \frac{x+v}{b})\ .\end{aligned}$$

Proof.

$$\begin{aligned}&\|\frac{1}{a}L(\mathbf{x})L(\mathbf{y}) - \frac{1}{b}L(\mathbf{u})L(\mathbf{v})\|_1\\ &\le \|\frac{1}{a}L(\mathbf{x})L(\mathbf{y}) - \frac{1}{b}L(\mathbf{x})L(\mathbf{y})\|_1\\ &+ \|\frac{1}{b}L(\mathbf{x})L(\mathbf{y}) - \frac{1}{b}L(\mathbf{x})L(\mathbf{v})\|_1\\ &+ \|\frac{1}{b}L(\mathbf{x})L(\mathbf{v}) - \frac{1}{b}L(\mathbf{u})L(\mathbf{v})\|_1\\ &\le |\frac{1}{a} - \frac{1}{b}|\ \|L(\mathbf{x}\|_1\ \|L(\mathbf{y})\|_1\\ &+ \frac{1}{b}\|L(\mathbf{x})\|_1\|L(\mathbf{y}) - L(\mathbf{v})\|_1\\ &+ \frac{1}{b}\|L(\mathbf{v})\|_1\|L(\mathbf{x}) - L(\mathbf{u})\|_1\\ &\le \epsilon(xy/(ab) + x/b + v/b)\ .\end{aligned}$$

□

We only need to apply (3.4) to matrices Y approximating to the matrices $A_g^{-1} = (I - h_g A)^{-1}$, for small $|h_g|$, and this immediately implies upper bounds $O(1)$ on x, y, $1/a$ and $1/b$ in the subsequent applications of Proposition B.1.

is an unknown auxiliary polynomial. We observe the following identities, immediately implied by the equations (A.2) and (A.3):

$$h'_r(\lambda)\,/\,(1+h_r(\lambda)\,) = h'_r(\lambda) \bmod \lambda^{2r+1}\ ,$$

$$(\ell n(1+g_{2r}(\lambda)))' = \frac{g'_{2r}(\lambda)}{1+g_{2r}(\lambda)} =$$
$$\frac{g'_r(\lambda)}{1+g_r(\lambda)} + \frac{h'_r(\lambda)}{1+h_r(\lambda)} \bmod \lambda^{2r+1}\ .$$

Combining these identities with the equation (A.1) for $k = 2r+1$, we obtain that

$$\begin{aligned} &\frac{g'_r(\lambda)}{1+g_r(\lambda)} + h'_r(\lambda) \\ &= -\sum_{j=1}^{2r+1} s_j \lambda^{j-1} \bmod \lambda^{2r+1}\ . \end{aligned} \tag{A.4}$$

We assumed that we know the coefficients of $g_r(\lambda)$, as well as the values s_j for all $j \le 2r+1$. Now, we compute the polynomial $1/(1+g_r(\lambda))$ modulo λ^{2r+1}, multiply it by $g'_r(\lambda)$ modulo λ^{2r+1}, subtract the result from the right side of the equation (A.4) to obtain $h'_r(\lambda)$, and compute first $h_r(\lambda)$, by using the equation (A.3), and then $g_{2r}(\lambda)$, by using the equation (A.2). We start this recursive evaluation of $g_{2r}(\lambda)$ with $r = 1$ and $r = 2$, for the polynomials $g_1(\lambda)$ and $g_2(\lambda)$ are readily available. Then we recursively compute the polynomials $g_4(\lambda)$, $g_8(\lambda)$, $g_{16}(\lambda)$, and so on, until this procedure yields the desired coefficient vector $\mathbf{c}$ of the characteristic polynomial of A. We recursively use each reciprocal polynomial $1/(1+g(\lambda)) \bmod \lambda^{r+1} = 1/(1+g_r(\lambda)) \bmod \lambda^{r+1}$ when we compute modulo λ^{2r+1} the reciprocal of $1+g(\lambda)$ in the next iteration, so, the overall cost of the computation is $O_A(\log^2 n, n/\log n)$.

Appendix B. Inversion of Toeplitz Matrices. For a Toeplitz matrix A, we may modify the iteration (5.2) and let $X_k = X(Y_k)$ where $X(Y)$ is defined by (3.4). This way we may dispense with Algorithm 4.1 and avoid the computation of the SVDs, thus simplifying the entire computation (compare [PR], [P89]).

Appendix A. From Power Sums of the Eigenvalues to the Characteristic Polynomial. We will assume that we are given the vector

$$\mathbf{s} = [s_1, s_2, \ldots, s_n]^T$$

of the first n power sums $s_k = \text{ trace}(A^k)$, $k = 1, 2, \ldots, n$, of all the eigenvalues of an $n \times n$ matrix A and will compute the coefficient vector

$$\mathbf{c} = [c_{n-1}, \ldots, c_0]^T$$

of the characteristic polynomial of A,

$$\det(\lambda I - A) = \lambda^n + \sum_{i=0}^{n-1} c_i \lambda^i = \prod_{j=1}^{n} (\lambda - \lambda_j) \ ,$$
$$c_0 = (-1)^n \ \det \ A \ .$$

