A DETERMINISTIC ALGORITHM FOR RATIONAL FUNCTION INTERPOLATION

DIMA YU. GRIGORIEV
STEKLOV INSTITUTE OF MATHEMATICS
SOV. ACAD. OF SCIENCES
LENINGRAD 191011

MAREK KARPINSKI *
DEPT. OF COMPUTER SCIENCE
UNIVERSITY OF BONN
5300 BONN 1

Abstract.

We construct a deterministic polynomial time (deterministic boolean NC [C 85]) algorithm for interpolating arbitrary t-sparse rational functions. It is the first deterministic polynomial time algorithm for this problem. Given an arbitrary rational function f (in the most general setting given by a *black box*), such that $f = P/Q$, $P, Q \in \mathbb{Z}[x_1, \cdots, x_n]$ for P, Q t-sparse, relatively prime, with coefficients bounded in absolute value by 2^n, and such that $deg_{x_i}(P)$, $deg_{x_i}(Q) < d$, the algorithm works in $O(\log^3(ndt))$ boolean parallel time and $O(n^4\, d^{7.5}\, t^{17.5} \log n\, \log d\, \log t)$ boolean processors. (In general, if the coefficients of P and Q are bounded in the absolute value by the function S, $S : \mathbb{N} \to \mathbb{N}$, $S(n) \geq n$, there exists an implementation (cf. [BCP 83]) of our algorithm in $NC^3(\log S)$.)

Having established the above deterministic circuit complexity upper bounds for the Rational Function Interpolation, the very challenging practical matter arises to improve on the number of boolean processors (or the sequential deterministic time) of our algorithm.

*Supported in part by Leibniz Center for Research in Computer Science, by the DFG Grant KA 673/2-1, and by the SERC Grant GR-E 68297.

1. Introduction.

The problem of interpolating rational functions over fields of characteristic 0 belongs to one of the most fundamental questions in computational mathematics. The interpolation formulas of Lagrange, Newton, and Hermite laid foundations for the whole area of numeric analysis. The recent algorithmic developments in problems like polynomial factorization and GCD computations (cf. [G 83], [K 85], [G 86] [KT 88], [BT 88]) lead to the new general problem of a *black box sparse* interpolation of multivariate polynomials and rational functions.

Very recently, the problem of interpolating polynomials given by determinants over fields of characteristic 0 was solved in [GK 87], a general polynomial interpolation in fields of characteristic 0 in [BT 88], and over arbitrary finite fields in [GKS 88].

The first randomized algorithm for the Sparse Rational Interpolation was designed by Kaltofen and Trager [KT 88]. Their method of solution depends on the Padé approximation and the Hilbert irreducibility theorem, and is intrinsically probabilistic. For a given $\varepsilon > 0$, it gives a randomized algorithm not failing with probability greater than $1 - \varepsilon$ and running in time polynomial in $\log(\frac{1}{\varepsilon})$.

In this paper, we give a deterministic solution of the Sparse Rational Interpolation. Given (as a black box) an arbitrary rational function $f = P/Q$ with $P, Q \in \mathbb{Z}[x_1, \cdots, x_n]$ (we assume P, Q are relatively prime, and with coefficients bounded in absolute value by 2^n) and P, Q having at most t nonzero terms and $deg_{x_i}(P)$, $deg_{x_i}(Q) < d$, there exists a deterministic polynomial time (deterministic boolean NC) algorithm for computing P and Q. Our algorithm depends on the new method developed here of solving linear systems of equations with polynomials as indeterminates containing black boxes as coefficients. This extends in an interesting way techniques of Mulmuley [M 86], and might also be of independent interest.

The algorithm will be formulated and analysed in terms of uniform boolean circuits to give better insight into both the sequential and the parallel time and space requirements (cf. [C 79], [C 85]). The boolean circuit complexity of this problem could also be of independent interest. The Rational Interpolation algorithm works in $O(\log^3(ndt))$ deterministic boolean time and $O(n^4\, d^{7.5}\, t^{17.5} \log n \log d \log t)$ boolean processors.

