DOUBLING ALGORITHMS FOR TOEPLITZ AND RELATED EQUATIONS*

M. Morf

Information Systems Laboratory
Stanford University
Stanford, California 94305

ABSTRACT

A new class of doubling or halving algorithms for solving Toeplitz and related equations is presented. For scalar n by n Toeplitz matrices, they require $O(n \log^2 n)$ computations, similarly to the HGCD (half-greatest-common-divisor) based algorithm of Gustavson and Yun. However, these new algorithms are based on the notions of "shift" or displacement rank $1 \leq \alpha \leq n$, an index of how close a matrix is to being Toeplitz, requiring $O(\alpha^d\, n \log^2 n)$ operations, $(d \leq 2)$. A basic version of a doubling algorithm for such "α-Toeplitz matrices" is presented, and the applications of these results to related problems are mentioned, such as the inversion of banded-, block- and Hankel matrices.

1. INTRODUCTION

Using the HGCD (half-greatest-common-divisor) algorithm of Aho, Hopcroft, Ullman [1], Gustavson and Yun [2] constructed an algorithm to invert a Toeplitz matrix in $O(n \log^2 n)$. The main principle used in the HGCD as well as in our new algorithm is the "divide and conquer" or "doubling" approach. Roughly, we divide the problem to be solved into two sub-problems of "half" size. If this subdivision requires $m(n)$ computations, the total number of operations for carrying out our basic algorithms is $O(m(n) \log n)$. In HGCD algorithms the main operations are multiplication of two polynomials of degree n, which can be done in $O(n \log n)$ using fast Fourier transforms, (FFT's), hence the total number of operations is $O(n \log^2 n)$. If fast convolution is used instead [3], we get $O(n \log n)$ multiplications. Gustavson and Yun's algorithm is based on the Euclidean algorithm in matrix form, see e.g. [4], and the fact that the Pade' approximation requires the inversion of a Toeplitz matrix, see for instance [5] and [6]. Using those facts an implicit form of the inverse of an $n \times n$ Toeplitz matrix T is found in $O(n \log^2 n)$ operations.

We are interested in using the idea of doubling directly on Toeplitz matrices without referring to the HGCD algorithm and to find expressions for inverting a more general class of matrices that is related to Toeplitz matrices. The objective was to reduce the $O(n^2)$ operations of the well-known algorithm of Levinson [7] and its α-Toeplitz matrix matrix extensions [8-12]. Early references to algorithms for Toeplitz matrices requiring less than $O(n^2)$ operations can be found in [1] and [8, p. 99, p. 126], and for banded Toeplitz matrices of bandwidth p in $O(n \log p)$ [10], and later (apparently unaware of previous results) in $O(n \log n)$ [13]. Using scattering theory ideas, this can be improved to at most $O(p^3 \log n)$ [14], not only for banded but also for matrices arising from p-state models.

2. PROPERTIES OF α-TOEPLITZ MATRICES

First we recall the definition and properties of displacement ranks required in the proof of the basic algorithm.

Let M be an $n \times n$ matrix. In [11] we defined the displacement ranks

$$\alpha_+ = \min \alpha \quad , \quad \text{such that} \quad M = \sum_{i=1}^{\alpha} L_i U_i \quad ,$$

where L_i, U_i are lower and upper triangular Toeplitz respectively, and

$$\alpha_- = \min \alpha \quad , \quad \text{such that} \quad M = \sum_{i=1}^{\alpha} \bar{U}_i \bar{L}_i \quad ,$$

where $\bar{U}_i$, $\bar{L}_i$ are upper and lower triangular Toeplitz respectively. We also gave an operational definition of α using the lower shift matrix, $[Z_{ij}] = 1$ if $i-j=1$ else 0;

$$\alpha_+(M) = \text{rank } (M - Z M Z^T) \quad , \quad \text{and}$$

$$\alpha_-(M) = \text{rank } (M - Z^T M Z) \quad .$$

We can also state formally the following conversion lemma, that enables the conversion of the α_- into the α_+-representation and vice versa [10].

