Int. J. Adv. Appl. Math. and Mech. 10(3) (2023) 66 – 77 (ISSN: 2347-2529)

Journal homepage: www.ijaamm.com

IJAAMM

International Journal of Advances in Applied Mathematics and Mechanics

An efficient free-division Algorithm for computing polynomial subresultants using non-homogeneous Bezout Matrix

Research Article

Morou AMIDOU[a, *], Maimouna SALOU[b], Ousmane MOUSSA TESSA[b]

[a] *IREM, Abdou Moumouni University, P.O. Box 10896, Niamey, NIGER*

[b] *Department of Mathematics and Computer Science, Abdou Moumouni University, P.O. Box 10662, Niamey, NIGER*

Received 05 December 2023; accepted (in revised version) 20 February 2023

Abstract: In this paper we propose an efficient free-division method for computing the principal subresultant sequence of two polynomials over an integral domain. We introduce a new algorithm based on the non-homogeneous Bezout Matrix and the improved Samuelson-Berkowitz algorithm. This method shows a better practical behaviour when the polynomials are dense with parametric coefficients.

MSC: 13C10 • 19A13 • 14Q20 • 03F65

Keywords: Subresultants • Samuelson-Berkowitz algorithm • Non-Homogeneous Bezout matrix • Free-division algorithm

1. Introduction

The computation of polynomial subresultants sequences is a standard tool in Computer Algebra and geometry. The concept of resultant of two univariate polynomials was originally given by means of determinant of Bezout matrix since 1748. In the cases where the basic ring is a field, algorithms based on euclidean remainder sequence have been used extensively. Nevertheless, for polynomials with coefficients in a domain, as computations have to take place in the related fraction field, we encounter steadily growth of their size, doubled with the problem of avoiding division. To deal with such situation, many authors rely on free-division algorithms, e.g. for computing the principal subresultants sequence of two polynomials over a domain [6]. Such algorithms are based on an application of the Samuelson-Berkowitz algorithm to Hybrid Bezout and Symmetric Bezout matrices [1]. They turn out to show better practical behavior than the usual methods based on Sylvester matrices, specially in the case with parametric coefficients. Our goal is to present a matrix whose minors give the subresultants sequence when the coefficients are in a domain.

In Section 2, we present some classical algorithms intended to efficient subresultant polynomials computations. So, we revisit the representation of the GCD of two univariate polynomials through Sylvester matrix, hybrid and non-homogeneous Bezout matrices to compute polynomial subresultants, as well the parametrization of the GCD degree.

In Section 3, we propose our main result with relying on a modified Samuelson-Berkowitz algorithm [6] to non-homogeneous Bezout Matrix. Henceforth, we yield an efficient algorithm in computing subresultants polynomial with parametric coefficients. We give demonstrative examples explaining the application of our method. Finally, in Section 4, we discuss the conclusions of the current study, pointing out some limits and prospects.

* Corresponding author.

E-mail address(es): moorou_a@yahoo.fr (Morou AMIDOU).

2. Materials and Methods

2.1. Classical polynomial subresultants computation

Let us recall classical definitions related to the concept of polynomial subresultants [1, 8], where D is a integral domain.

Definition 2.1.
Let $P(X), Q(X) \in D[X]$ such that
$P(x) = p_0X^n + p_1X^{n-1} + \ldots + p_n$ et
$Q(X) = q_0X^m + q_1X^{m-1} + \ldots + q_m$
with $n = \deg(P(X)) \geq m = \deg(Q(X))$.
For $i \in \{0,\ldots,m\}$ if $n > m$ and $i \in \{0,\ldots,m-1\}$ if $n = m$, we define the Sylvester matrix of index i associated to $P(X)$ and $Q(X)$ as the $(n+m-2i) \times (n+m-i)$ matrix:

$$\mathbf{Sylv_i}(P,Q) = \begin{pmatrix} p_0 & \cdots & p_n & & \\ & \ddots & & \ddots & \\ & & p_0 & \cdots & p_n \\ q_0 & \cdots & q_m & & \\ & \ddots & & \ddots & \\ & & q_0 & \cdots & q_m \end{pmatrix}$$

For $i = 0$, we have the classic Sylvester matrix.
The polynomial subresultants can be obtained from the previous matrices using the notions of polynomial determinant :

Definition 2.2.
Let A be a matrix of type (m,n) with $n \geq m$. The polynomial determinant of A is the polynomial **detpol**(A) defined as follow :

$$\mathbf{detpol}(A) = \sum_{k=0}^{n-m} det(\Delta_k)x^{n-m-k}$$

where Δ_k is the square matrix of type m obtained from the $m-1$ first column and the $(m+k)^{th}$ column of A.
If A is a square matrix, then **detpol**$(A) = det(A)$.

