2011

TR-2011003: Partial Fraction Decomposition, Sylvester Matrices, Convolution and Newton's Iteration

Victor Y. Pan

Partial Fraction Decomposition, Sylvester Matrices, Convolution and Newton's Iteration *

Victor Y. Pan

Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA
victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/

Abstract

Both Sylvester matrix and convolution are defined by two polynomials. If one of them has small degree, then the associated Sylvester linear system can be solved fast by using its PFD interpretation of the convolution equation. This can immediately simplify the refinement of approximate convolution by means of Newton's iteration, where we also incorporate the PFD refinement techniques or alternatively least-squares solution of a linear system associated with the convolution. The process is naturally extended to polynomial factorization and root-finding.

Key words: Sylvester matrices, PFD, Newton's iteration, convolution, polynomial factorization

1 Introduction

Factorization of a degree n polynomial

$$p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - z_j), \quad p_n \neq 0, \tag{1.1}$$

into the product of n linear factors $x - z_1, \ldots, x - z_n$ is an important subject in its own right (see Wilson (1969), Box and Jenkins (1976), Barnett (1983), Demeure and Mullis (1989 and 1990), Van Dooren (1994)), but it can be also extended to

*Supported by PSC CUNY Awards 61406–0039 and 62230–0040.

approximation of the n zeros or roots $z_1, \ldots, z_n$ or to their isolation from each other in the case where they are distinct and the polynomial $p(x)$ has integer coefficients. Existence of such complex roots and factors is the Fundamental Theorem of Algebra, and their computation or isolation is the fundamental problem of Computer Algebra. Bibliography on polynomial root-finding, root isolation and factorization includes thousands of papers (cf. McNamee (1993, 1997, 2002 and 2007)), dozens of them appeared in 2010.

The algorithms in Pan (1995, 1996, 2001a, and 2002) solve these three problems in nearly optimal arithmetic and Boolean time, up to polylogarithmic factors in the degree n and in the parameter b denoting either the precision of the output approximation in the cases of factorization and root-finding or the maximal length of the input coefficients in the case of root isolation. The algorithms first split the polynomial $p(x)$ into the product of two nonconstant factors L_1 and L_2, then recursively split each nonlinear factor in similar fashion, and stop when complete factorization in (1.1) is approximated within a fixed error tolerance. If the tolerance is small enough, the factorization can be immediately extended to root-finding and root isolation. This splitting scheme was proposed and explored in Schönhage (1982), where the same computational problems have been solved albeit slower by the factor n.

Kirrinnis (1998) extends this approach to refine approximate splitting into m factors for any m, $2 \leq m \leq n$. As by-product his algorithm refines the respective approximate partial fraction decomposition of the polynomial reciprocal $1/p(x)$. Hereafter we will use the acronym "*PFD*" for "partial fraction decomposition".

We observe that for the factors of degree one the Kirrinnis' algorithm is closely linked to Newton's classical iteration for approximating a root of a univariate equation. Generally, shifting to Newton's multivariate iteration tends to enhance the convergence power, e.g. in Durand–Kerner iteration (due to Weierstrass (1903)), Aberth or Ehrlich–Aberth iteration (due to Börsch-Supan (1963)), and matrix methods involving eigenvectors (see Pan and Zheng (2011) and the bibiliography therein), and so we try to employ Newton's multivariate iteration based on the convolution equation.

Its every iteration step essentially amounts to the solution of a Sylvester linear system whose matrix is defined by the current approximate factors of the input polynomial. At this point our study of the associaited PFD helps us to accelerate the solution of this system in the important case where one of the approximate factors has small degree. We also employ the associated PFD or alternatively the least squares solution of the convolution equation to compute or to refine one of the factors where another factor is fixed. Our study suggests that combining the convolution equation and its associated PFD can enhance the power of iterations for polynomial factorization and root-finding.

2 Computation of PFDs

To simplify the notation we will write u for a polynomial $u(x) = \sum_i u_i x^i$ wherever this causes no confusion.