2. Sparse Rational Interpolation Problem

We consider the general problem of interpolating rational functions f defined by quotients of polynomials with at most t nonzero terms. In the most general setting we are given a black box for f capable of producing values of f for its arguments. (The very special cases of interpolations are those where the black boxes are given by straight-line programs, determinants, formulas, etc. (cf. [KT 85a]).) In this setting, we are given by a black box a rational function $f = P/Q$, information of its sparsity t, and degree d. (From now on, we shall assume that the absolute value of coefficients is bounded by 2^n.) The Sparse Rational Interpolation Problem $IP(t,d)$ is the problem of computing all nonzero terms of P and Q from the values of f.

We say that the interpolation problem $IP(t,d)$ is in NC^k (cf. [C 85]) if there exists a class of uniform $n^{0(1)}$-size and $O(\log^k n)$- depth boolean circuits with oracle nodes S (returning values of a black box) computing for arbitrary n-variate rational function $f = P/Q$ ($GDC(P,Q) = 1$) all the nonzero coefficients and monomial vectors of P and Q, with the oracle S_f, defined by $S_f(x_1, \cdots, x_n, y)$ iff $f(x_1, \cdots, x_n) = y$. It is obvious that if the computation of $f(x_1, \cdots, x_n)$ performed by a black box (straight-line program, determinant, formula, etc.) is in NC (in P), then the interpolation problem $IP(t,d)$ itself lies in NC (in P).

Theorem. The Rational Sparse Interpolation Problem $IP(t,d)$ is in NC^3. Given any rational function f, such that $f = P/Q$, P, Q t-sparse, relatively prime, and such that $deg_{x_i}(P)$, $deg_{x_i}(Q) < d$, there exists a deterministic parallel algorithm for interpolating f working in $O(n^4 d^{7.5} t^{17.5} \log n \log d \log t)$ boolean processors and $O(\log^3(ndt))$ parallel time.

PROOF. In Sections 3 and 4.

Let us also note that there exists an obvious implementation ([BCP 83])of our algorithm in $NC^3(\log S)$ for the arbitrary upper bounds S, $S : I\!N \to I\!N$, on the absolute values of the coefficients.

3. Rational Interpolation Algorithm.

Input: Given a black box for a t-sparse rational function f, $f = P/Q$, where $P, Q \in \mathbb{Z}[x_1, \cdots, x_n]$ and P, Q are both t-sparse, P, Q are relatively prime and $deg_{x_i}(P), deg_{x_i}(Q) < d$.

Output: P, Q (all nonzero coefficients, and monomial vectors).

Assume $n = 2^m$. Define $S^{(1)}_{\alpha,\beta} = \{I = (i_1, \cdots, i_{2^{\alpha-1}}) : x^{i_1}_{\beta 2^{\alpha-1}+1} \cdots x^{i_{2^{\alpha-1}}}_{\beta 2^{\alpha-1}+2^{\alpha-1}}$ occurs in P$\}$.
$S^{(2)}_{\alpha,\beta} = \{J = (j_1, \cdots, j_{2^{\alpha-1}}) : x^{j_1}_{\beta 2^{\alpha-1}+1} \cdots x^{j_{2^{\alpha-1}}}_{\beta 2^{\alpha-1}+2^{\alpha-1}}$ occurs in Q$\}$.

Algorithm:

The algorithm by recursion on α calculates $S^{(1)}_{\alpha,\beta}$, $S^{(2)}_{\alpha,\beta}$ for all $0 \le \beta < 2^{m+1-\alpha}$.

Basis $\alpha = 1$. For each $0 \le k_1, k_2 < d$ and for every variable $x_j, 1 \le j \le n$ in parallel, we substitute instead of x_j the pairwise distinct numbers $a_0, \cdots, a_{2d}$ in the rational function