---

*Work supported by the DARPA Contract MDA903-78-C-0179 and in part by AFOSR Contract AF49620-79-C-0058, NSF Grant ENG-78-1003 and JSEP Contract DAAG29-79-C-0047.

Conversion Lemma

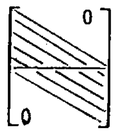
$$\sum_{i=1}^{\alpha_+} L_i U_i = T - \sum_{i=1}^{\alpha_-} \bar{U}_i \bar{L}_i \quad ,$$

where T is a Toeplitz matrix.

Proof: expand L_i and U_i to upper and lower banded Toeplitz matrices as follows

L_i extends to $[\bar{U}_i : L_i] = U_{Bi}$:

U_i extends to $\begin{bmatrix} \bar{L}_i \\ U_i \end{bmatrix} = L_{Bi}$:



.

Since the product of upper- times lower-banded Toeplitz matrices is Toeplitz

$$L_i U_i = U_{Bi} L_{Bi} - \bar{U}_i \bar{L}_i = T_i - \bar{U}_i \bar{L}_i \quad ,$$

and summing over i, with T the sum of the Toeplitz matrices T_i, the lemma follows. ■

Since $T = T_+ + T_-$, where $T_+(T_-)$ are the lower (upper) triangular parts of T, we have the bound $\alpha_-(M) \leq \alpha_+ + 2$.

Reversing the argument yields $|\alpha_+ - \alpha_-| \leq 2$, see also [11]. There we also prove the inversion property

$$\alpha_+(M) = \alpha_-(M^{-1}) \quad , \quad \text{and} \quad \alpha_-(M) = \alpha_+(M^{-1}) \quad .$$

This concludes the summary of our previously obtained results needed here.

3. DISPLACEMENT RANK PROPERTIES OF PARTITIONED MATRICES

Now partition the matrix M into

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad ,$$

where A and D are square, and their respective inverses exist if needed. Also, let

$$M^{-1} = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} \quad .$$

Using the α_+-decomposition

$$M = \sum_{i=1}^{\alpha_+} L_i U_i$$

$$M = \sum_{i=1}^{\alpha_+} \begin{bmatrix} L_{11}^i & 0 \\ L_{21}^i & L_{22}^i \end{bmatrix} \begin{bmatrix} U_{11}^i & U_{12}^i \\ 0 & U_{22}^i \end{bmatrix}$$

$$= \sum_{i=1}^{\alpha_+} \begin{bmatrix} L_{11}^i U_{11}^i & L_{11}^i U_{12}^i \\ L_{21}^i U_{11}^i & L_{21}^i U_{12}^i + L_{22}^i U_{22}^i \end{bmatrix}$$

where L_{11}^i, L_{22}^i are lower triangular Toeplitz, U_{11}^i, U_{22}^i are upper triangular Toeplitz, and L_{21}^i, U_{12}^i are full Toeplitz matrices. Then by our partition

$$A = \sum_{i=1}^{\alpha_-} L_{11}^i U_{11}^i \quad \text{so that} \quad \alpha_+(A) \leq \alpha_+(M) \quad .$$

Decomposing U_{12}^i into $(U_{12}^i)_+ + (U_{12}^i)_-$, we get

$$B = \sum_{i=1}^{\alpha_+} L_{11}^i (U_{12}^i)_+ + \sum_{i=1}^{\alpha_+} L_{11}^i (U_{12}^i)_-$$

$$= L + \sum_{i=1}^{\alpha_+} L_{11}^i (U_{12}^i)_- \quad , \qquad \text{hence}$$

$$\alpha_+(B) \leq \alpha_+(M) + 1 \quad .$$

Similarly $\alpha_+(C) \leq \alpha_+(M) + 1$, and $\alpha_+(D) \leq \alpha_+(M) + 2$, since

$$D = T + \sum_{i=1}^{\alpha_+} (L_{21}^i)_+ \, (U_{12}^i)_- \quad ,$$

where T, a sum of products of upper- times lower-banded Toeplitz matrices, is Toeplitz. Now we can state a summary of the displacement rank bounds of submatrices contained in a matrix M and its inverse.

