Some Recent Algebraic/Numerical Algorithms

Victor Y. Pan *
Mathematics and Computer Science Department
Lehman College, City University of New York
Bronx, NY 10468
Internet: VPAN@LCVAX.LEHMAN.CUNY.EDU

December 17, 1998

Abstract

Combination of algebraic and numerical techniques for improving the computations in algebra and geometry is a popular research topic of growing interest. We survey some recent progress that we made in this area, in particular, regarding polynomial rootfinding, the solution of a polynomial system of equations, the computation of an approximate greatest common divisor of two polynomials as well as various computations with dense structured matrices and their further applications to polynomial and rational interpolation and multipoint polynomial evaluation. In some cases our algorithms reach nearly optimal time bounds and/or improve the previously known methods by order of magnitude, in other cases we yield other gains, such as improved numerical stability.

Key words: algebraic/numerical algorithms, polynomial rootfinding, solution of a polynomial system of equations, approximate gcd, dense structured matrices, Toeplitz matrices, Cauchy matrices, polynomial interpolation, rational interpolation, polynomial evaluation, Trummer's problem.

1991 Mathematics Subject Classification : 68Q40, 65Y20, 65D99.

*Supported by NSF Grants CCR 9625344 and CCR9732206 and PSC CUNY Awards Nos. 6683665 and 669363

1 Introduction

Historically, the algorithms for numerical computations and for symbolic and algebraic computations were designed and developed by two distinct groups of people having rather little interaction with each other. The basic principles, tools and techniques used in these two groups were also quite different from each other. Numerical computations are typically performed with a fixed (single or double) finite precision and the propagation and magnification of roundoff errors is usually avoided by means of various techniques of error and perturbation analysis and approximation theory. Symbolic and algebraic computations typically assume exact computations with no roundoff errors, that is, for an exact input, the output yields exact solution to the computational problem. To maintain exact solution computation, the typical tools and techniques of rational computations are applied, such as the Chinese remainder algorithm, p-adic (Newton-Hensel's) lifting and the extended Euclidean (continued fraction approximation) algorithm. In spite of using these sophisticated techniques, a certain price is to be paid for reaching the exact rather than an approximate solution: typically, the algebraic computations require substantially more resources of computer time and memory space than numerical computations for the problems of the same size.

The recent trend, however, is to merge both groups of algebraic and numerical tools and techniques to develop most effective algorithms for both exact and approximate solution of algebraic and geometric computational problems. Such a development is of benefit in both cases, accentuanting the power of each group of techniques. The reader is referred to [P92] and [BP94] for some further material on this subject.

An important point is that the customary tools and algorithms for the computations in applied linear algebra were typically developed by numerical analysts whereas advanced polynomial computations belong to the traditional domain of computer algebra based on the tools of symbolic computation. An effective bridge between numerical computation with matrices and symbolic computation with polynomials can be built based on the manipulation with dense structured matrices (such as Toeplitz, Hankel, Vandermonde and Cauchy matrices and the matrices having similar structures). The latter matrices can be treated by the known numerical techniques ensuring numerically stable computations. On the other hand, there is a tight link between the dense structured matrices and polynomials, in terms of the associated operators of displacement and scaling. Such a link enables equivalent interpretation of various polynomial computation problems in terms of computations with dense structured matrices. These observations are exploited explicitly or implicitly in most of our algorithms.

In the present paper we survey our recent progress in the study of symbolic/numeric computations. We cover the following particular topics: solution of a polynomial equation and a polynomial system of equations, computation of approximate polynomial greatest common divisors and computations with dense structured matrices with their further applications to polynomial and rational interpolation and multipoint polynomial evaluation. Each of these topics is a well-known classical algebraic subject, important for the theory and practice of computing and still highly popular among the researchers. We seek approximate solutions by combining algebraic and numerical techniques to improve the known algorithms. In some cases we yield output approximations in nearly optimal time and/or improve by order of magnitude the known computational complexity estimates; in other cases our gain is the improved numerical stability. We omit or keep very brief the discussion on the (usually well-known) importance of the presented computational problems and the full description of the algorithms, referring the reader to our bibliography, which also contains further references.

Our range of topics is quite broad but personal and by no means complete, even regarding our own research. For instance, we do not cover important progress recently achieved by combining numerical and algebraic techniques for the computation of the sign of algebraic and algebraic-geometric predicates. This progress includes in particular the computation of the sign of matrix determinant, which is a major problem of practical geometric computations [BEPP97], [BEPP99], [PY99].

We organize the paper as follows. In the next section, we will specify our models of computing. Then we cover polynomial rootfinding in section 3, solution of a polynomial system of equations in section 4, computation of approximate polynomial gcd in section 5 and computations with dense structured matrices (with applications to polynomial and rational interpolation and multipoint evaluation) in section 6.

2 Models of computing

Stating our computational complexity estimates, we will assume the customary computational models of arithmetic and Boolean Random Access Machines (RAM modeles) and their parallel modifications (PRAM models) [AHU74], [KR90], [Qui94]. The estimates can be restated immediately in terms of the model of [BSS89] as well as the arithmetic and Boolean circuits [Gat86]. Moreover, to large extent, our results on computational complexity are model independent. They are reduced to invocation of a few fundamental operations such as the computation of the inner products and convolutions of vector pairs and discrete Fourier transform, for which effective algorithms are available under all the above models as well as under some more realistic models of parallel computing [Lei92].

We will write $O_A(t(N))$ and $O_B(T(N))$ to denote the sequential arithmetic and Boolean time bounds $O(t(N))$ and $O(T(N))$, respectively, for the input size N. Under parallel models of computing, we will write $O_A(t(N), p(N))$ or $O_B(t(N), P(N))$ to denote the simultaneous time and arithmetic or Boolean processor bounds $O(t(N))$ and $O(p(N))$, respectively. We will assume Brent's scheduling principle, according to which a simple processor needs time $O(pt)$ to simulate the work of p processors performed in time t. In other words, the parallel cost bounds $O_A(t, p)$ or $O_B(t, p)$

imply the sequential cost bounds $O_A(pt)$ or $O_B(pt)$, respectively.