Example 2.1.
If $A = \begin{pmatrix} 1 & 2 & 0 & 4 & 0 \\ 0 & 2 & 1 & 3 & -1 \\ 1 & 1 & 2 & 4 & 0 \end{pmatrix}$ with $n = 5, m = 3$, we have

$$\mathbf{detpol}(A) = \begin{vmatrix} 1 & 2 & 0 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \end{vmatrix} x^2 + \begin{vmatrix} 1 & 2 & 4 \\ 0 & 2 & 3 \\ 1 & 1 & 4 \end{vmatrix} x + \begin{vmatrix} 1 & 2 & 0 \\ 0 & 2 & -1 \\ 1 & 1 & 0 \end{vmatrix} = 5x^2 + 3x - 1.$$

In the following part, we will see how the Sylvester matrix can give a parameterization of the GCD degree.

2.2. Parametrization of the GCD degree

Let $P(X)$ and $Q(X)$ be two polynomials in $D[\alpha_1,\ldots,\alpha_r][X]$ where $\alpha_1,\ldots,\alpha_r$ are parameters with values in the algebraic closure of D.

Proposition 2.1.
If $0 \leq i \leq \min\{\deg(P),\deg(Q)\}$, *the parametrization of the GCD degree exhibits a sequence of coefficients* $a_i(P,Q) \in D[\alpha_1,\ldots,\alpha_r]$ *such as :*

$$deg(GCD(P,Q)) = i \iff \begin{cases} a_0(P,Q) = \ldots = a_{i-1}(P,Q) = 0 \\ \\ a_i(P,Q) \neq 0 \end{cases} \qquad (PS)$$

Example 2.2.
: A known sequence which verify this property (PS) is the sequence of the following polynomial subresultants $psc_i(P,Q)$ where $0 \le i \le \min\{deg(P), deg(Q)\}$ called the principal subresultant coefficient of index i. It is the determinant of the square matrix obtained from Sylvester matrices as

$$psc_i(P,Q) = \begin{pmatrix} p_0 & \cdots & \cdots & \cdots & \cdots & p_{n-2i+m-1} \\ & \ddots & & & & \vdots \\ & & p_0 & \cdots & \cdots & p_{n-i} \\ q_0 & \cdots & \cdots & \cdots & \cdots & q_{m-2i+n-1} \\ & \ddots & & & & \vdots \\ & & \ddots & & & \vdots \\ & & & q_0 & \cdots & q_{m-i} \end{pmatrix}$$

Remark 2.1.
The Structure theorem [3, 8] furnishes an algorithm improving the subresultants algorithms in the defective case. This theorem shows that the minors of Bezout matrices also give (PS) sequence from the Samuelson-Berkowitz algorithm.

As well, based on relation between the matrices $Sylv(P(X), Q(X))$ and $Sylv(P(X), XQ(X))$, the Flipflop algorithm improves the subresultant algorithm in the non-defective case.

Definition 2.3.
If $0 \le i \le \min\{deg(P), deg(Q)\}$, the polynomial subresultant of index i is the polynomial determinant of the corresponding matrix $\mathbf{Sylv_i(P,Q)}$, i.e.

$$\mathbf{Sres_i}(P,Q) = \mathbf{detpol}(Sylv_i(P,Q)).$$

Remark 2.2.
Sylvester matrices and polynomial subresultants provide a way to compute the greatest common divisor of two univariate polynomials as

$$deg(GCD(P,Q) = i \Leftrightarrow rang(\mathbf{Sylv}(P,Q)) = n+m-i$$

and

$$GCD(P,Q) = \mathbf{Sres}_i(P,Q)$$

The Sylvester matrices are not the only whose minors give polynomial subresultants. The determinant of Bezout matrices also gives the resultant of two univariate polynomials. A more general definition of Bezout is the following one.

Definition 2.4.
Let $P(X), Q(X) \in D[X]$ with $deg(P) = n$, $deg(Q) = m$ and $n \ge m$. The Bezout matrix associated to $P(X)$ and $Q(X)$ is the symmetric matrix:

$$\mathbf{Bez}(P,Q) = \begin{pmatrix} c_{0,0} & \cdots & c_{0,n-1} \\ \vdots & \cdots & \vdots \\ c_{n-1,0} & \cdots & c_{n-1,n-1} \end{pmatrix}$$

where the coefficients $c_{i,j}$ verify :

$$\frac{P(X)Q(Y) - P(Y)Q(x)}{x-Y} = \sum_{i,j=0}^{n-1} c_{i,j} X^i Y^j.$$

Definition 2.5.
Let $P(X), Q(X) \in D[X]$ with $deg(P) = n$, $deg(Q) = m$ and $n \ge m$. The hybrid Bezout matrix associated to $P(X)$ and $Q(X)$ is the $n-$square matrix $\mathbf{Hbez}(P,Q)$ with entries are defined as follows:

- if $1 \le i \le m, 1 \le j \le n$, then $(\mathbf{Hbez}(P,Q))_{i,j}$ is the coefficient of X^{n-j} of the polynomial $(p_0X^{m-i} + \ldots + p_{m-i}) \times (q_{m-i+1}X^{n-m+i-1} + \ldots + q_mX^{n-m}) - (p_{m-i+1}X^{n-m+i-1} + \ldots + p_n) \times (q_0X^{m-i} + \ldots + q_{m-i})$.
- if $m+1 \le i \le n, 1 \le j \le n$, the entry $(\mathbf{Hbez}(P,Q))_{i,j}$ of $\mathbf{Hbez}(P,Q)$ is the coefficient of X^{n-j} of the polynomial $X^{n-i}Q(X)$.