$\frac{\sum_{0\leq i\leq k_1} x_j^i P_i}{\sum_{0\leq i\leq k_2} x_j^i Q_i} = f$, where $P_i, Q_i \in \mathbb{Z}[x_1,\cdots,x_{j-1},x_{j+1},\cdots,x_n]$. So the algorithm considers the linear system $\sum_{0\leq i\leq k_1} a_\ell^i P_i = f\cdot\sum_{0\leq i\leq k_2} a_\ell^i Q_i, 0\leq \ell\leq 3d$, considering P_i, Q_i as indeterminates with coefficients "containing" black boxes $f(x_1,\cdots,x_{j-1},a_\ell,x_{j+1},\cdots,x_n)$. The algorithm applies to this linear system a method [M 86] and solves it in the following sense. During performing [M 86] we need to test whether some polynomial in the black boxes mentioned vanishes as the polynomial in the variables $x_1,\cdots,x_{j-1},x_{j+1},\cdots,x_n$. We do it with the aid of zero-test from [T 87], [GK 87] (cf. also [GKS 88]). As a result of solving a system (provided that it is solvable), we express P_i, Q_i and test for each of them its nonvanishing, again relying on zero-test from [T 87]. If $P_{k_1}\not\equiv 0, Q_{k_2}\not\equiv 0$ (we check the latter again involving zero-test), then the pair k_1, k_2 *fits* (so there exists a representation $P/Q = f$ with $deg_{x_1}P = k_1, deg_{x_1}Q = k_2$) and at the end we choose among such pairs the minimal pair (e.g. with the least k_1). Then we yield $S_{1,j}^{(1)}, S_{1,j}^{(2)}$ using zero-test according to nonzero terms just for this pair k_1, k_2.

Let us prove the correctness of the algorithm for $\alpha = 1$. We have to prove that $\frac{\sum_{0\leq i\leq k_1} x_j^i P_i}{\sum_{0\leq i\leq k_2} x_j^i Q_i} = f$. Represent $f = \frac{\sum_{0\leq i<d} x_j^i P_i^{(1)}}{\sum_{0\leq i<d} x_j^i Q_i^{(1)}}$ in nonreducible way where $P_i^{(1)}, Q_i^{(1)} \in \mathbb{Z}[x_1,\cdots,x_{j-1},x_{j+1},\cdots,x_n]$. Then $(\sum_{0\leq i\leq k_1} a_\ell^i P_i)(\sum_{0\leq i\leq d} a_\ell^i Q_i^{(1)}) = (\sum_{0\leq i\leq k_2} a_\ell^i Q_i)(\sum_{0\leq i<d} a_\ell^i P_i^{(1)}) \in \mathbb{Z}[x_i,\cdots,x_{j-1},x_{j+1},\cdots,x_n]$ for each $0\leq \ell\leq d$. Hence $(\sum_{0\leq i\leq k_1} x_j^i P_i)(\sum_{0\leq i<d} x_j^i Q_i^{(1)}) = (\sum_{0\leq i\leq k_2} a_j^i Q_i)(\sum_{0\leq i<d} x_j^i P_i^{(1)})$, because the degrees in x_j in both sides are less than $2d$, therefore $\frac{\sum_{0\leq i\leq k_1} x_j^i P_i}{\sum_{0\leq i\leq k_2} x_j^i Q_i} = f$. Moreover, as k_1, k_2 is chosen as the least pair, $S_{1,j}^{(1)}, S_{1,j}^{(2)}$ correspond to the nonreducible representation of f. □

Recursion Step.

Assume that we have already produced $S_{\alpha,\beta}^{(1)}, S_{\alpha,\beta}^{(2)}$ for all $0\leq\beta<2^{m-\alpha+1}$. Now we produce $S_{\alpha+1,\beta}^{(1)}, S_{\alpha+1,\beta}^{(2)}$ for fixed $o\leq\beta<2^{m-\alpha}$. For each element from $S_{\alpha,2\beta}^{(*)}$ and the element from $S_{\alpha,2\beta+1}^{(*)}$ where either $* = 1$ or $* = 2$, consider the corresponding product $x^{\boldsymbol{k}} = x_{\beta 2^\alpha+1}^{k_1}\cdots x_{\beta 2^\alpha+2^\alpha}^{k_{2^\alpha}}$. For all such products (observe that the number of them is at most $2t^2$ since $|S_{\alpha,2\beta}^{(*)}|, |S_{\alpha,2\beta+1}^{(*)}|\leq t$) we can write $\frac{\sum_{\boldsymbol{k}^{(1)}} x^{\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}}}{\sum_{\boldsymbol{k}^{(2)}} x^{\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}}} = f$ where $P_{\boldsymbol{k}^{(1)}}, Q_{\boldsymbol{k}^{(2)}} \in \mathbb{Z}[x_1,\cdots,x_{\beta 2^\alpha},x_{(\beta+1)2^\alpha+1},\cdots,x_n]$. So the algorithm takes pairwise different primes $p_1,\cdots,p_{2^\alpha}$ and in parallel for every $0\leq\ell<2t^3$ substitutes instead of $x_{\beta 2^\alpha+j}$ the number p_j^ℓ for $1\leq j\leq 2^\alpha$. Then the algorithm looks over all pairs $0\leq k_1,k_2<nd$ and for