Containment Bound Theorems

The α_+-decomposition of M yields the bounds

$$\alpha_+(A) \leq \alpha_+(M) \qquad \alpha_+(B) \leq \alpha_+(M) + 1 \quad ,$$

$$\alpha_+(C) \leq \alpha_+(M) + 1 \quad , \qquad \alpha_+(D) \leq \alpha_+(M) + 2 \quad ;$$

and the α_--decomposition of M yields

$\alpha_-(A) \leqq \alpha_-(M) + 2$, $\alpha_-(B) \leqq \alpha_-(M) + 1$,

$\alpha_-(C) \leqq \alpha_-(M) + 1$, $\alpha_-(D) \leqq \alpha_-(M)$.

Considering the partition of M^{-1}, we get

$\alpha_+(\tilde{A}) \leqq \alpha_+(M^{-1}) = \alpha_-(M)$,

$\alpha_-(\tilde{D}) \leqq \alpha_-(M^{-1}) = \alpha_+(M)$, and

$\alpha_-(\bar{A}) \leqq \alpha_-(M)$, $\alpha_+(\bar{D}) \leqq \alpha_+(M)$,

denoting $\bar{A} \hat{=} \tilde{A}^{-1} = A - B D^{-1} C$ and $\bar{D} \hat{=} \tilde{D}^{-1} = D - C A^{-1} B$, both so-called Schur complements.

The last identities say that the displacement ranks of the Schur complements of a matrix are inherited, a crucial property for the Fast Toeplitz Inversion Algorithm below, as well as other algorithms, e.g., distributed forms [15]. It is instructive for the reader to verify this fact for $\alpha_+ = 1$, say.

On a historical note, this property of Schur complements was the missing link in proving the lower complexity of the doubling algorithms referred to in [8, p. 99, p. 126], i.e., we had the same basic algorithm, but the upper bound of the displacement rank of nested submatrices would have grown to equal their size. Our results above on the bounds were finally announced by Kailath in [12].

4. FAST TOEPLITZ MATRIX INVERSION ALGORITHM

We now assume for simplicity that $n = 2^k$ for some integer k and we always use equal partition parts, i.e., the dimension of all elements of the partitioned matrix are half of the original matrix. We stress that this assumption is not necessary, but simply a convenience, see [15].

Consider the partitioned matrix

$$M_{2n} = \begin{bmatrix} A_n & B_n \\ C_n & D_n \end{bmatrix} ,$$

where the indices indicate the dimensions of matrices, and its inverse

$$M_{2n}^{-1} = \begin{bmatrix} \tilde{A}_n & \tilde{B}_n \\ \tilde{C}_n & \tilde{D}_n \end{bmatrix} .$$

The Schur complements of the submatrices A_n and D_n were defined as

$$\bar{A}_n = A_n - B_n D_n^{-1} C_n \quad , \qquad \bar{D}_n = D_n - C_n A_n^{-1} B_n \quad .$$

Now we have the following relations

$$\tilde{A}_n = \bar{A}_n^{-1} = A_n^{-1} + A_n^{-1} B_n \tilde{D}_n C_n A_n^{-1}$$

$$\tilde{B}_n = - A_n^{-1} B_n \tilde{D}_n = - \tilde{A}_n B_n D_n^{-1}$$

$$\tilde{C}_n = - \tilde{D}_n C_n A_n^{-1} = - D_n^{-1} C_n \tilde{A}_n$$

$$\tilde{D}_n = \bar{D}_n^{-1} = D_n^{-1} + D_n^{-1} C_n \tilde{A}_n B_n D_n^{-1} \quad .$$

Remark: From the above expressions, we can obtain M_{2n}^{-1} from say A_n^{-1} and $\bar{D}_n^{-1}$, and then we can get the solution of the system

$$M_n x_n = a_n \qquad \text{or} \qquad x_n = M_n^{-1} a_n$$

in $O(\alpha\, m(n))$, if M^{-1} is obtained in its α decomposed form, where $m(n)$ is the number of operations required to multiply a vector times a (triangular) Toeplitz matrix. This requires α convolutions that take each $m(n) = O(n \log n)$ operations using FFT's.