Remark 2.3.
The rank of the Bezout matrices is linked to the degree of the GCD as :

$$\begin{aligned} deg(GCD(P,Q)) &= n - rang(\mathbf{Bez}(P,Q)) \\ &= n - rang(\mathbf{Hbez}(P,Q)). \end{aligned}$$

It is also possible to obtain the polynomial subresultants from the minors of the Barnett matrix [9] and the Hankel matrix associated to (P,Q). We refer the reader to [1] for more details.

2.3. Classical Samuelson-Berkowitz Algorithm

Among several algorithms for computing the characteristic polynomials, we rely on the Samuelson-Berkowitz algorithm [5–7], due to its smooth sequential version and steady behavior.

Definition 2.6.
Let $A = (a_{ij}) \in D^{n\times n}$ be a n-square matrix, for $1 \le r \le n$, we denote by :

- $A_n = A$;
- I_r the identity matrix of order r;
- $A_r = (a_{i,j})$ the $r \times r$ matrix of elements $a_{i,j}$, whenever $1 \le i, j \le r$, as the leading principal submatrix of order r of A.
- $P_r(X) = \det(A_r - XI_r) = \sum_{i=0}^{r} \lambda_{r-i} X^i$
  with $\lambda_0 = 1$ and $\lambda_r = (-1)^r \det(A_r)$.
- R_r the $1 \times r$ matrix of elements $a_{r+1,j}$, whenever $1 \le j \le r$ and $r \le n-1$.
- S_r the $r \times 1$ matrix of elements $a_{i,r+1}$, whenever $1 \le i \le r$ and $r \le n-1$.

Definition 2.7.
Let $P(X) = \sum_{k=0}^{d} a_k X^k$ be a polynomial with coefficients in D.

The **Toeplitz matrix** associated to the coefficients of $P(X)$ is a $(d+1) \times d$ lower triangular matrix :

$$Toep(P(X)) = \begin{pmatrix} a_0 & 0 & 0 & \dots & 0 \\ a_1 & a_0 & 0 & \dots & 0 \\ a_2 & a_1 & a_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{d-1} & a_{d-2} & \dots & a_1 & a_0 \\ a_d & a_{d-1} & a_{d-2} & \dots & a_1 \end{pmatrix}$$

Remark 2.4.
Let $P(X) = \sum_{k=0}^{d} a_k X^k$ be in $D[X]$.

- We have $\overrightarrow{P} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix}$ with coefficients of $P(X)$.
- For $r \ge 2$, the Samuelson formula can be written as an $(r+2)$-vector equality :

$$\overrightarrow{P_{r+1}(X)} = Toep(Q_{r+1}(X)) \times \overrightarrow{P_r(X)} \quad \text{where}$$

$$\begin{aligned} Q_{r+1}(X) &= -X^{r+1} + a_{r+1,r+1}X^r + (R_r S_r)X^{r-1} + \\ &\dots + (R_r A_r^i S_r)X^{r-1-i} + \dots + (R_r A_r^{r-1} S_r). \end{aligned}$$

- $\overrightarrow{P_1(X)} = \begin{pmatrix} -1 \\ a_{11} \end{pmatrix} = Toep(Q_1(X))$

3. Results and Discussion

3.1. Non-Homogeneous Bezout matrix

Definition 3.1.
The non-homogeneous Bezout Matrix associated to $P(x)$ and $Q(x)$) is the n-square matrix $\mathbf{Nhbez}(P,Q)$ which first m rows are those of the corresponding Bezout Matrix and the last $n-m$ are the coefficients of $Q(x)$

$$\mathbf{Nhbez}(P,Q) = \begin{pmatrix} c_{0,0} & \dots & c_{0,m} & \dots & c_{0,n-1} \\ \vdots & & \vdots & & \vdots \\ c_{m-1,0} & \dots & c_{m-1,m} & \dots & c_{m-1,n-1} \\ q_m & \dots & q_0 & & \\ & \ddots & & \ddots & \\ & & q_m & \dots & q_0 \end{pmatrix}$$

The computation of subresultans can be highly related to non-homogeneous Bezout minors.

