Appears in the Proc. of the 33rd Annual Symp. on Foundations of Comput. Sci., IEEE, 714–723 (1992).

Processor-Efficient Parallel Solution of Linear Systems II The Positive Characteristic and Singular Cases*

(Extended Abstract)

Erich Kaltofen[1] *and Victor Pan*[2]

[1]Department of Computer Science, Rensselaer Polytechnic Institute
Troy, New York 12180-3590; Inter-Net: kaltofen@cs.rpi.edu
[2]Department of Mathematics and Computer Science
Lehman College, City University of New York
Bronx, New York 10468; Inter-Net: vpan@lcvax.bitnet

Abstract. We show that over any field, the solution set to a system of n linear equations in n unknowns can be computed in parallel with randomization simultaneously in poly-logarithmic time in n and with only as many processors as are utilized to multiply two $n \times n$ matrices. A time unit represents an arithmetic operation in the field. For singular systems our parallel timings are asymptotically as fast as those for non-singular systems, due to our avoidance of binary search in the matrix rank problem, except when the field has small positive characteristic; in that case, binary search is avoided at a somewhat higher processor count measure.

1. Introduction

Processor-efficient parallel algorithms have been constructed in (Kaltofen and Pan 1991) for the problems of solving a non-singular linear system, computing the determinant of a matrix, and inverting a non-singular matrix; specifically, for an n-dimensional input the algorithms solve the designated problem in $O((\log n)^2)$ time using as many processors as are asymptotically required by the n-dimensional matrix multiplication problem (see Karp and Ramachandran 1990 for the notion of a processor-efficient parallel algorithm). Each processor performs addition, subtraction, multiplication, and division over the field of entries; division by zero is avoided for non-singular inputs by randomization. Our solution utilizes the Le Verrier (1840)/ Csanky (1976) algorithm (see also Preparata and Sarwate 1978 and Galil and Pan 1989) and is thus invalid for fields of small positive characteristic, in which case we have presented an n^3-processor algorithm, based on the Berkowitz (1984)/Chistov (1985) approach.

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-90-06077 and under Grant No. CDA-88-05910 (first author), and under Grant No. CCR-90-20690 and by the PSC CUNY Awards #661340 and #662478 (second author).

In §2 we shall show how to overcome that restriction. Specifically, if the field of entries, denoted by K, is of characteristic $1 < p \leq n$, we give processor-efficient Las Vegas randomized algorithms that compute the solution of an n-dimensional non-singular linear system, the determinant of a matrix, and the inverse of a non-singular matrix in parallel time at most $O((\log n)^3(\log\log n)/(\log p))$. Our algorithm picks for sufficiently large fields uniformly $O(n)$ random elements from a subset $S \subset \mathsf{K}$, and then reports failure with probability $O(n^2/\mathrm{card}(S))$. In that case, the parallel time is also asymptotically reduced by a factor $\log\log n$. If K is a small finite field, we need to work in a Galois extension and therefore asymptotically require a multiple of at most $\log n$ more random elements.

A second contribution focuses on the problem of solving a singular system. In (Kaltofen and Pan 1991) we have obtained a processor-efficient Las Vegas randomized parallel algorithm in the case $p = 0$ or $p > n$. However, our solution there required asymptotically more time, namely $O((\log n)^3)$. The extra $\log(n)$ factor was introduced by performing binary search in order to determine the rank of the system, which until now seems to have been the only way to accomplish processor-efficiency. In §3 we shall show how to compute the rank of an $n \times n$ matrix directly, i.e., in random $O((\log n)^2)$ time without utilizing more processors. If binary search is to be avoided for fields of characteristic $1 < p < n$, our processor count is currently $O(n^3 \log n \ \log\log n)$.

Our solutions are facilitated by several innovations in the design of parallel linear algebra algorithms, and draw from recent developments on a diversity of topics. The key underpinnings of the results in (Kaltofen and Pan 1991) are the coordinate recurrence method for solving sparse linear systems over arbitrary fields by Wiedemann (1986) and parameterized Newton iteration for the inversion of the matrix associated with the characteristic polynomial of a Toeplitz matrix (Pan 1990). The latter parallel algorithm explores the well-known

formula for the inverse of a Toeplitz matrix by Gohberg and Semencul (1972) (see also Trench 1964). A far-reaching generalization of the Trench/Gohberg& Semencul approach is the theory of the displacement rank of a matrix by Kailath et al. (1979) and the corresponding parallel algorithms by Pan (1990) and Bini and Pan (1992). Although both our small positive characteristic and our rank results can rely on these advances, several additional innovations come to bear on our solutions.

Our key idea in making linear system solving over a field of small positive characteristic p processor-efficient is that those Newton identities in Le Verrier's linear system for the coefficients of a characteristic polynomial that turn useless for such characteristic p can be replaced by additional linear equations obtained from the coordinate recurrences. Le Verrier's system can then be reduced to a new linear system of dimension $\lfloor n/p \rfloor$, which we may solve recursively. We call our technique "recursive parallel triangulation," since the recurring size-reduction of the intermediately arising linear systems is based on block-elimination by triangular matrices.

Our fast rank algorithm first uses the reduction by Kaltofen and Saunders (1991) of the rank determination of a general matrix to that of a certain Toeplitz matrix. The parallel solution of the latter problem is then based on Mulmuley's (1985) method. Mulmuley's algorithm, as given, multiplies the arising Toeplitz matrix by a certain diagonal matrix, which does not keep the displacement rank constant; we would have no means of computing in parallel the characteristic polynomial of a such perturbed matrix, which is needed by that algorithm, even with as many processors as are used by general matrix multiplication. We by-pass this difficulty by switching to the corresponding Hankel matrix and by modifying the perturbation such that the Hankel structure is preserved. Finally, the characteristic polynomial of the perturbed matrix can be computed in parallel with few processors by the algorithms for matrices of small displacement rank cited above.

The number of arithmetic processors used by our algorithms depends both on processor count of $n \times n$ parallel matrix multiplication and the processor count of computing in parallel the characteristic polynomial of an $n \times n$ Toeplitz or Hankel matrix. In light of the possibility that the former is asymptotically smaller than the latter, we shall introduce as our processor count measure

$$M^*(n) := \max\{M(n)\ \log n,\ n^2 \log n\ \operatorname{loglog} n\},$$

where the first of the arguments to the max-operator corresponds to the processor complexity $M(n)$ of parallel matrix multiplication; at this time we have $M(n) = n^\omega$ with $\omega < 2.3755$ (Coppersmith and Winograd 1990). Note that we restrict each processor to a constant number of arithmetic operations; without this restriction $M^*(n)$ can be reduced by $\log n$ factors. It should also be noted that our solutions use matrix multiplication as a black-box. Therefore, the processor count and especially the constant in the big-O estimate is directly related to the particular matrix multiplication algorithm used.

