Proceedings of PASCO'94

Parallel Solution of Toeplitz and Toeplitz-Like Linear Systems Over Fields of Small Positive Characteristic[1]

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

Victor Pan
Department of Mathematics
and Computer Science
Lehman College
City University of New York
Bronx, New York 10468
Inter-Net: `VPAN@lcvax.lehman.cuny.edu`

Abstract: We show that over a field of characteristic p the solution to a non-singular system of n linear equations in n unknows, with $2 \leq p < n$, whose coefficient matrix is of displacement rank α and which is given as a sum of α LU-products of Toeplitz matrices, can be computed in parallel with randomization simultaneously in $O((\log n)^3)$ time and a total work of $O(\max\{\alpha n, p^2\}\, n \times \log n \operatorname{loglog} n)$. A time unit represents an arithmetic operation in the coefficient field. Our solution is based on our recursive parallel triangulation technique for processor-efficient parallel linear system solving over fields of characteristic p. In particular, we show that our recursive parallel triangulation technique can be implemented in a way that preserves Toeplitz-likeness.

Keywords: Parallel algorithm, processor-efficiency, Toeplitz system, displacement rank, abstract coefficient field, positive characteristic, Le Verrier method, Wiedemann method

1 Introduction

In a Toeplitz matrix a single constant entry runs down along each diagonal. We seek parallel algorithms that solve $n \times n$ systems of linear equations in poly-log parallel time; that is, in $(\log n)^{O(1)}$ parallel arithmetic operations in the coefficient field. Furthermore, the algorithms are to minimize the number of processors. An algorithm is presented that in poly-log parallel time with $n \cdot \max\{p^2, n\}$ processors computes the solution of a linear system whose coefficient matrix is Toeplitz and whose field of coefficients has characteristic p, where $2 \leq p < n$. Our result is achieved by drawing on a far-reaching generalization of the notion of Toeplitz matrices to matrices of small displacement rank, so-called Toeplitz-like matrices (Kailath et al. 1979).

The notion of Toeplitz-like and Hankel-like matrices captures a wide class of matrices including block Toeplitz matrices, such as the Sylvester matrix of two polynomials, matrices that are a sum of a Toeplitz-like plus a Hankel-like matrix, and products and inverses of such matrices. The key idea is to represent a matrix as a sum of α matrix products, where α is much smaller than the row and column dimensions and where each product is, in the Toeplitz-like case, for instance, that of a lower triangular times an upper triangular Toeplitz matrix. We shall refer to this representation as the ΣLU representation of a matrix, and to α as the displacement rank. For pure Toeplitz matrices, for example, we have $\alpha \leq 2$. Toeplitz-like matrices are ubiquitous in symbolic computation as resultants and subresultants have this form (Brown and Traub 1971, Sasaki and

[1] This material is based on work supported in part by the National Science Foundation under Grant No. CCR-9319776 (first author) and Grant No. CCR-9020620 (second author) and by the PSC CUNY Award 664334 (second author).

Furukawa 1984, Hong 1993). Furthermore, block matrices are used, for instance, as a parallelization technique by reducing the dimensions while increasing the running time of the arithmetic operations on the individual entries, which now are matrices rather than field elements. This approach leads to coarse grain parallel algorithms, in which the individual block operations are carried out on different computers. Coppersmith's block Wiedemann algorithm (Coppersmith 1994, Kaltofen 1995) for solving sparse linear systems in parallel is an example of this approach. An intermediately arising subproblem there is the solution of a block Toeplitz matrix.

We give a parallel randomized algorithm that computes the exact solution of a Toeplitz-like non-singular linear system whose $n \times n$ coefficient matrix is given in ΣLU representation and whose coefficient field has small positive characteristic p, where $2 \leq p < n$. Our algorithm can solve a Toeplitz-like system of displacement rank α in $O((\log n)^3)$ expected parallel time with roughly $n \cdot \max\{\alpha n, p^2\}$ processors, each of which performs coefficient field arithmetic in unit time. The significance of our result lies in the fact that we can accomplish a poly-log parallel algorithm with a number of processors that is only quadratic in the dimension. For very small fields, such as the Galois field with 2 elements, the parallel time increases by a factor of $\log n$. Our processor estimates are rough in the sense that we will not specify a concrete parallel model and instead follow the work-time presentation framework of JáJá (1992, §1.5). That approach incorporates the notion of "scalability" of the parallel algorithm: with $n^{1.5}$ processors, for example, our algorithm takes $O(\max\{\alpha\sqrt{n}, p^2/\sqrt{n}\}\,(\log n)^3)$ parallel time. For now, but not in our later theorems, we also ignore factors like $\operatorname{loglog} n$ in the processor count.

