Processor-Efficient Parallel Computation of Polynomial Greatest Common Divisors*

Erich Kaltofen

Department of Computer Science, Rensselaer Polytechnic Institute,
Troy, NY 12180-3590; kaltofen@cs.rpi.edu

Preliminary Report (July 1, 1989)

1. Introduction

We present a parallel algebraic PRAM algorithm that can compute the greatest common divisor of two polynomials of maximal degree n with coefficients from a field simultaneously in $O(\log(n)^2)$ time using $n^{\omega+1}$ processors, where ω denotes the matrix multiplication exponent (currently the best being 2.37 (Coppersmith and Winograd 1987)). Our algorithm actually can find all subresultants of the input polynomials and the corresponding multiplier polynomials in the extended Euclidean scheme on an algebraic circuit of

$$\text{size} = O(n^{\omega+1}\log(n)) \;\text{ and depth} = O(\log(n)^2)$$

This more general result also leads to processor-efficient poly-log time parallel solutions to the Berlekamp-Massey problem (Massey 1969) of finding a linear generating recurrence for a sequence of field elements, or for the problem of counting the number of real zeros of a rational polynomial in a given interval. The processor/size measure that we achieve compares favorable to the best known sequential solutions if one restricts to algebraic computations without divisions.

The first poly-log solution to the polynomial greatest common divisor problem was described by Borodin, von zur Gathen, and Hopcroft (1982). In terms of network size or processor count, depending on the model of parallel algebraic computation, their algorithm computes $O(n)$ determinants of dimension $O(n)$. Using the most processor efficient poly-log determinant algorithms known to-date (Chistov 1985) and (Galil and Pan 1989), one obtains as the size of their network

$$O(n^{\omega+2}\log(n)) \;\text{ or }\; O(n^{\omega+1.5-\epsilon})$$

for arbitrary or zero characteristic, respectively. Our solution essentially reduces this size to the size of a single arbitrary characteristic n-dimensional determinant computation. We remark that Pan and Reif (1987) have investigated this problem for polynomials with rational coefficients and for iterative algorithms, i.e., for algorithms that compute high precision floating point approximations to the GCD before converting the result to the exact answer. Even though they obtain a better processor count, it must be realized that each processor has to carry out high precision floating point arithmetic, where the number of bits in the

*This material is based on work supported in part by the National Science Foundation under Grant No. CCR-87-05363 and under Grant No. CDA-88-05910.

precision depends on the input degree. Such an approach excludes a modular GCD algorithm (Brown 1971), which is an excellent means to parallelize the arithmetic itself and to control intermediate coefficient length growth, and it does not apply to finite coefficient fields, which is, e.g., the situation if one wants a poly-log parallel solution for the Berlekamp-Massey decoding problem.

We accomplish our processor/size improvement by making use of a drawback in the best-known parallel algorithms for computing matrix determinants, e.g., Chistov's (1985) algorithm. That algorithm can be made to actually produce, at a processor cost of $n^{\omega+1}$, not only the characteristic polynomial of the input matrix, but the characteristic polynomials of all principal submatrices. From a sequential algebraic complexity point of view the cost for this augmented problem might actually be optimal. We can construct a matrix, similar to the Sylvester matrix of the two polynomials, that has the matrices whose determinants are the leading coefficients of all subresultants in principal submatrix position. Using the construction of the subresultant as a polynomial matrix we observe that the coefficients of the polynomial multipliers in the Euclidean scheme that corresponds to a subresultant are actual entries in the first row of the adjoints of all these principal submatrices. These first rows we can obtain from the corresponding characteristic polynomials by the Cayley-Hamilton theorem and a processor efficient matrix power times vector prefix sum algorithm. Of course, the fundamental theorem of subresultants plays a crucial rolé for deciding the degrees of the remainders in the Euclidean sequence.