3 Polynomial rootfinding

Problem 3.1, *polynomial rootfinding.* Given a positive b and real or complex coefficients $p_0, \ldots, p_{n-1}$ of the monic polynomial $p(x) = \sum_{i=0}^{n} p_i x^i = \prod_{j=1}^{n}(x - z_j)$ with zeros $z_1, \ldots, z_n$ lying in the unit disc $D(0,1) = \{x : |x| \leq 1\}$, compute approximations to the zero, $z_1^*, \ldots, z_n^*$, satisfying

$$|z_j^* - z_j| < 2^{-b},\ j = 1, \ldots, n.$$

If some zeros of $p(x)$ are not in the unit disc $D(0,1)$, we may compute the values $r^* = \max_{k\geq 1} |p_{n-k}/p_n|^{1/k}$, $r^+ = 1.62r^*$, recall from [He74], pp. 451-457, that $r^+ > \max_j |z_j|$, and scale the variable x, that is, shift from $p(x)$ to the monic polynomial $q(y) = (r^+)^{-n} p(yr^+)$ with zeros $z_j/r^+, j = 1, \ldots, n$.

For a few millennea, the study of this problem was crucial for the development of mathematics and applied mathematics; presently this subject remains highly important for computer algebra (cf. [P97], [P98a]). The solution of this algebraic problem typically combines numerical and algebraic techniques, in some cases also involving geometric search on the complex plane. In particular, this is the case for the recent algorithms of [P95] (cf. also [P96], [P97], [P98a]), which support the following results.

Theorem 3.1 *Problem 3.1 can be solved at the computational cost bounded by $O_A((n\log^2 n)(\log^2 n + \log b))$ as well as by $O_A((\log^3 n)(\log^3 n + \log b), n/\log n)$ and $O_B((\log^3 n)\log^2(bn), M((b\log(bn)+n)n^2)/(\log^2(bn)(\log^2 n)))$, where $M(d)$ is the Boolean complexity of multiplication of a pair of integers modulo 2^d, $M(d) = O((d\log d)\log\log d)$.*

The estimates of Theorem 3.1 are optimal (up to polylogarithmic factors in n and b) under both arithmetic and Boolean models and in both cases under both sequential and parallel models of computing, because they are within polylogarithmic factors from the number of input coefficients as well as from the number of binary bits required to represent these coefficients in order to enable obtaining the output zeros within the error bound 2^{-b}. Indeed, we need to process all the n coefficients $p_0, \ldots, p_{n-1}$ in order to obtain even a single zero z_j. Moreover, we have to process order of bn^2 bits of $n/2$ trailing coefficients of such a polynomial as $(x - 1/7)^n$ in order to approximate within 2^{-b} its single (n-multiple) zero $z_1 = 1/7$.

The algorithms of [P95], [P96] supporting Theorem 3.1 involve sophisticated and advanced algebraic, numerical and geometric techniques, and this slows down the process of their numerical testing and practical implementation. The reader is referred to [P97] on the currents most competitive practical algorithms for polynomial rootfinding.

4 Solution of a polynomial system of equations

The next problem, highly important in computer algebra, is a natural extension of Problem 3.1.

Problem 4.1, *solution of a polynomial system of equations.* Given a positive b, a positive integer D, a vector norm $||\cdot||$, and the coefficients of a system of n polynomial equations,

$$p_i(x_1, \ldots, x_n) = 0, \; i = 1, \ldots, n, \tag{4.1}$$

in n variables $x_1, \ldots, x_n$ having at most D unknown common roots, $Z = (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(d)})$, $\mathbf{z}^{(i)} = (z_1^{(i)}, \ldots, z_n^{(i)})$, $i = 1, \ldots, d$; $d \leq D$, compute approximations $\tilde{\mathbf{z}}^{(1)}, \ldots, \tilde{\mathbf{z}}^{(d)}$ to the roots satisfying $||\tilde{\mathbf{z}}^{(i)} - \mathbf{z}^{(i)}|| \leq 2^{-b} \max \{1, ||\mathbf{z}^{(i)}||\}$, $i = 1, \ldots, n$.

Problem 4.1 is much harder than Problem 3.1. The known upper and lower bounds even on the sequential arithmetic computational cost of its solution remain far apart from each other; namely, they are $O_A(D^3)$ [Ren89] and $\Omega(D)$, respectively.

Practically, one frequently needs a solution of various subproblems of Problem 4.1, such as the following ones.

Problem 4.2, *counting the roots.* Compute the number of the roots of system (4.1).

Problem 4.3, *counting the real roots.* Compute the number of the real roots of system (4.1).

Problem 4.4, *counting the roots that lie in a fixed polytop.* Compute the number of the roots of system (4.1) lying in a fixed polytop bounded by H hyperplanes in the complex n-dimensional space $\mathbb{C}^n$.

As a particular case of Problem 4.4 for $H = 2n$, one may compute the number of nearly real roots having imaginary parts of all their n coordinates at the distance at most ϵ from 0, for a fixed small positive ϵ.

Problem 4.5. *Approximate all roots of system (4.1) that lie in a fixed polytop in $\mathbb{C}^n$ bounded by H hyperplanes in $\mathbb{C}^n$.*

Problem 4.6. *Approximate a root of the system (4.1) maximizing the value $|h(\mathbf{x})|$ of a fixed polynomial $h(\mathbf{x}) = h(x_1, \ldots, x_n)$.*

Problem 4.7. *Approximate a root of the system (4.1) minimizing the value $|h(\mathbf{x})|$ of a fixed polynomial $h(\mathbf{x})$.*

Until recently, $O_A(D^3)$ was the record known upper bound on the sequential arithmetic complexity of all these problems, even assuming *generic system* (4.1) whose solution is only sought for an algebraically open set in the input coefficient space. A substantial asymptotic acceleration, however, was achieved in our joint work with Bernard Mourrain [MP98], [MP98a], [MP,a] for Problems 4.2-4.7. We summarize this progress in the next theorem.

Theorem 4.1 *There are algorithms supporting the following arithmetic cost estimates : $O_A(3^n D^2 \log D)$ for Problems 4.2 and 4.3, $O_A(3^n H D^2 \log D)$ for Problem 4.4, $O_A(3^n(\delta + H)(D^2 \log D) \log b)$ for Problem 4.5, and $O((D^2 \log D) \log^2 b)$ for Problems*

4.6 and 4.7, assuming all these problems for generic polynomial system of equations (4.1).