every k_1, k_2 considers a linear system $\sum_{|\boldsymbol{k}^{(1)}|\le k_1} p^{\ell\cdot\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}} = f \cdot \sum_{|\boldsymbol{k}^{(2)}|\le k_2} p^{\ell\cdot\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}}$ for $0 \le \ell < 2t^3$ considering $P_{\boldsymbol{k}^{(1)}}, Q_{\boldsymbol{k}^{(2)}}$ as indeterminants, where $|\boldsymbol{k}^{(1)}|$ is a sum of coordinates of the multi-index $\boldsymbol{k}^{(1)}$, and solves it involving [M 86] similarly as above in the basis step. This would lead to an expressing $P_{\boldsymbol{k}^{(1)}}, Q_{\boldsymbol{k}^{(2)}}$ (provided that the linear system is solvable) as black boxes. Involving zero-test [Ti 87], we check whether $P_{\boldsymbol{k}^{(1)}} \not\equiv 0, Q_{\boldsymbol{k}^{(2)}} \not\equiv 0$ for some $|\boldsymbol{k}^{(1)}| = k_1, |\boldsymbol{k}^{(2)}| = k_2$. Then this pair k_1, k_2 *fits* and we take among such pairs the least one (e.g. with the least k_1). Then we yield $S^{(1)}_{\alpha,\beta}, S^{(2)}_{\alpha,\beta}$ using zero-test just for this pair k_1, k_2.

We prove the correctness of the recursive step of the algorithm: We have to prove that $\dfrac{\sum_{|\boldsymbol{k}^{(1)}|\le k_1} x^{\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}}}{\sum_{|\boldsymbol{k}^{(2)}|\le k_2} x^{\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}}} = f$. Represent $f = \dfrac{\sum_{|\bar{\boldsymbol{k}}^{(1)}|<nd} x^{\bar{\boldsymbol{k}}^{(1)}} \bar{P}_{\bar{\boldsymbol{k}}^{(1)}}}{\sum_{|\bar{\boldsymbol{k}}^{(2)}|<nd} x^{\bar{\boldsymbol{k}}^{(2)}} \bar{Q}_{\bar{\boldsymbol{k}}^{(2)}}}$ in nonreducible way and denominator and nominator are t-sparse where $\bar{P}_{\bar{\boldsymbol{k}}^{(1)}}, \bar{Q}_{\bar{\boldsymbol{k}}^{(2)}} \in \mathbb{Z}[x_1, \cdots, x_{\beta 2^\alpha}, x_{(\beta+1)2^\alpha+1}, \cdots, x_n]$. Then $(\sum_{|\boldsymbol{k}^{(1)}|\le k_1} p^{\ell\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}})(\sum_{|\bar{\boldsymbol{k}}^{(2)}|<nd} p^{\ell\bar{\boldsymbol{k}}^{(2)}} \bar{Q}_{\bar{\boldsymbol{k}}^{(2)}}) = (\sum_{|\boldsymbol{k}^{(2)}|\le k_2} p^{\ell\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}})(\sum_{|\bar{\boldsymbol{k}}^{(1)}|<nd} p^{\ell\bar{\boldsymbol{k}}^{(1)}} \bar{P}_{\bar{\boldsymbol{k}}^{(1)}})$ for $0 \le \ell < 2t^3$.

Hence (1)

$$(\sum_{|\boldsymbol{k}^{(1)}|\le k_1} x^{\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}})(\sum_{|\bar{\boldsymbol{k}}^{(2)}|<nd} x^{\bar{\boldsymbol{k}}^{(2)}} \bar{Q}_{\bar{\boldsymbol{k}}^{(2)}}) = (\sum_{|\boldsymbol{k}^{(2)}|\le k_2} x^{\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}})(\sum_{|\bar{\boldsymbol{k}}^{(1)}|<nd} x^{\bar{\boldsymbol{k}}^{(1)}} \bar{P}_{\bar{\boldsymbol{k}}^{(1)}})$$