The time measures of our algorithms are for sufficiently large fields of small positive characteristic p asymptotically slower by a factor of $\log_p n$. As said above, for fields of small cardinality we work in sufficiently large Galois extensions, the purpose being the guarantee of a positive success probability of our randomizations. This introduces a further slow-down in terms of base field arithmetic but does not increase the processor count. Finally, in case $1 < p < n$ an additional $\log n$ factor is introduced by the binary search for the rank. Therefore, we shall introduce the multiplier function

$$\gamma_{\mathsf{K}}(n,p) := \begin{cases} 1 \quad \text{for } p = 0 \text{ or } p \geq n, \\ \lceil (\log n)^2/(\log p) \rceil \\ \qquad \text{for } 1 < p < n \text{ and } \operatorname{card}(\mathsf{K}) \geq n, \\ \lceil (\log n)^3/(\log p\ \operatorname{log}(\operatorname{card}\mathsf{K})) \rceil \\ \qquad \text{for } \operatorname{card}(\mathsf{K}) < n, \end{cases}$$

which becomes universal for any field. By $\operatorname{card}(\mathsf{K})$ we shall denote the cardinality of the field K; note that in the last case, by carrying out the arithmetic in the Galois extension field of algebraic degree $d_{\mathsf{K}}(n) = O(\log_{\operatorname{card}\mathsf{K}} n)$ in parallel, we can improve the multiplier $\gamma_{\mathsf{K}}(n,p)$ to $\lceil \log_p n\ \log d_{\mathsf{K}}(n) \rceil$, but we then need to increase the number of processors by a factor of $d_{\mathsf{K}}(n)\ \operatorname{loglog} d_{\mathsf{K}}(n)$. In any case, the multiplier is no more than

$$\gamma_{\mathsf{K}}(n,p) = O((\log n)^3).$$

Our main result can now be stated as follows: a Las Vegas-randomized algorithm is constructed that in parallel time $O(\gamma_{\mathsf{K}}(n,p)\,(\log n)^2)$ using $M^*(n)$ many processors finds the rank of a matrix and the solution of a linear system, which in the singular case consists of both one specific solution vector and a basis for the right null space of the coefficient matrix. For non-singular systems over fields of characteristic $1 < p < n$ our parallel time is actually a factor of $\log n$ faster.

We shall not be very specific about the parallel model of computation for which these results hold, although a suitable parallel adaptation of the probabilistic algebraic random access machine defined formally in (Kaltofen 1988) yields the stated complexity measures. In (Kaltofen and Pan 1991) we used algebraic circuits, which for the rank problem need to be amended appropriately to account for the zero-tests (cf. von zur Gathen's (1986) arithmetic networks).

2. Recursive Parallel Triangulation

We shall presume a certain familiarity of the reader with our methods in (Kaltofen and Pan 1991) for the non-singular case over fields of characteristic $p = 0$ or $p > n$. We now discuss how one can salvage the case $2 \leq p \leq n$ in a processor efficient manner. The key and only place where our previous methods fail is in the Le Verrier/Csanky transition from

$$s_i := \mathrm{Trace}(T^i) = \lambda_1^i + \lambda_2^i + \cdots + \lambda_n^i, \quad 1 \leq i \leq n-1,$$

where $\lambda_1, \ldots, \lambda_n$ are all eigenvalues of a non-singular Toeplitz matrix T, to the coefficients of the characteristic polynomial of T,

$$\mathrm{Det}(\lambda I - T) =: \lambda^n - c_1\lambda^{n-1} - c_2\lambda^{n-2} - \cdots - c_{n-1}\lambda - c_n. \quad (1)$$

The Newton identities,

$$\begin{bmatrix} 1 & & & & 0 \\ s_1 & 2 & & & \\ s_2 & s_1 & \ddots & & \\ \vdots & & & & \\ s_{n-2} & \ldots & & n-1 & \\ s_{n-1} & s_{n-2} & \ldots & s_1 & n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{n-1} \\ s_n \end{bmatrix} \quad (2)$$

lead to a triangular non-singular system for the c_i only if in the field of entries one can divide by $n!$. For fields of small positive characteristic it is in general impossible to determine the fundamental symmetric functions from the sums of powers.

Our solution hinges on the fact that we can find other linear equations for the unknown c_i. Let us suppose for a moment that the minimum polynomial of T is equal to the characteristic polynomial (1). Then by Wiedemann's (1986) theory of coordinate recurrences we have for the column vectors $u, v \in \mathsf{K}^n$ and for the field elements $\bar{a}_i := u^{\mathrm{tr}} T^i v$ that for all $j \geq 0$

$$c_n\bar{a}_j + c_{n-1}\bar{a}_{j+1} + \cdots + c_1\bar{a}_{j+n-1} = \bar{a}_{j+n}. \quad (3)$$

If the entries in u and v are chosen randomly and uniformly from a sufficiently large subset of K, then with a high probability the first n equations in (3) form a non-singular system (Kaltofen and Pan 1991, remarks after Theorem 1).