Denote the reverse characteristic polynomial of A as follows:

$$1 + g(\lambda) = \lambda^n \ \det(I - (1/\lambda)A)$$
$$= 1 + \sum_{i=1}^{n} c_{n-i} \lambda^i = \prod_{j=1}^{n} (1 - \lambda \, \lambda_j) \ ,$$

and deduce that

$$(\ell n(1 + g(\lambda)))' = g'(\lambda)/(1 + g(\lambda)) =$$
$$- \sum_{j=1}^{n} \lambda_j / (1 - \lambda \, \lambda_j) = - \sum_{j=1}^{k} s_j \lambda^{j-1} \ \text{mod} \ \lambda^k \qquad (A.1)$$

for all $k \leq n + 1$. Suppose that the polynomial $g_r(\lambda) = g(\lambda) \text{mod } \lambda^{r+1}$ is available and that we want to compute $g_{2r}(\lambda) = g(\lambda) \text{mod } \lambda^{2r+1}$. (The polynomials $g_1(\lambda) = c_{n-1}\lambda$ and $g_2(\lambda) = c_{n-1}\lambda + c_{n-2}\lambda^2$ are readily available, for $c_{n-1} = -s_1$, $2c_{n-2} = s_1^2 - s_2$.) We will express $g_{2r}(\lambda)$ as follows:

$$1 + g_{2r}(\lambda) = (1 + g_r(\lambda))(1 + h_r(\lambda)) \ \text{mod} \ \lambda^{2r+1} \ , \qquad (A.2)$$

where

$$h_r(\lambda) = h_{r+1} \lambda^{r+1} + \ldots + h_{2r} \lambda^{2r} \qquad (A.3)$$

Baltimore, Maryland, 1989.

[GP] Z. Galil and V. Pan, "Parallel Evaluation of the Determinant and of the Inverse of a Matrix," *Inform. Proc. Lett.*, vol. 30, pp. 41-45, 1989.

[vzG] J. von zur Gathen, "Parallel Algorithms for Algebraic Problems," *SIAM J. on Comp.*, vol. 13,4, pp. 802-824, 1984.

[GSe] I. C. Gohberg and A. A. Semencul, "On the Inversion of Finite Toeplitz Matrices and their Continuous Analogs," *Mat. Issled.*, vol. 2, pp. 201-233 (in Russian), 1972.

[Kai] T. Kailath, "Signal Processing Applications of Some Moment Problems," *Proc. AMS Symp. in Applied Math.*, vol. 37, pp. 71-100, 1987.

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

[PR] V. Pan and J. Reif, "Some Polynomial and Toeplitz Matrix Computations," *Proc. 28th Ann. IEEE Symp. FOCS*, pp. 173-181, 1987.

[P88] V. Pan, "New Effective Methods for Computations with Toeplitz-like Matrices," *Technical Report 88-28, Computer Science Dept., SUNY Albany*, 1988.

[P89] V. Pan, "Fast and Efficient Parallel Inversion of Toeplitz and Block Toeplitz Matrices," *Operator Theory; Advances and Applications,* vol. 40, pp. 359-389, 1989.

[P89a] V. Pan, "On Some Computations with Dense Structured Matrices," *Proc. ACM-SIGSAM Symposium ISSAC-89*, pp. 34-42, 1989 (to appear in *Math. of Computations*).

[PS] F. P. Preparata and D. V. Sarwate, An improved parallel processor bound in fast matrix inversion, *Inform. Proc. Lett.* vol. 7 (3) (1978) 148-150.

[Sc] A. Schönhage, "The Fundamental Theorem of Algebra in Terms of Computational Complexity," manuscript, *Dept. of Math., University of Tübingen*, Tübingen, West Germany, 1982.

[T64] W. F. Trench, "An Algorithm for Inversion of Finite Toeplitz Matrices," *J. of SIAM*, vol. 12,3, pp. 515-522, 1964.

which are the k-th power sums of the eigenvalues of A. Then, we will compute the vector $\mathbf{c} = [c_{n-1}, \ldots, c_0]^T$ of the coefficients of the characteristic polynomial of A [see (2.4)] for the cost of $O_A(\log^2 n, n/\log n)$ by means of an algorithm of [Sc], which we complete and reproduce in Appendix A. This also gives us det A and $A^+\mathbf{b}$ for the desired computational cost of $O_A(\log^2 n, n^2)$, due to (2.5), Corollary 2.2 and Fact 2.2. The vectors $A^k\mathbf{b}$ for $k = 1, \ldots, n$, required for computing $A^+\mathbf{b}$, are obtained from the vectors $(I - h_g A)^{-1}\mathbf{b}$ by using the approach of the previous section.

Acknowledgements. Supported by NSF Grant CCR-8805782 and by PSC-CUNY Awards #668541 and #669290.

References

[BG] D. Bini and L. Gemignani, "On the Euclidian Scheme for Polynomials," *Proc. 2nd Ann. ACM SPAA*, 1990.

[BA] R. R. Bitmead and B. D. O. Anderson, "Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations," *Linear Algebra and its Applics.*, vol. 34, pp. 103-116, 1980.

[BGH] A. Borodin, J. von zur Gathen, and J. Hopcroft, "Fast Parallel Matrix and GCD Computation," *Information and Control*, vol. 52,3, pp. 241-256, 1982.

[CKL-A] J. Chun, T. Kailath, and H. Lev-Ari, "Fast Parallel Algorithm for QR-factorization of Structured Matrices," *SIAM J. on Scientific and Statistical Computing*, vol. 8,6, pp. 899-913, 1987.

[Cs] L. Csanky, "Fast Parallel Matrix Inversion Algorithm," *SIAM J. on Computing,* vol. 5,4, pp. 618-623, 1976.

[FKML] 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 and its Applics.*, vol. 27, pp. 31-60, 1979.