because polynomials in both sides contain at most t^3 terms in $x_{\beta\cdot 2^\alpha+1}, \cdots, x_{\beta\cdot 2^\alpha+2^\alpha}$, each of them; so we can write the latter identity in the form $\sum_{\boldsymbol{k}} x^{\boldsymbol{k}} T_{\boldsymbol{k}} = 0$ where $T_{\boldsymbol{k}} \in \mathbb{Z}[x_1, \cdots, x_{\beta 2^\alpha}, x_{(\beta+1)2^\alpha+1}, \cdots, x_n]$ and the latter polynomial contains at most $2t^3$ terms. On the other hand, $\sum p^{\ell\boldsymbol{k}} T_{\boldsymbol{k}} = 0$ for $0 \le \ell < 2t^3$ and $p^{\tilde{\boldsymbol{k}}} \ne p^{\tilde{\tilde{\boldsymbol{k}}}}$ for $\tilde{\boldsymbol{k}} \ne \tilde{\tilde{\boldsymbol{k}}}$ and the latter system can be written in the form $UT = 0$ where U is a Vandermonde matrix and the vector $T = (\{T_{\boldsymbol{k}}\}_{\boldsymbol{k}})$, therefore $T = 0$, that proves the identity (1). Thus $f = \dfrac{\sum_{|\boldsymbol{k}^{(1)}|\le k_1} x^{\boldsymbol{k}^{(1)}} P_{\boldsymbol{k}^{(1)}}}{\sum_{|\boldsymbol{k}^{(2)}|\le k_2} x^{\boldsymbol{k}^{(2)}} Q_{\boldsymbol{k}^{(2)}}}$.

Moreover, as k_1, k_2 were chosen as the least possible, produced $S^{(1)}_{\alpha+1,\beta}, S^{(2)}_{\alpha+1,\beta}$ correspond to the nonreducible representation of f. □

4. Analysis of the Algorithm.

We start with the analysis of the identity-to-zero algorithm. For a given t-sparse polynomial $P \in \mathbb{Z}[x_1, \cdots, x_n]$ given by a black box, the problem of checking identity-to-zero of P, whether $P \equiv 0$, can be solved by the following algorithm (see [T 87], [GK 87]).

Input. Black box for P.
Step 1. Generate n first prime numbers $p_1, \cdots, p_n$.
Step 2. Compute the numbers α_i (values of P at the points $(p_1^i, \cdots, p_n^i)$, $1 \leq i \leq t$:
$\alpha_i = P(p_1^i, \cdots, p_n^i), 1 \leq i \leq t$.
Output. Yes $(P \equiv 0)$ iff $\forall i[\alpha_i = 0]$.

By the Erastosthenes sieve method, Step 1 costs $O(n \log n)$ boolean processors (and $O(\log n)$ boolean time). Step 2 costs $O(nt \log n)$ processors. Thus, the whole identity-to zero-subroutine works in $O(nt \log n)$ processors and $O(\log n)$ parallel time.

Taking into account the complexity of the above subroutine and the costs of computing the rank of matrices [M 86], we get the following processor count of our algorithm. The Basis Step works in $O(n^2 d^{5.7} t \log d \log n)$ processors and $O(\log nt + \log^2 d)$ parallel time. The Recursion Step works in $O(n^4 d^2 t^{17.5} \log n \log t)$ processors and $O(\log^2 t + \log nd)$time. For every β, $0 \leq \beta \leq 2^{m-\alpha}$, the Recursion Step works in $O(\log^2(ndt))$ parallel time and $O(n^4 d^2 t^{17.5} \log n \log t)$ processors. As a consequence, the deterministic *Rational Interpolation Algorithm* works in $O(n^4 d^{7.5} t^{17.5} \log n \log d \log t)$ boolean processors and $O(\log^3(ndt))$ parallel time. □

5. Conclusions.

Our paper solves an open problem of Ben-Or and Tiwari [BT 88], and gives the first deterministic polynomial time (boolean NC) algorithm for the Sparse Multivariate Rational Interpolation Problem. The Rational Interpolation is connected to the restricted problem of determining whether there is a sparse vector in the null space of a given matrix; the unrestricted problem is known to be NP-complete (cf. [BT 88]. It it also connected in an interesting way to the seminal problem of Strassen [S 73] of computing the numerator and denumerator of general rational functions given by straight-line programs. Recently, Kaltofen [K 85b] was able to put this problem in random polynomial time, proving the existence of *non-uniform* polynomial size circuits for computing numerators and denominators. The algorithm in this paper proves the existence of *deterministic* uniform boolean circuits of polynomial size and polylogarithmic depth for computing numerators and denumerators of sparse rational functions given in the most general black-box oracle form for their values. The algorithm transforms deterministically arbitrary sparse rational straight-line programs into equivalent programs where only one division is allowed at the end of a computation. Having put the status of the above problem in P and in the deterministic boolean NC, an important practical problem arises to improve substantially on the number of processors of the algorithm.