For simplicity, we shall continue the argument for the case that the characteristic of K is $p = 2$. We remove all even numbered rows from (2), which have 0 on their diagonal position. We also reorder the columns such that the odd numbered columns precede the even numbered ones. Thus we obtain the first $\lceil n/2 \rceil$ rows of our new system,

$$[\, L_1 \mid L_2 \,] \begin{bmatrix} c' \\ c'' \end{bmatrix} = [\, s' \,], \quad \text{where}$$

$$L_1 := \begin{bmatrix} 1 & & & 0 \\ s_2 & 1 & & \\ s_4 & s_2 & 1 & \\ \vdots & & & \ddots \end{bmatrix}, \quad L_2 := \begin{bmatrix} 0 & & & 0 \\ s_1 & 0 & & \\ s_3 & s_1 & 0 & \\ \vdots & & & \ddots \end{bmatrix},$$

$$\text{and} \quad c' := \begin{bmatrix} c_1 \\ c_3 \\ c_5 \\ \vdots \end{bmatrix}, \quad c'' := \begin{bmatrix} c_2 \\ c_4 \\ c_6 \\ \vdots \end{bmatrix}, \quad s' := \begin{bmatrix} s_1 \\ s_3 \\ s_5 \\ \vdots \end{bmatrix}.$$

Finally, after rearranging the columns in the system (3) appropriately as

$$[\, \bar{A}_1 \mid \bar{A}_2 \,] \begin{bmatrix} c' \\ c'' \end{bmatrix} = \bar{b}, \quad (4)$$

where $\bar{A}_1 \in \mathsf{K}^{n \times \lceil n/2 \rceil}$, $\bar{A}_2 \in \mathsf{K}^{n \times \lfloor n/2 \rfloor}$, and $\bar{b} \in \mathsf{K}^n$, we compress the system (4) to $\lfloor n/2 \rfloor$ equations by multiplying both the left-hand side coefficient matrix and the right-hand side vector by a $\lfloor n/2 \rfloor \times n$ matrix C whose entries are chosen randomly from a sufficiently large subset of K. This randomization guarantees that with a high probability the system

$$\left[\begin{array}{c|c} L_1 & L_2 \\ \hline C\bar{A}_1 & C\bar{A}_2 \end{array}\right] \begin{bmatrix} c' \\ c'' \end{bmatrix} = \begin{bmatrix} s' \\ \hline C\bar{b} \end{bmatrix} \quad (5)$$

is non-singular. The heart of argument lies in the fact that in (4) there are $\lfloor n/2 \rfloor$ rows that are linearly independent of the rows in $[\, L_1 \mid L_2 \,]$, because (3) has by our assumption full rank. Thus there exists a specific matrix C that selects those rows, and hence a matrix $\mathcal{C}$ with indeterminate elements also has the property that the corresponding determinant is non-zero, now as polynomial in the unknown entries of $\mathcal{C}$. By a lemma of Zippel (1979)/Schwartz (1980) a matrix with random elements chosen uniformly from a sufficiently large subset of K also has with high probability that property.

We now can proceed recursively. First, we eliminate the block $C\bar{A}_1$ by pre-multiplying (5) with

$$\left[\begin{array}{c|c} I_{\lceil n/2 \rceil} & 0^{\lceil n/2 \rceil \times \lfloor n/2 \rfloor} \\ \hline -C\bar{A}_1 L_1^{-1} & I_{\lfloor n/2 \rfloor} \end{array}\right].$$

Since L_1 is Toeplitz and lower triangular, its inverse can be computed easily in $O((\log n)^2)$ time with $n^2 \times \log n \log\log n$ processors. This leads to a $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ non-singular system for c'' which we may solve by recursive application of the entire algorithm. Finally, we determine c' by back-substituting into the first row of blocks. The algorithm clearly has depth $O((\log n)^3)$, and it uses

$$P(n) \le M^*(n) + P(\lfloor n/2 \rfloor), \quad \text{i.e., } O(M^*(n))$$

many processors; constant rescaling reduces the number of processors to $M^*(n)$. For characteristic $p > 2$ the recursion is on blocks of size $\lfloor n/p \rfloor$, hence the general slow-down in time is by the factor $\gamma_{\mathsf{K}}(n,p)$.

It is possible perform the needed compression by C with fewer random elements. Suppose we first eliminate $\bar{A}_1$ in (4) using L_1^{-1}:

$$\left[\begin{array}{c|c} I_{\lceil n/2 \rceil} & 0^{\lceil n/2 \rceil \times n} \\ \hline -\bar{A}_1 L_1^{-1} & I_n \end{array}\right] \left[\begin{array}{c|c} L_1 & L_2 \\ \hline \bar{A}_1 & \bar{A}_2 \end{array}\right] = \left[\begin{array}{c|c} L_1 & L_2 \\ \hline 0^{n \times \lceil n/2 \rceil} & -\bar{A}_1 L_1^{-1} L_2 + \bar{A}_2 \end{array}\right]$$

Then by (Kaltofen and Saunders 1991, Theorem 2) it suffices to pre-multiply the $n \times \lfloor n/2 \rfloor$ matrix $-\bar{A}_1 \times L_1^{-1} L_2 + \bar{A}_2$ by a random $\lfloor n/2 \rfloor \times n$ unit upper triangular Toeplitz matrix C and thereby obtain with high probability a $\lfloor n/2 \rfloor \times \lfloor n/2 \rfloor$ compressed matrix of full rank. Applying these transformations to the right-hand side vector of our system also, we have thus established the need of only $O(n)$ many random field elements for that phase of the construction. Note that all matrices involved in the smaller system have a special structure. Therefore, our recursive triangulation approach may also be applicable to the processor-efficient solution of structured systems, such as Toeplitz and Sylvester systems.

At last, we shall consider the case where the minimum polynomial of T is a proper divisor of the characteristic polynomial, in which case the first n equations in (3) are linearly dependent. The needed sub-problem in (Kaltofen and Pan 1991), namely the determination of the coefficients $\widetilde{c}_i$ of the characteristic polynomial of the matrix $\widetilde{A}$ (ibid. p. 185, bottom of the second column), is actually the computation of $T^{-1}b$ for a known vector b, which there is solved in a processor-efficient way via the coefficients of (1). However, we may instead work with

$$\widetilde{T} := VTW \quad \text{and} \quad \widetilde{b} := Vb,$$

where

$$V := \begin{bmatrix} 1 & v_2 & v_3 & \dots & v_n \\ & 1 & v_2 & \dots & v_{n-1} \\ & & 1 & \ddots & \vdots \\ & 0 & & \ddots & v_2 \\ & & & & 1 \end{bmatrix} \tag{6}$$

and

$$W := \begin{bmatrix} w_1 & & & & \\ w_2 & w_1 & & 0 & \\ w_3 & w_2 & w_1 & & \\ \vdots & & \ddots & \ddots & \\ w_n & w_{n-1} & \dots & w_2 & w_1 \end{bmatrix}, \tag{7}$$

because then $T^{-1}b = W(\widetilde{T}^{-1}\widetilde{b})$ can still be determined efficiently. If we choose the entries v_i and w_j randomly from a sufficiently large subset of K, then with high probability the minimum polynomial of $\widetilde{T}$ agrees with its characteristic polynomial. Furthermore, we may also compute for $1 \le i < n$ all $\mathrm{Trace}(\widetilde{T}^i)$ in the same asymptotic complexity measures as those we have for the original s_i. The proofs for these two claims are, however, quite involved. The argument for minimum = characteristic polynomial can be deduced from the perturbation theory developed in Wiedemann (1986) and Kaltofen and Saunders (1991), while the argument for computing the traces of the powers of the perturbed matrix follows from the parallel algorithms by Pan (1990) associated with the theory of matrices of fixed displacement rank by Kailath et al. (1979). The matrix $\widetilde{T}$, being a product of three Toeplitz matrices, has displacement rank at most 7.

