Asymptotically Fast Solution of Toeplitz-Like Singular Linear Systems*

Erich Kaltofen

Department of Computer Science, Rensselaer Polytechnic Institute
Troy, New York 12180-3590; Inter-Net: kaltofen@cs.rpi.edu

Extended Abstract

1. Introduction

The Toeplitz-likeness of a matrix (Kailath et al. 1979) is the generalization of the notion that a matrix is Toeplitz. Block matrices with Toeplitz blocks, such as the Sylvester matrix corresponding to the resultant of two univariate polynomials, are Toeplitz-like, as are products and inverses of Toeplitz-like matrices. The displacement rank of a matrix is a measure for the degree of being Toeplitz-like. For example, an $r \times s$ block matrix with Toeplitz blocks has displacement rank $r+s$ whereas a generic $N \times N$ matrix has displacement rank N. A matrix of displacement rank α can be implicitly represented by a sum of α matrices, each of which is the product of a lower triangular and an upper triangular Toeplitz matrices. Such a ΣLU representation can usually be obtained efficiently.

We consider the problem of computing a solution to a possibly singular linear system $Ax = b$ with coefficients in an arbitrary field, where A is an $N \times N$ matrix of displacement rank α given in ΣLU representation. By use of randomization we show that if the system is solvable we can find a vector that is uniformly sampled from the solution manifold in $O(\alpha^2 N (\log N)^2 \operatorname{loglog} N)$ expected arithmetic operations in the field of entries. In case no solution exists, this fact is discovered by our algorithm. In asymptotically the same time we can also compute the rank of A and the determinant of a non-singular A.

Toeplitz and Toeplitz-like matrices and the corresponding linear systems are ubiquitous in control theory, of course, but also in symbolic computation. Examples are Sylvester resultants and subresultants, extended subresultants (Sasaki and Furukawa 1984), and slope resultants (Hong 1993). We have encountered them in Coppersmith's block Wiedemann algorithm (Coppersmith 1994, Kaltofen 1993 and 1994), where a bottleneck subproblem is the computation of a non-zero solution to a homogeneous block linear system with Toeplitz blocks. That system is derived from a problem for finding a linear recursion for a sequence of matrices, whose solution is a sequential component of the otherwise parallelizable algorithm and whose running time grows linearly in the number of processors used when solved by a generalization of the Berlekamp/Massey algorithm (see Díaz et al. 1993, Figures 4 and 5). Toeplitz-like systems appear in coding theory problems (Feng et al. 1994). Macaulay matrices and their sparse counterparts (Canny and Emiris 1993) also have Toeplitz-like properties, although the displacement operators used here do not directly apply. It is an important open problem to develop the corresponding theory and algorithms.

Our results build on work by Bitmead and Anderson (1980) and Morf (1980). We contribute in two ways. First, we remove by use of randomization the restriction that the input matrices are in general position. That condition is necessitated by the Bit-Anderson/Morf approach for two reasons. One is the necessary invertibility of the consecutively computed Schur complements and the other is the need to compute a minimum length ΣLU representation for the occurring Schur complements from one with more terms than is the known displacement rank of those matrices. We apply a randomization from Kaltofen and Saunders (1991) to solve both problems: by multiplying an arbitrary matrix from the left with a random unit upper triangular Toeplitz matrix and from the right with a random unit lower triangular Toeplitz matrix, the resulting product matrix has a generic rank profile with high probability. This means that such a matrix can be triangularized without

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-90-06077.

ISAAC 94 - 7/94 Oxford England UK
© 1994 ACM 0-89791-638-7/94/0007..$3.50

row or column permutations, a property which is needed for $O(\alpha N^2)$ Levinson-type algorithms as well (see, e.g., Gohberg et al. 1986 and Delsarte et al. 1985). Our second contribution is the generalization of algorithms to singular systems. Again, we make use of ideas in Kaltofen and Saunders (1991).

Our algorithm yields a unified approach for computing in "soft"-linear time polynomial greatest common divisors and polynomial Euclidean schemes (Moenck 1973), Sylvester resultants (Schwartz 1980), and solutions to Toeplitz systems (Brent et al. 1980). As said before, our application is to a bottleneck substep of the block Wiedemann method (Díaz et al. 1993). For instance, when solving a sparse linear system with 20 000 equations 20 000 unknowns, and 1.3 million non-zero entries in the coefficient matrix over the finite field GF(32 749) on eight computers, the solution of the arising singular block Toeplitz system by the block Berlekamp/Massey algorithm consumes more than 50% of the total time (ibid., Figure 4). It is, however, unknown to us if a careful implementation of the methods presented here can result in superior performance on such actual problems than the quadratic-time procedures which we currently use.

The next section presents sufficient detail of the theory of Toeplitz-like matrices, which is intended to make this paper self-contained. In particular, we give a constructive proof of the fact that the product of Toeplitz-like matrices remains Toeplitz-like (Proposition 2). Our algorithm utilizes recent randomization techniques from non-numeric linear algebra. In §3 we introduce the necessary ideas and give the key algorithm for reducing a ΣLU representation to minimum length (Proposition 4). In §4 we describe the main divide-and-conquer algorithm for inverting the leading principal of a Toeplitz-like matrix that has generic rank profile. We also prove that the arising possibly singular Schur complements remain Toeplitz-like.