Let $m,\ n,\ n_1,\dots,n_m$ denote positive integers such that $2 \le m \le n$ and $n_1+\ldots+n_m = n$. For a monic polynomial p of degree n and a polynomial $T = T(x)$ of degree at most $n-1$ and coprime with p we seek pairwise prime monic polynomials $L_1,\dots,L_m$ and polynomials $V_1,\dots,V_m$, $\deg\ V_i < \deg\ L_i = n_i$, $i = 1,\dots,m$, defining the factorization $p = L_1\cdots L_m$ and the PFD

$$\frac{T}{p} = \frac{V_1}{L_1} + \cdots + \frac{V_m}{L_m}. \tag{2.1}$$

Multiply this PFD by p and obtain the equivalent representation as the polynomial equation $Q_1V_1+\cdots+Q_mV_m = T$ where

$$Q_i = p/L_i,\ V_i = (TL_i/p) \mod L_i,\ i = 1,\dots,m. \tag{2.2}$$

By solving the PFD problem for $T = 1$ we obtain the polynomials $W_i = (L_i/p) \bmod L_i$ and then we can readily obtain $V_i = TW_i \mod L_i$ for all i.

Alternatively the coefficient vectors of the polynomials $V_1,\dots,V_m$ can be obtained from a linear system of equations

$$S(Q_1,\dots,Q_m)\mathbf{V} = \mathbf{T}. \tag{2.3}$$

Here $\mathbf{V}^T = (\mathbf{V}_1^T,\dots,\mathbf{V}_m^T)$, $\mathbf{V}_j$ denoting the coefficient vectors of the polynomials V_j for $j = 1,\dots,m$; $\mathbf{T}$ is the coefficient vector of T, so that $\mathbf{T} = \mathbf{e}_n = (0,\dots,0,1)^T$ is the nth coordinate vector of dimension n for $T = 1$, the coefficient matrix $S(Q_1,\dots,Q_m) = [C_{n-n_1}(Q_1) \mid \ldots \mid C_{n-n_m}(Q_m)]$ is the $1\times m$ block matrix with the blocks $C_{n-n_1}(Q_1),\dots,C_{n-n_m}(Q_m)$, and $C_k(w) = \begin{pmatrix} w_l & & & O \\ \vdots & \ddots & & \\ \vdots & \ddots & \ddots & \\ \vdots & \ddots & \ddots & w_l \\ w_0 & \vdots & \ddots & \vdots \\ \ddots & \ddots & \vdots & \\ & & \ddots & \vdots \\ O & & & w_0 \end{pmatrix}$ denotes the kth convolution matrix of a polynomial $w(x) = \sum_{i=0}^{l} w_i x^i$, that is the $(k+l)\times(k+1)$ Toeplitz matrix defined by its first column $(w_l, w_{l-1},\dots,w_0,0\dots,0)^T$ and its first row $(w_l,0,\dots,0)$. $S(Q_1,\dots,Q_m)$ is a generalized Sylvester matrix, becoming Sylvester matrix $S(Q_1,Q_2) = S(L_2,L_1)$ in the case where $m = 2$.

Equation (2.2) enables us to express the solution of a generalized Sylvester linear system (2.3) via its solution for $T = 1$. The extension from this special case takes $O((n\log n)\log m)$ arithmetic operations (see Problem 4.1 (POL·MODULI) in Bini and Pan (1994)). The cost bound decreases to $O(n)$ where $m = O(1)$ and the factors $L_1,\dots,L_{m-1}$ have degrees in $O(1)$. In particular in the case where $m = 2$ and $\deg L_1$ is in $O(1)$ we obtain an algorithm that supports the following theorem.