The algorithms of [MP98], [MP98a], [MP,a] supporting Theorem 4.1 combine several advanced algebraic and numerical techniques involving dual spaces of linear forms (maps) on the primal linear space of polynomials in $\mathbb{C}^n$, algebraic residues, structured matrices, and the iterative processes of repeated squaring and sign iteration in the primal and dual spaces. So far, involvement of such sophisticated tools slowed down the experimental testing and practical implementation of the algorithms, but their simplified versions for Problems 4.6 and 4.7 (supporting the bounds $O_A(D^2 \log D)$ in terms of D) have been successfully tested numerically (cf. [BMP98]).

Some distinct algorithms for the fast and memory space efficient solution of sparse polynomial systems (4.1) were presented in [EP97].

5 Approximate polynomial gcds

Problem 5.1, *computation of approximate polynomial greatest common divisors (gcds).* Given two polynomials, $u(x) = \sum_{i=0}^{m} u_i x^i$ and $v(x) = \sum_{i=0}^{n} v_i x^i$, a fixed polynomial norm $\|\cdot\|$, and a positive ϵ, compute an ϵ-gcd of $u(x)$ and $v(x)$, that is, a nonunique polynomial $d^*(x)$ of the maximum degree d_ϵ that divides two polynomials $u^*(x)$ and $v^*(x)$ satisfying

$$\deg u^*(x) \leq m, \quad \deg v^*(x) \leq n, \tag{5.1}$$

$$\| u^*(x) - u(x) \| \leq \epsilon \| u(x) \|, \quad \| v^*(x) - v(x) \| \leq \epsilon \| v(x) \| . \tag{5.2}$$

The study of this problem is motivated by its applications to control linear systems, network theory and computer aided design (see some bibliography in [CGTW95], [EGL96], [CCC98], [P98], [P,a]).

Effective algorithms are known that estimate the weighted least-squares distances from $u(x)$ and $v(x)$ to two polynomials $u^*(x)$ and $v^*(x)$ satisfying (5.1) and divisible by a given polynomial $d^*(x)$ [CGTW95], [P,a]. The computation of such a candidate polynomial $d^*(x)$ is a harder problem, however. Only exponential time algorithms are known for its solution [Hou77], [KL96], [CCC98]; they are based on using the (weighted) least-squares norm and quadratic programming approach. There are several alternative techniques, which define the polynomial neighborhoods (5.2) based on the coefficient vector norms. These techniques give us heuristic solutions of Problem 5.1, that is, produce a common ϵ-divisor $d_\epsilon(x)$ of $u(x)$ and $v(x)$ and, furthermore, for some input pairs $u(x)$ and $v(x)$ verify that the degree of such a divisor is maximum but for other input pairs leave the latter question open. These techniques include Euclidean algorithm [Sc85], [NS91], [HS96], [EGL96], [EGL97], [SaSa97], computations with subresultant matrices [CGTW95], [EGL96], [EGL97] and Lazard's algorithm [L81], having an equivalent matrix pencil version [KM94].

We propose two alternative approaches, based on *Padé approximation* and *polynomial rootfinding* (cf. [P98], [P,a]). The former one is similar to the subresultant approach. It leads to a little simpler computations (involving symmetric Hankel or Bezout matrices of smaller sizes, versus unsymmetric subresultant matrices) but to a little less efficient techniques for the insurance of the ϵ-divisiblity. This approach relies on the observation that the polynomial $g(x) = \gcd(u(x), v(x))$ satisfies the equations

$$g(x) = \frac{u(x)}{w(x)} = \frac{v(x)}{z(x)} \tag{5.3}$$

where $(z(x),\ w(x))$ is the (m, n) Padé approximation of the power series $h(x) = \sum_{i=0}^{\infty} h_i x^i = \frac{u(x)}{v(x)}$. Here, with no loss of generality, we assume that $v(0) \neq 0$.

The computation of the gcd by this approach consists of the three successive stages of the computation of the following polynomials:

1. $h_N(x) = h(x) \bmod x^{N+1}, N = m + n$,
2. $z(x)$ and $w(x)$, being the (m, n)-Padé approximation of $h_N(x)$,
3. $g(x)$ of (5.3).

For the computation of an ϵ-gcd, stages 2 and 3 are modified as follows [P98], [P,a]. At first, the pairs $(z_d(x), w_d(x))$ of polynomials being the $(m - d, n - d)$ Padé approximations of $h_N(x)$ are computed for d recursively decreasing from a given upper bound d^+ on the degree of ϵ-gcds (say, $d^+ = \min\{m, n\}$) down to an unknown nonnegative integer d^-; each computed polynomial $w_d(x)$ is tested for being an ϵ-divisor of $u(x)$; the process continues recursively for $d = d^+,\ d^+ - 1, \ldots,$ until we either arrive at $d = 0$ (in which case the value $d^- = 0$ and the constant polynomial $g^-(x) = 1$ are output) or satisfy the inequalities $||u(x) - w_d(x)g^-(x)|| \leq \epsilon||u(x)||$, $||v(x) - z(x)g^-(x)|| \leq \epsilon||v(x)||$ for some polynomials $g^-(x)$ of degree at most d and $z(x)$ of degree at most $n - d$ (in which case $d^- = d$ and $g^-(x)$ are output).

Such an algorithm outputs a common ϵ-divisor of $u(x)$, $v(x)$ having degree d^-. If other methods show that d^- is actually an upper (and consequently the tight) bound on the degree of ϵ-gcd, then the computed ϵ-divisor g^- is a certified ϵ-gcd of $u(x)$ and $v(x)$.

The arithmetic cost of performing the above computations is bounded by $O_A((d^+ - d^- + 1)n \log n)$, for a generic input pair $(u(x),\ v(x))$, and by $O_A((d^+ - d^- + 1)n \log^2 n)$, for any pair $(u(x),\ v(x))$, provided that the Padé approximations are computed by means of the extended Euclidean algorithm [BGY80], [BP94]. An alternative way is to obtain them via the computations with the associated structured (Hankel or Bezout) matrices, that is, via the computation of the rank r of such an $n \times n$ matrix H_n or B_n followed by solving a nonsingular linear system of r equations with a structured coefficient matrix [BGY80], [BP94]. In the context of ϵ-gcd computation, one computes the rank numerically and obtains an upper bound d^+ on the degree of ϵ-gcds equal to the number of the eigenvalues of H_n and B_n lying in the interval $(-\sigma(\epsilon), \sigma(\epsilon))$ for a positive function $\sigma(\epsilon)$. This way of computation is slower but allows numerical stabilization and defines an upper bound d^+ on the degree of an ϵ-gcd.