We can now present the algorithm HTI (half α-Toeplitz matrix inversion) with arguments M, the matrix itself, n the dimension of the matrix and α_+ the (+)-displacement rank of the matrix. For simplicity we assume here that the matrix M is given in its decomposed form, for a discussion of how to get this representation see [9], [11]. Briefly, the main step is the decomposition of $(M - ZMZ^T)$ into LDU, where L is n by α_+ and its columns are the first columns of the L_i below, D is an α_+ by α_+ identity (or signature matrix in the symmetric case, where $U = L^T$), and U is α_+ by n, its rows are the first rows of the U_i below. This would require $O(\alpha_+^d n)$ operations $(d \leqq 2)$, see e.g., [16]; however, as explained in [9], [11], in many applications this step can be avoided. Now, by assumption we are given

$$L_i,\ U_i \qquad \text{such that} \qquad M = \sum_{i=1}^{\alpha_+} L_i U_i \quad .$$

The algorithm can then be given in the form of the following recursive procedure in the style of a higher level language computer program (e.g., ALGOL, or APL type), similarly to the algorithms discussed in [1]. The internal representation of the matrices is assumed to be in terms of the first columns and rows of the L_i's and U_i's of each matrix, and the various conversions, additions and multiplications of the matrices, are performed directly in these representations. For simplicity of the description of the basic algorithm, we can assume that these operations are implemented in the computer language used here, or (as in APL) defined separately, c.f., following discussion.

Procedure HTI(M, n, α_+)

1. If $n = 1$, return $M^* = 1/M$,

   (* denotes representation of the inverse)

2. $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ (partition the matrix)

3. $A^* = HTI(A, n/2, \alpha_+)$ (note: $\alpha_+(A) \leq \alpha_+(M)$)

   $\bar{D} = D - C A^* B$

4. $\tilde{D} = \bar{D}^* = HTI(\bar{D}, n/2, \alpha_+)$

   (note: $\alpha_+(\bar{D}) \leq \alpha_+(M)$)

   $\tilde{A} = A^* + A^* B \tilde{D} C A^*$

   $\tilde{B} = - A^* B \tilde{D}$

   $\tilde{C} = - \tilde{D} C A^*$

5. return $M^* = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}$

End HTI

6. COMPUTATIONAL COMPLEXITY

At each iteration, the main operation is to find an explicit representation of the Schur complement of D, $\bar{D} = D - C A^* B$. Assuming that

$$B = \sum_{i=1}^{\alpha+1} L_{bi} U_{bi} \quad \text{and} \quad A^* = \sum_{i=1}^{\alpha+2} L_{ai} U_{ai} \quad ,$$

where $\alpha = \alpha_+(M)$, the representation of the product

$$A^* B = \sum_{i=1}^{\alpha+2} \sum_{j=1}^{\alpha+1} L_{ai} U_{ai} L_{bj} U_{bj}$$

is obtained by converting $U_{ai} L_{bj}$ into $T_{ij} - L_i U_j$, see *conversion lemma*, implying that

$$A^* B = \sum_{i=1}^{\alpha+2} \sum_{j=1}^{\alpha+1} L_{ai} (T_{ij} - L_i U_j) U_{bj} \quad .$$

Note: Since T_{ij} is Toeplitz, we can construct it from its first column and first row. The first column and the first row of T_{ij} require each $m(n)$ operations, i.e., $O(n \log n)$ via FFT's, hence for the product $A^* B$ we have a total of at most $O(\alpha^2 m(n))$ computations.

$$A^* B = \sum_{i=1}^{\alpha+2} \sum_{j=1}^{\alpha+1} L_{ai} T_{ij} U_{bj} - L U \quad ,$$

where

$$L = \sum_{i=1}^{\alpha+2} L_{ai} L_i \quad , \qquad U = \sum_{j=1}^{\alpha+1} U_j U_{bj} \quad ,$$

and again we get L and U with $O(\alpha\, m(n))$ operations.

