SresToolsNPB: Computing Resultant and Subresultants in Non-power Basis with Maple

JING YANG* and WEI YANG*†, School of Mathematic and Physics, Guangxi Minzu University, China

Abstract. We present a software package called SresToolsNPB for computing resultant and subresultants of polynomials in non-power basis, including the Newton or Bernstein basis. It is required that the produced resultant and subresultants are also defined in the same basis, which often has a simpler form than those in power basis. The package SresToolsNPB is implemented in Maple based on some basis-preserving algorithms recently developed for constructing resultant and subresultants of univariate polynomials in non-power basis. A detailed description of the usage of the functions contained in the package and illustrative examples are provided to show the capability of the implemented package.

CCS Concepts: • **Mathematics of computing** → **Solvers**.

Additional Key Words and Phrases: resultant, subresultant, non-power basis, basis-preserving algorithm, Newton polynomial, Bernstein polynomial

Recommended Reference Format:
Jing Yang and Wei Yang. 2021. SresToolsNPB: Computing Resultant and Subresultants in Non-power Basis with Maple. *Maple Trans.* X, Y, Article Z (2021), 8 pages. https://doi.org/XXXXXXXXX

1 Introduction

Resultant theory serves as a fundamental pillar within the domain of computer algebra and has been extensively studied in the past centuries (just list a few [3, 5, 8, 9, 11, 14, 15, 16, 17, 18, 19, 21, 23]). Most of the studies so far have concentrated on the study of polynomials in power basis (also known as standard basis or monomial basis). However, the increasing application of basis-preserving algorithms across a spectrum of computational tasks ([1, 6, 10, 12, 13]) has underscored the necessity for a deeper investigation into the resultant and subresultants within the context of non-power bases (see [2, 4, 20, 24]). Nevertheless, the development of theories and software tools which are specialized for computing resultant and subresultants for polynomials in non-power basis has been lagging, although significant strides have been taken, as evidenced by the works of [1, 7, 25, 26, 29], which have substantially bridged the aforementioned gap. Furthermore, we are not aware of any software tools or packages which are publicly available for computing resultant and subresultants in non-power basis.
In this paper, we introduce a Maple package named SresToolsNPB, which is specifically tailored for computing resultant and subresultants of polynomials expressed in Bernstein basis and Newton basis (which are two typical types of non-power bases). It should be pointed out that the package SresToolsNPB is designed with the requirement of maintaining the same basis for both the input and output polynomials. In other words, when calling the functions in the package, the output resultant/subresultants are represented in the same basis as used to formulate the input polynomials.

*Both authors contributed equally to this research.
†Corresponding author.

Authors' address: Jing Yang, yangjing0930@gmail.com; Wei Yang, weiyang020499@163.com, School of Mathematic and Physics, Guangxi Minzu University, 188 Daxue East Road, Xixiangtang District, Nanning, Guangxi, China, 530006.

The package SresToolsNPB is implemented based on the most recent algorithms developed by the authors and other colleagues for computing resultant/subresultant polynomials in non-power basis [22, 24, 27, 28]. These algorithms are basis-preserving, which indicates that they do not rely on basis transformation and thus can avoid extra computation cost and numerical instability caused by the transformation. In the package, nine functions are provided for computing resultant matrices, resultant and subresultant polynomials.
The rest of the paper is structured as follows. Section 2 provides a brief review of resultant, resultant matrices and subresultant polynomials in power basis. In Section 3, we give a formal description of the problems tackled by the package as well as the methods the implementation relies on. Then we illustrate the usage of public functions provided by SresToolsNPB and provide some examples for readers' reference in Section 4.

2 Review

In this section, we review three types of resultant and subresultants of polynomials. For this purpose, we assume $F, G \in \mathcal{K}[x]$ be of degree m and n, respectively, where $\mathcal{K}$ stands for an integral domain hereinafter.
Sylvester type. The *Sylvester matrix* of F and G with respect to x is defined as a matrix $\mathrm{Syl}(F,G)$ such that

$$\mathrm{Syl}(F,G)\cdot\begin{bmatrix}x^{m+n-1} & \cdots & x^0\end{bmatrix}^T = \begin{bmatrix}x^{n-1}F & \cdots & x^0F & x^{m-1}G & \cdots & x^0G\end{bmatrix}^T$$