In our second approach, we rely on the observation that the behavior of the gcd of the perturbed pairs of polynomials is directly governed by the perturbation of their zeros and only indirectly by the perturbation of their coefficients. Thus, in this approach, we first recall Theorem 3.1 to compute some higher precision approximations, y_i^* and z_j^*, to all the zeros y_i of $u(x)$ and z_j of $v(x)$, $i = 1, \ldots, m$; $j = 1, \ldots, n$. Then we define a bipartite graph with the two sets of vertices, $Y = \{y_i^*\}$ and $Z = (z_j^*)$, and with the edges connecting $y_{i(k)}^*$ and $z_{j(k)}^*$ if and only if $|y_{i(k)}^* - z_{j(k)}^*| < \delta(\epsilon)$ for a specified positive $\delta(\epsilon)$. We compute the maximum cardinality matching $M = \{(y_{i(k)}^*, z_{j(k)}^*\}$ in the graph and the candidate ϵ-gcd, $g(x) = \prod_k (x - (y_{i(k)}^* + z_{j(k)}^*)/2)$, where the product is over all k such that $(y_{i(k)}^*, z_{j(k)}^*) \in M$. Finally, we test $g(x)$ for being a common ϵ-divisor of $u(x)$ and $v(x)$ and compare its degree with the known upper bound d^+, set equal to $\min\{m, n\}$ or obtained by different approaches.

The computational cost of our second approach is quite low; it is defined by Theorem 3.1 and by the well known bound of $O(N^{2.5})$ comparisons, $N = m + n$, on the cost of the matching computation. We also show that the latter bound can be decreased to $O(mn)$ comparisons (by replacing the matching stage by the computation of the connected components in a bipartite graph) if *elastic choice* of ϵ is allowed (cf. [P98], [P,a]).

This approach leads to a natural extension of the definition of ϵ-gcd where the norm (of the perturbation) of a polynomial is defined in terms of the absolutely maximum (perturbation of) its zeros rather than its coefficients. In this case our approach always outputs correctly an ϵ-gcd. Furthermore, the resulting solution of such an ϵ-gcd problem can be extended to a heuristic algorithm for the problem of computing a Hankel matrix of a lower rank r^- approximating a given Hankel matrix of rank r, with $r^- < r$, this problem has important applications to signal and image processing [Cad88], [CaW90], [DeM94], [vdV96].

6 Computations with dense structured matrices, correlations to polynomial and rational interpolation and multipoint evaluation

We consider Toeplitz, Hankel, Vandermonde and Cauchy matrices as well as the matrices having similar structures. These are the most popular classes of dense structured matrices, omnipresent in scientific, engineering and electrical engineering computations. For readers convenience, we recall the definitions (cf. [BP94], [PCZ,a]).

Definition 6.1 *A matrix $T = (t_{i,j})$ is called a* Toeplitz matrix *if*

$$t_{i+1,j+1} = t_{i,j}$$

for all pairs (i, j) for which $t_{i,j}$ and $t_{i+1,j+1}$ are defined.

Toeplitz matrices are a special class (where $r \leq 2$) of the following important and well-studied class of Toeplitz-like matrices having a small displacement rank r (cf. [BP94], pp. 174-211).

Definition 6.2 *(cf. e.g. [BP94], Definition 11.1). For an $n \times n$ matrix T, define the two displacement generators,*

$$F_-(T) = T - Z^T T Z, \quad F_+(T) = T - Z T Z^T, \tag{6.1}$$

where

$$Z = \begin{pmatrix} 0 & & & & 0 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & & \\ 0 & & & 1 & 0 \end{pmatrix} \tag{6.2}$$

is a down-shift $n \times n$ matrix. If for $F = F_+$ or $F = F_-$ we have

$$F(T) = GH^T, \tag{6.3}$$

*where G and H are $n \times r$ matrices, then the pair of matrices (G, H) is called an F-*generator *or a* displacement generator *of T of length r and will be denoted $d.g._r(T)$. The minimum r allowing the above representation (6.3) is called the F-*rank *or the* displacement rank *of T. T is called a* Toeplitz-like matrix *if $r = O(1)$ as $n \to \infty$.*

Next, we will recall a basic property of Toeplitz-like matrices, which will enable us to manipulate with them by means of operating with a few entries of their short displacement generators, rather than with their own more numerous entries.

Theorem 6.1 *[KKM79]. Let F_-, F_+, T, G, H, and r be as in (6.1) and (6.3). Then $F(T) = GH^T = \sum_{k=1}^{r} \mathbf{g}^{(k)} \mathbf{h}^{(k)T}$ if and only if*

$$T = \sum_{k=1}^{r} L^T(\mathbf{g}^{(k)}) L(\mathbf{h}^{(k)}) \ for\ F = F_-, \ T = \sum_{k=1}^{r} L(\mathbf{g}^{(k)}) L^T(\mathbf{h}^{(k)}) \ for\ F = F_+,$$

where $G = (\mathbf{g}^{(1)}, \ldots, \mathbf{g}^{(r)})$, $H = (\mathbf{h}^{(1)}, \ldots, \mathbf{h}^{(r)})$, and $L(\mathbf{v})$ is a lower triangular Toeplitz matrix with the first column $\mathbf{v}$.

Hankel matrices and the *Hankel-like matrices* of displacement rank r are obtained from Toeplitz matrices and Toeplitz-like matrices of displacement rank r, respectively, by their pre-multiplication (or post-multiplication) by the *reflection matrix* J having ones on its antidiagonal and zero entries elsewhere. (Note that J^2 is the identity matrix.)