The parallel solution of Toeplitz and Toeplitz-like linear systems with few processors is extensively investigated by Bini and Pan (see Bini et al. 1991, Pan 1992b, Bini and Pan 1993 and 1995). Indeed, Pan's algorithms for solving a Toeplitz system in $O((\log n)^2)$ parallel time with roughly n^2 processors is an important substep in the $O((\log n)^2)$ parallel time solution of general linear systems with roughly n^3 processors (Kaltofen and Pan 1991). The Le Verrier/Csanky approach in all the methods leads to technical difficulties when the coefficient fields have small positive characteristic. The difficulties are overcome by the "recursive parallel triangulation" technique of Kaltofen and Pan (1992). In this paper we show that the recursive parallel triangulation can be realized in a manner that keeps intermediately computed matrices Toeplitz-like. For fields of characteristic $p = 2$ this is quite clear from the randomizations invented by Kaltofen and Saunders (1991). For this case we present the entire algorithm in §3. When $p > 2$ additional complications arise, which we overcome by switching to block Toeplitz matrices. With this change in mind, we will describe in §2 the basic displacement operators for entries from a non-commutative algebra. The intricate recursive triangulation algorithm with block triangular matrices is then explained in §4. We note that if the input matrix is a pure Toeplitz matrix, the matrices computed by the recursive invocations do not remain Toeplitz and the full theory of Toeplitz-likeness comes to bear. We remark that the positivity of the field characteristic rules out an iterative approach like in (Pan 1992a).

Singular Toeplitz and Toeplitz-like systems can be solved with the same number of processors by determining the rank by binary search for non-singular leading principal submatrices (see, e.g., Kaltofen and Pan 1992, §3). As a consequence we can compute the greatest common divisor of two polynomials of degree n over a field of characteristic p, where $2 \leq p < 2n$, in $O((\log n)^4)$ expected parallel arithmetic operations with roughly $n \cdot \max\{n, p^2\}$ processors (cf. Kaltofen 1994, Example in §4). However, it is not known to us how to extend to small positive characteristic the techniques in (Kaltofen and Pan 1992, §3) that avoid the extra $\log n$ factor in the parallel time which is induced by binary search.

We remark that no processor-efficient parallel algorithm (in the sense of Karp and Ramachandran 1990) for Toeplitz-like linear systems is known. Recently, it has been shown (Kaltofen 1994) that Toeplitz-like linear systems can be solved in $O(\alpha^2 n(\log n)^2 \operatorname{loglog} n)$ sequential field operations. Thus a processor-efficient poly-log time parallel algorithm can use no more than roughly $\alpha^2 n$ processors. No direct (meaning non-iterative) parallel algorithm of this sort is known even when $p = 0$ and the coefficient matrix is Toeplitz.

2 Displacement Operators Over Non-Commutative Algebras

We now introduce well-known tools from the theory of Toeplitz-like matrices (Kailath et al. 1979). We consider $n \times n$ matrices over a non-commutative algebra $\mathbb{A}$. This generalization is necessary because in §4 we must deal with block matrices. Define the lower-shift matrix

$$Z = \begin{bmatrix} 0 & & \\ 10 & 0 & \\ 1\ddots & & \\ 0 & \ddots & \\ & & 10 \end{bmatrix}$$

and define the matrix shift operators

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

The matrix $\downarrow A$ is equal to A after being shifted down by one row, filling the first row by zeros, and the matrix $\Rsh A$ 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

$$\phi_+(A) = A - \downarrow(\Rsh A) = A - Z\,A\,Z^{\text{tr}}.$$

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,

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

has the unique solution

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

where $L[\![y]\!]$ denotes a lower-triangular Toeplitz matrix whose first column is y and $U[\![z^{\text{tr}}]\!]$ denotes an upper triangular Toeplitz matrix whose first row is z^{tr}. We shall call the vectors $y_1, \ldots, y_\alpha$ and $z_1, \ldots, z_\alpha$ in

$$Y = \sum_{j=1}^{\alpha} y_j z_j^{\text{tr}} = [\, y_1 \mid y_2 \mid \ldots \mid y_\alpha \,] \cdot \begin{bmatrix} z_1^{\text{tr}} \\ \hline z_2^{\text{tr}} \\ \hline \vdots \\ \hline z_\alpha^{\text{tr}} \end{bmatrix} \tag{3}$$

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 $O(\alpha n)$ ring elements.

A ubiquitous problem in our algorithms will be to derive the ΣLU representation for the product of Toeplitz-like matrices given by their ΣLU representations. 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

$$\phi_+(X) = X - Z_m X Z_n^{\text{tr}} \qquad \text{for} \quad X \in \mathbb{A}^{m \times n}.$$

For $m \le n$ we may remove the last $n - m$ rows from L in (2) and y in (3), while for $m \ge n$ we may remove the last $m - n$ columns from U in (2) and z in (3). Suppose now that we are given α generators of $\phi_+(G)$, where $G \in \mathbb{A}^{l \times m}$, and β generators for $\phi_+(H)$, where $H \in \mathbb{A}^{m \times n}$. We may compute $\alpha + \beta + 1$ generators for $\phi_+(GH)$ as follows (Pan 1992b, Proposition A.3): First, observe that $I_m = Z_m^{\text{tr}} Z_m + e_m e_m^{\text{tr}}$, where I_m is the $m \times m$ identity matrix and e_m is the m^{th} unit vector. Therefore

