Università di Pisa
Dipartimento di Informatica

Technical Report : TR-97-03

GCD of polynomials and Bezout matrices

Gemignani L.

January 13, 1997

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

GCD of polynomials and Bezout matrices

Luca Gemignani *

Dipartimento di Informatica, Università di Pisa

Corso Italia 40, 56125 Pisa, Italy.

Abstract

A new algorithm is presented for computing an integer polynomial similar to the GCD of two polynomials $u(x)$ and $v(x)$ $\in \mathbf{Z}[x]$, $\deg(u(x)) = n \geq \deg(v(x))$. Our approach uses structured matrix computations involving Bezout matrices rather than Hankel matrices. In this way we reduce the computational costs showing that the new algorithm requires $O(n^2)$ arithmetical operations or $O(n^4(\log^2 n + l^2))$ Boolean operations, where $l = \max\{\log(\| u(x) \|_\infty), \log(\| v(x) \|_\infty)\}$.

1 Introduction

The problem we want to study in this paper is the efficient computation of an integer polynomial similar to the GCD of two polynomials $u(x)$ and $v(x) \in \mathbf{Z}[x]$, $\deg(u(x)) = n \geq \deg(v(x))$. Polynomial GCD-algorithms occur in rational function arithmetic to keep the "quolynomials" simplified [14], in squarefree and shiftfree factorizations [15], [17], and in the Hensel construction [18].

Recently many authors [4], [16], [1], [3], [2] and [7] have proposed GCD-algorithms based on a matrix approach involving Hankel matrices constructed starting from the coefficients of $u(x)$ and $v(x)$. These algorithms also give a way for evaluating the inertia and the signature of Hankel matrices which is of great interest in most of the basic algorithms of real algebraic geometry. The focus of all these studies is on the equivalence between the computation of a triangular factorization of the Hankel matrix $H_n = (h_{i+j-1})$, where $v(x)/u(x) = \sum_{i=1}^{2n-1} h_i x^{-i} + o(x^{-2n})$, and Euclid's algorithm applied to $u(x)$ and $v(x)$. As shown in [7], an integer triangular factorization of H_n can be computed using $O(n^2)$ arithmetical operation or $O(n^4 \log^2(\max_{1\leq i\leq 2n-1} |h_i|))$ Boolean time in the average case and $O(n^5 \log^2(\max_{1\leq i\leq 2n-1} |h_i|))$ in the worst case. Here and hereafter we assume that the classical algorithm with cost $O(n^2)$ is used for multiplications over $\mathbf{Z}$ of integers of length $O(n)$. Since the maximum length of the entries of H_n is dominated by nl where $l = \max\{\log(\| u(x) \|_\infty), \log(\| v(x) \|_\infty)\}$, this means that we are able to solve the polynomial GCD problem in time $O(n^6(l^2 + \log^2 n))$ or $O(n^7(l^2 + \log^2 n))$ in the worst case. Comparable upper bounds are obtained in [4] and [16].

In this paper we develop a new algorithm which leads to a better upper bound to the Boolean time. This improvment is achieved by replacing H_n with the Bezout matrix B_n associated with the polynomials $u(x)$ and $v(x)$, that is

$$B_n = (b_{i,j}), \quad \frac{u(x)v(y) - v(x)u(y)}{x - y} = \sum_{i,j=1}^{n} b_{i,j} x^{i-1} y^{j-1}.$$

The solution of linear algebra problems involving Bezout matrices has been given a new interest in connection with quantifier elimination problems (see [13] and [10]) and real root counting problems (see [8] and [9]).

Specifically, we show that an integer block triangular factorization of $J_n B_n J_n$, where J_n is the permutation matrix having 1 on the antidiagonal, yields the coefficients which express an integer polynomial similar to the GCD of $u(x)$ and $v(x)$ with respect to the polynomial basis defined by

$$p_0(x) = u_n, \quad p_1(x) = xp_0(x) + u_{n-1},$$
$$p_j(x) = xp_{j-1}(x) + u_{n-j}, \quad j = 2, \ldots, n.$$

By means of structured matrix techniques such a factorization can be computed efficiently using only $O(n^2)$ arithmetical operations. Moreover, using Hadamard's inequality, we estimate an essentially linear growth rate for the length of the entries of the triangular factors by proving that the length of the largest entry is bounded by $n \log(\max_{1\leq i,j\leq n} |b_{i,j}|)$. Since the entries of B_n are bounded by $2n \| u(x) \|_\infty \| v(x) \|_\infty$, our method determines an integer polynomial similar to the GCD of $u(x)$ and $v(x)$ at the cost of $O(n^4(l^2 + \log^2 n))$ Boolean operations in the average case and $O(n^5(l^2 + \log^2 n))$ Boolean operations in the worst case.

Finally, we observe that $J_n B_n J_n$ is congruent to H_n and, therefore, within the same computational cost our approach also determines the inertia and the signature of H_n.