We first prove that with high probability the minimum polynomial of $\widetilde{T}$ is its characteristic polynomial.

Proposition 1. *Let $M \in \mathsf{K}^{n \times n}$ be a non-singular matrix and let the entries $v_2, \dots, v_n$ of V in (6) and $w_1, \dots, w_n$ of W in (7) be randomly and uniformly selected from a subset $S \subset \mathsf{K}$. Then the minimum polynomial of $\widetilde{M} := VMW$ is equal to the characteristic polynomial of $\widetilde{M}$ with probability no less than $1 - 4n^2/\mathrm{card}(S)$.*

Proof. We follow the proof of Wiedemann (1986, Lemma in §V) and show that with high probability the characteristic polynomial of $\widetilde{M}$ is square-free. Let $\mathcal{V}$ and $\mathcal{W}$ be Toeplitz matrices of the form V and W, respectively, but with their entries being indeterminates. We first prove the square-freeness of the characteristic polynomial of $\mathcal{M} := \mathcal{V}M\mathcal{W}$ for such generic multiplier matrices, and then consider with what probability randomly chosen values for these indeterminates will preserve this condition. In order to prove the square-freeness of the generic case, we need the additional condition that all $i \times i$ submatrix blocks in the right upper corner of $\mathcal{V}M$, which we denote by $(\mathcal{V}M)_{i\neg}$, are non-singular. This condition

follows from the proof of Theorem 2 in (Kaltofen and Saunders 1991):

Let $\mathcal{V}_{I,J}$ be the determinant of the submatrix of $\mathcal{V}$ formed by including only the rows contained in the set I and only the columns contained in the set J. It is shown there that for all $i = 1, \ldots, n$ the set of all polynomials

$$\{\mathcal{V}_{\{1,\ldots,i\},J} \mid \mathrm{card}(J) = i\}$$

is linearly independent over K. Furthermore, by the Cauchy-Binet formula,

$$(\mathcal{V}M)_{\{1,\ldots,i\},\{n-i+1,\ldots,n\}} = \sum_{\substack{J=\{j_1,\ldots,j_i\} \\ 1\le j_1<\cdots<j_i\le n}} \mathcal{V}_{\{1,\ldots,i\},J}\, M_{J,\{n-i+1,\ldots,n\}}. \quad (8)$$

Since M is non-singular, there exists a set of row indices J_0 such that $M_{J_0,\{n-i+1,\ldots,n\}} \neq 0$, hence the sum (8) of linearly independent polynomials also does not vanish.

The rest of the argument is by induction on n, as in Wiedemann's lemma. Consider the characteristic polynomial $\mathrm{Det}(\lambda I - \widetilde{\mathcal{M}})$ as a polynomial in λ with coefficients being polynomials in the variable entries $v_2, \ldots, w_n$ of $\mathcal{V}$ and $\mathcal{W}$. If we set $w_1 = 0$, this polynomial is equal to

$$\lambda\, \mathrm{Det}(\lambda I - (\mathcal{V}M)_{n-1\neg}\, \mathcal{W}_{\llcorner\, n-1}),$$

where $\mathcal{W}_{\llcorner\, n-1}$ is the $(n-1)\times(n-1)$ sub-matrix at the lower left of $\mathcal{W}$. Since $(\mathcal{V}M)_{n-1\neg}$ is non-singular, as was shown above, by induction hypothesis the characteristic polynomial of $(\mathcal{V}M)_{n-1\neg}\mathcal{W}_{\llcorner\, n-1}$ must be square-free; it is also not divisible by λ. Therefore, the characteristic polynomial of the genericly randomized matrix $\mathcal{V}M\mathcal{W}$ is square-free even for $w_1 = 0$, hence must be square-free for an indeterminate w_1.

It remains to estimate the probability of success. We appeal to the Zippel (1979)/Schwartz (1980) lemma and must therefore estimate the total degree of the discriminant in λ of $\mathrm{Det}(\lambda I - \widetilde{\mathcal{M}})$, which is a polynomial in the entries of $\mathcal{V}$ and $\mathcal{W}$. However, that total degree is certainly less than $4n^2$. ⊠

Wiedemann chooses a random diagonal matrix in place of our W of Lemma 1. That our choice of a Toeplitz matrix can be proven to work is somewhat fortunate, since otherwise we would not know how to compute the required $\mathrm{Trace}(\widetilde{T}^i)$ efficiently. The underlying reason is the fact that a Toeplitz matrix has fixed displacement rank, while for a random diagonal matrix the displacement rank is n. A similar problem has to be overcome in §3 in regard to Mulmuley's randomization. We now shall explain the theory of Toeplitz-like matrices in some more detail.

We consider $n \times n$ matrices; 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

$$\downarrow A := ZA, \quad \curvearrowleft A := AZ, \quad \text{and} \quad \curvearrowright A := AZ^{\mathrm{tr}}.$$

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

$$\phi_+(A) := A - \downarrow(\curvearrowright A) = A - ZA\,Z^{\mathrm{tr}}$$

and

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

the latter being the displacement rank of A. 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,

$$X - \downarrow(\curvearrowright X) = \sum_{j=1}^{\alpha} y_j z_j^{\mathrm{tr}} \quad (9)$$

has the unique solution

$$X = \sum_{j=1}^{\alpha} L[\![y_j]\!]\, U[\![z_j^{\mathrm{tr}}]\!], \quad (10)$$

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 is a sum of α products of lower and upper triangular Toeplitz matrices. We shall call

$$Y = \sum_{j=1}^{\alpha} y_j z_j^{\mathrm{tr}} = [\, y_1 \mid y_2 \mid \ldots \mid y_\alpha \,] \left[\begin{array}{c} z_1^{\mathrm{tr}} \\ \hline z_2^{\mathrm{tr}} \\ \hline \vdots \\ \hline z_\alpha^{\mathrm{tr}} \end{array}\right] \quad (11)$$

the *generators* of the resulting $n \times n$ matrix Y, which is in the above case the displaced matrix $\phi_+(X)$; the representation (10) we shall call the *Gohberg-Semencul representation* for X. That representation requires only the storage of $O(\alpha n)$ field elements. Clearly, one may derive a generator (11) for Y by choosing the vectors y_j to be the 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.