$$\begin{aligned} &\phi_+(GH) \\ &= GH - Z_l G I_m H Z_n^{\text{tr}} \\ &= GH - (Z_l G Z_m^{\text{tr}})(Z_m H Z_n^{\text{tr}}) - Z_l G e_m e_m^{\text{tr}} H Z_n^{\text{tr}} \\ &= (G - Z_l G Z_m^{\text{tr}})H + Z_l G Z_m^{\text{tr}}(H - Z_m H Z_n^{\text{tr}}) \\ &\qquad\qquad\qquad\qquad\qquad\qquad - g h^{\text{tr}} \\ &= \phi_+(G)H + Z_l G Z_m^{\text{tr}} \phi_+(H) - g h^{\text{tr}}, \end{aligned} \tag{4}$$

where $g = Z_l G e_m \in \mathbb{A}^l$ and $h = Z_n H^{\text{tr}} e_m \in \mathbb{A}^n$. In (4) the product $\phi_+(G)H$, for example, requires the multiplication the right generators of G by the ΣLU representation of H, which can be accomplished by $O(\alpha\beta)$ triangular Toeplitz matrix-times-vector products over $\mathbb{A}$.

In our algorithms it will also be necessary to invert triangular Toeplitz matrices over $\mathbb{A}$. Since

$$\begin{bmatrix} a_1 & & & \\ a_2 & a_1 & & 0 \\ a_3 & a_2 & a_1 & \\ \vdots & & \ddots\ddots & \\ a_n a_{n-1} & \ldots & a_2 & a_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

is equivalent to

$$(a_1 + \cdots + a_n t^{n-1})(x_1 + \cdots + x_n t^{n-1}) \equiv b_1 + \cdots + b_n t^{n-1} \pmod{t^n}$$

we have

$$\begin{bmatrix} a_1 & & & \\ a_2 & a_1 & & 0 \\ a_3 & a_2 & a_1 & \\ \vdots & & \ddots\ddots & \\ a_n a_{n-1} & \ldots & a_2 & a_1 \end{bmatrix}^{-1} =$$

$$\begin{bmatrix} c_1 & & & 0 \\ c_2 & c_1 & & \\ c_3 & c_2 & c_1 & \\ \vdots & & \ddots & \ddots \\ c_n & c_{n-1} & \dots c_2 & c_1 \end{bmatrix}$$

with

$$\begin{aligned} (c_1 + \cdots + c_n t^{n-1})&(a_1 + \cdots + a_n t^{n-1}) \\ &\equiv 1 \pmod{t^n}. \end{aligned}$$

The entries c_i can always be found provided that a_1 is invertible in $\mathbb{A}$. In that case, Newton iteration computes the power series inverse as

$$\begin{aligned} &c^{(0)}(t) = a_1^{-1}, \\ &c^{(i)}(t) = c^{(i-1)}(t)(2 - a(t)c^{(i-1)}(t)) \bmod t^{2^i}, \end{aligned}$$

where $a(t) = a_1 + a_2 t + \cdots$ and $c^{(i)}(t) = c_1 + c_2 t + \cdots + c_{2^i} t^{2^i - 1}$. Note that

$$\begin{aligned} 1 - c^{(i)}(t)a(t) &\equiv (1 - c^{(i-1)}(t)a(t))^2 \\ &\equiv 0 \pmod{t^{2^i}}. \end{aligned}$$

In conclusion, we can compute the entries c_i in $O((\log n)^2)$ parallel time and $O(n \log n \operatorname{loglog} n)$ work in terms of operations in $\mathbb{A}$, the latter by the polynomial multiplication algorithm of Cantor and Kaltofen (1991).

3 Coefficient Fields of Characteristic 2

We first present the case where the field of entries has characteristic 2. This case is a direct application of the "recursive parallel triangulation" paradigm of (Kaltofen and Pan 1992) and the theory of Toeplitz-like systems (Kailath et al. 1979).

Algorithm 1

Input: Vectors $y_1, \ldots, y_\alpha,\ z_1, \ldots, z_\alpha \in \mathbb{K}^n$ such that $A = \sum_{j=1}^{\alpha} L[\![y_j]\!] U[\![z_j^{\text{tr}}]\!] \in \mathbb{K}^{n \times n}$ is non-singular, where $\mathbb{K}$ is a field of characteristic 2. Furthermore, a vector $b \in \mathbb{K}^n$.

Output: The vector $A^{-1}b$.

Step 1: In later steps it is necessary that the coefficient matrix has no multiple eigenvalues. A key technique is to precondition the matrix A (see Kaltofen and Pan 1992, Proposition 1): *Compute a ΣLU representation of length $\alpha + 4$ for $\widetilde{A} = VAW$ where*

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

and

$$W = \begin{bmatrix} w_1 & & & 0 \\ w_2 & w_1 & & \\ w_3 & w_2 & w_1 & \\ \vdots & & \ddots & \ddots \\ w_n & w_{n-1} & \dots w_2 & w_1 \end{bmatrix}$$

have random entries from a set $S \subset \mathbb{K}$.