2. Subresultants

We now describe the theory of subresultants (Collins 1967), (Brown and Traub 1971) that underlies all known poly-log algorithms for computing polynomial greatest common divisors (Borodin et al. 1982). Consider a polynomial pseudo-remainder sequence for polynomials over an integral domain D:

$$\beta_i f_i(x) = \alpha_i f_{i-2}(x) - q_i(x) f_{i-1}(x), \quad 2 \le i \le k+1,$$

where

$$f_i, q_i \in \mathsf{D}[x], \quad d_i := \deg(f_i) < d_{i-1}, \quad \alpha_i, \beta_i \in \mathsf{D} \setminus \{0\}, \quad f_{k+1} = 0.$$

Then $f_k(x)$ is a scalar multiple of the GCD of $f(x) := f_0(x)$ and $g(x) := f_1(x)$ computed over the field of quotients $\mathrm{QF}(\mathsf{D})$ of D. The *Fundamental Theorem of Subresultants* relates the coefficients of $f_i(x)$ to minors in the Sylvester matrix of $f_0(x)$ and $f_1(x)$: Let

$$f(x) =: a_n x^n + \cdots + a_0, \quad f_1(x) := b_m x^n + \cdots + b_0, \quad n := d_0 \ge m := d_1.$$

Define as the j-th subresultant $S_j(f, g)$, $0 \le j \le m$, the determinant of the $(n + m - 2j) \times$

$(n+m-2j)$ matrix with entries in $\mathsf{D}[x]$,

$$\left(\begin{array}{cccccccccc}
a_n & a_{n-1} & a_{n-2} & \cdots & a_{n-(m-j-1)} & \cdots & a_0 & & & x^{m-j-1}f(x) \\
 & a_n & a_{n-1} & \cdots & a_{n-(m-j-2)} & \cdots & a_1 & a_0 & & x^{m-j-2}f(x) \\
 & \ddots & & & & & & \ddots & & \vdots \\
 & & & & & & & & a_0 & x^{j+1}f(x) \\
 & & & \ddots & & & & & \vdots & \vdots \\
 & & & & a_n & \cdots & \cdots & \cdots & a_{j+1} & f(x) \\
b_m & b_{m-1} & b_{m-2} & \cdots & b_{j+1} & \cdots & b_0 & & & x^{n-j-1}g(x) \\
 & b_m & b_{m-1} & \cdots & \cdots & \cdots & \cdots & b_0 & & x^{n-j-2}g(x) \\
 & \ddots & & & & & & \ddots & & \vdots \\
 & & & & & & & & b_0 & x^{j+1}g(x) \\
 & & & & \ddots & & & & \vdots & \vdots \\
 & & & & & b_m & \cdots & \cdots & b_{j+1} & g(x)
\end{array}\right). \tag{1}$$

Performing minor expansion along the last row of (1) produces polynomials

$$u_j f + v_j g = S_j(f,g), \quad u_j(x), v_j(x) \in \mathsf{D}[x], \quad \deg(u_j) < m-j, \deg(v_j) < n-j.$$

In fact, the coefficients of u_j and v_j are the $(n+m-2j-1)\times(n+m-2j-1)$ minors along the last column of (1). The fundamental theorem states (Brown and Traub 1971):
Let $\delta_i := d_i - d_{i-1}$ and c_i be the leading coefficient of f_i. Then for all $3 \le i \le k$,

$$\left.\begin{array}{l}
S_{d_{i-1}-1} = \gamma_i f_i \text{ with} \\
\qquad \gamma_i = (-1)^{\mu_i} c_{i-1}^{1-\delta_{i-1}} \prod_{l=3}^{i} \left(\dfrac{\beta_l}{\alpha_l}\right)^{d_{l-1}-d_{i-1}+1} c_{l-1}^{\delta_{l-2}+\delta_{l-1}} \\
\qquad \text{and } \mu_i = \sum_{l=3}^{i} (d_{l-2}-d_{i-1}+1)(d_{l-1}-d_{i-1}+1), \\
S_{d_i} = \vartheta_i f_i \text{ with} \\
\qquad \vartheta_i = (-1)^{\nu_i} c_i^{\delta_{i-1}-1} \prod_{l=3}^{i} \left(\dfrac{\beta_l}{\alpha_l}\right)^{d_{l-1}-d_i} c_{l-1}^{\delta_{l-2}+\delta_{l-1}} \\
\qquad \text{and } \nu_i = \sum_{l=3}^{i} (d_{l-2}-d_i)(d_{l-1}-d_i), \\
S_j(f,g) = 0 \text{ for all } j \text{ with } d_{i-1}-1 > j > d_i \text{ or } d_k > j \ge 0.
\end{array}\right\} \tag{2}$$