Lemma 3.1.
(Theorem 2.2 of [1]) With the previous notations, let assume that $n \geq m$. Then the polynomial subresultants of $P(X)$ and $Q(X)$ are :

$$(-1)^{k(k-1)/2}\mathbf{Sres_{n-k}}(P,Q) = (-1)^{(n-m)(n-m-1)/2} \times (Nh_{k,0}X^{n-k} + Nh_{k,1}X^{n-k-1} + \ldots + Nh_{k,n-k}$$

for each $k \in \{n-m+1,\ldots,n)$, where $Nh_{k,t}$ denotes the minor of order k extracted from the last $k-1$ rows and the $(n-k-t+1)^{th}$ row of the matrix $\mathbf{Nhbez}(P,Q)$ *(for each $0 \leq t \leq n-k$).*

Example 3.1.
Let $P(X) = 6X^5 - 9X^4 - 3X^3 - 5X^2 - 4X + 7$ and $Q(X) = X^4 + 7X^3 + 9X^2 + 3X - 6$.

$$\mathbf{Nhbez}(P,Q) = \begin{pmatrix} 45 & 93 & 67 & 61 & -36 \\ 93 & 88 & 80 & -59 & 18 \\ 67 & 80 & -51 & -58 & 54 \\ 61 & -59 & -58 & 6 & 42 \\ -6 & 3 & 9 & 7 & 1 \end{pmatrix}$$

3.2. Modified Samuelson-Berkowitz Algorithm

Definition 3.2.
The backward identity matrix of order n is

$$J_n = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Let's quote that the non-homogeneous Bezout matrix minors taking into account are those extracted from the lower right hand corner. Henceforth, we need to modify the classical Samuelson-Berkowitz Algorithm as follows :

Input : A square matrix $B \in D^{n\times n}$.

Output : The characteristic polynomial of

$J_n \times B \times J_n$.

1. Initialization :

$$A = J_n \times B \times J_n; \text{Vect} = \begin{pmatrix} 1 \\ -a_{11} \end{pmatrix}$$

2. Update : For r from 1 to $n-1$,

(a) Compute the entries $\{R_r A_r^{k-1} S_r\}_{k=1}^{r}$ of $Toep(Q_{r+1})$;
(b) $Vect := Toep(Q_{r+1}) \times Vect$;

3. Return P_n the unique polynomial such that $\vec{P}_n = Vect$

Remark 3.1.
As noted in [1], this algorithm compute the characteristic polynomial of n-square matrix over an integral domain and can be presented as an uniform family of arithmetic parallel circuits with size $O(n^{\alpha+1}\log n)$ and depth $O(\log^2 n)$. Here α is the exponent for the $n \times n$ fast parallel multiplication with $O(\log n)$-depth circuit.

In fact, the computations are done relying only on dot-products and matrix-vector multiplications instead of multiplying matrix-matrix. In such process, multiplying by J_n does not change the complexity.

3.3. Non-Homogeneous Bezout Algorithm

Our main result stands as the following algorithm

Input: P, Q

Output: The sequence of principal minors of **Nhbez**(P, Q).

1. Compute the matrix **Nhbez**(P, Q).
2. Apply the Sequential modified Samuelson-Berkowitz algorithm in order to compute the intended principal minors of **Nhbez**(P, Q).

In the appendix, we share implementation of the algorithm in Maple code.

3.4. Complexity

Let $P(X)$ and $Q(X)$ be two polynomials in $D[\alpha_1, \ldots, \alpha_r][X]$ where $\alpha_1, \ldots, \alpha_r$ are parameters with values in the algebraic closure of D. We denote $deg(P) = n$, $deg(Q) = m$ and $n \geq m$, r the number of parameters and N the total degree of $P(X)$ and $Q(X)$ with respect to the parameters.

Proposition 3.1.
Without parameters, that is $r = 0$,

- *the Bezout algorithms requires :*
  $O(n^4)$ *arithmetic operations in* **D**
- *The improved subresultant algorithm and the Flipflop algorithm require* $O(n^2)$ *arithmetic operations in* **D**.

Proof. The cost of the Bezout algorithms corresponds essentially to the cost of the computation of Samuelson-Berkowitz sub-algorithm. □

Remark 3.2.
In the case where there is no parameter and n is higher, it is clear that the usual pseudo-division based methods, and particularly the Flipflop algorithm have a better practical behavior.

Proposition 3.2.
The use of the non-homogeneous Bezout algorithm requires $O(n^{2r+3}N^{2r})$ *arithmetic operations in* D *to compute the PS sequence.*

Proof. We have just to reproduce the same of the symmetric and hybrid Bezout algorithms([7]). Note that the use of the non-homogeneous Bezout matrix requires only $O(n^2 N^{2r})$ arithmetic operations in **D**. □

Remark 3.3.
If the coefficients are parameters, the Bezout algorithms have a better practical behaviour as pointed out in [7]. When the polynomials are dense, the Non-Homogeneous Bezout Algorithm have a good practical behavior as shown with the following experimental results.

3.5. Experimental Results

In $Z[a,b,c][X]$, we consider the pair of integer polynomials in coefficients with parameters a, b and c.