[GL] G. H. Golub, and C. F. van Loan, *Matrix Computations*, Johns Hopkins Univ. Press,

approximate to A^k for $k = 2, 3, \dots, n$ within the absolute error norms

$$\begin{aligned} e_{k,m} &= h^m \|A^{k+m}(I - (hA)^m)^{-1}\|_2 \\ &< 2h^m \|A\|_2^{1.5m} \le 2(4n^2\|A\|_2)^{-n} \ . \end{aligned} \tag{6.2}$$

Since we already know approximations to the displacement generators of lengths at most $r = O(1)$ for A_g^{-1} for $g = 0, 1, \dots, m-1$, we may approximate to any fixed entry (i, j) of $A(k)$ for $k = 1, 2, \dots, n$ for the cost of FFT at m points, that is, for the cost of $O_A(\log n, n)$. This will give us an approximation to the (i, j) entry of A^k for all $k \le n$ within errors at most

$$E = m\, e_m + e_{k,m} \tag{6.3}$$

(compare (6.1), (6.2)), which is clearly less than $1/2$, and thus, we may obtain the exact value of this entry of A^k by means of the round-off, provided that A is filled with integers. (If A is filled with rational numbers, we may scale A by a common multiple of their denominators.)

We may similarly compute all the diagonal entries of the matrices A^k and all the components of the vectors $A^k\mathbf{b}$ for $k = 1, 2, \dots, n$, given displacement generators of lengths $O(1)$ for A_g^{-1} for $g = 0, 1, \dots, m-1$ and a vector $\mathbf{b}$ filled with integers and such that $2E\|\mathbf{b}\|_2 < 1$ (compare (6.3)). The cost bound increases to $O_A(\log n, n^2)$ for these computations but is still dominated by the bound $O_A(\log^2 n, n^2)$ on the cost of approximating to the generators of lengths $O(1)$ for A_g^{-1} for $g = 0, 1, \dots, m-1$.

Remark 6.1. We also observe that the precision of $O(n \log(n\|A\|_2) + \log \|\mathbf{b}\|_2)$ binary bits suffices in all the above computations, which is roughly the output precision, and the term $\log \|\mathbf{b}\|_2$ can be deleted where we compute the diagonal entries.

7. Computing the Characteristic Polynomial of a Toeplitz-like Matrix and a Least-Squares Solution of a Toeplitz-like Linear System: Summing the diagonal entries of A^k (available due to the previous section), we compute

$$s_k = \text{ trace}(A^k) \ , \quad k = 1, 2, \dots, n \ ,$$

$\|A_g^{-1}\|_2 \le 2$, $\mathrm{cond}_2\ A_g \le 3$) and

$$3\big(1+2(2r+3)n\big)\left(|h_g|\,\|A\|_2\right)^{1-\nu} \le 1\ ,$$

$$\text{for a positive } \nu < 1\ .$$

We will assume that $r = O(1)$ and will choose h_g satisfying these two bounds and such that, moreover,

$$h_g = h\omega^g\ ,\quad g = 0, 1, \ldots, m-1\ ,$$

where ω is a primitive m-th root of 1, $\omega^m = 1$, $\omega^s \ne 1$ for $0 < s < m$, where

$$O < h < 1/(2n\,\|A\|_2^2)\ .$$

(To bound $\|A\|_2$, we may use (2.1).) Then $\|I - A_g\|_2 = h\|A\|_2 \le 1/(2n\,\|A\|_2)$, so that we may apply Proposition 5.1 (with A_g replacing A) and thus approximate to A_g^{-1}, for all g, within the error norm

$$e_m = 2\big(2n\,\|A\|_2\big)^{-m} \tag{6.1}$$

in $O(\log n)$ steps (5.2), where we set $X_0 = I$ and replace A by A_g.

Now, observe that

$$\begin{aligned}\omega^{-gk}A_g^{-1} &= \sum_{s=0}^{\infty}\omega^{-gk}(h_gA)^s\\ &= \sum_{s=0}^{\infty}\omega^{g(s-k)}(hA)^s\ .\end{aligned}$$

Therefore,

$$\begin{aligned}\sum_{g=0}^{m-1}\omega^{-gk}A_g^{-1} &= \sum_{s=0}^{\infty}\Big(\sum_{g=0}^{m-1}\omega^{g(s-k)}\Big)(hA)^s\\ &= m\sum_{q=0}^{\infty}(hA)^{k+mq}\\ &= m(hA)^k(I+(hA)^m(I-(hA)^m)^{-1})\ .\end{aligned}$$

Thus, the matrices

$$A(k) = (mh^k)^{-1}\sum_{g=0}^{m-1}\omega^{-gk}A_g^{-1}$$

Thus, we may replace $\rho(0)$ in (5.5) by $\rho(1)$, then recursively repeat applying the same argument and deduce that

$$\rho(k) \le (\rho(k-1))^{1+\nu} \le (\rho(0))^{(1+\nu)^k} ,$$
$$k = 1, 2, \dots .$$

This bound immediately implies (5.8).

6. Extensions to Approximate Matrix Inversion and to Computing Matrix Powers: To apply Proposition 5.1, we need to have X_0 satisfying (5.1) and (5.5), that is, we need a good initial approximation to A^{-1} having a low displacement rank. In some cases, it is easy to precompute such an X_0, for instance, by setting $X = B^{-1}$ if A is close to a band Toeplitz matrix B (as this was pointed out to me by E. Linzer). Generally, we may obtain the desired X_0 by using some additional techniques, and we should include the cost of computing X_0 into the overall cost estimates. This way, an algorithm of [P88] computes, for the cost of $O_A(t,p)$, a displacement generator of length $O(1)$ for an approximation $\tilde{A}^{-1}$ to A^{-1} such that $\|\tilde{A}^{-1} - A^{-1}\|_2 \le 2^{-d}\|A^{-1}\|_2$ for a fixed d, provided that A is given with its displacement generator of length $O(1)$. Here t and p depend on d and $\mathrm{cond}_2\ A$, and we have that $t = (\log n \log\log n)^2$, $p = n \log n$ if $\log d = O(\log n)$, $\log\ \mathrm{cond}_2\ A = O(\log n)$, which means an NC solution with the order of n processors. The algorithm, however, becomes inefficient and slow as $\mathrm{cond}_2\ A$ grows large, provided that A is filled with integers or rational numbers, in which case we have a superior algorithm. We will next present this algorithm and prove the estimate $O_A(\log^2 n, n^2)$ for its cost.