*Supported by 40% funds of the Progetto Analisi numerica e matematica computazionale of MURST and by G.N.I.M. of C.N.R..

2 Preliminaries

Let $u(x) = \sum_{i=0}^{n} u_i x^i$ and $v(x) = \sum_{i=0}^{m} v_i x^i$ be two polynomials $\in \mathbf{Z}[x]$ of degree n and m, respectively. For the sake of notational simplicity assume that $u(x)$ is a monic polynomial, that is, $u_n = 1$. We associate a family of Hankel matrices $H_k \in Z^{k\times k}$, $k = 1, 2, \ldots$,

$$H_k = \begin{pmatrix} h_1 & h_2 & \ldots & h_k \\ h_2 & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ h_k & h_{k+1} & \ldots & h_{2k-1} \end{pmatrix},$$

with the polynomials $u(x)$ and $v(x)$ by considering the formal power series

$$\frac{v(x^{-1})}{xu(x^{-1})} = \sum_{i=1}^{+\infty} h_i x^{i-1}.$$

Analyzing the properties of the matrix H_n allows us to give a matrix formulation of computing polynomial GCD. In fact it can be shown that the greatest common divisor of $u(x)$ and $v(x)$ is univocally determined by a suitable vector in the kernel of the matrix H_n [2]. Such a vector can be determined by means of a recursive procedure which finds a generalized triangular factorization of H_n.

Let $0 = m_0 < m_1 < \ldots < m_l \leq n$ be the integers such that $\det H_{m_i} \neq 0$, $1 \leq i \leq l$ and $\det H_j = 0$ if $j \neq m_i,\ j \leq n$. The block triangular factorization algorithm [11] applied to H_n produces an upper triangular matrix $R \in \mathbf{R}^{n\times n}$,

$$R = (\mathbf{r}_1, \ldots, \mathbf{r}_n),\ \ \mathbf{r}_1 = \begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix},\ \ \mathbf{r}_i = \begin{pmatrix} \bar{\mathbf{r}}_i \\ 1 \\ \mathbf{0} \end{pmatrix},\ \ i \geq 2,$$

such that

$$R^T H_n R = D,$$

where

$$D = \begin{pmatrix} D_1 & & & \\ & \ddots & & \\ & & D_l & \\ & & & 0 \end{pmatrix}$$

is a block diagonal matrix. The diagonal block $D_i \in \mathbf{R}^{\delta_i \times \delta_i}$, $\delta_i = m_i - m_{i-1}$, is a lower triangular (with respect to the antidiagonal) Hankel matrix

$$D_i = \begin{pmatrix} & & & d_1^{(i)} \\ & \bigcirc & d_1^{(i)} & d_2^{(i)} \\ & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ d_1^{(i)} & d_2^{(i)} & \ldots & d_{\delta_i}^{(i)} \end{pmatrix}, \quad d_1^{(i)} \neq 0.$$

It is interesting to point out that this block triangular factorization is not generally unique. However, since

$$H_{m_i} \begin{pmatrix} \bar{\mathbf{r}}_{m_{i-1}+1} \\ 1 \\ \mathbf{0} \end{pmatrix} = d_1^{(i)} \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}, \quad 2 \leq i \leq l, \tag{1}$$

and

$$H_{m_i} \bar{\mathbf{r}}_{m_i+1} = -\begin{pmatrix} h_{m_i+1} \\ \vdots \\ h_{2m_i} \end{pmatrix}, \quad 1 \leq i \leq l, \tag{2}$$

then any upper triangular factor $\tilde{R} = (\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_n)$ satisfies $\tilde{\mathbf{r}}_{m_i+1} = \mathbf{r}_{m_i+1}$.

Let us introduce the polynomials

$$q_i(x) = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{m_i-1} \end{pmatrix}^T \bar{\mathbf{r}}_{m_i+1} + x^{m_i}, \quad 0 \leq i \leq l.$$

It is well known (see [6]) that $\{q_i(x)\}_{i=0}^{l-1}$ defines a set of orthogonal polynomials with respect to the symmetric bilinear form

$$\langle \phi, \psi \rangle = \mathbf{p}^{\mathbf{t}} H_n \mathbf{q},\ \ \phi = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix}^T \mathbf{p},\ \ \psi = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix}^T \mathbf{q},$$

on the space Π_n of polynomials of degree less than n. Moreover, these polynomials satisfy a three-term recurrence relation, that is,

$$q_{-1}(x) = 0, \quad q_0(x) = 1, \tag{3}$$

$$q_{j+1}(x) = c_{j+1}(x) q_j(x) - \beta_j q_{j-1}(x),\ 0 \leq j \leq l-1, \tag{4}$$