Example 3.2.
We consider the pair $\mathscr{P}_1$ of integer univariate polynomials such that $deg(p_1) = 7$, $deg(q_1) = 9$, 2 the number of parameters and $N = 7$ the total degree of $p_1(X)$ and $q_1(X)$ with respect to the parameters.

$$\mathscr{P}_1 = \begin{cases} p_1[X] &= 6a^7X^{10} - 5b^3acX^2 - 4X - 7 \\ q_1[X] &= cX^9 + 3X - 6 \end{cases}$$

Example 3.3.

$$\mathscr{P}_2 = \begin{cases} p_2[X] &= 6aX^{12} + X^{11} + X^{10} + X^9 \\ &\quad + X^8 + X^7 + X^6 + X^5 + X^4 \\ &\quad + X^3 - 5X^2 - 4X - 7 \\ q_2[X] &= bX^3 + X^2 + 3X - 6 \end{cases}$$

Example 3.4.

$$\mathscr{P}_3 = \begin{cases} p_3[X] &= 6a^3bX^{12} + X^{11} + X^{10} + X^9 \\ &\quad + X^8 + X^7 + X^6 + X^5 \\ &\quad + X^4 + X^3 - 5X^2 - 4X - 7 \\ q_3[X] &= X^5 + bX^4 + X^3 + X^2 + 3X - 6 \end{cases}$$

Example 3.5.

$$\mathscr{P}_4 = \begin{cases} p_4[X] &= a^3bX^{10} + aX^9 + X^8 + X^7 + \\ &\quad X^6 + X^5 + X^4 + X^3 \\ &\quad - 5X^2 - 4X - 7 \\ q_4[X] &= bX^5 + X^4 + X^3 + X^2 + 3X - 6 \end{cases}$$

In the following table, we have the required time of computing the PS sequence using the share algorithm. The first pair of polynomials are sparse and the three last pairs have been chosen dense. Note that the good behavior of the non-homogeneous algorithm when the polynomials are dense. The computations are done with the computer system Maple. In the appendix, the details of the case of $\mathscr{P}_1$ are given.

4. Conclusions

In this paper, we proposed the application of a modified Samuelson-Berkowitz Algorithm in order to compute the sequence of principal minors of the non-homogeneous Bezout MatriX associated to pairs of polynomials.

In the same vein, as done in [7], it's well-known that combining classical subresultant and Flipflop algorithms yields better results than Bezout free-division approach.

Nevertheless, performing in domains with indeterminates reveals the occurrence of inflation in arithmetic's operations, and at the same time, we know that the required divisions can be dramatically expensive. Therefore, using free-division approach seems more appropriate to handle such situations. Indeed, the size of manipulated objects turns to decrease with the Berkowitz-method, particularly when dealing with dense pair of polynomials.

In control theory, a major deal for computing of the GCD is that the process frequently with a very large number of polynomials. Bearing this in mind, such requirement makes our method not suitable for such context, even relying on

Table 1. Comparison of the average computing time

Polynomials (P_i, n, m, r, N)	PS Sequence by Non-homogeneous	PS Sequence by Hybrid Bezout	PS Sequence by Symmetric Bezout
($\mathscr{P}_1$, 10, 9, 3, 7)	9,110s	13,301s	13,859
($\mathscr{P}_2$, 12, 3, 2, 1)	89,500s	685,266s	$> 1h$
($\mathscr{P}_3$, 12, 5, 2, 4)	29,362s	234,601s	$> 1h$
($\mathscr{P}_4$, 10, 5, 2, 4)	11,702s	86,563s	$> 1h$

the pairwise type approaches for GCD. Perhaps, one can modify Barnett's approach method [2, 9] for computing the GCD of several polynomials reckoning on our method using the transformed companion matrix of $J_n \times A \times J_n$. Also, we hope that our method can provide means for a more efficient implementation of the classical Bézout-QR method [9] with less computational complexity.

References

[1] G. Diaz-Toca, L. Gonzalez-Vega. Various new expressions for subresultants and their applications. Applicable Algebra in Engineering, Communication and Computing, 2004.

[2] L. Gonzales-Vega. An elementary proof of Barnett's theorem about the greatest common divisor of several univariate polynomials. Linear Algebra and its Applications, 247:185-202, 1996.

[3] H. Lombardi, M. F. Roy, M. Safey. New structure theorem for subresultants. Journal of symbolic computation, Vol. 29, 663-689, 2000.

[4] H. Hong, J. Yang. Subresultant of several univariate polynomials (2022), arxiv.org:2112.15370, preprint, 35 pages. Online available from https://arxiv.org/abs/2112.15370

[5] J. Abdeljaoued. Algorithmes rapides pour le calcul du polynôme caractéristique. Thèse de doctorat, Université de Franche-comté, 1997.

[6] J. Abdeljaoued. Berkowitz algorithm, maple and computing the characteristic polynomial in an arbitrary commutative ring. MapleTech,4(3), Birkhauser, 1997.

[7] J. Abdeljaoued, L. Gonzalez-Vega, G. Diaz-Toca. Minors of Bezout matrices, subresultants and the parameterization of the degree of the polynomial greatest common divisor. Springer-Verlag, 2004.

[8] S. Basu, R. Pollack, M. F. Roy. Algorithms in real algebraic geometry. Algorithms in mathematics. Vol. 10. Springer-verlag, 2003.