Step 2: This step utilizes the fundamental idea of parameterized Newton iteration for the inversion of the characteristic matrix of the Toeplitz-like matrix (Pan 1992b, Proposition 3.1). *Compute the ΣLU representation of*

$$\widetilde{A}(\lambda) = \sum_{i=0}^{n-1} \lambda^i \widetilde{A}^i \equiv (I - \lambda \widetilde{A})^{-1} \pmod{x^n}. \quad (5)$$

Step 3: *From the ΣLU representation of (5) compute $s_i = \operatorname{Trace} \widetilde{A}^i$. Furthermore, pick two random vectors $u, y \in S^n$ and from the ΣLU representation of (5) compute $a_i = u^{\text{tr}} \widetilde{A}^i y$ for all $0 \le i \le 2n - 1$.*

Step 4: The quantities s_i and a_i computed in Steps 2 and 3 result in the $2n \times n$ nonsingular system for the coefficients of the characteristic polynomial

$$\begin{aligned} \operatorname{Det}(\lambda I - \widetilde{A}) = \lambda^n - c_1 \lambda^{n-1} - c_2 \lambda^{n-2} - \\ \cdots - c_{n-1}\lambda - c_n \quad (6) \end{aligned}$$

of $\widetilde{A}$:

$$\begin{bmatrix} 1 & & & & \\ s_1 & 2 & & \mathbf{0} & \\ s_2 & s_1 & \ddots & & \\ \vdots & & & & \\ s_{n-2} & \cdots & & n-1 & \\ s_{n-1} & s_{n-2} & \cdots & s_1 & n \\ & & & & \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \\ a_n & a_{n-1} & \cdots & a_2 & a_1 \\ \vdots & a_n & \ddots & \vdots & a_2 \\ & \vdots & & & \vdots \\ a_{2n-3} & & & a_{n-1} & \\ a_{2n-2} & a_{2n-3} & \cdots & a_n & a_{n-1} \end{bmatrix} \times \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \\ \\ a_n \\ a_{n+1} \\ a_{n+2} \\ \vdots \\ \\ a_{2n-1} \end{bmatrix} \tag{7}$$

In (7) the first n equations are the Newton identities and the second n equations express the fact that the characteristic polynomial of $\widetilde{A}$ linearly generates the sequence $a_0, a_1, \ldots$ Since $\widetilde{A}$ has no multiple eigenvalues, with high probability the second n equations are linearly independent. The system is rearranged as follows: first remove every row of the first n rows with an even integer on the diagonal; second, reorder the columns such that the odd numbered columns precede the even numbered ones; and finally, reorder the rows numbered $n+1, \ldots, 2n$ to $n+1, n+3, \ldots, n+2, n+4, \ldots$ We obtain the system

$$\left[\begin{array}{c|c} L_1 & L_2 \\ \hline T_1 & T_2 \\ \hline T_3 & T_4 \end{array}\right] \cdot \left[\begin{array}{c} c' \\ \hline c'' \end{array}\right] = \left[\begin{array}{c} s' \\ \hline a' \\ \hline a'' \end{array}\right] \tag{8}$$

where

$$L_1 = \begin{bmatrix} 1 & & \\ s_2 & 1 & \mathbf{0} \\ s_4 & s_2 & 1 \\ \vdots & & \ddots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil \times \lceil n/2 \rceil},$$

$$L_2 = \begin{bmatrix} 0 & & \\ s_1 & 0 & \mathbf{0} \\ s_3 & s_1 & 0 \\ \vdots & & \ddots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil \times \lfloor n/2 \rfloor},$$

$$T_1 = \begin{bmatrix} a_{n-1} & a_{n-3} & \cdots \\ a_{n+1} & a_{n-1} & \\ \vdots & \ddots & \ddots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil \times \lceil n/2 \rceil},$$

$$T_2 = \begin{bmatrix} a_{n-2} & a_{n-4} & \cdots \\ a_n & a_{n-2} & \\ \vdots & \ddots & \ddots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil \times \lfloor n/2 \rfloor},$$

$$T_3 = \begin{bmatrix} a_n & a_{n-2} & \cdots \\ a_{n+2} & a_n & \\ \vdots & \ddots & \ddots \end{bmatrix} \in \mathbb{K}^{\lfloor n/2 \rfloor \times \lceil n/2 \rceil},$$

$$T_4 = \begin{bmatrix} a_{n-1} & a_{n-3} & \cdots \\ a_{n+1} & a_{n-1} & \\ \vdots & \ddots & \ddots \end{bmatrix} \in \mathbb{K}^{\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor},$$

and

$$c' = \begin{bmatrix} c_1 \\ c_3 \\ c_5 \\ \vdots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil}, \quad c'' = \begin{bmatrix} c_2 \\ c_4 \\ c_6 \\ \vdots \end{bmatrix} \in \mathbb{K}^{\lfloor n/2 \rfloor},$$

$$s' = \begin{bmatrix} s_1 \\ s_3 \\ s_5 \\ \vdots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil},$$

$$a' = \begin{bmatrix} a_n \\ a_{n+2} \\ a_{n+4} \\ \vdots \end{bmatrix} \in \mathbb{K}^{\lceil n/2 \rceil}, \; a'' = \begin{bmatrix} a_{n+1} \\ a_{n+3} \\ a_{n+5} \\ \vdots \end{bmatrix} \in \mathbb{K}^{\lfloor n/2 \rfloor}.$$