2. Toeplitz-like matrices

In this section we introduce well-known tools from the theory of Toeplitz-like matrices (Kailath et al. 1979) needed in our linear system solver. We first define the notion of the displacement rank of a matrix. We consider $N \times N$ matrices over a field $\mathbb{K}$; define the lower-shift matrix

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

and define the matrix shift operators

$$\downarrow A = ZA \quad \text{and} \quad \uparrow A = AZ^{\mathrm{tr}}.$$

The matrix $\downarrow A$ is equal to A after being shifted down by one row, filling the first row by zeros, and the matrix $\uparrow A$ is equal to A after being shifted to the right by one column, filling the first column by zeros. Following Kailath et al. (1979), we define

$$\phi_+(A) = A - \downarrow(\uparrow A) = A - ZAZ^{\mathrm{tr}}$$

and

$$\alpha_+(A) = \operatorname{rank} \phi_+(A),$$

the latter being the *displacement rank of A with respect to the displacement operators ϕ_+*. The fundamental property is that given 2α column vectors $y_1, \ldots, y_\alpha$ and $z_1, \ldots, z_\alpha$ the functional equation in the matrix X,

$$X - \downarrow(\uparrow X) = \sum_{j=1}^{\alpha} y_j z_j^{\mathrm{tr}} \tag{1}$$

has the unique solution

$$X = \sum_{j=1}^{\alpha} L[\![y_j]\!]\, U[\![z_j^{\mathrm{tr}}]\!], \tag{2}$$

where $L[\![y]\!]$ denotes a lower-triangular Toeplitz matrix whose first column is y and $U[\![z^{\mathrm{tr}}]\!]$ denotes an upper triangular Toeplitz matrix whose first row is z^{tr}. Therefore a matrix of displacement rank α w.r.t. ϕ_+ is a sum of α products of lower and upper triangular Toeplitz matrices. We shall call the vectors $y_1, \ldots, y_\alpha$ and $z_1, \ldots, z_\alpha$ in

$$Y = \sum_{j=1}^{\alpha} y_j z_j^{\mathrm{tr}} = \underbrace{[\, y_1 \mid y_2 \mid \ldots \mid y_\alpha \,]}_{y} \cdot \left.\begin{bmatrix} z_1^{\mathrm{tr}} \\ \hline z_2^{\mathrm{tr}} \\ \hline \vdots \\ \hline z_\alpha^{\mathrm{tr}} \end{bmatrix}\right\} z \tag{3}$$

the *left* and *right generators* of the $N \times N$ matrix Y. For our purpose, the matrix Y will be a displaced matrix such as $\phi_+(X)$. Furthermore, we shall call the representation (2) the *ΣLU representation* for X. That representation requires only the storage of $O(\alpha N)$ field elements. Clearly, one may derive a generator (3) for Y by choosing the vectors y_j to be α linearly independent columns of Y, and the entries in each column of the right factor matrix with the rows z_j^{tr} to be the linear combination that yields the corresponding column of Y.

The main property of matrices of small displacement rank is that their inverses also have small displacement rank. Clearly, the inverse of a Toeplitz matrix is not Toeplitz but, as we will see, it is Toeplitz-like. However, the displacement operator ϕ_+ does not directly apply to the inverse; instead, a dual operator is used, which we

now introduce. Consider the shift operators

$$\uparrow A = Z^{\mathrm{tr}} A \quad \text{and} \quad \leftarrow A = AZ.$$

The matrix $\uparrow A$ is equal to A after being shifted up by one row, filling the last row by zeros, and the matrix $\leftarrow A$ is equal to A after being shifted to the left by one column, filling the last column by zeros. Now define

$$\phi_-(A) = A - \uparrow(\leftarrow A) = A - Z^{\mathrm{tr}} A Z$$

and

$$\alpha_-(A) = \operatorname{rank} \phi_-(A),$$

the latter being the *displacement rank with respect to the displacement operator* ϕ_-. By transposition along the anti-diagonal of the matrix X in (1), one obtains a dual to the ΣLU representation; namely,

$$X - \uparrow(\leftarrow X) = \sum_{k=1}^{\bar{\alpha}} \bar{y}_k \bar{z}_k^{\mathrm{tr}}$$

$$\Longleftrightarrow \quad X = \sum_{k=1}^{\bar{\alpha}} U[\![(\bar{y}_k^{\mathrm{rev}})^{\mathrm{tr}}]\!] L[\![\bar{z}_k^{\mathrm{rev}}]\!], \quad (4)$$

where z^{rev} is the reverse of a vector z; that is,

$$z^{\mathrm{rev}} = J \cdot z \quad \text{with} \quad J = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \iddots & & \vdots \\ 1 & & 0 & 0 \end{bmatrix} \in \mathbb{K}^{N \times N}.$$

We will call the right side of (4) the ΣUL representation of X. There is an explicit formula for converting a ΣLU representation to a ΣUL representation (cf. Bitmead and Anderson 1980, Lemma 5), which we will need later: for $y, z \in \mathbb{K}^N$

$$L[\![y]\!]U[\![z^{\mathrm{tr}}]\!] = IL[\![\hat{z}]\!] + U[\![\hat{y}^{\mathrm{tr}}]\!]I - U[\![(ZJy)^{\mathrm{tr}}]\!]L[\![ZJz]\!], \quad (5)$$

where $\hat{z}^{\mathrm{tr}}$ is the reversed last row of $L[\![y]\!]U[\![z^{\mathrm{tr}}]\!]$, and $\hat{y}$ the reversed last column of $L[\![y]\!]U[\![z^{\mathrm{tr}}]\!]$ but with the first entry set to 0. Note that I is the $N \times N$ identity matrix. From (5) and the dual formula

$$U[\![z^{\mathrm{tr}}]\!]L[\![y]\!] = L[\![\check{y}]\!]I + IU[\![\check{z}^{\mathrm{tr}}]\!] - L[\![ZJz]\!]U[\![(ZJy)^{\mathrm{tr}}]\!], \quad (6)$$

for $y, z \in \mathbb{K}^N$, where $\check{z}^{\mathrm{tr}}$ is the first row of $U[\![z^{\mathrm{tr}}]\!]L[\![y]\!]$, and $\check{y}$ is the first column of $U[\![z^{\mathrm{tr}}]\!]L[\![y]\!]$ with its first entry set to 0, we conclude that for any square matrix A the inequalities $-2 \le \alpha_+(A) - \alpha_-(A) \le 2$ must hold. Moreover, the conversions (5) and (6) can be carried out in $O(N \log N \operatorname{loglog} N)$ arithmetic operations in $\mathbb{K}$. We finally can formulate the closure property with respect to matrix inversion.