We will start with computing the traces, that is, the sums of the diagonal entries, of the matrices $A, A^2, \dots, A^n$, as well as the products of these matrices by a vector.

Observe that (5.5) holds for the auxiliary matrices

$$A_g = I - h_g A , \quad g = 0, 1, \dots, m-1, \quad m = 2n ,$$

provided that $r = 1 +$ rank $F_{ZZ^T}(A^{-1})$, $X_0 = I$, $|h_g|\,\|A\|_2 \le \frac{1}{2}$ (so that $\|A_g\|_2 \le 3/2$,

Proposition 5.1. *Let* (5.1) *and* (5.2) *hold, and let*

$$\big(1+2(r+r^*+3)n\big)\ (\mathrm{cond}_2 A)\,\rho(0)^{1-\nu} \le 1 \tag{5.5}$$

for $\rho(0)$ *of* (5.3) *and for some fixed positive* $\nu < 1$. *Then, for* $k = 0, 1, \ldots$, *we have:*

$$\mathrm{rank}\ F_{ZZ^T}(X_k) \le r\ , \tag{5.6}$$

$$\mathrm{rank}\ F_{ZZ^T}(Y_k) \le 2r + r^* + 3\ , \tag{5.7}$$

$$\|X_k - A^{-1}\|_2 \le \rho(0)^{(1+\nu)^k}\|A^{-1}\|_2\ , \tag{5.8}$$

and the matrices Y_1, X_1, Y_2, X_2, ..., Y_k, X_k *can be computed for the overall cost of* $O_A(k\log n, n)$.

Proof. (5.6) is implicit in the definition of X_k and Algorithm 4.1; (5.7) follows from Proposition 3.2, which, in combination with (5.7) and with the complexity estimates for Algorithm 4.1, also implies the estimate $O_A(k\log n, n)$ of Proposition 5.1.

It remains to prove (5.8). Apply (5.4) for $k = 0$ and obtain that $\|I - AY_1\|_2 \le (\rho(0))^2$,

$$\|A^{-1} - Y_1\|_2 \le \|A^{-1}\|_2(\rho(0))^2\ .$$

Substitute the bound of Fact 4.1 (for $B_r = Y_1$, $R \le 2r + r^* + 3$ and $X = A^{-1}$) and deduce that

$$\|A^{-1} - X_1\|_2 \le \|A^{-1}\|_2\big(1+2(r+r^*+3)n\big)(\rho(0))^2\ .$$

Therefore [see (2.2)],

$$\rho(1) \le (\mathrm{cond}_2 A)\,(1+2(r+r^*+3)n)\,(\rho(0))^2\ .$$

Now, substitute (5.5) and deduce that

$$\rho(1) \le (\rho(0))^{1+\nu} < \rho(0)\ .$$

for all i [note that $F_{ZZ^T}(B) = W$].

Combining (4.4)-(4.6), we deduce that

$$\sum_{i=r+1}^{R} \sigma_i^2(W) \le 2(R-r)\Delta \ .$$

Substitute the latter bound in (4.3) and deduce that

$$\begin{aligned} \|X - B_r\|_2 &\le \|X - B\|_2 + \|B - B_r\|_2 \\ &\le \big(1 + 2(R-r)n\big)\Delta \ , \end{aligned}$$

which proves Fact 4.1. $\square$

5. Modified Newton Iteration for the Inversion of Toeplitz-like Matrices: In this section, we will assume that an $n \times n$ nonsingular matrix A and an initial approximation X_0 to $X = A^{-1}$ have been given with their F_{ZZ^T}-generators of lengths r^* and r_0^*, respectively, such that

$$\begin{aligned} &r_0^* \le r^* \le r \le r^* + 2 = O(1) \ , \\ &r \ge \text{ rank } F_{ZZ^T}(A^{-1}) \ . \end{aligned} \tag{5.1}$$

We will recursively define matrices Y_1, X_1, Y_2, X_2, ... as follows:

$$Y_{k+1} = X_k(2I - AX_k) \ , \quad k = 0, 1, \ldots \tag{5.2}$$

where X_k for $k \ge 1$ denotes the output matrix of Algorithm 4.1 applied to $B = Y_k$, $W = F_{ZZ^T}(Y_k)$ and r of (5.1). [Actually, (5.2) is a modified Newton iteration for the inversion of A.]

We will denote hereafter that

$$\rho(k) = \|I - AX_k\|_2 \ , \quad \text{for all } k \ , \tag{5.3}$$

and will deduce from (5.2) that $I - AY_{k+1} = (I - AX_k)^2$, so that

$$\|I - AY_{k+1}\|_2 \le (\rho(k))^2 \ . \tag{5.4}$$

Stage 2. Compute and output the desired F_{ZZ^T}-generator $G = U_r\Sigma_r$, $H = V_r\Sigma_r$ of length r for the matrix B_r of (3.3). [G, H is a generator for $W_r = U_r\Sigma_r^2V_r^H = F_{ZZ^T}(B_r)$; compare Proposition 3.1.]

The complexity of these computations is dominated by the complexity of stage 1, at which we start with a generator M, N of length at most R for W and, for the cost of $O_A(\log n, n/\log n)$, reduce computing SVD of $W = MN^T$ to computing SVD of an $R \times R$ matrix. The reduction is by means of a recursive orthogonal factorization of W (from pages 198-199 of [GL]); here and hereafter, we will assume for simplicity that $R = O(1)$, so that $r = O(1)$ too. In the remainder of this section, we will estimate Δ_r.