We eliminate T_1 and T_3 by

$$\underbrace{\left[\begin{array}{c|c|c} I & 0 & 0 \\ \hline -T_1L_1^{-1} & I & 0 \\ \hline -T_3L_1^{-1} & 0 & I \end{array}\right] \cdot \left[\begin{array}{c|c} L_1 & L_2 \\ \hline T_1 & T_2 \\ \hline T_3 & T_4 \end{array}\right]}_{\left[\begin{array}{c|c} L_1 & L_2 \\ \hline 0 & \Delta_1 \\ \hline 0 & \Delta_2 \end{array}\right]} \cdot \left[\begin{array}{c} c' \\ \hline c'' \end{array}\right]$$

$$= \underbrace{\left[\begin{array}{c} s' \\ \hline a' - T_1L_1^{-1}s' \\ \hline a'' - T_3L_1^{-1}s' \end{array}\right]}_{\left[\begin{array}{c} s' \\ \hline \delta' \\ \hline \delta'' \end{array}\right]},$$

where

$$\Delta_1 = T_2 - T_1L_1^{-1}L_2 \quad \text{and} \quad \Delta_2 = T_4 - T_3L_1^{-1}L_2.$$

Compute the ΣLU representation of the Schur complements Δ_1 and Δ_2 and compute the vectors δ' and δ''. Note that L_1^{-1} has as the inverse of a triangular Toeplitz matrix displacement rank one. Therefore, by the product formula (4) the displacement ranks of each Δ_i is no more than 8.

Step 5: Here we compress the non-singular $n \times \lfloor n/2 \rfloor$ system

$$\left[\begin{array}{c} \Delta_1 \\ \hline \Delta_2 \end{array}\right] \cdot c'' = \left[\begin{array}{c} \delta' \\ \hline \delta'' \end{array}\right]$$

to a $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ non-singular system. *Select a random $\lfloor n/2 \rfloor \times n$ unit upper triangular Toeplitz matrix*

$$E = [\, E_1 \mid E_2 \,] = \begin{bmatrix} 1 e_2\, e_3 \ldots e_{\lfloor n/2 \rfloor + 1} \cdots & & e_n \\ 1\; e_2 \ldots & \ddots & e_{n-1} \\ 0 \; \ddots \; \ddots & & \vdots \\ 1 & e_2 & \ldots e_{n - \lfloor n/2 \rfloor + 1} \end{bmatrix},$$

where $E_1 \in \mathbb{K}^{\lfloor n/2 \rfloor \times \lceil n/2 \rceil}$ and $E_2 \in \mathbb{K}^{\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor}$ with the entries e_i being random elements in the set S. By Theorem 2 of Kaltofen and Saunders (1991) the matrix $\Delta = E_1\Delta_1 + E_2\Delta_2 \in \mathbb{K}^{\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor}$ is with high probability non-singular. *Compute a ΣLU representation for the matrix Δ and compute the vector $\delta = E_1\delta' + E_2\delta''$.* By (4) the matrix Δ has displacement rank no more than 21.

Step 6: *By recursive application of the entire algorithm, solve $\Delta\, c'' = \delta$.*

Step 7: We can now back-substitute in (8) to determine the coefficients of (6): *Compute $c' = L_1^{-1}(s' - L_2c'')$.*

Singularity of A can be discovered in two ways: either, Δ becomes zero during the triangulation process, or $c_n = 0$. In the former case, A is singular with high probability, whereas in the later case A is definitely singular.

Step 8: By the Cayley/Hamilton theorem we have

$$\widetilde{A}^{-1} = \frac{1}{c_n}\Big(\widetilde{A}^{n-1} - c_1\widetilde{A}^{n-2} - \cdots - c_{n-1}I\Big).$$

Compute $\tilde{b} = Vb$ and, with the help of the ΣLU representation for (5) computed in Step 2, $\tilde{b}^{[i]} = \widetilde{A}^i\, \tilde{b}$ for all $1 \le i \le n-1$. Finally, determine

$$A^{-1}b = W\widetilde{A}^{-1}\tilde{b} = \frac{1}{c_n}W\Big(\tilde{b}^{[n-1]} - c_1\tilde{b}^{[n-2]} - \cdots - c_{n-1}\tilde{b}\Big). \quad \square$$

Aside from the basic techniques of the theory of Toeplitz-like matrices, we think that there are as many as 5 distinct ideas incorporated in the above algorithm: first, the Le Verrier/Csanky approach to linear system solving; second, Wiedemann's coordinate recurrence projections; third, Pan's parameterized Newton iteration for Toeplitz-like matrices; fourth, preconditioning á la Borodin et al. (1982) which by (Kaltofen and Saunders 1991) can be restricted to triangular Toeplitz multipliers; and fifth, our recursive parallel triangulation. The parallel running time, the work, which is the total number of all arithmetic operations performed by all processors, and the success probability of Algorithm 1 is summarized in the following theorem.

Theorem 1. *Algorithm 1 picks $O(n)$ random elements from a subset $S \subset \mathbb{K}$. It performs $O(\alpha n^2 \log n \operatorname{loglog} n)$ arithmetic operations in total work, and has a parallel time of $O((\log n)^3)$. With probability no less than $1 - 9n^2/\mathrm{card}(S)$ it returns the correct answer.*

Note that if the field $\mathbb{K}$ has fewer than n^3, say, elements we must perform the entire algorithm in a finite algebraic extension of $\mathbb{K}$ in order to guarantee success with a positive probability. Therefore, the cost of a single arithmetic operation for our algorithm costs at least $\Omega(\log n)$ bit operations.