Our parallel algorithm will not work in the case that $m < n$. However, it is easy to modify the theory such that we always have $m = n$ by padding the polynomial g with zero monomials of degree larger then m. Considering the formal definition of the subresultant (1) not assuming that $b_m \neq 0$, we have

$$S_j(f, 0\cdot x^n + \cdots + 0\cdot x^{m+1} + g(x)) = a_n^{n-m} S_j(f,g). \tag{3}$$

3. Parallel Characteristic Polynomial Computation

In this section we inspect Chistov's (1985) method of solving linear systems in parallel. We have the following parallel construction, which we formulate in the model of uniform algebraic circuits (von zur Gathen 1986).

Theorem 1. *There exists a uniform family of algebraic division-free circuits over an arbitrary integral domain* D *that computes, for an* $n \times n$ *matrix* $A := (a_{i,j})_{1 \le i,j \le n}$ *over* D*, the coefficients of the characteristic polynomials of all principal submatrices* $A_k := (a_{i,j})_{1 \le i,j \le k}$*,* $1 \le k \le n$*, and whose circuit size is of order* $O(n^{\omega+1} \log(n))$ *and whose circuit depth is of order* $O(\log(n)^2)$*.*

The improvement in size, which was pointed out to me by von zur Gathen, hinges on the following lemma (Keller-Gehrig 1985).

Lemma 1. *There exists a uniform family of algebraic circuits over an arbitrary integral domain* D *that computes, for an* $n \times n$ *matrix* $A := (a_{i,j})_{1 \le i,j \le n}$ *over* D *and an* n*-dimensional column vector* b*, all vectors*

$$Ab, A^2 b, \ldots A^n b,$$

and whose circuit size is of order $O(n^{\omega} \log(n))$ *and whose circuit depth is of order* $O(\log(n)^2)$*.*
⊠

It is important to notice that in terms of sequential circuit complexity the construction is optimal, i.e., we do not know any circuit—even with divisions—that can compute the characteristic polynomials of all A_k and whose size is $o(n^{\omega+1})$. In terms of division-free sequential computation the circuits are optimal for positive characteristic even if one only requires the computation of the characteristic polynomial of A, or for that matter its constant coefficient, namely the $\mathrm{Det}(A)$. For fields of characteristic zero the algorithm by Galil and Pan (1989) improves the size of a circuit for the characteristic polynomial by a factor of a bit more than $\sqrt{n}$. However, we do not know if the same improvement can be obtained for computing the characteristic polynomials of all principal submatrices, with or without poly-log circuit depth.

Sketch of Proof of Theorem 1. If we denote by I_n an n-dimensional identity matrix, and if $(B)_{i,j}$ denotes the element in row i and column j in the matrix B, then Chistov's algorithm is based on the identities

$$\begin{aligned}
\frac{1}{\mathrm{Det}(I_k - \lambda A_k)} &= \prod_{l=1}^{k} \left((I_l - \lambda A_l)^{-1}\right)_{l,l} \\
&= \prod_{l=1}^{k} \left(\sum_{m=0}^{\infty} \lambda^m A_l^m\right)_{l,l} \\
&\equiv \prod_{l=1}^{k} \sum_{m=0}^{n} (A_l^m)_{l,l} \lambda^m \pmod{\lambda^{n+1}} \\
&=: \varphi_k(\lambda).
\end{aligned}$$

One computes the coefficients $(A_l^m)_{l,l}$ of the polynomials of the right-hand-side product by lemma 1, namely one finds

$$A_l e_l^{(l)}, A_l^2 e_l^{(l)}, \ldots, A_l^n e_l^{(l)},$$

where $e_l^{(l)}$ is the l-dimensional unit vector with identity in the l-dimension. Then the elements $(A_l^m)_{l,l}$ are the last entries in the vectors $A_l^m e_l^{(l)}$. Thus we obtain in $O(n^{\omega+1}\log(n))$ size and $O(\log(n)^2)$ depth polynomials

$$\psi_l(\lambda) := \sum_{m=0}^{n} (A_l^m)_{l,l}\lambda^m.$$

By parallel prefix, we then obtain

$$\varphi_k(\lambda) = \left(\prod_{l=1}^{k} \psi_l(\lambda)\right) \bmod \lambda^{n+1}$$