[9] G.M. Diaz-Toca, L. Gonzalez-Vega. Barnett's Theorems about the Greatest Common Divisor of several univariate polynomials through Bezout-like matrices. J. Symb. Comput. 34(1), 59–81 (2002). Online available from https://doi.org/10.1006/jsco.2002.0542

[10] B. Chi, A. Terui. The GPGCD algorithm with the Bézout matrix. In: Computer Algebra in Scientific Computing (CASC 2020), Lecture Notes in Computer Science, Vol. 12291, pp. 170–187. Springer International Publishing, Cham (2020). Online available from https://doi.org/10.1007/978-3-030-60026-6 10

[11] B. Chi, A. Terui. The GPGCD algorithm with the Bézout matrix for multiple univariate polynomials (2022), arXiv:2205.02984, preprint, 12 pages. Online available from https://arxiv.org/pdf/2205.02984.pdf

[12] S. Uygun. The binomial transforms of the generalized (s,t)-Jacobsthal matrix. Int. J. Adv. Appl. Math. and Mech. 6(3)(2019) 14-20.

[13] Y. Soykan. A study on generalized Fibonacci polynomials: Some formulas. Int. J. Adv. Appl. Math. and Mech. 10(1)(2022) 39-118.

[14] Y. Soykan. On binomial transform of the generalized reverse 3-primes sequence. Int. J. Adv. Appl. Math. and Mech. 8(2)(2020) 35-53.

Appendix

Maple codes

```
with(linalg):
Bezout:=proc(P,Q,x)# Computation of the
 matrix Jn.Bez.Jn #
local B,Bez,a,i,j;
B:=normal((P*subs(x=y,Q)-subs(x=y,P)*Q)/(x-y));
B:=collect(B,[x,y],distributed);
Bez:=array(1..degree(P,x),1..degree(P,x),[]);
for i from 1 to degree(P,x) do
for j from 1 to degree(P,x) do
a[i]:=coeff(B,x,degree(P,x)-i);
Bez[i,j]:=sort(coeff(a[i],y,degree(P,x)-j))
 od;od;
evalm(Bez)
end proc:

berkosam:=proc(A::matrix,x::name)
## Samuelson Berkowitz ##
local n,r,i,j,k,v,C,u,S,p,Q;
n:=coldim(A);v:=table([1=1,2=-A[1,1]]);
C[1]:=1;
for r from 2 to n do
for i to r-1 do S[i]:= -A[i,r] od;
C[2]:=-A[r,r];
for i from 1 to r-2 do
C[i+2]:=sum('A[r,k]*S[k]','k'=1..r-1);
for j to r-1 do
Q[j]:=sum('A[j,k]*S[k]','k'=1..r-1) od;
for j to r-1 do S[j]:=Q[j] od;
od;
C[r+1]:=sum('A[r,k]*S[k]','k'=1..r-1);
for i to r+1 do
Q[i]:=sum('C[i+1-k]*v[k]','k'=1..min(r,i));
od;p[1]:=x-A[1,1];
for i to r+1 do v[i]:=Q[i] od;
 p[r]:=sum('v[k]*x^(r-k+1)','k'=1..r+1)
od;
seq(sort(normal((-1)^r*tcoeff(p[r],x))),r=1..n);
end proc:

amb:=proc(P,Q,x)## Principal subresultant
sequence with symetric Bezout matrix ##
local s,n,m,t,k,a,b;
n:=degree(P,x);m:=degree(Q,x);a:=lcoeff(P,x);
b:=lcoeff(Q,x);
s[m]:=((-1)^((n-m)*(n-m-1)/2))*(b^(n-m-1))*Q;
for k from n-m+1 to n do
s[n-k]:=sum('berkosam(swaprow(Bezout(P,Q,x),
k,k+t),x)[k]*x^(n-k-t)','t'=0..n-k)/
((-1)^(k*(k-1)/2)*a^(n-m));
od;
seq(sort(s[k],x),k=0..m)
end proc:

amhb:=proc(P,Q,x)## Principal subresultant sequence
with Hybrid Bezout matrix ##
local s,n,m,t,k,Hb,i,j,b,J;
n:=degree(P,x);m:=degree(Q,x);b:=lcoeff(Q,x);
```

```
J:=array(1..n,1..n);## computation of Jn ##
for i from 1 to n do
for j from 1 to n do
if j=n+1-i then
J[i,j]:=1 else J[i,j]:=0;fi;od;od;
Hb:=evalm(J&*bezout(P,Q,x));
s[m]:=((-1)^((n-m)*(n-m-1)/2))*(b^(n-m-1))*Q;
for k from 0 to m-1 do
s[k]:=sum('((-1)^((n-k)*(n-k-1)/2))*berkosam
(swaprow(Hb,n-k,n-k+t),x)
[n-k]*x^(k-t)','t'=0..k)
od;
seq(s[k],k=0..m)
end proc:

with(linalg):
nonhB:=proc(P,Q,x)## Determination of the
non homogeneous Bezout matrix ##
local B,Nonh,a,i,j;
B:=normal((P*subs(x=y,Q)-subs(x=y,P)*Q)/(x-y));
B:=collect(B,[x,y],distributed);
Nonh:=array(1..degree(P,x),1..degree(P,x),[]);
for i from 1 to degree(P,x) do
for j from 1 to degree(P,x) do
a[i]:=coeff(B,x,i-1);
Nonh[i,j]:=sort(coeff(a[i],y,j-1))
 od;od;
evalm(Nonh)
end proc:
with(linalg):
NH:=proc(P,Q,x)
local B,Nh,a,i,j,n,m;
n:=degree(P,x);m:=degree(Q,x);
Nh:=array(1..n,1..n,[]);
for i from 1 to m do
for j from 1 to n do
Nh[i,j]:=nonhB(P,Q,x)[i,j];od;od;
for i from 1 to n-m do
for j from 1 to n do
Nh[m+i,j]:=coeff(x^(i-1)*Q,x,j-1);od;od;
evalm(Nh);
end proc:


amnb:=proc(P,Q,x)## Principal subresultant
sequence with nonhomogeneous Bezout matrix ##
local s,n,m,t,k,nh,i,j,b,J;
n:=degree(P,x);m:=degree(Q,x);b:=lcoeff(Q,x);
J:=array(1..n,1..n);##computation of Jn ##
for i from 1 to n do
for j from 1 to n do
if j=n+1-i then
J[i,j]:=1 else J[i,j]:=0;fi;od;od;
nh:=evalm(J&*(NH(P,Q,x)&*J));
s[m]:=((-1)^((n-m)*(n-m-1)/2))*(b^(n-m-1))*Q;
for k from 0 to m-1 do
s[k]:=sum('((-1)^((n-m)*(n-m-1)/2))/((-1)^((n-k)
*(n-k-1)/2))*berkosam(swaprow(nh,
n-k,n-k+t),x)[n-k]*x^(k-t)','t'=0..k)
od;
seq(s[k],k=0..m)
```

```
end proc:
```