Theorem 2.1. *Suppose $S(L_2, L_1)$ denotes the $n \times n$ Sylvester matrix defined by two coprime polynomials L_1 and L_2 where $d_1 = \deg L_1$, $d_2 = \deg L_2 = n - d_1$, d_1 is in $O(1)$. Then a Sylvester linear system of n equations with this matrix can be solved by using $O(n)$ arithmetic operations.*

Remark 2.1. *Note that $V_j = T(z_j)/p'(z_j)$ where $L_j = x - z_j$ is a monic linear factor of p. In this case for $j = 1$ we obtain from equation (2.1) for $m = 2$ and $T = 1$ that*

$$\frac{V_2}{L_2} = \frac{1}{p} - \frac{1}{p'(z_j)(x - z_j)} \text{ and so } V_2 = \frac{1 - L_2/p'(z_j)}{x - z_j}. \tag{2.4}$$

3 Factorization via PFD and Newton's Iteration

Assume that the variable x has been scaled to bring the zero set of the polynomial p into the unit disc $D(0,1) = \{x : \ |x| \leq 1\}$ and let us be given sufficiently close approximations $l_i = l_i^{(0)}$ to L_i, $v_i = v_i^{(0)}$ to V_i in (2.1) for $i = 1, \ldots, m$ and $l^{(0)} = l_1^{(0)} \cdots l_m^{(0)}$ to p satisfying the PFD

$$\frac{1}{l^{(0)}} = \frac{v_1^{(0)}}{l_1^{(0)}} + \cdots + \frac{v_m^{(0)}}{l_m^{(0)}}. \tag{3.1}$$

Kirrinnis (1998) recursively improves the initial approximations by the polynomials $l_1^{(0)}, \ldots, l_m^{(0)}$ to $L_1, \ldots, L_m$ by preserving the equation. He sets

$$l_i^{(k+1)} = l_i^{(k)} + \Delta_i^{(k)}, \ i = 1, \ldots, m; \ k = 0, 1, \ldots \tag{3.2}$$

and computes the Newton's corrections $\Delta_1^{(k)}$, …, $\Delta_m^{(k)}$ from the PFD

$$\frac{p - l^{(k)}}{l^{(k)}} = \frac{\Delta_1^{(k)}}{l_1^{(k)}} + \cdots + \frac{\Delta_m^{(k)}}{l_m^{(k)}}, \ \deg \ \Delta_i^{(k)} < \deg \ l_i^{(k)}, \ i = 1, \ldots, m. \tag{3.3}$$

Alternatively one can first compute the PFD

$$\frac{1}{l^{(k)}} = \frac{v_1^{(k)}}{l_1^{(k)}} + \cdots + \frac{v_m^{(k)}}{l_m^{(k)}}, \ \deg v_i^{(k)} < \deg \ l_i^{(k)}, \ i = 1, \ldots, m$$

and then apply equation (2.2) to recover the correction values so that

$$\Delta_i^{(k)} = (v_i^{(k)} p) \bmod l_i^{(k)}, \ i = 1, \ldots, m. \tag{3.4}$$

In the case where $l_i^{(k)} = x - z_i^{(k)}$ is a monic linear factor, we arrive at the Newton's correction $\Delta_i^{(k)} = p(z_i^{(k)})/p'(z_i^{(k)})$, so that $l_i^{(k+1)} = l_i^{(k)} + p(z_i^{(k)})/p'(z_i^{(k)})$ $z_i^{(k+1)} = z_i^{(k)} - p(z_i^{(k)})/p'(z_i^{(k)})$. This is Newton's classical iteration having local quadratic convergence. Kirrinnis (1998) generalizes it to splitting $p(x)$ into m nonlinear factors, extends to this case the classical results on local quadratic convergence of Newton's iteration, and specifies the Boolean (that is bitwise) operation complexity provided that the factors $L_1, \dots, L_m$ as well as their initial approximations $l_1^{(0)} \approx l_m^{(0)}$ have pairwise isolated zero sets. In the case of such factors he proposes to replace the above recipes for updating $l_i^{(k)}$ for $i = 1, \dots, m-1$ by the following expressions,

$$q_i^{(k)} = l^{(k)}/l_i^{(k)}, \;\; l_i^{(k+1)} = l_i^{(k)} + ((2 - v_i^{(k)} q_i^{(k)}) v_i^{(k)} p \bmod l_i^{(k)}), \; i = 1, \dots, m$$

with the goal of improving numerical stability of the computations.

4 Newton's Iteration for Convolution Equation

In view of its close link to the classical Newton's univariate root-finder, the PFD factorization method above preserves the benefits and limitations of this root-finder, but we expect to enhance the convergence power by applying Newton's multivariate iteration to refine the initial solution $l_1^{(0)} \approx L_1$ and $l_2^{(0)} \approx L_2$ to the convolution equation $p = L_1 L_2$.