Proof. Algorithm 1 fails to produce an answer if the minimum polynomial of $\widetilde{A}$ is not its characteristic polynomial, that with probability no more than $4n^2/\text{card}(S)$ (Kaltofen and Pan 1992, Proposition 1), or if the linear generator of a_i is not the characteristic polynomial, that with probability no more than $2n/\text{card}(S)$ (Kaltofen and Pan 1991, Lemma 2), or if the matrix E of Step 5 constructs a singular Δ, that with probability no more than $n(n+1)/(2\,\text{card}\,S)$ (cf. Kaltofen and Saunders 1991, Theorem 2), or if the recursive call of Step 6 fails. Therefore, the probability of failure, $Pf(n,\alpha)$ is bounded as

$$\begin{aligned} &Pf(n,\alpha)\\ &\qquad \le Pf(\lfloor n/2\rfloor, 21) + (9n+5)n/(2\,\text{card}\,S). \end{aligned}$$

Step 1 by the product formula (4) reduces to $O(\alpha^2)$ triangular Toeplitz matrix-times-vector products all of which in $O(\log n)$ parallel time cost $O(\alpha^2 n \log n \operatorname{loglog} n)$ work. The cost of Steps 2 and 3 is by known methods (see, e.g., Bini and Pan 1995) no more than $O(\alpha n^2 \log n \operatorname{loglog} n)$ work in $O((\log n)^2)$ time. Step 3 requires the inversion of a triangular Toeplitz matrix, that in $O(n \log n \operatorname{loglog} n)$ work and $O((\log n)^2)$ parallel time via a power series reciprocal (see §2), and product constructions which because of constant displacement ranks cost only $O(n \log n \operatorname{loglog} n)$ work and $O(\log n)$ parallel time. The same is true for Step 5. Step 7 essentially leads to 2 triangular Toeplitz matrix-times-vector products and its running time is negligible. Finally, the computation of $\tilde{b}^{[i]}$ in Step 8 is similar to the computation of a_i in Step 3 and the final sum for $A^{-1}b$ requires $O(n^2)$ work and $O(\log n)$ time. Therefore, we have the following recursive relations to the work $Wk(n,\alpha)$ and parallel time $T(n,\alpha)$:

$$\begin{aligned} Wk(n,\alpha) &\le Wk(\lfloor n/2\rfloor, 21)\\ &\qquad + d_1 \alpha n^2 \log n \operatorname{loglog} n,\\ T(n,\alpha) &\le T(\lfloor n/2\rfloor, 21) + d_2(\log n)^2, \end{aligned}$$

where d_1 and d_2 are positive constants, which easily yields the stated complexities. ⊠

4 Coefficient Fields of Characteristic > 2

Algorithm 1 can be generalized to coefficient fields $\mathbb{K}$ of characteristic p with $3 \le p < n$. All but Step 4 are valid for any field of sufficient cardinality. In this section we show how Step 4 is modified. We shall reduce the linear system (7) to a $\lfloor n/p\rfloor \times \lfloor n/p\rfloor$ linear system of displacement rank $O(p)$. We first proceed as in Step 4 and rearrange (7) as follows. We remove those rows among the first n rows that have an integer divisible by p, that is, a zero in $\mathbb{K}$ on the diagonal. We reorder the columns such that the p^{th}, $(2p)^{\text{th}}, \ldots$ columns are placed after columns whose number is not divisible by p. Finally, we move the rows numbered $n+p$, $n+2p, \ldots$ below the rows $n+1$, $n+2, \ldots, n+p-1$, $n+p+1, \ldots$ The resulting system has a block shape similar to (8):

$$\left[\begin{array}{c|c} \boldsymbol{L}_1 & \boldsymbol{L}_2 \\ \hline \boldsymbol{T}_1 & \boldsymbol{T}_2 \\ \hline \boldsymbol{T}_3 & \boldsymbol{T}_4 \end{array}\right] \cdot \left[\begin{array}{c} c' \\ \hline c'' \end{array}\right] = \left[\begin{array}{c} s' \\ \hline a' \\ \hline a'' \end{array}\right], \tag{9}$$

where $\boldsymbol{L}_1$ is a triangular matrix of dimensions $(n - \lfloor n/p\rfloor) \times (n - \lfloor n/p\rfloor)$, $\boldsymbol{L}_2$ is of dimensions $(n - \lfloor n/p\rfloor) \times \lfloor n/p\rfloor$, and $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ have $n - \lfloor n/p\rfloor$ rows and $\boldsymbol{T}_3$ and $\boldsymbol{T}_4$ have $\lfloor n/p\rfloor$ rows. The main difference to the case $p = 2$ occurs in $\boldsymbol{L}_1$. On the diagonal, $\boldsymbol{L}_1$ has the elements $1, 2, \ldots, p-1$, $1, 2, \ldots$ and is therefore not a Toeplitz matrix. However, by blocking this matrix into blocks of size $(p-1) \times (p-1)$, this matrix becomes a block Toeplitz matrix. Note that the blocks in the last row/column may have fewer than $p-1$ rows/columns. There are $\lceil (n - \lfloor n/p\rfloor)/(p-1)\rceil = n/p + O(1)$ blocks in this matrix. Similar blocking is possible in the remaining matrices: $\boldsymbol{T}_1$ also has $(p-1) \times (p-1)$ blocks, while $\boldsymbol{L}_2$ and $\boldsymbol{T}_2$ have $(p-1) \times 1$ blocks, $\boldsymbol{T}_3$ has $1 \times (p-1)$ blocks, and $\boldsymbol{T}_4$ has 1×1 blocks. Again the blocks in the last rows/columns may be of smaller dimensions. The usage of bold face in (9) indicates that these matrices are blocked in that way. It should be noted that number of blocks in the rows and columns of all matrices is $n/p + O(1)$ and that they are block Toeplitz matrices.