Fact 4.1. $\Delta_r \le (1 + 2n(R-r))\Delta$ (see (4.1), (4.2)).

Proof. We defined B_r in the form (3.3) at stage 2 of Algorithm 4.1; we now similarly represent B and deduce that

$$\begin{aligned} \|B - B_r\|_2 = \|\sum_{i=r+1}^{R} L(\mathbf{g}_i)\, L^T(\mathbf{h}_i)\|_2 \le \\ \sum_{i=r+1}^{R} \|L(\mathbf{g}_i)\|_2 \|L^T(\mathbf{h}_i)\|_2 \le n \sum_{i=r+1}^{R} \sigma_i^2(W) \end{aligned} \tag{4.3}$$

where $W = F_{ZZ^T}(B)$. The latter inequality follows because

$$\|\mathbf{g}_i\|_2 = \|\mathbf{h}_i\|_2 = \sigma_i(W)\ , \quad \text{for } i = 1, \dots, R$$

[compare (2.6) for $A = W$ and (3.3)].

Now, we observe that, surely,

$$\|F_{ZZ^T}(X) - F_{ZZ^T}(B)\|_2 \le 2\|X - B\|_2 = 2\Delta\ , \tag{4.4}$$

whereas Proposition 3.1 implies that

$$\sigma_i(F_{ZZ^T}(X)) = 0\ , \quad \text{for} \quad i > r\ . \tag{4.5}$$

On the other hand, we may deduce from an estimate on page 428 of [GL] that

$$\begin{aligned} |\sigma_i^2(F_{ZZ^T}(X)) - \sigma_i^2(F_{ZZ^T}(B))| \\ \le \|F_{ZZ^T}(X) - F_{ZZ^T}(B)\|_2\ , \end{aligned} \tag{4.6}$$

of X, respectively, x_0 denotes the first component of $\mathbf{x}$; the matrices J, Z and $L(\mathbf{v})$ be defined by Definitions 3.1 and 3.2. Then

$$x_0 X = L(\mathbf{x})L^T(J\mathbf{y}) - L(Z\mathbf{y})L^T(JZ\mathbf{x}) \ .$$

Note that for an arbitrary matrix Y, we may let $\mathbf{x}$, $\mathbf{y}$ and x_0 denote its first column, its last column, and the first component of $\mathbf{x}$, respectively, and then, if $x_0 \neq 0$, define

$$\begin{aligned} X(Y) = (1/x_0)\,(L(\mathbf{x})\,L^T(J\mathbf{y}) \\ - L(Z\mathbf{y})\,L^T(JZ\mathbf{x})) \ . \end{aligned} \tag{3.4}$$

Proposition 3.4 states that $x_0 Y = x_0 X(Y)$ if Y is the inverse of a Toeplitz matrix.

4. Approximation by a Matrix of a Bounded Displacement Rank: Let X be an unknown $n \times n$ matrix such that rank $F_{ZZ^T}(X) \leq r$ for a fixed r [say, let $X = A^{-1}$ for a given $n \times n$ nonsingular matrix A such that rank $F_{Z^TZ}(A) \leq r$], let an approximation B to X have been computed with its F_{ZZ^T}-generator of length $R > r$, and let

$$\Delta = \|X - B\|_2 \ . \tag{4.1}$$

We want to compute a matrix B_r with its F_{ZZ^T}-generator of length at most r and to estimate

$$\Delta_r = \|X - B_r\|_2 \tag{4.2}$$

as a function in Δ, n and $R - r$, choosing B_r so as to keep Δ_r smaller.

We will apply the following algorithm to compute the desired matrix B_r given a generator of length R for the matrix $W = F_{ZZ^T}(B)$ and a natural $r \leq R$:

Algorithm 4.1.

Stage 1. Compute the SVD of the matrix $W = F_{ZZ^T}(B)$ [compare (2.6) for $A = W$]. Let U_r and V_r denote the pair of $k \times r$ matrices formed by the first r columns of the matrices U and V of the SVD and let

$$\Sigma_r = \mathrm{diag}(\sigma_1(W), \ldots, \sigma_r(W)) \ .$$

Due to Corollary 3.1, we may immediately define an F_{ZZ^T}- (respectively, an F_{Z^TZ}-) generator of length $r+2$ for a matrix A given its F_{Z^TZ}- (respectively, its F_{ZZ^T}-) generator of length r.

It is economical to represent the matrices of smaller displacement ranks r by using the $2rn$ entries of their displacement generators of length r, and *we will hereafter identify computing a matrix with computing its displacement generator.*

Furthermore, we recall that multiplication of an $n \times n$ Toeplitz matrix by a vector is reduced to performing three discrete Fourier transforms at $O(n)$ points for the overall cost of $O_A(\log n, n)$. Due to Proposition 3.1, we may extend this bound to the bound $O_A(\log n, rn)$ on the complexity of computing the product of an $n \times n$ matrix by a vector, provided that the matrix has been given with its displacement generator of length r.

Finally, we will recall the following results:

Proposition 3.2. *Given a pair of F-generators of lengths a and b for a pair of $n \times n$ matrices A and B, we may immediately obtain an F-generator of length at most $a+b$ for $A + \alpha B$* (for any fixed scalar α); *furthermore, if F is a displacement operator of* (3.1) *or* (3.2), *we may also compute [for the cost of $O_A(\log n, abn)$] F-generators of lengths at most $a+b+1$ for AB and A^HA* (see [CKL-A], [P89a]).