Proposition 1. *For any nonsingular matrix $A \in \mathbb{K}^{N \times N}$ we have for the displacement ranks of the inverse matrix $\alpha_+(A^{-1}) = \alpha_-(A)$ and $\alpha_-(A^{-1}) = \alpha_+(A)$.*

An elegant proof of this property is found in (Pan 1992, Proposition A.4). Proposition 1 also provides the historical motivation for considering ΣLU representations (2): based on a Toeplitz matrix inversion algorithm by Trench (1964), Gohberg and Semencul (1973) developed a formula for the inverse of a Toeplitz matrix consisting of a sum of two Toeplitz LU-products.

Another property of Toeplitz-like matrices that we need for our algorithm is the fact that their products remain Toeplitz-like. Because we encounter rectangular matrices in our algorithm, we first have to extend the definitions of the displacement operators to such matrices. By subscripting Z_N we shall indicate that the shift matrix Z is of dimensions $N \times N$; we define a rectangular displacement operator

$$\phi_+(X) = X - Z_M X Z_N^{\mathrm{tr}} \quad \text{for } X \in \mathbb{K}^{M \times N}.$$

Again, $\phi_+(X)$ is generated by $\alpha = \alpha_+(X) = \operatorname{rank} \phi_+(X)$ vectors $y_1, \dots, y_\alpha \in \mathbb{K}^M$ and $z_1, \dots, z_\alpha \in \mathbb{K}^N$: $\phi_+(X) = \sum_{j=1}^{\alpha} y_j z_j^{\mathrm{tr}}$. We now have the following product rule (cf. Pan 1992, Proposition A.3).

Proposition 2. *Let $G \in \mathbb{K}^{L \times M}$ and $H \in \mathbb{K}^{M \times N}$ be rectangular matrices with displacement ranks $\gamma = \alpha_+(G)$ and $\delta = \alpha_+(H)$. Then $\phi_+(GH)$ can be generated by $\gamma + \delta + 1$ vectors.*

Proof. First, we observe that $I_M = Z_M^{\mathrm{tr}} Z_M + e_M e_M^{\mathrm{tr}}$, where I_M is the $M \times M$ identity matrix and e_M is the M^{th} unit vector. Therefore

$$\begin{aligned} &\phi_+(GH) \\ &= GH - Z_L G I_M H Z_N^{\mathrm{tr}} \\ &= GH - (Z_L G Z_M^{\mathrm{tr}})(Z_M H Z_N^{\mathrm{tr}}) - Z_L G e_M e_M^{\mathrm{tr}} H Z_N^{\mathrm{tr}} \\ &= (G - Z_L G Z_M^{\mathrm{tr}})H + Z_L G Z_M^{\mathrm{tr}}(H - Z_M H Z_N^{\mathrm{tr}}) - gh^{\mathrm{tr}} \\ &= \phi_+(G)H + Z_L G Z_M^{\mathrm{tr}} \phi_+(H) - gh^{\mathrm{tr}}, \end{aligned} \quad (7)$$

where $g = Z_L G e_M \in \mathbb{K}^L$ and $h = Z_N H^{\mathrm{tr}} e_M \in \mathbb{K}^N$. ⊠

3. Randomizations

Our algorithm utilizes randomization. We collect the necessary techniques here.

Theorem 1. *Let $F(x_1, \dots, x_\nu)$ be a non-zero ν-variate polynomial over an integral domain and let S be a subset of that domain. Then the probability of avoiding the zeros of F while evaluating in S is bounded as follows:*

$$\operatorname{Prob}\Big(F(s_1, \dots, s_\nu) \neq 0 \mid s_j \in S \text{ for all } 1 \le j \le \nu\Big) \ge 1 - (\deg F)/(\operatorname{card} S).$$

Here $\deg(F)$ denotes the total degree of F, i.e., the maximum of all term exponent sums and $\mathrm{card}(S)$ denotes the cardinality of S. The theorem in the above form was given by Schwartz (1980). Somewhat different versions are due to DeMillo and Lipton (1978) and Zippel (1979; 1993, §12).

In the following, we consider an $N \times N$ singular matrix A with entries from a field. By A_i we shall denote the leading $i \times i$ principal submatrix, i.e., the $i \times i$ submatrix located in the left upper corner of A, where $1 \leq i \leq N$. We say that A has *generic rank profile* (cf. Delsarte et al. 1985) if A_j is non-singular for all $1 \leq j \leq \mathrm{rank}\, A$. In such a case, no search for non-zero pivot elements would have to be performed during triangularization by Gaussian elimination. The following is Theorem 2 of Kaltofen and Saunders (1991).