Definition 6.3 *Let $x_0, \ldots, x_{n-1}$ denote n distinct values and let $s_0, t_0, \ldots, s_{n-1}, t_{n-1}$ denote $2n$ distinct values. Let $\mathbf{x} = (x_i)_{i=0}^{n-1}$, $\mathbf{s} = (s_i)_{i=0}^{n-1}$, $\mathbf{t} = (t_j)_{j=0}^{n-1}$. Then the $n \times n$ matrices $V = V(\mathbf{x}) = (x_i^j)_{i,j=0}^{n-1}$ and $C = C(\mathbf{s}, \mathbf{t}) = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$ are called* Vandermonde *and* Cauchy matrices*, respectively. Furthermore, the $n \times n$ matrices $V_f(\mathbf{x}, G, H)$ and $C(\mathbf{s}, \mathbf{t}, G, H)$ are said to be given with their $(f, \mathbf{x})$-*scaling/displacement generator (G, H) *and* $(\mathbf{s}, \mathbf{t})$-scaling generator (G, H), *both* of length r, *respectively, if*

$$V_f(\mathbf{x}, G, H) = \sum_{k=1}^{r} \text{diag } (f g_i^{(k)}/(f - x_i^n))_{i=0}^{n-1} V(\mathbf{x}) Z_{1/f}^T(\mathbf{h}^{(k)}), \tag{6.4}$$

$$C(\mathbf{s}, \mathbf{t}, G, H) = \sum_{k=1}^{r} \text{diag } (s_i g_i^{(k)})_{i=0}^{n-1} C(\mathbf{s}, \mathbf{t}) \text{ diag } (h_j^{(k)})_{j=0}^{n-1} \tag{6.5}$$

where

$$\mathbf{g}^{(k)} = (g_i^{(k)})_{i=0}^{n-1}, \mathbf{h}^{(k)} = (h_j^{(k)})_{j=0}^{n-1}, k = 0, \ldots, n-1, G = (\mathbf{g}^{(k)})_{k=0}^{n-1}, H = (\mathbf{h}^{(k)})_{k=0}^{n-1}, \tag{6.6}$$

$\text{diag}(\mathbf{w}_i)_{i=0}^{n-1}$ *denotes the diagonal matrix with the diagonal entries $w_0, \ldots, w_{n-1}$; $Z_q(\mathbf{y}) = (y_{i,j})_{i,j=0}^{n-1}$ is the q-*circulant matrix, $y_{i,j} = y_{i-j \bmod n}$ *for $i \geq j$, $y_{i,j} = q y_{i-j \bmod n}$ for $i < j$, $\mathbf{y} = (y_k)_{k=0}^{n-1}$, $q = 1/f$, and f is a scalar, $f \neq 0$, $f \neq x_i^n$ for $i = 1, \ldots, n-1$, so that $Z_0(y) = Z$ for the matrix Z of (6.2) and $\mathbf{y} = (0, 1, 0, \ldots, 0)^T$. 1-circulant matrices are called* circulant. *The minimum length r in the above representations of the matrices $V_f(\mathbf{x}, G, H)$ and $C(\mathbf{s}, \mathbf{t}, G, H)$ (for all pairs $(f, \mathbf{x})$ and $(\mathbf{s}, \mathbf{t})$, respectively) is called the* scaling/displacement rank *and the* scaling rank *of these matrices, respectively. If $r = O(1)$ as $n \to \infty$, then the above matrices are called* Vandermonde-like *and* Cauchy-like matrices*, respectively.*

Note that Vandermonde and Cauchy matrices have scaling/displacement and scaling ranks 1, respectively.

We also recall the next well-known results [BP94].

Theorem 6.2 *The cost of multplication of an $n \times n$ matrix by a vector is bounded by $O_A(n \log n)$ for Toeplitz and Hankel matrices and by $O_A(n \log^2 n)$ for Vandermonde and Cauchy matrices.*

Dut to Theorem 6.1 and equations (6.1)–(6.3), Theorem 6.2 is immediately extended to Toeplitz-like, Hankel-like, Vandermonde-like and Cauchy-like matrices.

As this was shown in [P89/90], some computations with dense structured matrices of the above four classes can be conveniently reduced to each other, so that any successful algorithm for some fundamental computations for matrices of one class can be effectively extended to the matrices of the other three classes. In [GKO95], [Hei95] such a reduction from Toeplitz and Toeplitz-like matrices to Cauchy-like matrices was simplified and exploited to yield an effective practical algorithm for fast and numerically stable solution of Toeplitz and Toeplitz-like linear systems of n equations by Gaussian elimination with pivoting (at the cost $O_A(n^2)$)), and surely, this is immediately extended to the Hankel and Hankel-like linear systems. The main point of this approach is that the row and/or column interchange destroys the structure of Toeplitz-like and Hankel-like matrices (which either blocks the application of pivoting in Gaussion elimination or slows down the computation) but preserves the Cauchy-like matrix structure. The latter property of Cauchy-like matrices can be observed either from matrix equation (6.5) or from the following equivalent representation,

$$C(\mathbf{s},\mathbf{t},G,H)=(\frac{\sum_{k=1}^{r} g_i^{(k)} h_j^{(k)}}{s_i-t_j})_{i,j=0}^{n-1}.$$

Another known approach to solving Toeplitz-like and Hankel-like linear systems of equations as well as to the computation of short displacement generators of the inverses of $n \times n$ Toeplitz, Hankel, Toeplitz-like and Hankel-like matrices, in all cases at the cost $O_A(n \log^2 n)$, is by the divide-and-conquer MBA algorithm of [M80], [BA80].

The algorithm relies on the following well-known factorization of 2×2 block matrices,

$$A=\begin{pmatrix} I & O \\ EB^{-1} & I \end{pmatrix}\begin{pmatrix} B & O \\ O & S \end{pmatrix}\begin{pmatrix} I & B^{-1}C \\ O & I \end{pmatrix}, \tag{6.7}$$

$$A^{-1}=\begin{pmatrix} I & -B^{-1}C \\ O & I \end{pmatrix}\begin{pmatrix} B^{-1} & O \\ O & S^{-1} \end{pmatrix}\begin{pmatrix} I & O \\ -EB^{-1} & I \end{pmatrix}, \tag{6.8}$$

where I and O denote the identity and null matrices of appropriate sizes,

$$A=\begin{pmatrix} B & C \\ E & J \end{pmatrix}, \qquad S=J-EB^{-1}C, \tag{6.9}$$

B is a $k \times k$ matrix, and $S=S(B,A)$ is called the *Schur complement of B in A*. (6.7) represents block Gauss-Jordan elimination applied to the 2×2 block matrix A of (6.9). If the matrix A is *strongly nonsingular* (that is, has nonsingular $k \times k$ leading principal submatrices for all k), then the matrix S of (6.9) can be obtained in $n-k$ steps of Gauss-Jordan elimination.