where $c_{j+1}(x)$ are monic polynomials of degree δ_{j+1},

$$c_{j+1}(x) = x^{\delta_{j+1}} + \sum_{i=0}^{\delta_{j+1}-1} c_{j+1,i} x^i.$$

Both the coefficients of $c_{j+1}(x)$ and β_j can be computed by imposing the orthogonality conditions. In fact if $i \neq j$ we have that

$$\langle x^k q_j(x), x^h q_i(x) \rangle = 0$$

for $k \leq \delta_{j+1}$ and $h \leq \delta_{i+1}$. If, otherwise, $i = j$ and $k, h \leq \delta_{j+1}$ then

$$\langle x^k q_j(x), x^h q_j(x) \rangle = \begin{cases} 0, & \text{if } k+h < \delta_{j+1} - 1; \\ d_{h+k+2-\delta_{j+1}}^{(j+1)}, & \text{elsewhere;} \end{cases}$$

holds. Hence, we obtain that

$$\beta_j = \frac{d_1^{(j+1)}}{d_1^{(j)}}, \tag{5}$$

and

$$c_{j+1,k} = -\frac{1}{d_1^{(j+1)}} \sum_{i=k+1}^{\delta_{j+1}} c_{j+1,i} \langle x^i q_j(x), x^{\delta_{j+1}-k-1} q_j(x) \rangle, \tag{6}$$

with $0 \leq k \leq \delta_{j+1} - 1$.

In order to compute these scalar products efficiently we may observe that (4) can be viewed in matrix form as

$$\mathbf{r}_{m_{j+1}+1} = (P^{\delta_{j+1}} + \sum_{i=0}^{\delta_{j+1}-1} c_{j+1,i} P^i) \mathbf{r}_{m_j+1} - \beta_j \mathbf{r}_{m_{j-1}+1}, \tag{7}$$

for $0 \leq j \leq l-2$, where P denotes the shift operator,

$$P = \begin{pmatrix} \mathbf{0}^T & 0 \\ I_{n-1} & \mathbf{0} \end{pmatrix},$$

and I_j is the $j \times j$ identity matrix. Since we have

$$H_n P - P^T H_n = \mathbf{e}_n \mathbf{h}^t - \mathbf{h}\mathbf{e}_n^t,$$

where $\mathbf{e}_j$ is the j-th column vector of I_n and

$$\mathbf{h} = \begin{pmatrix} h_{n+1} \\ \vdots \\ h_{2n-1} \\ 0 \end{pmatrix},$$

it follows that we may compute $H_n \mathbf{r}_{m_{j+1}+1}$ with the cost of $n\delta_{j+1}$ arithmetical operations. Thus, combining relations (5), (6) and (7) one readily obtains an algorithm for determining $q_l(x)$ or, equivalently, $\bar{\mathbf{r}}_{m_l+1}$ which requires $O(n^2)$ arithmetical operations.

In [7] this approach has been used for devising a fast algorithm for the gcd computation. Performing the computations over the integers we first determine an integer polynomial similar to $q_l(x)$, that is equal to $aq_l(x)$ for a certain $a \in \mathbf{Z}$. Then we find a polynomial similar to the GCD $r(x)$ of $u(x)$ and $v(x)$ by observing that [2]

$$u(x) = r(x) q_l(x). \tag{8}$$

In this paper we propose an alternative approach based on the computation of a generalized triangular factorization of Bezout matrices rather than Hankel ones. The reduction of the Euclidean scheme to computing a generalized block triangular factorization of a Bezout matrix has been already considered in [2]. However, differently from [2] where a divide and conquer strategy is used for the fast parallel computation of a polynomial remainder sequence over an arbitrary field, in the next section we develop a sequential algorithm which runs over an integral domain.

A Bezout matrix $B_n = (b_{i,j})$ of order n can be easily associated with the polynomials $u(x)$ and $v(x)$ by setting

$$\sum_{i,j=1}^{n} b_{i,j} x^{i-1} y^{j-1} = \frac{u(x)v(y) - v(x)u(y)}{x-y}.$$

A classical result relating the matrix H_n and B_n says that

$$B_n = \hat{B}_n H_n \hat{B}_n, \tag{9}$$

where $\hat{B}_n$ denotes the $n \times n$ Bezout matrix associated with $u(x)$ and 1, namely,

$$B(u,1) = \begin{pmatrix} u_1 & u_2 & \dots & u_n \\ u_2 & \iddots & \iddots & \\ \vdots & \iddots & O & \\ u_n & & & \end{pmatrix}.$$