Theorem 2. *For an $N \times N$ matrix A of rank r consider the matrix product $\widetilde{A} = VAW$ with*

$$V = \begin{bmatrix} 1 & v_2 & v_3 & \dots & v_N \\ & 1 & v_2 & \dots & v_{N-1} \\ & & 1 & \ddots & \vdots \\ & 0 & & \ddots & v_2 \\ & & & & 1 \end{bmatrix}$$

and

$$W = \begin{bmatrix} 1 & & & & \\ w_2 & 1 & & & 0 \\ w_3 & w_2 & 1 & & \\ \vdots & & \ddots & \ddots & \\ w_N & w_{N-1} & \dots & w_2 & 1 \end{bmatrix},$$

where the elements of the unit upper triangular Toeplitz matrix V and the elements of the unit lower triangular Toeplitz matrix W are randomly and uniformly selected from a subset S of the field of entries. Then $\widetilde{A}$ has generic rank profile with probability no less than $1 - r(r+1)/\mathrm{card}(S)$.

As we will see later, it is often useful to work with the pre-conditioned matrix $\widetilde{A}$, which has generic rank profile, instead of with A. The following simple lemma from Kaltofen and Saunders (1991, Theorem 4) shows how to find random non-zero solutions to inhomogeneous singular linear systems.

Proposition 3. *Let A be an $N \times N$ matrix of rank r and suppose that the leading $r \times r$ principal submatrix A_r is non-singular. Then for a random vector y with coordinates from the field of entries, the vector*

$$x = \left[\frac{A_r^{-1} b'}{0^{N-r}}\right] - y,$$

is a random solution to $Ax = b$, where the vector b' consists of the first r coordinates of $b + Ay$.

A key problem in the Bitmead-Anderson/Morf approach of inverting Toeplitz-like matrices is the reduction of a ΣLU representation of §2 for a matrix X to one with a minimum number of terms under the sum (2). We can solve this problem by the randomizations discussed above. Consider that we are given $\beta \geq \alpha$ generators for a matrix $Y = \phi_+(X)$,

$$Y = \hat{y} \cdot \hat{z}^{\mathrm{tr}}, \quad \hat{y}, \hat{z} \in \mathbb{K}^{N \times \beta},$$

and we wish to determine the displacement rank $\alpha = \alpha_+(X)$ and a ΣLU representation of length α for X. We pick random matrices V and W as in Theorem 2. Then the matrix $\widetilde{Y} = VYW$ has, with high probability, generic rank profile. Since $\mathrm{rank}(Y) = \mathrm{rank}(\widetilde{Y})$, every column to the right of the first α columns of $\widetilde{Y}$ is a linear combination of the first α columns. These linear combinations determine generators for $\widetilde{Y}$; namely, $\widetilde{Y} = \tilde{y} \cdot \tilde{z}^{\mathrm{tr}}$, where $\tilde{y}, \tilde{z} \in \mathbb{K}^{N \times \alpha}$. Here $\tilde{y}$ are the first α columns of $\widetilde{Y}$ and each column in $\tilde{z}^{\mathrm{tr}} = [I_\alpha \mid \dots]$ corresponds to the linear combination, yielding the column of $\widetilde{Y}$ in the same position. The minimum-length generators for Y are then obtained as $Y = (V^{-1}\tilde{y}) \cdot (\tilde{z}^{\mathrm{tr}} W^{-1})$. The running time of this method is stated in the next proposition, whose proof can be found in (Kaltofen 1994, Appendix A).

Proposition 4. *From a ΣLU representation of $X \in \mathbb{K}^{N \times N}$ of length β, namely, $X = \sum_{k=1}^{\beta} L[\![\hat{y}_k]\!] U[\![\hat{z}_k^{\mathrm{tr}}]\!]$, one can compute in $O(\alpha\beta N + \beta N \log N \operatorname{loglog} N)$ arithmetic steps in $\mathbb{K}$ a ΣLU representation $X = \sum_{j=1}^{\alpha} L[\![y_j]\!] U[\![z_j^{\mathrm{tr}}]\!]$, where $\alpha = \mathrm{rank}\, \phi_+(X)$ is minimum. The algorithm is randomized and requires $2N - 2$ uniformly sampled elements from a set $S \subset \mathbb{K}$; it returns with probability no less than $1 - \alpha(\alpha+1)/\mathrm{card}(S)$ a correct result.*

4. Fast inversion of a Toeplitz-like matrix

At task is to compute the ΣUL representation for the inverse of a non-singular matrix A given in ΣLU representation. If A is singular, but of generic rank profile, we seek the ΣUL representation for the largest non-singular leading principal submatrix. The algorithm follows a divide-and-conquer matrix partitioning à la Strassen: let

$$A = \left[\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline A_{2,1} & A_{2,2} \end{array}\right], \tag{8}$$

where $A_{1,1} \in \mathbb{K}^{M \times M}$, $A_{1,2}, A_{2,1}^{\mathrm{tr}} \in \mathbb{K}^{M \times (N-M)}$, and $A_{2,2} \in \mathbb{K}^{(N-M)\times(N-M)}$. If $A_{1,1}$ is non-singular, we consider the *Schur complement* $\Delta = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$. If both $A_{1,1}$ and A are non-singular, the inverse can be computed as $A^{-1} =$

$$\left[\begin{array}{c|c} A_{1,1}^{-1} + A_{1,1}^{-1}A_{1,2}\Delta^{-1}A_{2,1}A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}\Delta^{-1} \\ \hline -\Delta^{-1}A_{2,1}A_{1,1}^{-1} & \Delta^{-1} \end{array}\right].$$

The key property is:

Proposition 5. *If A, $A_{1,1}$, and Δ are the matrices defined above, $A_{1,1}$ is non-singular and if the top-left entry of A, $A[1,1] \neq 0$, then $\alpha_+(\Delta) \leq \alpha_+(A)$.*

Proof. In case that A is a non-singular matrix, the stated displacement rank inequality for the Schur complement is proven (without the condition on $A[1,1]$) in Bitmead and Anderson (1980, Lemma 8) and it is also stated in Morf (1980). We will reduce the singular case to the non-singular case. Consider a minimum length ΣLU representation of A, namely, $A = \sum_{j=1}^{\alpha} L^{(j)}U^{(j)}$, and suppose without loss of generality that $(L^{(1)}U^{(1)})[1,1] \neq 0$. The latter condition is necessitated by our assumption on the non-vanishing of the top-left entry of A. Therefore, the parameterized matrix

$$\begin{aligned} A(\lambda) &= (L^{(1)} + \lambda I)U^{(1)} + \sum_{j=2}^{\alpha} L^{(j)}U^{(j)} \\ &= A + \lambda U^{(1)} \in \mathbb{K}(\lambda)^{N\times N} \end{aligned}$$

is non-singular of displacement rank α w.r.t. ϕ_+. Partitioning $A(\lambda)$ corresponding to (8), we obtain a parameterized Schur complement

$$\Delta(\lambda) = A_{2,2}+\lambda U_{2,2}^{(1)} - A_{2,1}(A_{1,1}+\lambda U_{1,1}^{(1)})^{-1}(A_{1,2}+\lambda U_{1,2}^{(1)}).$$

It follows from the non-singular version of this proposition that $\alpha_+(\Delta(\lambda)) \leq \alpha$. We may write $\Delta(\lambda)$ as power series in λ with matrix coefficients,

$$\begin{aligned} \Delta(\lambda) = \Delta + \lambda(&U_{2,2}^{(1)} + A_{2,1}A_{1,1}^{-1}U_{1,1}^{(1)}A_{1,1}^{-1}A_{1,2} \\ &- A_{2,1}A_{1,1}^{-1}U_{1,2}^{(1)}) \\ &+ \text{higher order terms in } \lambda, \end{aligned}$$

using the series expansion

$$\begin{aligned} &(G + \lambda H)^{-1} \\ &\quad = (I + \lambda G^{-1}H)^{-1}G^{-1} \\ &\quad = G^{-1} - \lambda G^{-1}HG^{-1} + \lambda^2(G^{-1}H)^2G^{-1} + \cdots. \end{aligned}$$

Since no negative powers of λ occur, the constant term in this power series for $\Delta(\lambda)$, which is Δ, cannot have a higher displacement rank than $\Delta(\lambda)$, which proves the proposition. ⊠

We can now sketch the main algorithm (cf. Bitmead and Anderson 1980, p. 110).

Algorithm *Leading Principal Inverse*

Input: Vectors $y_1, \ldots, y_\alpha$, $z_1, \ldots, z_\alpha \in \mathbb{K}^N$ such that $A = \sum_{j=1}^{\alpha} L[\![y_j]\!]U[\![z_j^{\mathrm{tr}}]\!] \in \mathbb{K}^{N\times N}$ has generic rank profile.

Output: An integer $r \leq N$ and vectors $\bar{y}_1, \ldots, \bar{y}_{\bar{\alpha}}$, $\bar{z}_1, \ldots, \bar{z}_{\bar{\alpha}} \in \mathbb{K}^r$ with $\bar{\alpha} \leq \alpha$ such that with high probability

$$r = \mathrm{rank}(A) \quad \text{and} \quad A_r^{-1} = \sum_{k=1}^{\bar{\alpha}} U[\![\bar{y}_k^{\mathrm{tr}}]\!]L[\![\bar{z}_k]\!],$$

where A_r is the largest non-singular leading principal submatrix of A.

If $N \leq \alpha$ then expand the ΣLU representation of A and compute A_r^{-1} explicitly; finally, from $\phi_-(A_r^{-1})$ explicitly determine the ΣUL representation and return.

Now, let the matrix A be partitioned as (8) with $M = \lceil N/2 \rceil$.

Step 1: *Call the algorithm recursively to process $A_{1,1}$.* Note that the ΣLU representation of $A_{1,1}$ is given by the first M entries of y_j and z_j. If the returned rank of $A_{1,1}$ is less than M, we are done. Otherwise, the algorithm has produced a ΣUL representation of $A_{1,1}^{-1}$.

Step 2: *Compute a ΣLU representation of length no more than α for the Schur complement $\Delta = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$.* We further explain this task in the analysis of the algorithm. *If $\Delta[1,1] = 0$, then $M = \mathrm{rank}(A)$; else perform the next steps.*

Step 3: *Call the algorithm recursively to process Δ.* Note that, with high probability, $\mathrm{rank}(A) = M + \mathrm{rank}(\Delta) = r$.