%

With the pair $\mathscr{P}_1 = \begin{cases} p_1[X] &= 6a^7X^{10} - 5b^3acX^2 - 4X - 7 \\ q_1[X] &= cX^9 + 3X - 6 \end{cases}$,
the Sequential modified Samuelson-Berkowitz algorithm yields **Nhbez**(p_1, q_1) as

1. The matrix **Nhbez**(P,Q) =

$$\begin{bmatrix}
\begin{matrix}15ab^3bx^2c\\+45\end{matrix} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & -36a^7 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 \\
0 & 0 & 0 & 0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 \\
0 & 0 & 0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 \\
0 & 0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 & 0 \\
0 & 0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 & 0 & 0 \\
0 & 0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 & 0 & 0 & 0 \\
0 & \begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 & 0 & 0 & 0 & 0 \\
\begin{matrix}5ab^3bx^2c^2\\+7c\end{matrix} & \begin{matrix}-36a^7\\+4c\end{matrix} & 18a^7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-6 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

2. The PS sequence is given by

Sres$_0(P,Q) = 609359740010496\,a^{63} - 1983592903680\,a^{57}b^3c - 4407984230400\,a^{51}b^6c^3 - 4285540224000\,a^{45}b^9c^5 - 2380855680000\,a^{39}b^{12}c^7 - 1201792820576256\,a^{56}c - 826686000000\,a^{33}b^{15}c^9 - 1487694677760\,a^{56} - 148108270141440\,a^{50}b^3c^3 - 183708000000\,a^{27}b^{18}c^{11} - 3305988172800\,a^{50}b^3c^2 - 25515000000\,a^{21}b^{21}c^{13} - 3214155168000\,a^{44}b^6c^4 - 2025000000\,a^{15}b^{24}c^{15} - 1785641760000\,a^{38}b^9c^6 - 70312500\,a^9b^{27}c^{17} - 620014500000\,a^{32}b^{12}c^8 + 939782237921280\,a^{49}c^2 - 137781000000\,a^{26}b^{15}c^{10} + 214391292272640\,a^{43}b^3c^4 - 19136250000\,a^{20}b^{18}c^{12} + 11999512627200\,a^{37}b^6c^6 - 1518750000\,a^{14}b^{21}c^{14} - 52734375\,a^8b^{24}c^{16} - 374041950750720\,a^{42}c^3 - 115150878581760\,a^{36}b^3c^5 - 11419289280000\,a^{30}b^6c^7 - 360067680000\,a^{24}b^9c^9 + 81522879338880\,a^{35}c^4 + 28849445533440\,a^{29}b^3c^6 + 3590217691200\,a^{23}b^6c^8 + 180700632000\,a^{17}b^9c^{10} + 2917215000\,a^{11}b^{12}c^{12} - 9884453029920\,a^{28}c^5 - 3539328125760\,a^{22}b^3c^7 - 447555553200\,a^{16}b^6c^9 - 23059890000\,a^{10}b^9c^{11} - 387161250\,a^4b^{12}c^{13} + 661231226880\,a^{21}c^6 + 210081392640\,a^{15}b^3c^8 + 21709900800\,a^9b^6c^{10} + 724416000\,a^3b^9c^{12} - 23753502720\,a^{14}c^7 - 5662863360\,a^8b^3c^9 - 334924800\,a^2b^6c^{11} + 424673280\,a^7c^8 + 55050240\,ab^3c^{10} - 40353607\,c^{10} - 2949120\,c^9$