Proposition 3.3 ([KKM]). *If A is a nonsingular matrix, then*

$$\text{rank } F_{ZZ^T}(A^{-1}) = \text{ rank } F_{Z^TZ}(A) \ .$$

Note that the latter result expresses no displacement generator of A^{-1} through one of A.

If A is a nonsingular Toeplitz matrix, we may apply the Gohberg-Semencul formulae to express its inverse X through a pair of columns of X (but we still need to compute these columns):

Proposition 3.4 (see [FKML], [GSe] and [T64] for the proofs and extensions). *Let T be a nonsingular Toeplitz matrix, $X = T^{-1}$, $\mathbf{x}$ and $\mathbf{y}$ denote the first and the last columns*

are called the *displacement* (or shift) *operators*; the associated F-ranks and F-generators are called *displacement ranks* and *displacement generators.*

Toeplitz matrices A can be defined by their property that all the entries of $F_{ZZ^T}(A)$ and of $F_{Z^TZ}(A)$ are zeros, except that the matrix $F_{ZZ^T}(A)$ shares its first row and column with A and the matrix $F_{Z^TZ}(A)$ shares its last row and column with A, so that rank$F(A) \leq 2$ if A is a Toeplitz matrix and if F is a displacement operator of (3.1) or (3.2); furthermore, for such operators F, if A is an $m \times n$ block matrix with Toeplitz blocks, then rank $F(A) \leq m + n$.

Definition 3.3. $L(\mathbf{v})$ denotes the lower triangular Toeplitz matrix with the first column $\mathbf{v}$.

Proposition 3.1 ([KKM], [CKL-A]). *A pair (G, H) of $n \times r$ matrices $G = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_r]$ and $H = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_r]$ is an F-generator of length r for an $n \times n$ matrix B_r if and only if*

$$B_r = \sum_{i=1}^{r} L(\mathbf{g}_i)\, L^T(\mathbf{h}_i) \tag{3.3}$$

provided that $F(B_r) = F_{ZZ^T}(B_r)$ (see (3.1)) and if and only if

$$B_r = \sum_{i=1}^{r} L^T(J\mathbf{g}_i)\, L(J\mathbf{h}_i)$$

provided that $F(B_r) = F_{Z^TZ}(B_r)$ [see (3.2)].

Corollary 3.1 (see [BA], Lemma 5). *For any pair of vectors $\mathbf{g}$ and $\mathbf{h}$ of the same dimension,*

$$L(\mathbf{g})L^T(\mathbf{h}) = L^T(\mathbf{a}) + L(Z^TZ\mathbf{b})$$
$$- L^T(ZJ\mathbf{g})\, L(ZJ\mathbf{h})\ ,$$
$$L^T(\mathbf{g})\, L(\mathbf{h}) = L^T(\mathbf{c}) + L(ZZ^T\mathbf{d})$$
$$- L(ZJ\mathbf{g})\, L^T(ZJ\mathbf{h})$$

where J and Z are the matrices of Definition 3.1, $\mathbf{a}^TJ$ is the last row and $J\mathbf{b}$ is the last column of $L(\mathbf{g})\, L^T(\mathbf{h})$, $\mathbf{c}^T$ is the first row and $\mathbf{d}$ is the first column of $L^T(\mathbf{g})\, L(\mathbf{h})$.

Multiply this matrix equation by

$(A^+)^{n-r}$, apply Fact 2.1 and deduce (2.7). Multiply the matrix equation (2.7) by A^+, again apply Fact 2.1 and deduce (2.8). $\square$

Fact 2.2. $A^+ = A(A^HA)^+$.

Corollary 2.3. *Given a $g \times h$ matrix A, the matrix A^+ can be computed for the cost of $O_A(\log^2 n, n^{\beta+0.5-\delta})$ where $n = \max\{g,h\}$, β is the exponent of matrix multiplication, $2 \le \beta < 2.38$, and δ is a positive value defined in* [GP], $\beta + 0.5 - \delta < 2.851$.

Corollary 2.3 immediately follows if we combine the algorithm of [GP] (improving [Cs] and [PS]) with Corollary 2.2 and Fact 2.2, and we immediately observe that *the estimate of Corollary 2.3, together with the sequential complexity bound of $O_A(n^\beta \log n, 1)$, apply to computing the least-squares solution $A^+\mathbf{b}$ to any linear system $A\mathbf{x} = \mathbf{b}$.*

3. Displacement Generators and Displacement Ranks: Next, we will recall the concepts and some properties of the displacement generators and the displacement ranks of matrices. Here, we will follow [KKM], [KVM], [CKL-A], [BA] and [P89].

Definition 3.1. J is the reversion matrix, filled with zeros except for its main antidiagonal, filled with ones. Z is the lower displacement matrix, filled with zeros except for its first subdiagonal, filled with ones, so that

$$J\mathbf{v} = [v_n, \dots, v_0]^T \ ,$$

$$Z\mathbf{v} = [0, v_0, \dots, v_{n-1}]^T \ ,$$

for a vector $\mathbf{v} = [v_0, \dots, v_n]^T$.

Definition 3.2. For a matrix $A \in \mathbf{C}_{m,n}$ and a linear operator F defined on $\mathbf{C}_{m,n}$, generators and the rank of $F(A)$ are called *F-generators* and *F-rank* of A, respectively.