Lemma 6.1 *(cf. [BP94], Exercise 4, p. 212). If A is strongly nonsingular, so are B and S.*

For a strongly nonsingular matrix A, the factorization (6.7)-(6.9) can be extended to the matrices B and S and then recursively to the matrices of smaller size until we arrive at 1×1 matrices. The important point is to preserve the structure of the input matrix A throughout the recursive process, which enables the application of Theorem 6.2 and respective decrease of the computational time to the cited level $O_A(n \log^2 n)$. The algorithm of [M80], [BA80] achieved this goal for Toeplitz-like matrices.

By using the reduction of [P89/90] from Cauchy-like to Toeplitz-like matrices, the latter cost estimate can be extended to the Cauchy-like case. Such a reduction, however, is more involved than the reduction into the opposite direction (from Toeplitz-like to Cauchy-like matrices), and its numerical properties are not clear. In [PZ96], [PTCPS98] and [OP98], the divide-and-conquer algorithm based on(6.7)-(6.9) was devised directly for the Cauchy-like inversion and the solution of Cauchy-like linear systems of equations, at the cost $O_A(n \log^3 n)$ (cf. Theorem 6.2). In [OP98], the algorithm was also applied to some major problems of rational interpolation, which are the basis for some fundamental signal processing computations and which can be expressed as the recursive factorization problem based on (6.7)–(6.9). The main technicality of the extension of the MBA algorithm is the extension to the Cauchy-like case of the results of [M80], [BA80] on maintaining the Toeplitz-like structure throughout the recursive factorization process.

If the inverses of the dense structured matrices of the above classes are computed numerically, with some bounded output errors due to roundoff, then the output approximations can be very rapidly refined by Newton's iteration. Straightforward application of such an iteration causes rapid growth of the length of the displacement or scaling generators and a slowdown of the computations as a results, but some nontrivial modifications avoid this problem [P92a], [P93], [PZHD97], [PBRZ,a].

The basic block of all these algorithms is the multiplication of a dense structured matrix of a given class by a vector (cf. Theorem 6.2 on the complexity of such a multiplication). In the Cauchy-like case, we have the reduction to the problem of multiplication of a Cauchy matrix by a vector, also known as *Trummer's* celebrated *problem*, having various applications to scientific and engineering computing (see bibliography in [PTCLS98]). By Theorem 6.2, its solution cost is bounded by $O_A(n \log^2 n)$, but the algorithm of [Ger87] supporting this bound is numerically unstable. For numerical solution of Trummer's problem, the methods of practical choice are the multipole algorithms (cf. [Rok85], [Ger87], [AGR88], [GR87], [War98]), which are numerically stable and compute an approximate solution of Trummer's problem at the cost $O_A(bn)$ where 2^{-b} bounds the output error norm.

The cited transitions among various classes of structured matrices have some interesting applications to polynomial computations. In particular, recall that polynomial interpolation and multipoint evaluation can be expressed as the computation of the vector $\mathbf{p}$ for a given pair of vectors $\mathbf{x}$ and $\mathbf{v}$ and of the vector $\mathbf{v}$ for a given pair of vectors $\mathbf{x}$ and $\mathbf{p}$, respectively, where the three vectors $\mathbf{x}$, $\mathbf{p}$ and $\mathbf{v}$ of the same dimension n satisfy the vector equation

$$V(\mathbf{x})\mathbf{p} = \mathbf{v}, \tag{6.10}$$

and $V(\mathbf{x})$ is defined in Definition 6.2. Here we assume a polynomial $p(x)$ of degree at most $n-1$ with the coefficient vector $\mathbf{p}$, whereas $\mathbf{x}$ is the vector of the n nodes of evaluation or interpolation, and $\mathbf{v}$ is the vector of the values of $p(x)$ at these nodes.

The known fast algorithms yield the cost bound $O_A(n \log^2 n)$ for both problems but are numerically unstable. The similarity with Trummer's problem can be continued from this point; furthermore, we reduce polynomial interpolation and multipoint evaluation to Trummer's problem based on the next equation (such a reduction was first proposed in [PLST93], [PZHY97]; cf. also [PTCLS98], [PTCPS98]):

$$V(\mathbf{x}) = -\frac{1}{\sqrt{n}} \operatorname{diag}(1 - x_i^n)_{i=0}^{n-1} \; C(\mathbf{x}, \mathbf{w}) \; \operatorname{diag}(w_i)_{i=0}^{n-1} \; F. \tag{6.11}$$

Here $\mathbf{w} = (w^i)_{i=0}^{n-1}$, $F = \frac{1}{\sqrt{n}}(w^{jk})_{j,k=0}^{n-1}$, is the matrix of discrete Fourier transform, $w = \exp(2\pi\sqrt{-1}/n)$ being a primitive n-th root of 1. At a cost $O_A(n \log n)$, equations (6.10) and (6.11) combined reduce the multipoint polynomial evaluation problem to Trummer's problem, where one may apply effective multipole algorithms. The resulting approximation algorithms for polynomial evaluation are fast and numerically stable, as this was confirmed by the results of extensive numerical experiments reported in [PZHY97]. To yield a similar reduction of the polynomial interpolation problem, one may invert equation (6.11) and then use the known expressions for the matrices $C^{-1}(\mathbf{x}, \mathbf{w})$ and $V^{-1}(\mathbf{x})$ (see [BP94], p. 131; [PTCLS98], [PTCPS98]). The constructions proposed in the papers [PTCLS98] and [PTCPS98] also involve and exploit some matrix equations extending (6.11), which enable the reverse reduction, that is, from Trummer's problem to the evaluation and interpolation, with the objective to improve the known algorithms for Trummer's problem.

References

[AGR88] J. Ambrosiano, L. Greengard, V. Rokhlin, The Fast Multipole Method for Gridless Particle Simulation, *Computer Physics Communications*, 48, 115–125, 1988.

[AHU74] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts, 1974 (first edition), 1976 (second edition).

[BA80] R. R. Bitmead, B. D. O. Anderson, Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations, *Linear Algebra Appl.,* 34, 103-116, 1980.

[BEPP97] H. Brönnimann, E. Z. Emiris, V. Y. Pan, S. Pion, Computing Exact Geometric Predicates Using Modular Arithmetic with Single Precision, *Proc.*

13th Ann. ACM Symp. on Computational Geometry, 174-182, ACM Press, New York, 1997.