We wish to argue that matrices of fixed displacement rank are closed under matrix multiplication. To show this, we introduce the alternate displacement operator (see Pan 1990)

$$\phi^+(A) := {}^{\uparrow}A - \downarrow A = AZ - ZA$$

and

$$\alpha^+(A) := \operatorname{rank} \phi^+(A).$$

We then have $\alpha_+(A) \le \alpha^+(A) + 1$, in fact we can find the generators for $\phi_+(A)$ from the generators for $\phi^+(A)$, since

$$\phi_+(A) = \phi^+(A)Z^{\mathrm{tr}} + (Ae_1)e_1^{\mathrm{tr}}, \tag{12}$$

where $e_1 := [1 \quad 0 \quad \dots \quad 0]^{\mathrm{tr}}$. Note that the vectors z_j^{tr} in (11) for $\phi^+(A)$ are shifted to the right by post-multiplication by Z^{tr}. Quite similarly we can easily obtain the inequality

$$\alpha^+(AB) \le \alpha^+(A) + \alpha^+(B),$$

namely from

$$\phi^+(AB) = A\phi^+(B) + \phi^+(A)B.$$

The vectors y_j generating $\phi^+(B)$ are pre-multiplied by A, and the vectors z_j are post-multiplied by B. Note that this approach allows us to efficiently compute the Gohberg-Semencul representation (10) for $X = \widetilde{T} = VTW$. Since $\alpha^+(V) = \alpha^+(T) = \alpha^+(W) = 2$, we we will compute $6 = ((2+2)+2)$ generators (11) for $\phi^+(\widetilde{T})$ and thus 7 for $\phi_+(\widetilde{T})$.

Finally, we have the following theorem at our disposal:

Proposition 2 (Pan 1990, Proposition 3.1)**.** *Let $\widetilde{T}$ be given in Gohberg-Semencul representation with a fixed displacement rank, i.e., with a fixed number of LU-products. Then the traces*

$$s_i = \operatorname{Trace}(\widetilde{T}^i) \quad \text{for all } i = 2, \dots, n$$

can be computed in parallel in $O((\log n)^2)$ time using $n^2 \log n \operatorname{loglog} n$ processors.

There is a very elegant proof for this important fact, due to Dario Bini, which we shall briefly sketch. Pan's (1990) approach computes a Gohberg-Semencul representation for

$$X_i := I + \lambda\widetilde{T} + \cdots + \lambda^{2^i-1}\widetilde{T}^{2^i-1} \equiv (I - \lambda\widetilde{T})^{-1} \pmod{\lambda^{2^i}} \tag{13}$$

(cf. §3 in Kaltofen and Pan 1991). This is done by "lifting" the representation of X_i to the squared modulus using Newton iteration, namely

$$X_{i+1} = X_i(2I - (I - \lambda\widetilde{T})X_i).$$

Bini's method is based on the identity

$$\begin{aligned}\phi^+(A^{-1}) &= A^{-1}Z - ZA^{-1} \\ &= -A^{-1}(AZ - ZA)A^{-1} \\ &= -A^{-1}\phi^+(A)A^{-1}.\end{aligned}$$

Then by (13) we have the congruence

$$\phi^+(X_{i+1}) \equiv -X_{i+1}\,\phi^+(I - \lambda\widetilde{T})\,X_{i+1} \pmod{\lambda^{2^{i+1}}}.$$

Thus, we obtain generators (11) for $Y = \phi^+(X_{i+1})$ from the generators (11) of $Y = \phi^+(I - \lambda\widetilde{T})$ by pre-multiplying the generators y_j by $X_{i+1} = X_i(2I - (I - \lambda\widetilde{T})X_i)$ and post-multiplying the generators z_j by the same matrix, all modulo $\lambda^{2^{i+1}}$. However, since we have the Gohberg-Semencul representation for X_i, this process can be carried out efficiently by using fast polynomial multiplication for the needed Toeplitz matrix times vector products. Finally, the Gohberg-Semencul representation (10) for X_{i+1} is deduced from the thus determined generators (11) of $Y = \phi^+(X_{i+1})$ by the conversion mechanism (12) to $\phi_+(X_{i+1})$ and the functional equivalence (10). In our situation there will be nine LU-products under the sum (seven for $\lambda\widetilde{T}$, one for I, and one produced by the conversion to ϕ_+).

Note that this approach does not require any generalization of the Gohberg-Semencul formula for $(I-\lambda\widetilde{T})^{-1}$, and is based solely on the solution (10) of the functional equation (9). Furthermore, the product formula for $\alpha^+(AB)$ is valid over any field, unlike Chun's et al. (1987, Lemma 3) formula for $\alpha_+(AB)$, where a division by 2 is performed.

We conclude this section by stating the main theorem.

Theorem 1. *Given a non-singular matrix $A \in \mathsf{K}^{n\times n}$ and a vector $b \in \mathsf{K}^n$, where K is a field of characteristic $1 < p < n$, a vector $x \in \mathsf{K}^n$ with $Ax = b$ can be computed (on a randomized algebraic PRAM) in*

$$O(\gamma_{\mathsf{K}}(n,p)\,\log n) \text{ time with } M^*(n) \text{ processors.}$$

In the case where $\operatorname{card}(\mathsf{K}) \ge n$, the algorithm uniformly chooses $O(n)$ many random elements from a subset $S \subset \mathsf{K}$ and with probability no more than $O(n^2/\operatorname{card}(S))$ reports "failure." In the case where $\operatorname{card}(\mathsf{K}) < n$, the algorithm uniformly chooses $O(n\,(\log n)/(\log \operatorname{card}\mathsf{K}))$ many field elements and with probability no more than $1/n^{O(1)}$ reports "failure."