The kth iteration step is essentially the solution of a Sylvester linear system with the Jacobian coefficient matrix $-S(l_2^{(k)}, l_1^{(k)})$ (see Zeng (2005), Bini and Boito (2010), Pan and Zheng (2011)). If $\deg L_i = \deg l_i^{(k)} = O(1)$ for $i = 1$ or $i = 2$, we can solve this linear system in $O(n)$ arithmetic operations by applying the algorithm that supports Theorem 2.1. We can yield further simplifications where $\deg L_i = \deg l_i^{(k)} = 1$ for $i = 1$ or $i = 2$ (see Remark 2.1).

The iteration rapidly refines a sufficiently close initial approximate factorization precomputed by another algorithm, but one can also try randomized heuristic initial approximations where a factor $l_i^{(k)}$ is defined by a single complex parameter. This is the case where we seek a zero of p or a pair of the complex conjugate zeros of a polynomial p with real coefficients.

E.g. random choices can be made near the origin, near the center of gravity $-p_{n-1}/(np_n)$ of the n zeros of p, on a large circle $\{x : \; |x| = R\}$ for $R \geq 2\max_{i>1}\{|p_{n-i}/p_n|\}$ (cf. Habbard, Schleicher, and Sutherland (2001)), or on the Bini's circles in Bini (1996) and Bini and Fiorentino (2000). One can try to apply the iteration successively or concurrently at a number of such initial points to increase the chances for fast convergence.

For each approximate zero $z_1^{(0)}$ one can immediately define the initial linear factor $l_1^{(0)} = x - z_1^{(0)}$ and then initialize the coefficient vector of the second factor $l_2^{(0)}$ either via the PFD $v_2^{(0)}/l_2^{(0)} = 1/l^{(0)} - p'(z_j)/(x - z_j)$ obtained by extending the PFD (2.4) or by setting this vector equal to least-squares solution of the overdetermined linear system $C_{n-1}(l_1^{(0)})\mathbf{l}_1^{(0)} = \mathbf{p}$ defined by the convolution equation $l_1^{(0)} l_2^{(0)} \approx p$ (cf. Corless et al. (1995)). Now one can refine this initial factorization by applying Newton's iteration and employing Theorem 2.1 and Remark 2.1.

5 Discussion

1. The convoltion equation $p = L_1 L_2$ defines two equivalent vector representations $C(L_1)\mathbf{L_2} = \mathbf{p}$ and $C(L_2)\mathbf{L_1} = \mathbf{p}$. Assuming a fixed approximation to L_1 (resp. L_2) one can approximate L_2 (resp. L_1) by computing the least squares solution of the former (resp. latter) vector equation. One can successively update approximations to both factors in this way, as a complementary or alternative approach to Newton's iteration for the convolution equation.

2. It is interesting to examine Newton's iteration for the convolution equation complemented with the equation $(x - z_j)V_2 + p'(z_j)L_2 = 1$. We obtain the latter equation by multiplying equation (2.4) by p. Would including the new equation enhance the convergence power of the iteration?

3. Seeking the zeros z_j of p that lie outside the unit disc $D(0, 1)$ one should seek the factors $x/z_j - 1$ of p or the factors $x - 1/z_j$ of the reverse polynomial $p_{\rm rev} = \sum_{i=0}^{n} p_i x^{n-i}$ (rather than the monic linear factors $x - z_j$ of p) to improve numerical stability of the computations (cf. Schönhage (1982)).

References

[1] Barnett, S. (1983), *Polynomial and Linear Control Systems*, Marcel Dekker, New York.