in $O(n\mathrm{M}(n))$ size and $O(\log(n)^2)$ depth, where $\mathrm{M}(n)$ is the asymptotic complexity of n-degree polynomial multiplication over D (currently the best being $n\log(n)\log(\log n)$ (Cantor and Kaltofen 1987)). The characteristic polynomial $\chi_k(\lambda) := \mathrm{Det}(\lambda I_k - A_k)$ is now the truncated power series inverse of φ_k, $\varphi_k(\lambda)^{-1} \bmod \lambda^{k+1}$ with the order of the coefficients reversed. Using Newton iteration, all n power series inversions can be carried out in $O(n\mathrm{M}(n))$ size and $O(\log(n)^2)$ depth. $\boxtimes$

We remark that theorem 1 also can proven using Berkowitz's (1984) approach to computing the determinant of a matrix in parallel.

4. Parallel Polynomial Subresultant Computation

We now present our algorithm that for two polynomials $f(x), g(x) \in \mathsf{D}[x]$, D an integral domain, $m := \deg(g) \le \deg(f) =: n$, computes all subresultants $S_{d_i}(f, g)$, where d_i ranges over the all degrees of remainders in the polynomial remainder sequence of f and g. If $m < n$ and the leading coefficient a_n of f is not a unit, the algorithm will perform exact divisions by a_n. The algorithm is realized by uniform circuits over D of

$$\text{size} = O(n^{\omega+1}\log(n)) \text{ and depth} = O(\log(n)^2). \tag{4}$$

Let $f(x) =: a_n x^n + \cdots + a_0$ and let $g(x) =: b_n x^n + \cdots + b_0$; note that $b_n = \cdots = b_{m+1} = 0$. We first consider the $(2n) \times (2n)$ Sylvester matrix of f and g:

$$R = \begin{pmatrix} a_n & a_{n-1} & \cdots\cdots & a_0 & & & \\ b_n & b_{n-1} & \cdots\cdots & b_0 & & & \\ & a_n & \cdots\cdots & a_1 & a_0 & & \\ & b_n & \cdots\cdots & b_1 & b_0 & & \\ & & \ddots & & \ddots & \ddots & \\ & & & a_n & \cdots\cdots\cdots & & a_0 \\ & & & b_n & \cdots\cdots\cdots & & b_0 \end{pmatrix}. \tag{5}$$

By the algorithm discussed in theorem 1 we can compute the characteristic polynomials $\chi_k(\lambda)$ of all $k \times k$ principal submatrices R_k of (5) within the complexity of (4). Comparing

R_{2j} with (1) and (3) we observe that by the adjoint formula for the matrix inverse, the first row of

$$\mathrm{Adj}(R_{2j}) := \mathrm{Det}(R_{2j})R_{2j}^{-1}$$

constitutes the coefficients of the polynomial multipliers $u_j, v_j \in \mathsf{D}[x]$ in the scheme

$$u_j f + v_j g = a_n^{n-m} S_j(f,g). \tag{6}$$

Notice, that v_j formally has $n-j$ coefficients, but its degree is only $m-j$.

By using lemma 1 (again), one can now compute from $\chi_{2j}(\lambda)$ the entries in the first rows of all $\mathrm{Adj}(R_{2j})$, $0 \le j \le m$, within the complexity (4); note that we know no way to compute all R_{2j}^{-1} within the same complexity, but luckily we only need the first rows. The argument is as follows: let

$$\chi_{2j}(\lambda) =: \lambda^{2j} + c_{2j,2j-1}\lambda^{2j-1} + \cdots + c_{2j,0};$$

then by the Cayley-Hamilton theorem

$$\mathrm{Adj}(R_{2j}) = -R_{2j}^{2j-1} - c_{2j,2j-1}R_{2j}^{2j-2} - \cdots - c_{2j,2}R_{2j} - c_{2j,1}I_{2j}.$$

Thus the first row of $\mathrm{Adj}(R_{2j})$ is obtained by multiplying the right-hand side by a unit row vector with the identity element in the first column, $(e_1^{(2j)})^{\mathrm{tr}}$. Lemma 1 now shows how to compute all

$$(e_1^{(2j)})^{\mathrm{tr}} R_{2j}^m, \quad 2 \le m \le 2j-1,$$

in $O(j^\omega \log(j))$ size and $O(\log(j)^2)$ depth.