Step 4: Consider the leading principal submatrix A_r partitioned as

$$A_r = \left[\begin{array}{c|c} A_{1,1} & A'_{1,2} \\ \hline A'_{2,1} & A'_{2,2} \end{array}\right],$$

where $A_{1,1} \in \mathbb{K}^{M\times M}$, $A'_{1,2}, A'^{\mathrm{tr}}_{2,1} \in \mathbb{K}^{M\times(r-M)}$, and $A'_{2,2} \in \mathbb{K}^{(r-M)\times(r-M)}$. At this point we have the ΣUL representations for $A_{1,1}^{-1}$ and for Δ'^{-1}, where $\Delta' = A'_{2,2} - A'_{2,1}A_{1,1}^{-1}A'_{1,2}$. *Compute (possibly non-minimum length) generators for $\phi_-(B'_{1,1})$, $\phi_-(B'_{1,2})$, and $\phi_-(B'_{2,1})$, where $B'_{1,2} = -A_{1,1}^{-1}A'_{1,2}\Delta'^{-1}$, $B'_{2,1} = -\Delta'^{-1}A'_{2,1}A_{1,1}^{-1}$, and $B'_{1,1} = A_{1,1}^{-1} - B'_{1,2}A'_{2,1}A_{1,1}^{-1}$. Finally, compute a minimum length ΣUL representation for*

$$A_r^{-1} = \left[\begin{array}{c|c} B'_{1,1} & B'_{1,2} \\ \hline B'_{2,1} & \Delta'^{-1} \end{array}\right]. \qquad \square$$

We can now state and prove the running time of the above algorithm.

Theorem 3. *Algorithm Leading Principal Inverse finishes after $O(\alpha^2 N(\log N)^2 \operatorname{loglog} N)$ arithmetic operations in $\mathbb{K}$. It requires $O(N \log N)$ random field elements that are uniformly sampled from a subset $S \subset \mathbb{K}$, and it returns with probability no less than $1 - 4N\alpha/\mathrm{card}(S)$ a correct rank and ΣUL representation of the largest leading principal submatrix.*

Proof. Let $T(\alpha, N)$ denote the maximum number of arithmetic operations required for any input of dimension N and of at most α displacement rank. Step 1 requires at most $T(\alpha, \lceil N/2 \rceil)$ arithmetic operations. By Proposition 5, Step 3 requires at most $T(\alpha, \lfloor N/2 \rfloor)$ arithmetic operations. We shall show that both Step 2 and Step 4 have arithmetic complexity $O(\alpha^2 N \log N \cdot \operatorname{loglog} N)$. Hence, for a constant C, we must have

$$T(\alpha, N) \leq T(\alpha, \lceil N/2 \rceil) + T(\alpha, \lfloor N/2 \rfloor) + C\alpha^2 N \log N \operatorname{loglog} N,$$

which yields the arithmetic complexity $T(\alpha, N) = O(\alpha^2 \cdot N(\log N)^2 \operatorname{loglog} N)$.

In Step 2, we first compute generators for $\phi_+(\Delta)$ of length $\beta \leq 4\alpha + 8$, which we then reduce by Proposition 4 and 5 to a length of no more than α. The former is accomplished as follows: generators of length no more than $\alpha + 2$ can be derived for $\phi_+(A_{2,2})$ from generators for $\phi_+(A)$ by correcting for the shift into $A_{2,2}$ of parts of row M and parts of column M of A. Similarly, generators of length no more than $\alpha + 1$ can be derived for $\phi_+(A_{2,1})$ and $\phi_+(A_{1,2})$ (cf. Bitmead and Anderson 1980, Lemma 8). The ΣUL representation for $A_{1,1}^{-1}$ can be converted to a ΣLU representation of length no more than $\alpha + 2$ by using formula (6). We do not have ΣLU representations for the rectangular matrices $A_{2,1}$ and $A_{1,2}$. However, we have the ΣLU representation of A restricted to these submatrices. Thus, we may effectively use the product rule (7) of Proposition 2. For instance, the generators

$$Z_{N-M} A_{2,1} Z_M^{\mathrm{tr}} \phi_+(A_{1,1}^{-1})$$

arising in the computation of generators for $\phi_+(A_{2,1}A_{1,1}^{-1})$ are found by multiplying each y vector of the generators of $\phi_+(A_{1,1}^{-1})$; first by Z_M^{tr}, then by $A_{2,1}$, and finally by Z_{N-M}. Clearly, from the ΣLU representation of A restricted to $A_{2,1}$, such multiplication can be carried out for a single vector in $O(\alpha N \log N \operatorname{loglog} N)$ arithmetic operations. Therefore, the computation of the generators for $A_{2,1}A_{1,1}^{-1}A_{1,2}$ dominates this step at a cost of $O(\alpha^2 N \log N \operatorname{loglog} N)$ arithmetic operations.

The tasks of Step 4 are carried out similarly. After converting the ΣLU representation of A to a ΣUL representation using formula (5), we can obtain generators of length no more than $\alpha + 3$ for $\phi_-(A'_{2,1})$ and $\phi_-(A'_{1,2})$. Note that here we need a generalized ϕ_- operator on rectangular matrices. Then, as in Step 2 with a product formula for $\phi_-(GH)$ dual to (7), we find generators for

$$\begin{aligned} &\phi_-(B'_{1,2}) \quad \text{with} \quad \alpha_-(B'_{1,2}) \leq 3\alpha + 5, \\ &\phi_-(B'_{2,1}) \quad \text{with} \quad \alpha_-(B'_{2,1}) \leq 3\alpha + 5, \\ &\phi_-(B'_{1,1}) \quad \text{with} \quad \alpha_-(B'_{1,1}) \leq 6\alpha + 10. \end{aligned}$$

Finally, the generators for the blocks can be "puzzled" together to a generator of $\phi_-(A_r^{-1})$ of length no more than $13\alpha + 22$. Note that the length is the sum of the individual lengths corrected by two extra generators, which make up for the "cross" of a row and a column missing in the shift of the individual blocks. Finally, we reduce the ΣUL representation of A_r^{-1} to minimum length, again appealing to a dual of Proposition 4. The overall cost in this step is again dominated by the implementation of the product formula, which is $O(\alpha^2 N \cdot \log N \operatorname{loglog} N)$.