[2] Bini, D. (1996), Numerical Computation of Polynomial Zeros by Means of Aberth's Method, *Numerical Algorithms*, **13**, 179–200

[3] Bini, D. A. and Boito, P. (2010), A Fast Algorithm for Approximate Polynomial GCD Based on Structured Matrix Computations, *Operator Theory: Advances and Applications*, **199**, 155–173, Birkhäuser Verlag

[4] Bini, D. A. and Fiorentino, G. (2000), Design, Analysis, and Implementation of a Multiprecision Polynomial Rootfinder, *Numerical Algorithms*, **23**, 127–173

[5] Bini, D. and Pan, V. Y. (1994), *Polynomial and Matrix Computations, Vol. 1: Fundamental Algorithms*, Birkhäuser, Boston

[6] Börsch-Supan, W. (1963), A-posteriori Error Bounds for the Zeros of Polynomials, *Numerische Math.* **5**, 380–398

[7] Box, G. E. P. and Jenkins, G. M. (1976), *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA

[8] Corless, R. M., Gianni, P. M., Trager, B. M. and Watt, S. M. (1995), The Singular Value Decomposition for Polynomail Systems, *Proc. Intern. Symposium on Symbolic and Algebriac Computation (ISSAC'95)*, 195–207, ACM Press, New York

[9] Demeure, C. J. and Mullis, C. T. (1989), The Euclid algorithm and the fast computation of cross–covariance and autocovariance sequences, *IEEE Trans. Acoust., Speech, Signal Processing* **37**, 545–552

[10] Demeure, C. J. and Mullis, C. T. (1990), A Newton–Raphson method for moving-average spectral factorization using the Euclid algorithm, *IEEE Trans. Acoust., Speech, Signal Processing* **38**, 1697–1709

[11] J. Habbard, D. Schleicher, S. Sutherland (2001), How to Find All Roots of Complex Polynomials by Newton's Method, *Invent. Math.*, **146**, 1–33

[12] Kirrinnis, P. (1998), Polynomial factorization and partial fraction decomposition by simultaneous Newton's iteration, *J. of Complexity* **14**, 378–444

[13] McNamee, J. M. (1993), Bibliography on Roots of Polynomials, *J. Computational and Applied Mathematics*, **47**, 391–394

[14] McNamee, J. M. (1997), A Supplementary Bibliography on Roots of Polynomials, *J. Computational and Applied Mathematics* **78**, 1

[15] McNamee, J. M. (2002), A 2002 update of the supplementary bibliography on roots of polynomials,

[16] J. M. McNamee, *Numerical Methods for Roots of Polynomials*, Elsevier, 2007.

[17] Pan, V. Y. (1995), Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros, *Proc. 27th Ann. ACM Symp. on Theory of Computing*, 741–750, ACM Press, New York

[18] Pan, V. Y. (1996), Optimal and nearly optimal algorithms for approximating polynomial zeros, *Computers and Math. (with Applications)* **31, 12**, 97–138

[19] Pan, V. Y. (2001), Univariate Polynomials: Nearly Optimal Algorithms for Factorization and Rootfinding, *Proc. International Symp. on Symbolic and Algebraic Computation (ISSAC '01)*, 253–267, ACM Press, New York

[20] Pan, V. Y. (2001a), *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser/Springer, Boston/New York

[21] Pan, V. Y. (2002), Univariate Polynomials: Nearly Optimal Algorithms for Factorization and Rootfinding, *Journal of Symbolic Computations* **33, 5**, 701–733

[22] Pan, V. Y. and Zheng, A. (2011), New Progress in Real and Complex Polynomial Root-Finding, *Computers and Math. (with Applications)* **61**, 1305–1334

[23] Schönhage, A. (1982), The fundamental theorem of algebra in terms of computational complexity, *Dept. of Math., University of Tübingen*, Tübingen, Germany

[24] Van Dooren, P. M. (1994), Some numerical challenges in control theory, *Linear Algebra for Control Theory, IMA Vol. Math. Appl* **62**

[25] Weierstrass, K. (1903), Neuer Beweis des Fundamentalsatzes der Algebra, *Mathematische Werker* **Tome III**, 251–269, Mayer und Mueller, Berlin

[26] Wilson, G. T. (1969), Factorization of the Covariance Generating Function of a Pure Moving-average Process, *SIAM J. on Numerical Analysis* **6**, 1–7

[27] Z. Zeng (2005), Computing multiple roots of inexact polynomials, *Mathematics of Computation* **74**, 869–903