Let us decompose each T_{ij} as $(T_{ij})_+ + (T_{ij})_-$, then

$$\sum_{i=1}^{\alpha+2} \sum_{j=1}^{\alpha+1} L_{ai} (T_{ij}) U_{bj} = \sum_{i,j} L_{ai} (T_{ij})_+ U_{bj}$$

$$+ \sum_{i,j} L_{ai} (T_{ij})_- U_{bj}$$

$$= \sum_{i=1}^{\alpha+2} L_{ai} \tilde{U}_i + \sum_{j=1}^{\alpha+1} \tilde{L}_j U_{bj} - L U$$

where each of the $\tilde{L}_i$'s and of $\tilde{U}_i$'s is obtained in $O(\alpha\, m(n))$ operations. From

$$A^* B = \sum_{i=1}^{\alpha+2} L_{ai} \tilde{U}_i + \sum_{j=1}^{\alpha+1} \tilde{L}_j U_{bj} - L U$$

we see $\alpha(A^* B) \leq 2(\alpha + 2)$ which is the upper bound on the displacement rank of $(A^* B)$.

If we use the same bounding argument for the product $(C A^* B)$, we get

$$\alpha(C A^* B) \leq (\alpha + 1) + 2(\alpha + 2) + 1 = 3(\alpha + 2) \quad ,$$

again an upper bound. Therefore, since $\bar{D} = D - (C A^* B)$ we have the bound $\alpha(\bar{D}) \leq 4(\alpha + 2)$, and a non-minimal representation of $\bar{D}$, as the sum of at most $4(\alpha + 2)$ lower- times upper-triangular Toeplitz matrices, is obtained via this procedure. However, one would prefer finding directly a minimal representation of at most α, L_i's and U_i's, since $\alpha(\bar{D}) \leq \alpha$. Keeping with the spirit of presenting here only the basic ideas, we refer to the more detailed discussion in [15], and introduce here only an outline. From the *inversion property* it is plausible that a minimal representation of $M^{-1} - Z M^{-1} Z^T$ can be found. By using the imbedding properties of submatrices, in particular $\bar{D}$, the Schur complement of D, we can get the desired minimal representation.

In summary, each iteration requires at most $O(\alpha^d\, m(n))$ operations, where $m(n)$ is the number of operations required for multiplying two

Toeplitz matrices, and $d \leq 2$. In order to get an upper bound, $K(n)$, on the total number of computations for the recursive procedure HTI, we have the inequality

$$K(n) \leq 2K\left(\frac{n}{2}\right) + O(\alpha^d\, m(n)) \quad . \qquad (*)$$

The term $2K\left(\frac{n}{2}\right)$ reflects the fact that the algorithm calls itself twice at each step. In the case where $\alpha = 1$, only one such call is required, because $\bar{D} = A$, c.f., comment in section 3, hence $K(n) = O(m(n))$. The inversion and the multiplication of triangular Toeplitz matrices have therefore the same complexity as convolution, as expected (see [1]). For $\alpha \geqq 2$ we clearly have

$$K(n) = O(\alpha^d\, m(n) \log n) \qquad \text{satisfies } (*) \quad ,$$

with $m(n) = O(n \log n)$ operations using FFT's. Alternatively, $m(n) = O(n)$ multiplications if fast convolution is used to multiply two Toeplitz matrices, see [3].

In conclusion, we have presented here just the most basic ideas for inverting an α-Toeplitz matrix in at most $O(\alpha^2 n \log^2 n)$ operations. However, the techniques used here are much more powerful and can be applied to a variety of Toeplitz matrix related problems, some of them are mentioned in the next section.

Computation Refinements

On the topic of computational refinements, it is clear that it would be desirable to cirectly obtain the minimal representation of the sum, the product, and partitioned matrices and the Schur complement:

i) The minimal representation of a sum of say r symmetric n by n matrices is obtained via

$$A = \sum_{i=1}^{r} A_i \leftrightarrow a \hat{=} (A - ZAZ^T)^{\frac{1}{2}} \quad ,$$

$$[a_1, a_2, \ldots, a_r]\tau = [a \,|\, O] \; : \; \left[\begin{array}{c|c} \begin{smallmatrix} * & 0 \\ * & * \end{smallmatrix} & 0 \end{array}\right] \quad , \qquad (+)$$

where $a_i \hat{=} (A_i - ZA_iZ^T)^{\frac{1}{2}}$, i.e., full-rank square-roots of size n by α_i, where α_i is the rank of a_i, τ is an orthogonal transformation ($\tau^T\tau = 1$) that triangularizes the matrix on its left, and the basis a is n by α_+, where

$$\alpha_+ \leq \sum_{i=1}^{r} \alpha_i$$

is the (+)-displacement rank of A. The required computational techniques resemble the ones presented for instance in [17].