Let J_n denote the $n \times n$ permutation matrix having 1 in the antidiagonal, i.e., its (i,j) entry is 1 for $j = n - i + 1$, $1 \le i, j \le n$; then (9) implies that

$$J_n B_n J_n = J_n \hat{B}_n H_n \hat{B}_n J_n.$$

Now, since $J_n \hat{B}_n$ is a lower triangular matrix there follows that a generalized block triangular factorization of H_n immeadiately gives an analogous decomposition of $J_n B_n J_n$, $U^T J_n B_n J_n U = D$, where

$$\hat{B}_n J_n U = R. \tag{10}$$

Thus we find that

$$\begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix}^T B_n J_n U \mathbf{e}_{m_i+1} = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{pmatrix}^T R\mathbf{e}_{m_i+1} = q_i(x),$$

which means that suitable columns of U define the coefficients which express the polynomials $q_i(x)$ with respect to the polynomial basis

$$p_0(x) = 1, \quad p_1(x) = x p_0(x) + u_{n-1}, \tag{11}$$
$$p_j(x) = x p_{j-1}(x) + u_{n-j}, \quad j = 2, \dots, n. \tag{12}$$

These polynomials form the so-called Horner basis (called also control basis in [12]) because they naturally appear in evaluation of a polynomial by means of the Ruffini-Horner algorithm. An algebraic viewpoint to Bezout matrices as matrices representing quadratic form with respect to the Horner basis appeared in [5]. This also leads to a generalization of Bezout matrices and their properties to the multivariate case. In a linear algebra setting the Horner basis can be associated with the characteristic polynomials of the leading principal submatrices of a certain Hessenberg matrix. In fact, (10) and (11) can be viewed in matrix form in the following way:

$$x \begin{pmatrix} p_0(x) \\ \vdots \\ p_{n-1}(x) \end{pmatrix} = F \begin{pmatrix} p_0(x) \\ \vdots \\ p_{n-1}(x) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ p_n(x) \end{pmatrix}, \tag{13}$$

which says that $p_1(x), \dots p_n(x)$ coincide with the leading principal submatrix of the modified Frobenius matrix

$$F = P^T + \begin{pmatrix} -u_{n-1} \\ \vdots \\ -u_0 \end{pmatrix} \mathbf{e}_1^T. \tag{14}$$

3 Constructing a p.r.s. associated with $J_n B_n J_n$

In this section we first describe a fast method for computing the matrix U and, therefore, for determining the coefficients of $q_l(x)$ in the Horner basis. We recall that in the view of (8) these coefficients univocally determine the GCD of $u(x)$ and $v(x)$. Then we consider a suitable variant of our method which only performs computations over $\mathbf{Z}$ and finally we present a cost analysis in a Boolean model of computation. If $z \in \mathbf{Z}$, we define $L(z)$, its length, by

$$L(z) = \begin{cases} 1, & \text{if } z = 0; \\ \lfloor \log |z| + 1 \rfloor, & \text{elsewhere.} \end{cases}$$

We assume that the classical algorithm is used for multiplications over $\mathbf{Z}$ and, therefore, cost$(a \times b)$=$L(a) \times L(b)$.

In the view of (7) and (10) we easily find that the $m_{j+1}+$ 1-st column $\mathbf{u}_{m_{j+1}+1}$ of U is equal to

$$J_n \hat{B}_n^{-1} (P^{\delta_{j+1}} + \sum_{i=0}^{\delta_{j+1}-1} c_{j+1,i} P^i) \hat{B}_n J_n \mathbf{u}_{m_j+1} - \beta_j \mathbf{u}_{m_{j-1}+1}.$$

By replacing the polynomial vector

$$\begin{pmatrix} p_0(x) \\ \vdots \\ p_{n-1}(x) \end{pmatrix}$$

with

$$J_n\hat{B}_n\begin{pmatrix}1\\ \vdots\\ x^{n-1}\end{pmatrix}$$

into the relation (13), we obtain that

$$(J_n\hat{B}_n)^{-1}FJ_n\hat{B}_n = P^T + \mathbf{e}_n\begin{pmatrix}-u_0\\ \vdots\\ -u_{n-1}\end{pmatrix}^T,$$

that is, $(J_n\hat{B}_n)^{-1}FJ_n\hat{B}_n$ coincides with the Frobenius matrix associated with $u(x)$. In this way we conclude that

$$F^T + \mathbf{w}\mathbf{e}_n^T = (\hat{B}_nJ_n)^{-1}P\hat{B}_nJ_n = \tilde{P},$$

which implies

$$(F^{T^{\delta_{j+1}}} + \sum_{i=0}^{\delta_{j+1}-1} c_{j+1,i}F^{T^i} + \sum_{i=0}^{\delta_{j+1}-1}\mathbf{w}_i^{(j)}\mathbf{e}_{n-i}^T)\mathbf{u}_{m_j+1} = \\ = \mathbf{u}_{m_{j+1}+1} - \beta_j\mathbf{u}_{m_{j-1}+1},$$

for certain vectors $\mathbf{w}_i^{(j)}$. Since we have

$$\mathbf{e}_{n-i}^T\mathbf{u}_{m_j+1} = 0, \quad 0 \le i \le \delta_{j+1}-1,$$

then we arrive at the following recursions for the columns of U,

$$\mathbf{u}_{m_{j+1}+1} = \sum_{i=0}^{\delta_{j+1}} c_{j+1,i}F^{T^i}\mathbf{u}_{m_j+1} - \beta_j\mathbf{u}_{m_{j-1}+1}, \qquad (15)$$

where we set $c_{j+1,\delta_{j+1}} = 1$.