We now have the coefficients of all polynomial multipliers in (6). By parallel polynomial multiplications and a scalar exact division we get all $S_j(f,g)$. The fundamental theorem of subresultants (2) now comes into play, because it accounts for all $S_j(f,g)$. Testing which of the $S_j(f,g)$ vanish (hence the change to the network model (von zur Gathen 1985) below), we can decide exactly which j is a degree d_i of a polynomial remainder, and what is its corresponding subresultant. We have the following theorem:

Theorem 2. *There exists a uniform family of algebraic networks over the integral domain* D *(with exact division) that is of size* $O(n^{\omega+1}\log(n))$ *and simultaneously of depth* $O(\log(n)^2)$ *and that for two polynomials* $f, g \in \mathsf{D}[x]$ *of degree* n *and* m*, respectively,* $m \le n$*, computes all "extended subresultant schemes" in the polynomial remainder sequence of* f *and* g*,*

$$u_{d_i} f + v_{d_i} g = S_{d_i}(f,g), \quad u_{d_i}, v_{d_i} \in \mathsf{D}[x],$$

where d_i *is the degree of the* i*-th remainder in the sequence, and* $\deg(u_{d_i}) < m - d_i$*,* $\deg(v_{d_i}) < n - d_i$*.* ⊠

This theorem implies the most processor-efficient poly-log time solution for the polynomial greatest common divisor problem that is currently known. For simplicity, we formulate the following theorem for polynomials over a field and for the computation taking place on an algebraic PRAM, which also saves (trivially) the $\log(n)$ factor in the size.

Theorem 3. *Let* $f, g \in \mathsf{K}[x]$ *have degree* $n \ge m$*, respectively. Then one can compute the greatest common divisor of* f *and* g *in time* $O(\log(n)^2)$ *on an algebraic PRAM with* $n^{\omega+1}$ *processors.* ⊠

In §5 we give further applications of theorem 2. It has also remarkable implications to the sequential complexity of the subresultant chain problem. Consider the problem of computing all $S_{d_i}(f, g)$. The optimal sequential solutions requires $O(n^2)$ arithmetic steps including exact divisions in D. By Strassen's (1973) elimination of divisions one can get all $S_{d_i}(f, g)$ in $O(n^2\mathrm{M}(n))$ additions, subtractions, and multiplications in D. The solution of theorem 2 only needs to divide by the leading coefficient of f. Hence, for monic f and classical polynomial and matrix arithmetic ($\mathrm{M}(n) = n^2$, $\omega = 3$), our parallel solution misses in size the best known sequential method by only a $\log(n)$ factor. Furthermore, one can realize our approach sequentially also in $O(n^2\mathrm{M}(n))$ by observing that a vR_k row vector v times matrix R_k product can be can be computed by a polynomial product-sum.

5. Applications

Theorem 2 computes more information than the GCD of the input polynomials alone. There are several applications of this "extended Euclidean remainder sequence," and we shall give two of them. The first makes use of all remainders computed.

Theorem 4. *Let $f(x) \in \mathbb{Q}[x]$ be a squarefree polynomial with rational coefficients, and let $a < b$ be two rational numbers. Then one can compute the number of real roots of f in the interval $\{\xi \mid a < \xi \leq b\}$ in time $O(\log(n)^2)$ on an algebraic PRAM over $\mathbb{Q}$ with $n^{\omega+1}$ processors and the additional comparison operation "greater than".*

Sketch of Proof. The argument is by classical Sturm sequences (see (Jacobson 1974, §5.3)). Let $f_0 := f$, $f_1 := f'$, and consider the polynomial remainder sequence

$$f_{i-2}(x) = q_i(x)f_{i-1}(x) - \frac{\beta_i}{\alpha_i}f_i(x), \quad 2 \leq i \leq k+1, f_{k+1} = 0, \frac{\beta_i}{\alpha_i} > 0. \tag{7}$$