The linear operators

$$F_{ZZ^T}(A) = A - ZAZ^T \ , \tag{3.1}$$

$$F_{Z^TZ}(A) = A - Z^TAZ \tag{3.2}$$

Definition 2.1. A pair of matrices $G \in \mathbf{C}_{m,r}$ and $H \in \mathbf{C}_{n,r}$ is a *generator of length* r for an $m \times n$ matrix $A = GH^T$. The minimum length of generators for A is called the *rank* of A and will be denoted rank A. The *singular value decomposition (SVD)* of $A \in \mathbf{C}_{m,n}$ is defined as a triple U, V, Σ such that

$$\begin{aligned} A &= U\Sigma^2 V^H \ , \quad U \in \mathbf{C}_{m,r} \ , \\ V &\in \mathbf{C}_{n,r} \ , \quad r = \text{ rank } A \ , \\ U^H U &= I \ , \quad V^H V = I \ , \\ \Sigma &= \text{ diag}(\sigma_1(A), \ldots, \sigma_r(A)) \ , \\ \sigma_1(A) &\geq \sigma_2(A) \geq \cdots \geq \sigma_r(A) > 0 \ , \end{aligned} \tag{2.6}$$

$\sigma_h^2(A)$ are called the *singular values* of A. The matrix $A^+ = V(\Sigma^+)^2 U^H$, where $\Sigma^+ = \text{ diag}(1/\sigma_1(A), \ldots, (1/\sigma_r(A))$ is called the *Moore-Penrose generalized (pseudo) inverse* of A.

Fact 2.1. *$AA^+A = A$, $A^+AA^+ = A^+$, and if $A = A^H$, then $A^+A = AA^+$. If A is nonsingular, then $A^+ = A^{-1}$.*

Corollary 2.2. *For an $n \times n$ Hermitian matrix A of rank r and for $c_n = 1$, we have:*

$$c_{n-r}A^+A = -\sum_{i=n-r+1}^{n} c_i A^{i-n+r} \ , \tag{2.7}$$

$$\begin{aligned} c_{n-r}A^+ &= -c_{n-r+1}A^+A - \sum_{i=n-r+1}^{n-1} c_{i+1}A^{i-n+r} \\ &= \sum_{i=n-r+1}^{n-1} \Big((c_{n-r+1}/c_{n-r})c_i - c_{i+1} \Big) A^{i-n+r} \\ &\quad + (c_{n-r+1}/c_{n-r})A^r \ . \end{aligned} \tag{2.8}$$

Proof. Since rank $A = r$,

$$c_0 = c_1 = \ldots = c_{n-r-1} = 0 \neq c_{n-r}$$

(this property defines rank A given the characteristic polynomial of A), and (2.3) takes the following form:

$$A^{n-r} \sum_{i=n-r}^{n} c_i A^{i-n+r} = O \ , \quad c_n = 1 \ .$$

in the ill-conditioned case. Such an extension is obtained in section 7. In Appendix A, we recall an auxiliary algorithm from [Sc] and include some details omitted in [Sc]. In Appendix B, we follow [PR] and [P89] and simplify the computations in the case where we deal with Toeplitz (and not just Toeplitz-like) matrices.

2. Definitions and Results from Matrix Theory: $\mathbf{C}_{m,n}$ denotes the space of $m \times n$ complex matrices, which are vectors if $m = 1$ or $n = 1$. O and I denote the null and the identity matrices of appropriate sizes, respectively. det A and trace A denote the determinant and the trace of a matrix A, respectively; A^T and A^H denote the transpose and the conjugate transpose of A, respectively. A is *Hermitian* if $A = A^H$. $\mathrm{diag}(v_1, v_2, \ldots, v_n)$ denotes the $n \times n$ diagonal matrix with the diagonal entries $v_1, v_2, \ldots, v_n$.

$\|A\|_h$, for $h = 1$ and $h = 2$, denotes the h-norm of a matrix A, so that

$$a \le \|A\|_h \le an \tag{2.1}$$

provided that $A \in \mathbf{C}_{n,n}$ and that a denotes the maximum absolute value of the entries of A. For a nonsingular matrix A, let

$$\mathrm{cond}_h A = \|A\|_h \|A^{-1}\|_h \text{ for, } h = 1, 2 \ . \tag{2.2}$$

Theorem 2. (Cayley-Hamilton). *For an $n \times n$ matrix A,*

$$A^n + \sum_{i=0}^{n-1} c_i A^i = O \tag{2.3}$$

where $c_0, c_1, \ldots, c_{n-1}$ are the coefficients of the characteristic polynomial of A,

$$\det(\lambda I - A) = \lambda^n + \sum_{i=0}^{n-1} c_i \lambda^i \ , \tag{2.4}$$

$$c_0 = \det(-A) \ , \quad (-1)^n c_0 = \ \det A \ . \tag{2.5}$$

Corollary 2.1. $A^{-1} = -(A^{n-1} + \sum_{i=1}^{n-1} c_i A^{i-1})/c_0$ *if* $c_0 \neq 0$, *that is, if A is a nonsingular matrix.*

(which also gives its rank) with a similar improvement of the previous parallel complexity estimates for these computations.

Our algorithms and our complexity estimates can be extended to computing the inverses and the determinants of Hilbert-like and Vandermonde-like matrices by using the results of [P89a], where we do actually need to handle the case of Toeplitz-like (and not just Toeplitz) matrices.

Technically, we exploit the approaches of [Cs], [PS], [GP], [PR], and [P89], to which we add two new techniques for the treatment of rank deficient and of Toeplitz-like (but not Toeplitz) cases. The simpler technique, for the rank deficient case, just expresses A^+ through the powers of A by using the Cayley-Hamilton theorem and the Moore-Penrose conditions. The Toeplitz-like case is treated by means of computing the SVDs for the auxiliary matrices of ranks $O(1)$, which is essentially reduced to computing the zeros of polynomials of degrees $O(1)$ having only real zeros.