ii) The minimal representation of a product, can be obtained in a computationally more attractive form from the "product" rule of the operator $\lrcorner M \underline{\Delta}\, M - ZMZ^T$

$$\lrcorner(AB) = (\lrcorner A)B + A(\lrcorner B) - (\lrcorner A)(\lrcorner B)$$

$$- ZAe_n e_n^T BZ^T \quad ,$$

where e_i is the i-th unit vector. Array forms similar to (+) above can be found easily.

iii) Finding the representation of a partitioned matrix, required in step 5 of the HTI-procedure, is the inverse of the problem discussed in Section 3. Using the same notation, the representation of M is of the form

$$\lrcorner M = \begin{bmatrix} \lrcorner A & \lrcorner B \\ \lrcorner C & \lrcorner D \end{bmatrix} + \text{low rank terms} \quad ,$$

where $\{\lrcorner A, \lrcorner B, \lrcorner C, \lrcorner D\}$ are the representations of $\{A, B, C, D\}$. Array forms can again be found using this expression.

iv) The minimal representation of a Schur complement is more involved. A crucial step involves representing $\Gamma(M^{-1}) \underline{\Delta}\, M^{-1} - Z^T M^{-1} Z$ in terms of $\lrcorner M$. We just indicate the form of such an expression

$$\Gamma(M^{-1}) = Q(e_1 e_1^T + \overline{\lrcorner M} - \overline{\lrcorner M}^2)Q^T \quad ,^*$$

where Q is a function of M and

$$\overline{\lrcorner M} \,\underline{\Delta}\, M^{-\frac{1}{2}}(\lrcorner M)(M^{-\frac{1}{2}})^T \quad , \qquad M = M^{\frac{1}{2}}(M^{\frac{1}{2}})^T \quad .$$

We note that it is not very hard to see from this expression that

$$\alpha_- = \text{rank}\{\Gamma(M^{-1})\} = \text{rank}\{\lrcorner M\} = \alpha_+ \quad .$$

For a discussion of further computational refinements and other topics we refer to [15], [18]. We only add here that as $\alpha \to n$, the upper bound given above becomes inefficient, i.e., $n^3 \log^2 n \geqq n^3$; however, using fast convolution and fast matrix multiplication, the exponent d and $m(n)$ can be lowered, see [15].

7. SOME APPLICATIONS

Matrix inversion algorithms that are based on partitioning lead to the problem of inverting the Schur complement of a submatrix. If one is interested in getting algorithms that can take advantage of the structure of the matrix to be inverted, the question arises, what properties are *invariant* under the Schur complement action. As an illustration,

* The quadratic feature of this expression is reminiscent of the δP lemma referred to in [17], a fact that hints a connection between the two problems.

we only sketch here a partial list of matrices that can be shown to have such invariants

- Hankel and Hankel plus Toeplitz type matrix equations;
- Banded and rational (ratio of banded) type matrices;
- Toeplitz block Toeplitz, or Hankel matrices;
- Banded block banded matrices and other sparse matrices;
- Matrices related to two-dimensional (2-D) and M-D problems;

For a discussion see [15], [18]. There are various other interesting topics that are related to doubling algorithms of this type, e.g.,

- Stochastic interpretation of doubling/halving algorithms;
- Fractal ladder form realization of matrix inversions.
- open problems, especially computational refinements,
- other applications: estimation, detection, pattern recognition, image reconstruction from projections, etc.