Again both the coefficients of $c_{j+1}(x)$ and b_j can be evaluated by imposing the orthogonality conditions. We observe that the scalar product $\langle x^kq_j(x), x^hq_i(x)\rangle$ can be evaluated in matrix form as $(P^k\mathbf{r}_{m_j+1})^TH_nP^h\mathbf{r}_{m_i+1}$. Equivalently we can write $\mathbf{u}_{m_j+1}^T\tilde{P}^{kT}J_nB_nJ_n\tilde{P}^h\mathbf{u}_{m_i+1}$ which gives $(F^{T^k}\mathbf{u}_{m_j+1})^TJ_nB_nJ_nF^{T^h}\mathbf{u}_{m_i+1}$. In the view of (5) and (6) we therefore have

$$\beta_j = \frac{(F^{T^{\delta_{j+1}-1}}\mathbf{u}_{m_j+1})^TJ_nB_nJ_n\mathbf{u}_{m_j+1}}{(F^{T^{\delta_j-1}}\mathbf{u}_{m_{j-1}+1})^TJ_nB_nJ_n\mathbf{u}_{m_{j-1}+1}}, \qquad (16)$$

and

$$\sum_{i=k}^{\delta_{j+1}} c_{j+1,i}(F^{T^i}\mathbf{u}_{m_j+1})^TJ_nB_nJ_nF^{T^{\delta_{j+1}-k-1}}\mathbf{u}_{m_j+1} = 0, \quad (17)$$

with $0 \le k \le \delta_{j+1}-1$.

In order to compute these scalar products efficiently we consider the displacement operator $\mathcal{K}: \mathbf{R}^{n\times n} \to \mathbf{R}^{n\times n}$,

$$\mathcal{K}(A) = FA - AF^T.$$

Observe that

$$\mathcal{K}(J_nB_nJ_n) = (\tilde{P}^T - \mathbf{e}_n\mathbf{w}^T)J_nB_nJ_n - J_nB_nJ_n(\tilde{P} - \mathbf{w}\mathbf{e}_n^T) = \\ = J_n\hat{B}_n(P^TH - HP)\hat{B}_nJ_n - \mathbf{e}_n\hat{\mathbf{w}}^T + \hat{\mathbf{w}}\mathbf{e}_n^T = \bar{\mathbf{w}}\mathbf{e}_n^T - \mathbf{e}_n\bar{\mathbf{w}}^T,$$

for certain vectors $\hat{\mathbf{w}}$ and $\bar{\mathbf{w}}$. This implies that

$$AF^T\mathbf{u}_{m_j+1} = FA\mathbf{u}_{m_j+1} + \mathbf{e}_n(\bar{\mathbf{w}}^T\mathbf{u}_{m_j+1}),$$

and, therefore, the vector $AF^T\mathbf{u}_{m_j+1}$ can be computed at the cost of $O(n)$ arithmetical operations whenever we know $A\mathbf{u}_{m_j+1}$ and $\bar{\mathbf{w}}$. In this way we are able to evaluate β_j and the coefficients of $c_{j+1}(x)$ at the overall computational cost of $O(n\delta_{j+1})$ which leads to an $O(n^2)$-algorithm.

A suitable variant of this algorithm only performs computations over $\mathbf{Z}$. Analogously with (1) and (2), we find that the columns of U solve certain linear systems, namely,

$$(J_nB_nJ_n)_{m_i}\begin{pmatrix}\bar{\mathbf{u}}_{m_{i-1}+1}\\ 1\\ \mathbf{0}\end{pmatrix} = d_1^{(i)}\begin{pmatrix}\mathbf{0}\\ 1\end{pmatrix}, \quad 2 \le i \le l, \quad (18)$$

and

$$(J_nB_nJ_n)_{m_i}\bar{\mathbf{u}}_{m_i+1} = -\begin{pmatrix}\tilde{b}_{1,m_i+1}\\ \vdots\\ \tilde{b}_{m_i,m_i+1}\end{pmatrix}, \quad 1 \le i \le l, \quad (19)$$

where we denote as A_j the $j\times j$ leading principal submatrix of a given matrix A and, moreover, $J_nB_nJ_n = \tilde{B}_n = (\tilde{b}_{i,j})$. Cramer's rule applied to these linear systems says that the vectors

$$\tilde{\mathbf{u}}_{m_{i-1}+1} = \frac{\det(\tilde{B}_n)_{m_i}}{d_1^{(i)}}\begin{pmatrix}\bar{\mathbf{u}}_{m_{i-1}+1}\\ 1\\ \mathbf{0}\end{pmatrix}, \quad 2 \le i \le l,$$

and

$$\hat{\mathbf{u}}_{m_i+1} = \det(\tilde{B}_n)_{m_i}\begin{pmatrix}\bar{\mathbf{u}}_{m_i+1}\\ 1\\ \mathbf{0}\end{pmatrix}, \quad 1 \le i \le l,$$

have integer entries. This also implies that

$$z_i = \frac{\det(\tilde{B}_n)_{m_i}}{d_1^{(i)}} \in \mathbf{Z}, \quad 2 \le i \le l.$$