Sres$_1(P,Q) = 2821109907456\,a^{56}cx + 33059881728\,a^{56}x - 66119763456\,a^{56} + 73466403840\,a^{50}b^3c^2x - 146932807680\,a^{50}b^3c^2 + 71425670400\,a^{44}b^6c^4x - 142851340800\,a^{44}b^6c^4 + 39680928000\,a^{38}b^9c^6x - 79361856000\,a^{38}b^9c^6 + 13778100000\,a^{32}b^{12}c^8x - 4427575271424\,a^{49}c^2x - 27556200000\,a^{32}b^{12}c^8 + 3061800000\,a^{26}b^{15}c^{10}x - 548549148672\,a^{49}c^2 - 533311672320\,a^{43}b^3c^4x - 6123600000\,a^{26}b^{15}c^{10} + 425250000\,a^{20}b^{18}c^{12}x - 850500000\,a^{20}b^{18}c^{12} + 33750000\,a^{14}b^{21}c^{14}x - 67500000\,a^{14}b^{21}c^{14} + 1171875\,a^8b^{24}c^{16}x - 2343750\,a^8b^{24}c^{16} + 2655130254336\,a^{42}c^3x + 746636341248\,a^{42}c^3 + 577754311680\,a^{36}b^3c^5x + 88885278720\,a^{36}b^3c^5 + 30862944000\,a^{30}b^6c^7x - 775947320064\,a^{35}c^4x - 371836749312\,a^{35}c^4 - 219126902400\,a^{29}b^3c^6x - 78186124800\,a^{29}b^3c^6 - 19717992000\,a^{23}b^6c^8x - 4000752000\,a^{23}b^6c^8 - 555660000\,a^{17}b^9c^{10}x + 118413032976\,a^{28}c^5x + 85140574848\,a^{28}c^5 + 36172513440\,a^{22}b^3c^7x + 22255611840\,a^{22}b^3c^7 + 3736416600\,a^{16}b^6c^9x + 1800338400\,a^{16}b^6c^9 + 145089000\,a^{10}b^9c^{11}x + 43218000\,a^{10}b^9c^{11} + 1500625\,a^4b^{12}c^{13}x - 9579596544\,a^{21}c^6x - 9460063872\,a^{21}c^6 - 2705270400\,a^{15}b^3c^8x - 2472845760\,a^{15}b^3c^8 - 243432000\,a^9b^6c^{10}x - 200037600\,a^9b^6c^{10} - 6860000\,a^3b^9c^{12}x - 4802000\,a^3b^9c^{12} + 404683776\,a^{14}c^7x + 510064128\,a^{14}c^7 + 88058880\,a^8b^3c^9x + 107251200\,a^8b^3c^9 + 4704000\,a^2b^6c^{11}x + 5488000\,a^2b^6c^{11} - 8331264\,a^7c^8x - 12644352\,a^7c^8 - 1003520\,ab^3c^{10}x - 1505280\,ab^3c^{10} + 65536\,c^9x + 114688\,c^9$

Sres$_2(P,Q) = -612220032\,a^{49}cx^2 + 1224440064\,a^{49}cx - 1190427840\,a^{43}b^3c^3x^2 + 2040733440\,a^{43}b^3c^3x - 992023200\,a^{37}b^6c^5x^2 + 1417176000\,a^{37}b^6c^5x - 459270000\,a^{31}b^9c^7x^2 + 524880000\,a^{31}b^9c^7x - 127575000\,a^{25}b^{12}c^9x^2 + 109350000\,a^{25}b^{12}c^9x - 21262500\,a^{19}b^{15}c^{11}x^2 + 12150000\,a^{19}b^{15}c^{11}x - 1968750\,a^{13}b^{18}c^{13}x^2 - 136048896\,a^{42}c^2x + 562500\,a^{13}b^{18}c^{13}x - 78125\,a^7b^{21}c^{15}x^2 - 238085568\,a^{42}c^2 - 226748160\,a^{36}b^3c^4x - 396809280\,a^{36}b^3c^4 - 157464000\,a^{30}b^6c^6x - 275562000\,a^{30}b^6c^6 - 58320000\,a^{24}b^9c^8x - 102060000\,a^{24}b^9c^8 - 12150000\,a^{18}b^{12}c^{10}x - 21262500\,a^{18}b^{12}c^{10} - 1350000\,a^{12}b^{15}c^{12}x - 2362500\,a^{12}b^{15}c^{12} - 62500\,a^6b^{18}c^{14}x - 109375\,a^6b^{18}c^{14}$

Sres$_3(P,Q) = 0$,

Sres$_4(P,Q) = 0$,

$\mathbf{Sres}_5(P,Q) = 0,$

$\mathbf{Sres}_6(P,Q) = 0,$

$\mathbf{Sres}_7(P,Q) = 0,$

$\mathbf{Sres}_8(P,Q) = -18\,a^7cx^2 + 36\,a^7cx - 5\,ab^3c^3x^2 - 4\,c^2x - 7\,c^2.$