3. Parallel Rank Computations

The results in Borodin et al. (1982) imply that a processor-efficient randomized parallel algorithm for the solution of singular linear systems can be reduced probabilistically to the problem of inverting a non-singular matrix and to the problem of determining the rank of a matrix that has the added property that all its leading principal sub-matrices of dimension no larger than its rank are non-singular. Their method picks random non-singular matrices $V, W \in \mathsf{K}^{n\times n}$ — by (Kaltofen and Saunders 1991, Theorem 2) these matrices may even be of unit upper and lower triangular Toeplitz form — and computes for the singular coefficient matrix $A \in \mathsf{K}^{n\times n}$ the product matrix $\widetilde{A} := VAW$, which can be shown to have with a certain probability non-singular $i \times i$ leading principal sub-matrices $\widetilde{A}_{\ulcorner i}$, where $i \le r$ and $r := \operatorname{rank} A$. Then

$$\widetilde{A}\,E = \left[\begin{array}{c|c} \widetilde{A}_{\ulcorner r} & 0^{r\times(n-r)} \\ \hline A' & 0^{(n-r)\times(n-r)} \end{array}\right]$$

for

$$\widetilde{A} =: \left[\begin{array}{c|c} \widetilde{A}_{\ulcorner r} & B \\ \hline A' & A'' \end{array}\right] \text{ and } E := \left[\begin{array}{c|c} I_r & -\widetilde{A}^{-1}_{\ulcorner r}B \\ \hline 0^{(n-r)\times r} & I_{n-r} \end{array}\right].$$

Hence the right null space of A is spanned by the columns of

$$WE\left[\begin{array}{c} 0^{r\times(n-r)} \\ \hline I_{n-r} \end{array}\right],$$

while for a vector b such that $Ax = b$ is solvable one gets

$$AWE\left[\begin{array}{c} \widetilde{A}^{-1}_{\ulcorner r}(Vb)_{\ulcorner r\urcorner} \\ \hline 0^{n-r} \end{array}\right] = b,$$

where $(Vb)_{\ulcorner r\urcorner}$ is the vector formed by the first r enties of Vb. We mention these standard linear algebra formulas since with them the probabilistically determined rank and the solvability of $Ax = b$ can be certified. Therefore the randomized method is of the Las Vegas kind.

Using a construction by Baur and Strassen (1983) (see also Linnainmaa 1976), it is shown in (Kaltofen and Pan 1991) that the determinant and thus by (Kaltofen and Singer 1991) also the inverse of a non-singular matrix can be computed in a randomized, parallel, and processor-efficient fashion. The results of §2 extend to the determinant and inverse computation. While for non-singular matrices the recursive triangulation algorithm is Las Vegas, the method cannot produce a proof that a matrix is singular, i.e., that its determinant is equal to zero. The reason for this is that for a singular matrix, the additional linear equations (4) may never be sufficient to yield a compressed non-singular system for the coefficients of (1). An example for this is $T = 0$. Therefore, we obtain a proof for the singularity of a matrix only after having probabilistically computed its rank.

It remains to show how to determine the rank of $\widetilde{A}$. At the cost of slowing the parallel time asymptotically by a factor of $\log n$, one may perform a (Monte Carlo) binary search for the largest non-singular leading principal sub-matrix. We now give a method that avoids this slow-down without utilizing asymptotically more processors, provided the characteristic of the field K is 0 or $>n$. In (Kaltofen and Saunders 1991, Lemma 2) the problem of determining the rank of a singular $\widetilde{A}$ is probabilistically reduced to the problem of determining the rank of a Toeplitz matrix. In particular, it is shown there that if the entries in the diagonal matrix D are selected randomly from a sufficiently large set, then with a high probability the degree of the minimum polynomial $f^{\widetilde{A}D}(\lambda)$ of $\widetilde{A}D$ is equal to $\operatorname{rank}(\widetilde{A}) + 1$. By the Wiedemann (1986) coordinate recurrence technique, for random column vectors u and v the Toeplitz matrix

$$T := \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ a_n & a_{n-1} & \dots & a_2 & a_1 \\ \vdots & a_n & \ddots & \vdots & a_2 \\ & \vdots & & & \vdots \\ a_{2n-3} & & & a_{n-1} & \\ a_{2n-2} & a_{2n-3} & \dots & a_n & a_{n-1} \end{bmatrix} \tag{14}$$

with the entries $a_i = u^{\mathrm{tr}}\widetilde{A}^i v$ has with a high probability the rank equal to $\deg(f^{\widetilde{A}D})$. We probabilistically find the rank of T by appealing to an idea of Mulmuley (1987). Consider the n-dimensional anti-diagonal unit matrix

$$J := \begin{bmatrix} 0 & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & 0 \end{bmatrix};$$

the matrix $H := TJ$ is Hankel, i.e., on each anti-diagonal there appears a single element. In particular, a Hankel matrix is symmetric. Furthermore, for the diagonal matrix

$$X := \begin{bmatrix} 1 & & & & 0 \\ & x & & & \\ & & x^2 & & \\ & & & \ddots & \\ 0 & & & & x^{n-1} \end{bmatrix},$$

the matrix $\widetilde{H} = XHX$ is also Hankel. Note that XTX is not Toeplitz, which is the reason for switching to the Hankel form. It follows by Mulmuley's (1987) arguments that, with the possible exception of at most n^2

field elements x, the multiplicity of λ in the characteristic polynomial $\mathrm{Det}(\lambda I - \widetilde{H})$ is equal to $n - \mathrm{rank}(H)$. The key property in his proof (loc. cit., Proof of Lemma 1) is the condition that for an indeterminate x

$$\mathrm{rank}(\widetilde{H}\,\widetilde{H}) = \mathrm{rank}(\widetilde{H}).$$

It is shown there that $\mathrm{rank}(HXH) = \mathrm{rank}\,H$, hence also $\mathrm{rank}(HX^2H) = \mathrm{rank}(H)$, which by pre- and post-multiplying with the non-singular matrix X yields the needed condition. Note that the trailing coefficient of the characteristic polynomial of $\widetilde{H}$ has, as a polynomial in x, degree no more than n^2.