Then one counts the number of sign variations $V[\![f]\!](a)$ and $V[\![f]\!](b)$ in the sequences

$$f_1(a), f_2(a), \ldots, f_k(a) \text{ and } f_1(b), f_2(b), \ldots, f_k(b),$$

respectively. The number of real zeros in the given interval is

$$V[\![f]\!](a) - V[\![f]\!](b).$$

The crucial observation is that by (2) we can from the subresultants set up a Sturm sequence (7), that is we can choose the correct sign of the leading coefficient in each of the f_i. ⊠

Our second application comes from coding theory, but it has direct relevance to algebraic problems such as sparse multivariate polynomial interpolation (Ben-Or and Tiwari 1988). Consider an infinite sequence $a_0, a_1, a_2, \ldots$ of elements in a field K, $a_0 \neq 0$. Assume the elements are generated by a linear recurrence, that is there exists a polynomial $\chi(\lambda) = \lambda^n - c_{n-1}\lambda^{n-1} - \cdots - c_0 \in \mathsf{K}[\lambda]$ such that

$$a_i = c_{n-1}a_{i-1} + \cdots + c_0 a_{i-n} \text{ for all } i \geq n.$$

The Berlekamp-Massey problem can now be stated as the problem to find a minimal degree linear recurrence that generates the sequence. It is not difficult to prove that such a minimal recurrence is unique, and that it is determined by only the first $2n$ elements in the sequence. We have the following processor-efficient parallel solution:

Theorem 5. *Given $a_0, \ldots, a_{2n-1} \in \mathsf{K}$ one can compute the minimal linear generating recurrence in time $O(\log(n)^2)$ on an algebraic PRAM over K with $n^{\omega+1}$ processors.*

Sketch of Proof. Consider the Euclidean scheme

$$U(x) \underbrace{x^{2n}}_{f_0(x)} + \Lambda(x) \underbrace{(a_0 x^{2n-1} + \cdots + a_{2n-1})}_{f_1(x)} = \underbrace{\Omega(x)}_{f_i(x)}.$$

If $\Omega(x)$ corresponds to the first remainder of degree no more than n, it is easy to see that the normalized multipier polynomial $\Lambda(x)$ is the minimal generating polynomial of the sequence. ⊠

Theorem 5 essentially shows that the processor complexity of the Berlekamp-Massey problem is bounded by that of a determinant computation. We like to point out that—in a certain sense—the reverse can also be establish. From lemma 1 and the approach by Wiedemann (1986) we can deduce the following reduction:

Theorem 6. *Given a poly-log time algebraic PRAM over an infinite field K with n^{ω} processors that can compute the minimal recurrence polynomial of a sequence of $2n$ elements, then one can construct a poly-log time randomized algebraic PRAM over K with n^{ω} processors that can compute in a Las Vegas fashion the determinant of an n-dimensional matrix over K.* ⊠

It should be noted that both the premise and the consequence statements in this theorem are at this time open problems. Nonetheless, it is the first-known parallel reduction from solving linear systems to computing polynomial GCDs.

Literature Cited

Ben-Or, M. and Tiwari, P., "A deterministic algorithm for sparse multivariate polynomial interpolation," *Proc. 20th Annual ACM Symp. Theory Comp.*, pp. 301–309 (1988).

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).

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., "On Euclid's algorithm and the computation of polynomial greatest common divisors," *J. ACM* **18**, pp. 478–504 (1971).

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`.

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).

Collins, G. E., "Subresultants and reduced polynomial remainder sequences," *J. ACM* **14**, pp. 128–142 (1967).

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

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).

Jacobson, N., *Basic Algebra I*; W. H. Freeman & Co., San Francisco, 1974.

Keller-Gehrig, W., “Fast algorithms for the characteristic polynomial,” *Theor. Comput. Sci.* **36**, pp. 309–317 (1985).

Massey, J. L., “Shift-register synthesis and BCH decoding,” *IEEE Trans. Inf. Theory* IT-**15**, pp. 122–127 (1969).

Pan, V. and Reif, J., “Some polynomial and Toeplitz matrix computations,” *Proc. 28th Annual IEEE Symp. Foundations Comp. Sci.*, pp. 173–184 (1987).

Strassen, V., “Vermeidung von Divisionen,” *J. reine u. angew. Math.* **264**, pp. 182–202 (1973). In German.

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