By multiplying (15) by

$$\theta_j = \frac{\det(\tilde{B}_n)_{m_{j+2}}}{d_1^{(j+2)}}\left(\frac{\det(\tilde{B}_n)_{m_{j+1}}}{d_1^{(j+1)}}\right)^{2\delta_{j+1}+1},$$

we obtain that

$$\sum_{i=0}^{\delta_{j+1}}\frac{\det(\tilde{B}_n)_{m_{j+2}}}{d_1^{(j+2)}}\left(\frac{\det(\tilde{B}_n)_{m_{j+1}}}{d_1^{(j+1)}}\right)^{2\delta_{j+1}}c_{j+1,i}F^{T^i}\tilde{\mathbf{u}}_{m_j+1} = \\ = \left(\frac{\det(\tilde{B}_n)_{m_{j+1}}}{d_1^{(j+1)}}\right)^{2\delta_{j+1}+1}\tilde{\mathbf{u}}_{m_{j+1}+1} + \theta_j\beta_j\mathbf{u}_{m_{j-1}+1}.$$

Since we have

$$\frac{\det(\tilde{B}_n)_{m_{j+1}}}{(d_1^{(j+1)})^{\delta_{j+1}}} = (-1)^{L_{j+1}}\det(\tilde{B}_n)_{m_j}$$

with $L_{j+1} = \delta_{j+1} + (\delta_{j+1}+1)\delta_{j+1}/2$, there follows that

$$\theta_j\beta_j = \frac{\det(\tilde{B}_n)_{m_{j+2}}}{d_1^{(j+2)}}\left(\frac{\det(\tilde{B}_n)_{m_{j+1}}}{d_1^{(j+1)}}\right)^{2\delta_{j+1}+1}\frac{d_1^{(j+1)}}{d_1^{(j)}} = \\ = (-1)^{L_{j+1}}z_{j+2}(\det(\tilde{B}_n)_{m_{j+1}})^{\delta_{j+1}}z_{j+1}^{\delta_{j+1}}z_j.$$

This gives the relations

$$z_{j+1}^{2\delta_{j+1}+1}\tilde{\mathbf{u}}_{m_{j+1}+1} = \sum_{i=0}^{\delta_{j+1}} z_{j+2}z_{j+1}^{2\delta_{j+1}} c_{j+1,i}F^{T^i}\tilde{\mathbf{u}}_{m_j+1}+$$

$$+(-1)^{L_{j+1}}z_{j+2}(\det(\tilde{B}_n)_{m_{j+1}})^{\delta_{j+1}}z_{j+1}^{\delta_{j+1}}\tilde{\mathbf{u}}_{m_{j-1}+1}. \qquad (20)$$

By dividing each side by $(d_1^{(j+2)})^{\delta_{j+2}-1}$ we obtain that

$$z_{j+1}^{2\delta_{j+1}+1}\hat{\mathbf{u}}_{m_{j+1}+1} =$$

$$= \sum_{i=0}^{\delta_{j+1}} \det(\tilde{B}_n)_{m_{j+1}} z_{j+1}^{2\delta_{j+1}} c_{j+1,i}F^{T^i}\tilde{\mathbf{u}}_{m_j+1}+$$

$$+(-1)^{L_{j+1}}(\det(\tilde{B}_n)_{m_{j+1}})^{\delta_{j+1}+1}z_{j+1}^{\delta_{j+1}}\tilde{\mathbf{u}}_{m_{j-1}+1}. \qquad (21)$$

Regarding at the computation of the quotients, we observe that the coefficients

$$\tilde{c}_{j+1,i} = \det(\tilde{B}_n)_{m_{j+1}} z_{j+1}^{2\delta_{j+1}} c_{j+1,i},$$

can be written as

$$\tilde{c}_{j+1,i} = (-1)^{L_{j+1}}\det(\tilde{B}_n)_{m_j}(\det(\tilde{B}_n)_{m_{j+1}})^{\delta_{j+1}} z_{j+1}^{\delta_{j+1}} c_{j+1,i}.$$

From (17) it follows that $-\tilde{c}_{j+1,k}$ is equal to

$$\frac{\sum_{i=k+1}^{\delta_{j+1}} \tilde{c}_{j+1,i}(F^{T^i}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n F^{T^{\delta_{j+1}-k-1}}\tilde{\mathbf{u}}_{m_j+1}}{\det(\tilde{B}_n)_{m_{j+1}} z_{j+1}}.$$

By induction we can then prove that

$$\tilde{c}_{j+1,k} \in \mathbf{Z}, \quad 0 \le k \le \delta_{j+1},$$

and, moreover $\tilde{c}_{j+1,k}$ is similar to

$$(-1)^{L_{j+1}}\det(\tilde{B}_n)_{m_j}(\det(\tilde{B}_n)_{m_{j+1}})^k z_{j+1}^k.$$

In this way we arrive at the following algorithm which computes over the integers the coefficients representing a p.r.s. associated with $\tilde{B}_n$ with respect to the Horner basis.