Finally, we argue that the algorithm produces, with the stated probability, the correct result. By Proposition 5, the displacement rank of the Schur complement Δ is no more than α. Furthermore, Δ has generic rank profile, as can be deduced from the factorization

$$A = \left[\begin{array}{c|c} I_M & 0 \\ \hline A_{2,1}A_{1,1}^{-1} & I_{N-M} \end{array}\right] \cdot \left[\begin{array}{c|c} A_{1,1} & A_{1,2} \\ \hline 0 & \Delta \end{array}\right]. \qquad (9)$$

Thus the algorithm produces a correct result if the randomizations of Proposition 4 needed in Steps 2 and 4 result in correct ΣLU representations and the recursive calls return correct ΣUL representations. The straightforward analysis yielding the given failure probability estimate and random element count can be found in (Kaltofen 1994, Appendix A). ⊠

From Theorem 2, Proposition 3, and the analysis of algorithm Leading Principle Inverse we immediately obtain the following corollary. Note that by Proposition 2 the displacement rank of $\tilde{A}$ in Theorem 2 satisfies $\alpha_+(\tilde{A}) \leq \alpha_+(A) + 4$.

Corollary 1. *Given a linear system $Ax = b$ over a field with N equations and N unknowns, where A is an $N \times N$ matrix of displacement rank α given in ΣLU representation, we have a randomized algorithm that either computes a solution vector x, which is randomly sampled in the solution manifold, or it determines that none exists in $O(\alpha^2 N(\log N)^2 \operatorname{loglog} N)$ expected field operations.*

Example: We shall apply Corollary 1 to the problem of computing extended Euclidean schemes. Consider two polynomials

$$f_{-1}(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_0 \in \mathbb{K}[x]$$

and

$$f_0(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_0 \in \mathbb{K}[x]$$

in $\mathbb{K}[x]$ of degree m and n, respectively. The following problem occurs, e.g., when computing Padé approximants: given an integer $l \leq \min\{m, n\}$, compute the

scheme

$$s_i f_{-1} + t_i f_0 = f_i, \quad s_i, t_i, f_i \in \mathbb{K}[x],$$

where f_i is that remainder in the polynomial remainder chain of f_{-1} and f_0 whose degree satisfies

$$d = \deg(f_i) \le l < \deg(f_{i-1});$$

note that $\deg(s_i) = n - \deg(f_{i-1})$. The main difficulty imposed is that one does not know the actual degree d in advance. One well-known solution is by Moenck (1973) (see also Brent et al. 1980 and Strassen 1983) and makes use of the Knuth/Schönhage half-GCD approach. Alternately, we can use our algorithms for Toeplitz-like linear systems, which we briefly explain now. Consider the linear system in the coefficients of the polynomials $S(x)$, $T(x)$, and $F(x)$,

$$Sf_{-1} + Tf_0 = F, \quad \begin{cases} \deg(F) \le l, \\ \deg(S) \le n - l - 1, \\ \deg(T) \le m - l - 1. \end{cases} \tag{10}$$

Comparing coefficients of the terms $1, x, x^2, \ldots, x^{m+n-l}$, we obtain the $(m+n-l+1) \times (m+n-l+1)$ coefficient matrix

$$\left[\begin{array}{ccccccccc} a_0 & & & 0 & b_0 & & 0 & -1 & & 0 \\ a_1 & a_0 & & & b_1 & \ddots & & & \ddots & \\ \vdots & a_1 & \ddots & & \vdots & \ddots & b_0 & 0 & & -1 \\ a_m & \vdots & & a_0 & & & & & & \\ 0 & a_m & & & b_n & & & & & \\ & 0 & \ddots & \vdots & 0 & \ddots & \vdots & & 0 & \\ & & \ddots & a_m & & \ddots & b_n & & & \\ \underbrace{0}_{n-l} & & & \underbrace{0}_{} & \underbrace{0}_{m-l} & & \underbrace{0}_{} & \underbrace{\quad}_{l+1} & & \end{array}\right].$$

This matrix has displacement rank no more than 3. A ΣLU representation is easily derived for this matrix. It can be shown that the triple (F, S, T) must be a polynomial multiple of (f_i, s_i, t_i). Note that this property remains true as long as $\deg(F) < \deg(f_{i-1})$ and $\deg(S) < n - d$, which then can be proven by induction on $\deg(F)$ (see Kaltofen 1992, Lecture 3.5, Lemma 1). Therefore, the dimension of the solution space of (10) is $l-d+1$ and the rank of the above coefficient matrix must be $m+n+d-l$. Its rank can be determined and a nonzero solution to (10) can be computed in $O(N(\log N)^2 \cdot \operatorname{loglog} N)$ arithmetic operations, where $N = m + n - l$. If one wishes to also have f_i, the system can be solved again for $l = d$. Alternately, one may perform a GCD computation $(f_i, s_i, t_i) = (F, S, T)/\mathrm{GCD}(S, T)$. Similar ideas can be applied to schemes of more than 2 polynomials (cf. Kalkbrener et al. 1993, Kaltofen 1993b). □

It is also easy to harness our algorithm for computing the determinant of a Toeplitz-like matrix for by (9) we have $\mathrm{Det}(A) = \mathrm{Det}(A_{1,1}) \cdot \mathrm{Det}(\Delta)$.

Corollary 2. *Given an $N \times N$ matrix A of displacement rank α in ΣLU representation, we have a randomized algorithm that computes the determinant of A in $O(\alpha^2 N (\log N)^2 \operatorname{loglog} N)$ expected field operations.*

Acknowledgement: Thanks to Martin Morf for his advice on the asymptotically fast Toeplitz-like solver, and to the referees for their suggestions.