[BEPP99] H. Brönnimann, E. Z. Emiris, V. Y. Pan, S. Pion, Sign Determination in Residue Number Systems, *Theoretical Computer Science,* 210, 1, 173-197, 1999.

[BGY80] R. P. Brent, F. G. Gustavson, D. Y. Y. Yun, Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximations, *J. of Algorithms*, 1, 259-295, 1980.

[BMP98] D. Bondifalat, B. Mourrain, V. Y. Pan, Controlled Iterative Methods for Solving Polynomial Systems, *Proc. Annual ACM-SIGSAM Intern. Symp. on Symb. and Algebr. Comp.* (*ISSAC'98*), 252-259, ACM Press, New York, 1998.

[BP94] D. Bini, V. Y. Pan, *Polynomial and Matrix Computations, vol. 1: Fundamental Algorithms*, Birkhäuser, Boston, Massachusetts, 1994.

[BSS89] L. Blum, M. Shub, S. Smale, On a Theory of Computations Over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines, *Bull. of Amer. Math. Soc.*, 21, 1-46, 1989.

[Cad88] J. A. Cadzow, Signal Enhancement: A Composite Property Mapping Algorithm. *IEEE Trans. on Acoust., Speech, Signal Processing*, ASSP-36, 39-62, 1988.

[CaW90] J. A. Cadzow, D. M. Wilkes. Signal Enhancement and the SVD, *Proceedings of the 2nd International Workshop on SVD and Signal Processing,* University of Rhode Island, Kingston, RI, 144-151, 1990.

[CCC98] P. Chin, R. M. Corless, G. F. Corliss, Optimization Strategies for the Approximate Gcd Problem, *Proc. Intern. Symp. on Symb. and Algebraic Comp.* (*ISSAC '98*), 228-235, ACM Press, New York, 1998.

[CGR87] J. Carier, L. Greengard, V. Rokhlin, A Fast Adaptive Multipole Algorithm for Particle Simulation, *SIAM J. Sci. Stat. Comput.,* 9, 4, 669-686, 1988.

[CGTW95] R. M. Corless, P. M. Gianni, B. M. Trager, S. M. Watt, The Singular Value Decomposition for Polynomial Systems, *Proc. Intern. Symp. on Symbolic and Algebraic Comp. (ISSAC '95)*, 195-207, ACM Press, New York, 1995.

[DeM94] B. De Moor, Total Least Squares for Affinely Structured Matrices and the Noisy Realization Problem, *IEEE Trans. Signal Processing*, 42, 3104-3113, 1994.

[EGL96] I. Z. Emiris, A. Galligo, H. Lombardi, Numerical Univariate Polynomial GCD, *AMS-SIAM Summer Seminar in Applied Math.: Mathematics of Numerical Analysis,* Park City, Utah, 1995, (J. Renegar, M. Shub, S. Smale editors), *Lectures in Applied Math.,* 32, 323–344, Amer. Math. Soc., Providence, R.I., 1996.

[EGL97] I. Z. Emiris, A. Galligo, H. Lombardi, Certified Approximate Polynomial GCDs, *J. of Pure and Applied Algebra*, 117/118, 229-251, 1997.

[EP97] I. Z. Emiris, V. Y. Pan, The Structure of Sparse Resultant Matrices, *Proc. of Annual ACM-SIGSAM International Symp. on Symb. and Alg. Comp. (ISSAC'97)*, 189-196, ACM Press, New York, 1997.

[Gat86] J. von zur Gathen, Parallel Arithmetic Computations: A Survey, *Proc. Math. Foundation of Computer Science, Lecture Notes in Computer Science*, 233, 93-112, Springer, Berlin, 1986.

[Ger87] A. Gerasoulis, A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors, *Math. of Comp.*, 50, 181, 179–188, 1987.

[GO94] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra and Its Applications*, 203, 163-192, 1994.

[GKO95] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gausian Elimination with Partial Pivoting for Matrices with Displacement Structure, *Math. of Computation*, 64, 1557-1576, 1995.

[GR87] L. Greengard, V. Rokhlin, A Fast Algorithm for Particle Simulation, *J. Comput. Physics*, 73, 2, 325–348, 1987.

[He74] P. Henrici, *Applied and Computational Complex Analysis*, 1, Wiley, New York, 1974.

[Hei95] G. Heinig, Inversion of Generalized Cauchy Matrices and the Other Classes of Structured Matrices, *Linear Algebra for Signal Processing, IMA Volume in Math. and Its Applications,* 69, 95-114, Springer, 1994.

[Hou77] D. G. Hough, Explaining and Ameliorating the Ill Condition of Zeros of Polynomials, Ph.D. Thesis, *Computer Science Dept., Univ. of California*, Berkeley, 1977.

[HS96] V. Hribernig, H.J. Stetter, Detection and Validation of Clusters of Polynomial Zeros, *J. Symb. Comp.*, 1996.

[KKM79] T. Kailath, S.Y. Kung, M. Morf, Displacement Ranks of Matrices and Linear Equations, *J. Math. Anal. Appl.*, 68, 2, 395-407, 1979.

[KL96] N. Karmarkar, Y. N. Lakshman, Approximate Polynomial Greatest Common Divisor and Nearest Singular Polynomials, *Proc. Ann. ACM-SIGSAM Intern. Symp. on Symb. and Algebraic Comp. (ISSAC'96),* 35–39, 1996.

[KM94] N. Karcanias, M. Mitrouli, A Matrix Pencil Based Numerical Method for the Computation of GCD of Polynomials, *IEEE Trans. on Automatic Control*, 39, 5, 977-981, 1994.

[KR90] R. Karp, V. Ramachandran, A Survey of Parallel Algorithms for Shared Memory Machines, *Handbook for Theoretical Computer Science* (J. van Leeuwen editor), 869–941, North Holland, Amsterdam, 1990.

[L81] D. Lazard, Resolution des Systems d'Equations Algebrique, *Theoretical Computer Science,* 15, 77–110, 1981.

[Lei92] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees & Hypercubes*, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[M80] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proc. IEEE Internat. Conf. on ASSP*, 954-959, IEEE Computer Society Press, 1980.

[MP98] B. Mourrain, V. Y. Pan, Asymptotic Acceleration of Solving Multivariate Polynomial Systems of Equations, *Proc. 30th Ann. ACM Symp. on Theory of Computing,* 488-496, ACM Press, New York, 1998.