We do not need this SVD stage where we deal with Toeplitz matrices (see Appendix B below); and we will (quite generously) assume that, when we need to perform this stage, parallel time $O(\log n)$ with n processors suffice. In all other stages, we will assume the customary PRAM model of parallel computing, where each processor performs at most one arithmetic operation in each step. We will write $O_A(t,p)$ where we need parallel time $O(t)$ and p processors.

We will organize the paper as follows: In section 2, we recall some matrix theory and present our parallel algorithms that evaluate A^+ and $A^+\mathbf{b}$. In section 3, we recall the concepts and some results on displacement ranks and displacement generators. In section 4, we show how to approximate to a matrix by a matrix of a lower displacement rank (this result may be of some independent interest). In section 5, we adapt Newton's iteration to the inversion of Toeplitz-like matrices. In section 6, we briefly recall the results from [P88] on the parallel numerical NC inversion using n processors of well-conditioned Toeplitz-like matrices and then start working on the extension to the $O_A(\log^2 n, n^2)$ bound

integer coefficients. The problem has been reduced in [BGH] and [vzG] to solving n linear systems with Toeplitz-like matrices, and thus can be solved by using $O(\log^2 n)$ steps and n^3 processors. If we only seek the gcd, we may reach the bounds $O(\log^3 n)$ and n^2, since we may apply $O(\log n)$ steps of the binary search of the degree of the gcd and thus only solve $O(\log n)$ systems. These bounds have been claimed already in [PR] and [P89], but the algorithms of [PR] and [P89] may fail here, because the auxiliary linear systems may be singular (as J. von zur Gathen pointed this out to me); moreover, they are Toeplitz-like but not Toeplitz systems, so that we do need to exploit our new results in order to support the above complexity estimates. There is a very recent alternate approach of [BG] to the gcd computation for the cost of $O(\log^2 n)$ steps and n^2 processors relying on [P89] and avoiding the above difficulties, but this approach seems to require a higher precision of computations, whereas it is sufficient for us to perform all our computations with the asymptotically optimum precision of the order of the output precision, provided that we seek the exact solution. This is not a realistic assumption only in the case where a *low precision approximation* to the inverse of a *well-conditioned* Toeplitz or Toeplitz-like matrix is sought, and in the latter casese, we suggest the superior parallel algorithms of [P88] and [P89].

We do not know any previous parallel NC algorithms for the least-squares solution of linear systems with rank deficient and/or Toeplitz-like matrices (in spite of the importance of these problems), except that Toeplitz-like systems can be treated as general linear systems and, in the rank deficient case, the solution can be obtained by means of first computing the rank of the input matrix and then the rank of its maximal submatrix of full rank, but these alternate NC algorithms enormously blow up the processor bounds for both general and Toeplitz-like systems, as well as the sequential arithmetic time for general systems, compared with our results.

Besides the generalized inverse, we compute (for the same asymptotic parallel cost bound) the determinant and the characteristic polynomial of a square Toeplitz-like matrix

Summary

We use $O(\log^2 n)$ parallel arithmetic steps and n^2 processors to compute the least-squares solution $\mathbf{x} = A^+\mathbf{b}$ to a linear system of equations, $A\mathbf{x} = \mathbf{b}$, given a $g \times h$ matrix A and a vector $\mathbf{b}$, both filled with integers or with rational numbers, provided that $g+h \le 2n$ and that A is given with its displacement generator of length $r = O(1)$ and thus has displacement rank $O(1)$. For a vector $\mathbf{b}$ and for a general $p \times q$ matrix A (with $p+q \le n$), we compute A^+ and $A^+\mathbf{b}$ by using $O(\log^2 n)$ parallel arithmetic steps and $n^{2.851}$ processors, and we may also compute $A^+\mathbf{b}$ by using $\mathrm{O}(n^{2.376})$ arithmetic operations.

1. Introduction: We present two related results: we first extend the algorithms and the record parallel complexity estimates of $O(\log^2 n)$ parallel arithmetic steps and $n^{2.851}$ processors under the PRAM models (so far known for the $n \times n$ matrix inversion, [Cs], [PS], [GP]) to computing the generalized (or pseudo-) inverse A^+ of any $g \times h$ matrix A with $g + h = O(n)$ (see Corollary 2.3 below), and this also implies computing the least-squares solution $\mathbf{x} = A^+\mathbf{b}$ to any linear system $A\mathbf{x} = \mathbf{b}$ for the same parallel cost. The same algorithm leads to computing $A^+\mathbf{b}$ in $\mathrm{O}(n^{2.376})$ arithmetic operations. Then we improve the complexity estimates for computing such a least-squares solution to $O(\log^2 n)$ steps and n^2 processors in the case where A is any $g \times h$ [with $g + h = O(n)$] (Toeplitz-like) matrix, filled with integers or with rational numbers and given with its displacement generator of length $r = O(1)$ as $n \to \infty$. The previous algorithms of [PR] and [P89] support the same complexity bounds but only for Toeplitz and some block Toeplitz (rather than Toeplitz-like) matrices and require their nonsingularity; both of these restrictions greatly inhibit the range of the applications of the algorithms of [PR] and [P89].

Indeed, there are numerous highly important applications, for instance, to signal processing (see [KKM], [FMKL], [Kai]) where Toeplitz-like linear systems of equations are encountered, and they are singular and/or nonToeplitz systems.

Also, for example, consider computing the greatest common divisor (gcd) or all the entries of the extended Euclidean scheme for two polynomials of degrees at most n with

Parallel Least-Squares Solution of General and Toeplitz-like Linear Systems

V. Pan

Department of Computer Science, Columbia University

New York, NY 10027

and

Department of Mathematics and Computer Science

Lehman College, CUNY, Bronx, NY 10468 and

Department of Computer Science, SUNYA, Albany, NY 12222

CUCS-023-90