Algorithm

Input: a Bezout matrix $B_n \in \mathbf{Z}^{n\times n}$.

Output: the integer entries of the vectors $\tilde{\mathbf{u}}_{m_i+1}$ and $\hat{\mathbf{u}}_{m_i+1}$ which define an integer block triangular factorization of $\tilde{B}_n = J_n B_n J_n$.

1. compute $h = \delta_{j+1} - 1$ such that

$$(F^{T^h}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n \tilde{\mathbf{u}}_{m_j+1} \ne 0;$$

2. compute

$$\frac{(F^{T^h}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n \tilde{\mathbf{u}}_{m_j+1}}{z_{j+1}} = \det(\tilde{B}_n)_{m_{j+1}};$$

3. determine the coefficients $\tilde{c}_{j+1,i}$, $0 \le i \le \delta_{j+1}$, and then compute $\hat{\mathbf{u}}_{m_{j+1}+1}$ by means of (21);

4. compute $k = \delta_{j+2} - 1$ such that

$$(F^{T^k}\hat{\mathbf{u}}_{m_{j+1}+1})^T J_n B_n J_n \hat{\mathbf{u}}_{m_{j+1}+1} \ne 0,$$

and then compute

$$\frac{((F^{T^h}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n \tilde{\mathbf{u}}_{m_j+1})^k}{(\det(\tilde{B}_n)_{m_{j+1}})^{2k-1}} =$$

$$= \det(\tilde{B}_n)_{m_{j+1}}(d_1^{(j+2)})^k = (-1)^{L_{j+2}}z_{j+2};$$

5. determine $\tilde{\mathbf{u}}_{m_{j+1}+1}$ as at step 3 by means of (20) and if $m_{j+1}+1 < n$ then return to step 1.

Concerning the cost analysis of this algorithm we first notice that Hadamard's inequality allows us readily to bound the largest entry a of the vectors $\tilde{\mathbf{u}}_{m_i+1}$ and $\hat{\mathbf{u}}_{m_i+1}$. Namely, we have that

$$L(a) \le \frac{(n+1)}{2}\log(n+1) + (n+1)\log(\max_{1\le i,j\le n}|b_{i,j}|).$$

Then we estimate the length of the coefficients of the quotients by observing that (17) says that the coefficients $\tilde{c}_{j+1,k}$, $0 \le k \le \delta_{j+1}-1$, satisfy a suitable linear system. This system has the form

$$T\mathbf{x} = \tilde{c}_{j+1,\delta_{j+1}}\mathbf{w},$$

where $T = (t_{h,k}), t_{h,k} = t_{h-k+1}$ is a $\delta_{j+1}\times\delta_{j+1}$ lower triangular Toeplitz matrix with integer entries

$$t_{h-k+1} = (F^{T^{h-k}}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n F^{T^{\delta_{j+1}-1}}\tilde{\mathbf{u}}_{m_j+1},$$

and $\mathbf{w}$ is an integer vector with entries w_i,

$$w_i = (F^{T^i}\tilde{\mathbf{u}}_{m_j+1})^T J_n B_n J_n F^{T^{\delta_{j+1}-1}}\tilde{\mathbf{u}}_{m_j+1}.$$

Let c be the maximum of the magnitude of the entries of T and $\mathbf{w}$ then we find that every coordinate of $\mathbf{x}$ is bounded in magnitude by

$$\delta_{j+1}^{\delta_{j+1}/2+1}c^{\delta_{j+1}+1}\tilde{c}_{j+1,\delta_{j+1}}.$$

Since we have

$$c \le n^{2(\delta_{j+1}+1)}|a|^2 \max_{1\le i,j\le n}|b_{i,j}|(\| u(x) \|_\infty)^{2\delta_{j+1}},$$

we obtain that

$$L(\max|c_{j+1,i}|) = O(\delta_{j+1}L(a)).$$

Therefore we may perform one step of our algorithm using only $O(n\delta_{j+1}^2(L(a))^2)$ Boolean time which gives the overall Boolean cost of $O(n^2\delta(L(a))^2)$ with $\delta = \max\delta_i$. Since δ is typically $O(1)$ the cost of our algorithm is

$$O(n^4\log^2(\max_{1\le i,j\le n}|b_{i,j}|))$$

and it can increase to

$$O(n^5\log^2(\max_{1\le i,j\le n}|b_{i,j}|))$$

in certain pathological cases.