We shall perform elimination as in Step 4 of Algorithm 1. The resulting compressed Schur complement

$$\begin{aligned} \Delta = \boldsymbol{E}_1(&\boldsymbol{T}_2 - \boldsymbol{T}_1 \boldsymbol{L}_1^{-1} \boldsymbol{L}_2)\\ &+ \boldsymbol{E}_2(\boldsymbol{T}_4 - \boldsymbol{T}_3 \boldsymbol{L}_1^{-1} \boldsymbol{L}_2), \end{aligned} \tag{10}$$

where $\boldsymbol{E}_1$ has $1 \times (p-1)$ blocks and $\boldsymbol{E}_2$ has 1×1 blocks, is a $\lfloor n/p\rfloor \times \lfloor n/p\rfloor$ matrix with 1×1 blocks.

The shift matrices used for the block matrices are block matrices $\boldsymbol{Z}$ with unit-diagonal matrices on the block subdiagonal. For the $(p-1) \times (p-1)$ blocks these unit-diagonal matrices are, of course, $(p-1) \times (p-1)$ identity matrices. For example,

$$\boldsymbol{\phi}_+(\boldsymbol{T}_3) = \boldsymbol{T}_3 - Z_{\lfloor n/p\rfloor} \boldsymbol{T}_3 \boldsymbol{Z}_{\boldsymbol{\nu}}^{\text{tr}}$$

where $\boldsymbol{\nu} = \lceil (n - \lfloor n/p \rfloor)/(p-1) \rceil$ and

$$\boldsymbol{Z_\nu} = \left[\begin{array}{c|c|c|c|c} 0 & \dots & & & 0 \\ \hline I_{p-1} & 0 & \dots & & 0 \\ \hline 0 & I_{p-1} & \ddots & & \vdots \\ \hline \vdots & & \ddots & 0 & 0 \\ \hline 0 & \dots & 0 & I' & 0 \end{array}\right]$$

is a $\boldsymbol{\nu} \times \boldsymbol{\nu}$ block sub-diagonal matrix, where the last sub-diagonal entry I' is a rectangular diagonal matrix of dimensions $(n - \lfloor n/p \rfloor \bmod (p-1)) \times (p-1)$ with 1's on the diagonal.

Block generators in the sense of (3) can be derived for all matrices. In case of $\boldsymbol{T}_3$, for example, we have $\boldsymbol{\phi}_+(\boldsymbol{T}_3) = \boldsymbol{y}_1 \boldsymbol{z}_1^{\mathrm{tr}} + \boldsymbol{y}_2 \boldsymbol{z}_2^{\mathrm{tr}}$ where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ have blocks of dimension $1 \times (p-1)$ and $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ have blocks of dimension $(p-1) \times (p-1)$, except the last block which may be smaller. Incidentally, since $\boldsymbol{T}_3$ is a block Toeplitz matrix $\boldsymbol{y}_1$ is the first column of $\boldsymbol{T}_3$ with the first block entry in $\boldsymbol{z}_1^{\mathrm{tr}}$ the identity matrix and the other blocks being 0; the matrix $\boldsymbol{z}_2^{\mathrm{tr}}$ contains the remaining first row of $\boldsymbol{T}_3$ and the matrix $\boldsymbol{y}_1$ has a single 1 in the first entry of the first block. We emphasize that these generators can be constructed although the notion of rank is ill-defined for the block matrix $\boldsymbol{\phi}_+(\boldsymbol{T}_3)$.

We shall now briefly discuss the algorithm for deriving the block ΣLU representation for $\boldsymbol{L}_1^{-1}$. From the Newton iteration algorithm presented in §2 it follows that the inverse $\boldsymbol{L}_1^{-1}$ can be computed in $O(n/p \, \log n \, \mathrm{loglog}\, n)$ block operations. Note that for this purpose we may extend the blocks in the last row and column to dimension $(p-1) \times (p-1)$; afterwards, we may reduce the resulting inverse matrix by the corresponding rows and columns, since the involved matrices are lower triangular. The block operations amount to matrix arithmetic and a single triangular matrix inverse. Therefore the block generators for $\boldsymbol{\phi}_+(\boldsymbol{L}_1^{-1})$ can be computed in $O((\log n)^2 \log p)$ parallel time and a total work of $O(p^2 n \log n \times \mathrm{loglog}\, n)$ arithmetic operations in $\mathbb{K}$, that without using asymptotically fast matrix multiplications.