References.

[AL 86] Adleman, L.M., and Lenstra, H.K., *Finding Irreducible Polynomials over Finite Fields*, Proc. ACM STOC (1986), pp. 350-355

[BT 88] Ben-Or, M., and Tiwari, P., *A Deterministic Algorithm for Sparse Multivariate Polynomial Interpolation*, Proc. 20^{th} ACM STOC (1988)

[BCP 83] Borodin, A., Cook, S.A., and Pippenger, N., *Parallel Computation for Well-Endowed Rings and Space-Bounded Probabilistic Machines*, Information and Control 58 (1983), pp. 113-136

[C 85] Cook, S.A., *A Taxonomy of Problems with Fast Parallel Algorithms*, Information and Control 64 (1985), pp. 2-22

[C 82] Cook, S.A., *Towards a Complexity Theory of Synchronous Parallel Computations*, Enseignement Mathematique 30 (1982), pp. 99-124

[G 83] von zur Gathen, J., *Factoring Sparse Multivariate Polynomials*, Proc. 24^{th} IEEE FOCS (1983), pp. 172-179

[G 84] von zur Gathen, J., *Parallel Algorithms for Algebraic Problems*, SIAM J. Comput., Vol. 13 (1984), pp. 808-824

[G 86] von zur Gathen, J., *Representations and Parallel Computations for Rational Functions*, SIAM J. Comput. 15 (1986), pp. 432-452

[GK 87] Grigoriev, D.Yu., and Karpinski, M., *The Matching Problem for Bipartite Graphs with Polynomially Bounded Permanents Is in NC*, Proc. 28^{th} IEEE FOCS (1987), pp. 166-172

[GKS 88] Grigoriev, D.Yu., Karpinski, M., and Singer, M.F., *Fast Parallel Algorithms for Sparse Multivariate Polynomial Interpolation over Finite Fields*, Research Report No. 8523-CS, University of Bonn, pp. 1-10 (May 1988)

[K 85a] Kaltofen, E., *Computing with Polynomials Given by Straight-Line Programs I: Greatest Common Divisors*, Proc. 17^{th} ACM STOC (1985), 131-142

[K 85b] Kaltofen, E., *Uniform Closure Properties of P-Computable Functions*, Proc. 18^{th} ACM STOC (1985), pp. 330-337

[KT 88] Kaltofen, E., and Trager, B., *Sparse Factorization and Rational Function Interpolation of Polynomials Given by Black-Boxes for their Evaluation*, Manuscript, 1988

[L 79] Lovasz, L., *On Determinants, Matchings, and Random Algorithms*, in: Fundamentals of Computation Theory, Akademie-Verlag, Berlin 1979, pp. 565-574

[M 86] Mulmuley, K., *A Fast Parallel Algorithm to Compute the Rank of a Matrix over an Arbitrary Field*, Proc. 18^{th} ACM STOC (1986), pp. 338-339

[S 73] Strassen, V., *Vermeidung von Divisionen*, Creller J. Reine und Angewandte Mathematik 264 (1973), pp. 184-202

[S 80] Schwartz, J.T., *Fast Probabilistic Algorithms for Verification of Polynomial Identities*, J. ACM 27,4 (1980), pp. 701-717

[T 87] Tiwari, P., *Deterministic Algorithms for Multivariate Polynomial Interpolation*, IBM T.J. Watson Research Center; Technical Report.

[Z 79] Zippel, R.E., *Probabilistic Algorithms for Sparse Polynomials*, Proc. EUROSAM '79, Springer Lecture Notes in Computer Science Vol. 72 (1979), pp. 216-226

[Z 88] Zippel, R.E., *Interpolating Polynomials from their Values*, Manuscript, 1988