[MP98a] B. Mourrain, V. Y. Pan, Multivariate Polynomials, Duality and Structured Matrices, Research Report No. 3513, *INRIA*, Sophia Antipolis, France, 1998, and MSRI preprint #1998-072, *Mathematical Science Research Institute*, Berkeley, California, 1998.

[MP,a] B. Mourrain, V. Y. Pan, to appear.

[NS91] M.-T. Noda, T. Sasaki, Approximate GCD and Its Application to Ill-Conditioned Algebraic Equations, *J. Computational and Applied Mathematics*, 38, 335-351, 1991.

[OP98] V. Olshevsky, V. Y. Pan, A Unified Superfast Algorithm for Boundary Rational Tangential Interpolation Problem and for Inversion and Factorization of Dense Structured Matrices, *Proc. 39th Annual IEEE Symp. on Foundation of Computer Science,* 192-201, IEEE Computer Society Press, 1998.

[P89/90] V. Y. Pan, Computations with Dense Structured Matrices, *Proc. ACM-SIGSAM Intern. Symp. on Symbolic and Alg. Comp. (ISSAC '89)*, 34-42, ACM Press, New York, 1989, and *Math. of Comp.,* 55, 191, 179-190, 1990.

[P92] V. Y. Pan, Complexity of Computations with Matrices and Polynomials, *SIAM Review,* 34, 9, 225-262, 1992.

[P92a] V. Y. Pan, Parallel Solution of Toeplitz-like Linear Systems, *J. of Complexity*, 8, 1–21, 1992.

[P93] V. Y. Pan, Concurrent Iterative Algorithm for Toeplitz-like Linear Systems, *IEEE Trans. on Parallel and Distributive Systems*, 4, 5, 592–600, 1993.

[P95] V. Y. Pan, Optimal (up to Polylog Factors) Sequential and Parallel Algorithms for Approximating Complex Polynomial Zeros, *Proc. 27th Ann. ACM Symposium on Theory of Computing*, 741-750, ACM Press, New York, 1995.

[P96] V. Y. Pan, Optimal and Nearly Optimal Algorithms for Approximating Polynomial Zeros, *Computers and Mathematics (with Applications)*, 31, 12, 97-138, 1996.

[P97] V. Y. Pan, Solving a Polynomial Equation: Some History and Recent Progress, *SIAM Review*, 39, 2, 187-220, 1997.

[P98] V. Y. Pan, Approximate Polynomial Gcds, Padé Approximation, Polynomial Zeros, and Bipartite Graphs, *Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms,* 68-77, ACM Press, New York, and SIAM Publications, Philadelphia, 1998.

[P98a] V. Y. Pan, Solving Polynomials with Computers, *American Scientist*, **86**, 62-69, January-February 1998.

[P,a] V. Y. Pan, Computation of Approximate Polynomial Gcds and an Extension, to appear in *Information and Computation.*

[PBRZ,a] V. Y. Pan, S. Branham, R. Rosholt, A. Zheng, Newton's Iteration for Structured Matrices and Linear Systems of Equations, accepted for the SIAM volume: *Fast Reliable Algorithms for Matrices with Structure* (T. Kailath and A. Sayed editors), SIAM Publications, Philadelphia, to appear.

[PCZ,a] V. Y. Pan, Z. Chen, A. Zheng, *The Complexity of the Algebraic Eigenproblem*, MSRI Preprint #1998-071, *Mathematical Science Research Institute*, Berkeley, California, 1998.

[PLST93] V. Y. Pan, A. Sadikou, E. Landowne, O. Tiga, A New Approach to Fast Polynomial Interpolation and Multipoint Evaluation, *Computers & Math. (with Applications)*, 25, 9, 25-30, 1993.

[PTCPS98] V. Y. Pan, M. A. Tabanjeh, Z. Chen, S. Providence, A. Sadikou, Transformations of Cauchy Matrices for Trummer's Problem and a Cauchy-like Linear Solver, *Proc, 5th Ann. International Symp. on Solving Irregularly Structured Problems in Parallel (Irregular '98), Lecture Notes in Computer Science*, 1457, 274-284, Springer, 1998.

[PTCLS98] V. Y. Pan, M. A. Tabanjeh, Z. Chen, E. Landowne, A. Sadikou, New Transformations of Cauchy Matrices and Trummer's Problem, *Computers & Mathematics (with Applications),* 35, 12, 1-5, 1998.

[PY99] V. Y. Pan, Y. Yu, Certified Numerical Computation of the Sign of a Matrix Determinant, *Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms,* ACM Press, New York, and SIAM Publications, Philadelphia, 1999.

[PZ96] V. Y. Pan, A. Zheng, Fast Algorithms for Cauchy-like Matrix Computations and an Extension to Singular Toeplitz-like Computations, Preprint, 1996.

[PZHD97] V. Y. Pan, A. Zheng, X. Huang, O. Dias, Newton's Iteration for Inversion of Cauchy-like and Other Structured Matrices, *J. of Complexity*, 13, 108-124, 1997.

[PZHY97] V. Y. Pan, A. Zheng, X. Huang, Y. Yu, Fast Multipoint Polynomial Evaluation and Interpolation via Computations with Structured Matrices, *Annals of Numerical Math.,* 4, 483-510, 1997.

[Qui94] M. J. Quinn, *Parallel Computing: Theory and Practice*, McGraw-Hill, New York, 1994.

[Ren89] J. Renegar, On the Worst-Case Arithmetic Complexity of Approximating Zeros of Systems of Polynomials, *SIAM J. Comput.,* 18, 350-370, 1989.

[Rok85] V. Rokhlin, Rapid Solution of Integral Equations of Classical Potential Theory, *J. Comput. Physics*, 60, 187–207, 1985.

[Sc85] A. Schönhage, Quasi-GCD Computations, *J. of Complexity*, 1, 118-137, 1985.

[SaSa97] T. Sasaki, M. Sasaki, Polynomial Remainder Sequence and Approximate GCD, *ACM SIGSAM Bulletin*, 31, 3, 4-10, ACM Press, New York, 1997.

[vdV96] A. van der Veen, A Schur Method for Low-Rank Matrix Approximation. *SIAM J. Matrix Anal. Appl.*, 17, 139-160, 1996.

[War98] A. F. Ware, Fast Approximate Fourier Transform for Irregularly Spaced Data, *SIAM Review*, 40, 4, 838-856, 1998.