The study of structures of sets that are inherited in subset is a topic by itself. Generally this leads to recursive function theory. A very interesting class of objects, perhaps up to now more of a curiosity, consists of the so-called fractals [19]. They are best thought of as objects created by recursive function calls or by "mirror galleries." For an example see the "fractal ladder realization" discussed in [15], they are based on doubling as well as projection ideas presented in another paper on ladder forms in this conference proceedings [20].

New applications of the ideas presented here can be obtained by suitable operations that convert structure of matrices into sparseness. Simple examples of this type are matrices that have higher order displacement ranks, see e.g., [11]. Such concpets are useful for matrices with n^{th}-order polynomial variation along diagonals and off-diagonals, compared to constants for Toeplitz matrices.

REFERENCES

1. A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA:1974.
2. F.G. Gustavson and D.Y.Y. Yun, "Fast Computations of the Rational Interpolation Tables and Toeplitz Systems of Equations via the Fast Extended Euclidean Algorithm," Reprint, IBM Report.
3. S. Winograd, "On Computing the Discrete Fourier Transform," Math. of Comp., Vol. 32, pp. 175-199:1978.
   S. Winograd, "Arithmetic Complexity of Computation," Preprint, IBM Report.
4. C.C. MacDuffee, *An Introduction to Abstract Algebra*, New York, Wiley:1940.
5. E.R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, NY:1968.
6. J.L. Massey, "Shift-Register Synthesis and BCH Decoding," IEEE Trans. Info. Theory, IT-15, pp. 122-127:1969.
7. N. Levinson, "The Wiener RMS (root mean square) Error Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, pp. 261-278:Jan. 1947. Reprinted in N. Wiener, "Time Series," MIT Press:1949.
8. M. Morf, "Fast Algorithms for Multivariable Systems," Ph.D. Dissertation, Stanford University:1974.
9. B. Friedlander, M. Morf, T. Kailath and L. Ljung, "New Inversion Formulas for Matrices Classified in Terms of Their Distance from Toeplitz Matrices," Linear Algebra and its Application, Elsevier North-Holland, Inc.:1978.
10 M. Morf and T. Kailath, "Recent Results in Least-Squares Estimation Theory," Annals of Economic and Social Measurement:June 3, 1977.
11. T. Kailath, S.Y. Kung and M. Morf, "Displacement Ranks of Matrices and Linear Equations," J. Math. Analysis and Applications, Vol. 68, No. 2, pp. 395-407:April 1979.
12. T. Kailath, "Survey of New Results on Fast Estimation Algorithms," seminar held at U. of Calif., Berkeley:May 1979; talk presented at the Int. Sym. on Math. Theory of Networks and Systems, Delft U. of Tech., The Netherlands, Vol. 3:July 3-6, 1979.
13. A.K. Jain, "Fast Inversion of Banded Toeplitz Matrices by Circular Decompositions," IEEE Trans. ASSP, Vol. ASSP-26, No. 2, pp. 121-126:April 1978.
14. M. Morf, J. Dobbins, J. Newkirk and T. Kailath, "Square-Root Doubling for Steady State Riccati Solution," Proc. 1978 Conf. Inf. Sci. and Systems, John Hopkins Univ., Baltimore, MD, P. 136:March 29-31, 1978.
15. M. Morf, "Doubling Algorithms for Toeplitz and Displaced Sparse Equations," to be submitted to the IEEE Proc. on Information Theory.
16. J.R. Bunch and B.N. Parlett, "Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations," SIAM J. Numer. Anal., Vol. 8, No. 4:1971.
17. M. Morf, J. Dobbins, B. Friedlander and T. Kailath, "Square Root Algorithms for Parallel Processing in Optimal Estimation," in Automatic:1979.
18. M. Morf and J.M. Delosme, "Doubling Algorithms for Block Toeplitz and Related Equations, with Applications to 2-D Problems," 1979 Workshop 2-D Digital Signal Proc., UC, Berkeley:1979.
19. B.B. Mandelbrot, *Fractals, Form, Chance and Dimension*, W.H. Freeman and Co., CA:1977.
20. D.T.L. Lee and M. Morf, "Recursive Square-Root Ladder Estimation Algorithms," Proc. 1980 ICASSP, Denver, CO:April 1980.