Therefore, we shall compute the characteristic polynomial of the Hankel matrix $\widetilde{H}$. Kailath et al. (1979, Footnote on p. 405) give a displacement operator that for fields of characteristic 0 or $>n$ leads to a $O((\log n)^2)$-time and $n^2(\mathrm{loglog}\,n)/(\log n)$-processors parallel algorithm (see also Bini 1983, Bini and Pan 1992). We shall present a direct method, which is also based on parallel algorithms for matrices with fixed displacement rank (see §2). Making some simplifying assumptions first, we shall sketch the entire method now. The key algorithm computes for a Toeplitz matrix $\widetilde{T}$ two lower triangular Toeplitz matrices $L_1(\lambda), L_2(\lambda)$ over $\mathsf{K}[\lambda]$ and two upper triangular Toeplitz matrices $U_1(\lambda), U_2(\lambda)$ such that

$$\begin{aligned} I + \lambda\widetilde{T} + \lambda^2\widetilde{T}^2 + \cdots + \lambda^{n-1}\widetilde{T}^{n-1} \\ \equiv L_1(\lambda)U_1(\lambda) + L_2(\lambda)U_2(\lambda) \pmod{\lambda^n}. \quad (15) \end{aligned}$$

This is the Gohberg-Semencul representation (10) of $(I - \lambda\widetilde{T})^{-1} \bmod \lambda^n$, as was explained following Proposition 2 in §2; just two LU-products are obtained by using the original Gohberg-Semencul formula (see Pan 1990 or Kaltofen and Pan (1991), §3), but this is not critical here. The trace of the right-hand-side matrix expression in (15) can now be computed in parallel with no more than $n^2\,\mathrm{loglog}(n)/\log(n)$ processors; for instance, we have

$$\begin{aligned} \mathrm{Trace}\Bigg(& \begin{bmatrix} w_1 & & & & \\ w_2 & w_1 & & 0 & \\ w_3 & w_2 & w_1 & & \\ \vdots & & \ddots & \ddots & \\ w_n & w_{n-1} & \dots & w_2 & w_1 \end{bmatrix} \\ & \times \begin{bmatrix} v_1 & v_2 & v_3 & \dots & v_n \\ & v_1 & v_2 & & v_{n-1} \\ & & \ddots & \ddots & \vdots \\ & 0 & & v_1 & v_2 \\ & & & & v_1 \end{bmatrix} \Bigg) \\ & = nw_1v_1 + (n-1)w_2v_2 + \cdots + w_nv_n. \quad (16) \end{aligned}$$

The coefficient of λ^i in the trace of (15) is then $s_i = \mathrm{Trace}(\widetilde{T}^i)$. Suppose for a moment that $\widetilde{T} = \widetilde{H}J$ would be also symmetric. Then $J\widetilde{T}J = \widetilde{T}$, hence for $i \geq 1$,

$$\widetilde{H}^{2i} = (\widetilde{T}J)^{2i} = \widetilde{T}^{2i}, \quad \widetilde{H}^{2i+1} = \widetilde{T}^{2i+1}J.$$

In this special case we therefore have

$$\mathrm{Trace}(\widetilde{H}^{2i}) = s_{2i}$$

and

$$\mathrm{Trace}(\widetilde{H}^{2i+1}) = \text{anti-Trace}(\widetilde{T}^{2i+1}),$$

with the anti-trace of a square matrix being the sum of all elements on the main anti-diagonal. The needed anti-traces can be computed from (15) like the traces; e.g., the anti-trace of the product-matrix in (16) is for odd n

$$\begin{aligned} \sum_{i=1}^{\lceil n/2 \rceil} w_{2i-1}(v_1 + v_3 + \cdots + v_{n+2-2i}) \\ + \sum_{i=1}^{\lfloor n/2 \rfloor} w_{2i}(v_2 + v_4 + \cdots + v_{n+1-2i}), \end{aligned}$$

where all the partial sums of the v_j's can be found by parallel prefix (Ladner and Fischer 1980). Thus we can compute $\mathrm{Trace}(\widetilde{H}^i)$ for all $1 \leq i < n$, from which for a field K of characteristic 0 or $>n$ we may determine the coefficients of the characteristic polynomial $\mathrm{Det}(\lambda I - \widetilde{H})$ by solving the Newton identities (2).

However, $\widetilde{H}$ is most likely not anti-symmetric. Therefore, we consider the "anti-symmetrized" block matrix

$$\underbrace{\left[\begin{array}{c|c} \widetilde{H} & 0^{n\times n} \\ \hline 0^{n\times n} & J\widetilde{H}J \end{array}\right]}_{\widetilde{H}_{\boxplus}} = \underbrace{\left[\begin{array}{c|c} 0^{n\times n} & \widetilde{H}J \\ \hline J\widetilde{H} & 0^{n\times n} \end{array}\right]}_{\widetilde{T}_{\boxplus}} \times \underbrace{\left[\begin{array}{c|c} 0^{n\times n} & J \\ \hline J & 0^{n\times n} \end{array}\right]}_{J_{2n}}$$

in place of $\widetilde{H}$ and $\widetilde{T}$. Then

$$\mathrm{Det}(\lambda I_{2n} - \widetilde{H}_{\boxplus}) = (\mathrm{Det}(\lambda I_n - \widetilde{H}))^2$$

and all $\mathrm{Trace}(\widetilde{H}^i_{\boxplus})$ can be determined from the traces and anti-traces of $\widetilde{T}_{\boxplus}$. Propostion 2 now allows the same approach for that block matrix as was discussed above. In fact $\alpha^+(\widetilde{T}_{\boxplus}) \leq 4$, thus one can compute five lower triangular Toeplitz matrices $L_j(\lambda)$ and five upper triangular Toeplitz matrices $U_j(\lambda)$ over $\mathsf{K}[\lambda]$ such that

$$\begin{aligned} I + \lambda\widetilde{T}_{\boxplus} + \lambda^2\widetilde{T}^2_{\boxplus} + \cdots + \lambda^{n-1}\widetilde{T}^{n-1}_{\boxplus} \\ \equiv \sum_{j=1}^{5} L_j(\lambda)U_j(\lambda) \pmod{\lambda^n}. \end{aligned}$$

In summary, we have the following theorem.

Theorem 2. *Given $A \in \mathsf{K}^{n\times n}$ and $b \in \mathsf{K}^n$, where K is a field of characteristic $p = 0$ or $p \geq n$, vectors $x_0, \ldots, x_{n-r} \in \mathsf{K}^n$ that determine the solution manifold*

$$A\,(x_0 + \mathsf{K}x_1 + \cdots + \mathsf{K}x_{n-r}) = \{b\},$$

can be computed (on a randomized algebraic PRAM) in

$$O((\log n)^2) \text{ time with } M^*(n) \text{ processors.}$$

In case the system $Ax = b$ is unsolvable the algorithm returns $\varnothing$. The algorithm uniformly chooses $O(n)$ random elements in $S \subset \mathsf{K}$, and with probability no more than $O(n^2/\mathrm{card}(S))$ reports "failure."