The coefficients of B_n are well bounded in terms of the coefficients of $u(x)$ and $v(x)$. Specifically, we find that

$$\max_{1 \le i,j \le n} |b_{i,j}| \le 2n \parallel u(x) \parallel_\infty \parallel v(x) \parallel_\infty .$$

This leads to an algorithm which computes the coefficients with respect to the Horner basis of an integer polynomial similar to the GCD of $u(x)$ and $v(x)$ at the cost of $O(n^4(l^2 + \log^2 n))$ in the average case and of $O(n^5(l^2 + \log^2 n))$ in the worst case, where

$$l = \max\{\log(\parallel u(x) \parallel_\infty), \log(\parallel v(x) \parallel_\infty)\}.$$

4 Conclusions and further extensions

In this paper we have developed an algorithm which solves the polynomial GCD problem for two polynomials $u(x)$ and $v(x)$ by constructing an integer block triangular factorization of a certain Bezout matrix B_n associated with $u(x)$ and $v(x)$. Since the size of the entries of B_n is well bounded in terms of the coefficients of $u(x)$ and $v(x)$, we find that our approach improves the computational performances of the other proposed algorithms involving Hankel matrices. Moreover, the computed decomposition of B_n yields informations about the inertia and the signature of B_n. These computations are important in solving root counting problems and quantifier elimination problems and, therefore, investigations of the use of our algorithm in those contexts would also be interesting. This will be the topic of subsequent papers.

References

[1] D. Bini, L. Gemignani, *On the Euclidean scheme for polynomials having interlaced real zeros*, Proc. 2nd Ann. ACM Symp. on Parallel Algorithms and Architectures, ACM Press, 1990, pp. 254-258.

[2] D. Bini, L. Gemignani, *Fast parallel computation of polynomial remainder sequence via Bezout and Hankel matrices* , SIAM J. Comput., 24 (1995), pp. 63-77.

[3] D. Bini, L. Gemignani, V. Pan, *Improved parallel computations with matrices and polynomials,* Proc. 18th ICALP, Lecture Notes in Computer Science, 510 (1991), Springer, Berlin, pp. 520-531.

[4] S. Cabay, P. Kossowski, *Power series remainder sequences*, J. Symbolic Comp., 10 (1990), pp. 139-163.

[5] J. P. Cardinal, *Dualite et algorithmes iteratifs pour la resolution de systemes polynomiaux*, Doctoral Dissertion, University of Rennes I, France (1993).

[6] R. Freund, H. Zha, *A look- ahead algorithm for the solution of general Hankel Systems*, Numer. Math., 64 (1993), pp. 295-322.

[7] L. Gemignani, *Solving Hankel Systems over the Integers,* J. Symbolic Computation, 18 (1994), pp. 573-584.

[8] L. Gemignani, *Computationally efficient applications of the Euclidean Algorithm to zero location*, Linear Algebra Appl., 249 (1996), pp. 79-91.

[9] L. González-Vega, *Solving the real root counting problem via Bezoutians*, TR. 12-1993, Departamento de Matematicas, Estadistica Y Computacion, Universidad de Cantabria, Spain.

[10] L. González-Vega, *A combinatorial algorithm solving some quantifier elimination problems*, "25 Years of Quantifier Elimination and Cylindrical Algebraic Decomposition", Texts and Monographs in Symbolic Computation, Springer-Verlag,(1993).

[11] W. B. Gragg, A. Lindquist, *On the partial realization problem*, Linear Algebra Appl., 50 (1983), pp. 277-319.

[12] U. Helmke, P. A. Fuhrman, *Bezoutians*, Linear Algebra Appl., 122 (1989), pp. 1039-1097.

[13] H. Hong, *Quantifier elimination for formulas constrained by quadratic equations*, Proceedings ISSAC-93, ACM-Press.

[14] R. Loos, *Generalized polynomial remainder sequences*, Computer algebra symbolic and algebraic computations, B. Buchberger, G. E. Collins, R. Loos, eds., Springer, 115-137 (1982).

[15] R. Loos, *Computing rational zeros of integral polynomials by p-adic expansion*, SIAM J. Comp., 12 (1983), pp. 286-293.

[16] J. R. Sendra, J. Llovet, *GCD of polynomials and Hankel matrices*, J. Symbolic Comp., 13 (1992), pp.25-39.

[17] D. Y. Y. Yun, *On square-free decomposition algorithms*, SYMSAC, 1976, pp. 26-35.

[18] D. Y. Y. Yun, *The Hensel Lemma in algebraic manipulation*, Ph. D. Thesis, Dept. of Ma., MAC-TR-138, M.I.T., 1974.