Once the block generators for all matrices in (10) are determined, we can apply the product construction (4). Again, the base operations are either $(p-1) \times (p-1)$ matrix products or matrix-times-vector products. All block matrices have a fixed number of block generators/block LU terms and are of dimensions $(n/p + O(1)) \times (n/p + O(1))$. By application of (4) we obtain no more than 21 block generators for Δ. The left generators have $1 \times (p-1)$ blocks and the right generators have $(p-1) \times 1$ blocks. For example, consider $\boldsymbol{E}_1 \boldsymbol{T}_1 \boldsymbol{L}_1^{-1} \boldsymbol{L}_2$:

matrix	left gen. block size	right gen. block size
$\boldsymbol{E}_1$	$1 \times (p-1)$	$(p-1) \times (p-1)$
$\boldsymbol{T}_1$	$(p-1) \times (p-1)$	$(p-1) \times (p-1)$
$\boldsymbol{L}_1^{-1}$	$(p-1) \times (p-1)$	$(p-1) \times (p-1)$
$\boldsymbol{L}_2$	$(p-1) \times (p-1)$	$(p-1) \times 1$

The shift matrices in $\boldsymbol{\phi}_+(\boldsymbol{E}_1 \boldsymbol{T}_1 \boldsymbol{L}_1^{-1} \boldsymbol{L}_2)$ have, luckily, 1×1 blocks. Therefore, the produced block generators actually are plain generators of length no more than $21(p-1)$. From known generators for $\boldsymbol{\phi}_+(\boldsymbol{L}_1^{-1})$ these generators can be computed in $O(\log n \; \log p)$ parallel operations in $\mathbb{K}$ with a total work of $O(p^2 n \log n \, \mathrm{loglog}\, n)$.

As there are only $O(\log_p n)$ recursive invocations we obtain by $\log n = (\log p)\,(\log_p n)$ the following fact:

Theorem 2. *For a field $\mathbb{K}$ of characteristic p with $3 \le p < n$ of cardinality at least $10n^2$ a modification of Steps 4 and 7 in Algorithm 1 allows the computation of $A^{-1}b$ in $O((\log n)^3)$ expected parallel time and a total expected work of $O(\max\{\alpha n, p^2\}\, n \log n \, \mathrm{loglog}\, n)$ arithmetic operations in $\mathbb{K}$.*

References

Bini, D., Gemignani, L., and Pan, V., "Improved parallel computations with matrices and polynomials," in *Proc. ICALP 91*, Springer Lect. Notes Comput. Sci. **510**, edited by J. Leach Albert, B. Monien, and E. Rodríguez Artalejo; pp. 520–531, 1991.

Bini, D. and Pan, V., "Parallel computations with Toeplitz-like and Hankel-like matrices," in *Proc. Internat. Symp. Symbolic Algebraic Comput. ISSAC '93*, edited by M. Bronstein; ACM Press, New York, N. Y., pp. 193–200, 1993.

Bini, D. and Pan, V., *Numerical and Algebraic Computations with Matrices and Polynomials*; Lecture Notes in Theor. Comput. Sci., edited by R. V. Book; Birkhäuser Boston, Inc., 1995. To appear.

Borodin, A., von zur Gathen, J., and Hopcroft, J. E., "Fast parallel matrix and GCD computations," *Inf. Control* **52**, pp. 241–256 (1982).

Brown, W. S. and Traub, J. F., "On Euclid's algorithm and the theory of subresultants," *J. ACM* **18**, pp. 505–514 (1971).

Cantor, D. G. and Kaltofen, E., "On fast multiplication of polynomials over arbitrary algebras," *Acta Inform.* **28**/7, pp. 693–701 (1991).

Available from anonymous@ftp.cs.rpi.edu in directory kaltofen.

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

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

JáJá, J., *An Introduction to Parallel Algorithms*; Addison-Wesley Publ. Comp., Reading, Massachusetts, 1992.

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

Kaltofen, E., "Asymptotically fast solution of Toeplitz-like singular linear systems," in *Proc. Internat. Symp. Symbolic Algebraic Comput. ISSAC '94*, edited by J. von zur Gathen and M. Giesbrecht; ACM Press, New York, N. Y., pp. 297–304, 1994. 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 (1995). Available from anonymous@ftp.cs.rpi.edu in directory kaltofen.

Kaltofen, E. and Pan, V., "Processor efficient parallel solution of linear systems over an abstract field," in *Proc. 3rd Ann. ACM Symp. Parallel Algor. Architecture*; ACM Press, pp. 180–191, 1991.

Kaltofen, E. and Pan, V., "Processor-efficient parallel solution of linear systems II: the positive characteristic and singular cases," *Proc. 33rd Annual Symp. Foundations of Comp. Sci.*, pp. 714–723 (1992).

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.

Karp, R. M. and Ramachandran, V., "Parallel algorithms for shared-memory machines," in *Handbook of Theoretical Computer Science, Algorithms and Complexity (Volume A)*, edited by J. van Leeuwen; Elsevier Science Publ., Amsterdam, pp. 869–941, 1990.

Pan, V., "Parallel solution of Toeplitzlike linear systems," *J. Complexity* **8**, pp. 1–21 (1992a).

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

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).