Finally, we comment on the case where the characteristic p of K is $1 < p < n$. For any Hankel matrix H, with

$$n^3 \log n \text{ loglog } n \text{ many processors}$$

we even know how to compute deterministically over any field K and in

$$O((\log n)^2) \text{ parallel time}$$

all characteristic polynomials

$$\mathrm{Det}(\lambda I_i - H_{\ulcorner i}), \quad i = 1, \ldots, n.$$

Hence, if we give ourselves slightly more processors, we can show how to determine the rank of matrices over fields of small positive characteristic in parallel time $O(\gamma_{\mathsf{K}}(n,p)/(\log_p n)\ \log n)$, which surpasses, e.g., our current processor-efficient solution for a Galois field with 2 elements by a factor of $(\log n)^2$. We make us of an identity by Chistov (1985), namely that for any $n \times n$ matrix M

$$\frac{1}{\mathrm{Det}(I_i - \lambda M_{\ulcorner i})} \equiv \prod_{j=1}^{i} \sum_{k=0}^{n} (M^k_{\ulcorner j})[j,j]\ \lambda^k \pmod{\lambda^{n+1}}.$$

Thus, if we can find for $M = H$ the entries in row and column j of the k-th power of $H_{\ulcorner j}$, the elements $(H^k_{\ulcorner j})[j,j]$ in the above formula, we can complete the computation by parallel prefix polynomial multiplication and power series inversion. But, by the above described algorithms, we can find for any Hankle matrix $\widetilde{H}$ the Gohberg-Semencul representation for

$$I + \lambda \widetilde{T}_{\boxplus} + \cdots + \lambda^n \widetilde{T}^n_{\boxplus},$$

that in $O((\log n)^2)$ parallel time and with $n^2 \log n \times$ loglog n many processors. By the symmetry of $\widetilde{T}_{\boxplus}$ we have

$$\widetilde{H}^{2k}_{\boxplus} = \widetilde{T}^{2k}_{\boxplus}, \quad \widetilde{H}^{2k+1}_{\boxplus} = \widetilde{T}^{2k+1}_{\boxplus} J_{2n},$$

hence we can find from the Gohberg-Semencul representation, similarly to the trace, all $(\widetilde{H}^k_{\boxplus})[n,n] = (\widetilde{H}^k)[n,n]$. The stated processor count now follows by using that approach for each of the Hankel matrices $\widetilde{H} = H_{\ulcorner i}$.

Literature Cited

Baur, W. and Strassen, V., "The complexity of partial derivatives," *Theoretical Comp. Sci.* **22**, pp. 317–330 (1983).

Berkowitz, S. J., "On computing the determinant in small parallel time using a small number of processors," *Inform. Process. Letters* **18**, pp. 147–150 (1984).

Bini, D., "On a class of matrices related to Toeplitz matrices," *Tech. Report.* **83-5**, Comput. Sci. Dept., State Univ. New York, Albany, N. Y., 1983.

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., *Numerical and Algebraic Computations with Matrices and Polynomials*; Lecture Notes in Theor. Comput. Sci., edited by R. V. Book; Birkhäuser Boston, Inc., 1992. 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).

Chistov, A. L., "Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic," *Proc. FCT '85, Springer Lec. Notes Comp. Sci.* **199**, pp. 63–69 (1985).

Chun, J., Kailath, T., and Lev-Ari, H., "Fast parallel algorithms for QR and triangular factorizations," *SIAM J. Sci. Stat. Comput.* **8**/6, pp. 899–913 (1987).

Coppersmith, D. and Winograd, S., "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.* **9**/3, pp. 251–280 (1990).

Csanky, L., "Fast parallel matrix inversion algorithms," *SIAM J. Comput.* **5**/4, pp. 618–623 (1976).

Galil, Z. and Pan, V., "Parallel evaluation of the determinant and of the inverse of a matrix," *Inform. Process. Letters* **30**, pp. 41–45 (1989).

von zur Gathen, J., "Parallel arithmetic computation: A survey," *Proc. MFCS '86, Springer Lec. Notes Comp. Sci.* **233**, pp. 93–112 (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.

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., "Greatest common divisors of polynomials given by straight-line programs," *J. ACM* **35**/1, pp. 231–264 (1988).

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 Saunders, B. D., "On Wiedemann's method of solving sparse linear systems," in *Proc. AAECC-5*, Springer Lect. Notes Comput. Sci. **536**; pp. 29–38, 1991.

Kaltofen, E. and Singer, M. F., "Size efficient parallel algebraic circuits for partial derivatives," in *IV International Conference on Computer Algebra in Physical Research*, edited by D. V. Shirkov, V. A. Rostovtsev, and V. P. Gerdt; World Scientific Publ., Singapore, pp. 133–145, 1991.

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.

Ladner, R. E. and Fischer, M. J., "Parallel prefix computation," *J. ACM* **27**/4, pp. 831–838 (1980).

Le Verrier, U.-J.-J., "Sur les variations séculaires des éléments elliptiques des sept planèts principales: Mercure, Vénus, la Terré, Mars, Jupiter, Saturne et Uranus," *J. Math. Pures Appl.* **5**, pp. 220–254 (1840). In French.

Linnainmaa, S., "Taylor expansion of the accumulated rounding error," *BIT* **16**, pp. 146–160 (1976).

Mulmuley, K., "A fast parallel algorithm to compute the rank of a matrix over an arbitrary field," *Combinatorica* **7**, pp. 101–104 (1987).

Pan, V., "Parameterization of Newton's iteration for computations with structured matrices and applications," *Tech. Report* **CUCS-032-90**, Comput. Sci. Dept., Columbia University, New York, N. Y., 1990. *Comput. and Math. (with Applic.)*, to appear.

Preparata, F. P. and Sarwate, D. V., "An improved parallel processor bound in fast matrix inversion," *Inform. Process. Letters* **7**/3, pp. 148–150 (1978).

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

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

Wiedemann, D., "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory* **IT-32**, pp. 54–62 (1986).

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