Literature Cited

Bitmead, R. R. and Anderson, B. D. O., "Asymtotically fast solution of Toeplitz and related systems of linear equations," *Linear Algebra Applic.* **34**, pp. 103–116 (1980).

Brent, R. P., Gustavson, F. G., and Yun, D. Y. Y., "Fast solution of Toeplitz systems of equations and computation of Padé approximants," *J. Algorithms* **1**, pp. 259–295 (1980).

Canny, J. and Emiris, I., "An efficient algorithm for the sparse mixed resultant," in *Proc. AAECC-10*, Springer Lect. Notes Comput. Sci. **673**, edited by G. Cohen, T. Mora, and O. Moreno; pp. 89–104, 1993.

Coppersmith, D., "Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm," *Math. Comput.* **62/205**, pp. 333–350 (1994).

Delsarte, P., Genin, Y. V., and Kamp, Y. G., "A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile," *IEEE Trans. Acoustics, Speech, and Signal Process.* ASSP-**33/4**, (1985).

DeMillo, R. A. and Lipton, R. J., "A probabilistic remark on algebraic program testing," *Information Process. Letters* **7/4**, pp. 193–195 (1978).

Díaz, A., Hitz, M., Kaltofen, E., Lobo, A., and Valente, T., "Process scheduling in DSC and the large sparse linear systems challenge," in *Proc. DISCO '93*, Springer Lect. Notes Comput. Sci. **722**, edited by A. Miola; pp. 66–80, 1993. Available from anonymous@ftp.cs.rpi.edu in directory kaltofen.

Feng, G. L., Wei, V. K., Rao, T. R. N., and Tzeng, K. K., "True designed-distance decoding of a class of algebraic-geometric codes, Part II: fast algorithms and Toeplitz-block Toeplitz matrices," *IEEE Trans. Inf. Theory*, to appear (1994).

Gohberg, I., Kailath, T., and Koltracht, I., "Efficient

solution of linear systems of equations with recursive structure," *Linear Algebra Applic.* **80**, pp. 81–113 (1986).

Gohberg, I. C. and Semencul, A. A., "On the inversion of finite Toeplitz matrices and their continous analogues," *Mat. Issled.* **2**, pp. 201–233 (1972). In Russian. Math. Rev. MR **50**#5524.

Hong, H., "Quantifier elimination for formulas constrained by quadratic equations via slope resultants," *The Computer J.* **36/5**, pp. 439–449 (1993).

Kailath, T., Kung, S.-Y., and Morf, M., "Displacement ranks of matrices and linear equations," *J. Math. Analysis Applications* **68**, pp. 395–407 (1979).

Kalkbrener, M., Sweedler, M., and Taylor, L., "Low degree solutions to linear equations with $K[x]$ coefficients," *J. Symbolic Comput.* **16/1**, pp. 75–81 (1993).

Kaltofen, E., "Efficient Solution of Sparse Linear Systems," *Lect. Notes*, Dept. Comput. Sci., Rensselaer Polytech. Inst., Troy, New York, 1992. Available from `anonymous@ftp.cs.rpi.edu` in directory `kaltofen`.

Kaltofen, E., "Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems," in *Proc. AAECC-10*, Springer Lect. Notes Comput. Sci. **673**, edited by G. Cohen, T. Mora, and O. Moreno; pp. 195–212, 1993. Available from `anonymous@ftp.cs.rpi.edu` in directory `kaltofen`.

Kaltofen, E., "Direct proof of a theorem by Kalkbrener, Sweedler, and Taylor," *SIGSAM Bulletin* **27/4**, p. 2 (1993b). Available from `anonymous@ftp.cs.rpi.edu` in directory `kaltofen`.

Kaltofen, E., "Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems," *Math. Comput.*, to appear (1994). Available from `anonymous@ftp.cs.rpi.edu` in directory `kaltofen`.

Kaltofen, E. and Saunders, B. D., "On Wiedemann's method of solving sparse linear systems," in *Proc. AAECC-9*, Springer Lect. Notes Comput. Sci. **539**; pp. 29–38, 1991. Available from `anonymous@ftp.cs.rpi.edu` in directory `kaltofen`.

Moenck, R. T., "Fast computation of GCDs," *Proc. 5th ACM Symp. Theory Comp.*, pp. 142–151 (1973).

Morf, M., "Doubling algorithms for Toeplitz and related equations," in *Proc. 1980 IEEE Internat. Conf. Acoust. Speech Signal Process.*; IEEE, pp. 954–959, 1980.

Pan, V., "Parameterization of Newton's iteration for computations with structured matrices and applications," *Computers Math. Applic.* **24/3**, pp. 61–75 (1992).

Sasaki, T. and Furukawa, A., "Secondary polynomial remainder sequence and an extension of the subresultant theory," *J. Inform. Process* **7/3**, pp. 176–184 (1984).

Schwartz, J. T., "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM* **27**, pp. 701–717 (1980).

Strassen, V., "The computational complexity of continued fractions," *SIAM J. Comput.* **12/1**, pp. 1–27 (1983).

Trench, W., "An algorithm for the inversion of finite Toeplitz matrices," *SIAM J. Appl. Math.* **12**, pp. 515–522 (1964).

Zippel, R., "Probabilistic algorithms for sparse polynomials," *Proc. EUROSAM '79, Springer Lec. Notes Comp. Sci.* **72**, pp. 216–226 (1979).

Zippel, R., *Effective Polynomial Computations*; Kluwer Academic Publ., Boston, Massachusetts, 1993; 384pp.