We call $\det \mathrm{Syl}(F,G)$ the *Sylvester resultant* of F and G w.r.t. x. Subresultants are often formulated in the form of determinant polynomial. The *determinant polynomial* of a matrix M with order $p\times q$ where $p \le q$ is defined by

$$\mathrm{detp}M := \sum_{0\le j\le q-p} c_j x^j$$

where $c_j = \det\begin{bmatrix}M_1 & \cdots & M_{p-1} & M_{q-j}\end{bmatrix}$ and M_k stands for the k-th column of M. Let Syl_k be a matrix satisfying

$$\mathrm{Syl}_k(F,G)\cdot\begin{bmatrix}x^{m+n-2k-1} & \cdots & x^0\end{bmatrix}^T = \begin{bmatrix}x^{n-k-1}F & \cdots & x^0F & x^{m-k-1}G & \cdots & x^0G\end{bmatrix}^T$$

Then the k-th *subresultant* of F and G w.r.t. x is defined as $S_k(F,G) := \mathrm{detp}\,\mathrm{Syl}_k(F,G)$ and the principal coefficient of $S_k(F,G)$ is called the k-th principal coefficient of subresultant of F and G w.r.t. x, denoted by $s_k(F,G)$. Obviously, $S_0(F,G)$ is the Sylvester resultant of F and G w.r.t. x.
Bézout type. Assume $m \ge n$. The *Bézout matrix* of F and G w.r.t. x is defined as a matrix $\mathrm{Bez}(F,G)$ of order $m\times m$ satisfying

$$\frac{F(x)G(y)-F(y)G(x)}{x-y} = \overline{x}^T\cdot \mathrm{Bez}(F,G)\cdot\overline{y}$$

where $\overline{x} = [x^{m-1},\ldots,x^0]^T$ and $\overline{y} = [y^{m-1},\ldots,y^0]^T$. We call $\det\mathrm{Bez}(F,G)$ the *Bézout resultant* of F and G w.r.t. x. Let Bez_k be the submatrix of $\mathrm{Bez}(F,G)$ obtained by deleting its last k rows. Then we have $S_k(F,G) = c\cdot\mathrm{detp}\,\mathrm{Bez}_k$ for some constant c. Thus we call $\mathrm{detp}\,\mathrm{Bez}_k$ the *Bézout subresultant* of F and G w.r.t. x.
Barnett type. The formulation of the Barnett-type resultant matrix needs a concept called *companion* matrix. The *companion matrix* of a polynomial P with degree t is a matrix C_P such that $x\cdot\overline{x} \equiv_P C_P\cdot\overline{x}$ where $\overline{x} = [x^{t-1},\ldots,x^0]^T$. For given polynomials $F, G\in\mathcal{K}[x]$, the *Barnett resultant matrix* of F and G w.r.t. x is defined as $\mathrm{Bar}(F,G) := G(C_F)$, whose determinant is called the *Barnett resultant.* Let Bar_k be the submatrix of C obtained by deleting its first k rows. Then we have $S_k = c\cdot\mathrm{detp}\,\mathrm{Bar}_k$ for some constant c. Thus we call $\mathrm{detp}\,\mathrm{Bar}_k$ the *Barnett subresultant* of F and G w.r.t. x.

From now on, when the variable which the resultant/subresultants are defined w.r.t. is clear from the context, we can omit it for the sake of simplicity.

3 Problems and Methods

It is seen that the three types of resultant matrix, resultant and subresultants in Section 2 are formulated in power basis. A natural question is: what is their equivalence in other basis? By equivalence, we mean that the resultant/subresultants in non-power basis are similar to those in power basis, only differing by a non-zero constant factor. In this section, we review three such types of resultant matrices, resultants and subresultants in non-power basis, including Sylvester type in Bernstein basis, Bézout type and Barnett type in Newton basis. It should be pointed out that the formulation of resultant matrices in non-power basis does not require the polynomials to be given in the same basis. However, these matrices display simpler forms when the polynomials are formulated in the same basis (see [27, Example 4]). Therefore, in the rest of the paper, we assume the polynomials are given in Bernstein/Newton form.

3.1 Sylvester matrices and resultants in Bernstein basis

Let $w_s = \begin{bmatrix} w_{s,s}(x) & w_{s,s-1}(x) & \cdots & w_{s,0}(x)\end{bmatrix}^T$ where

$$w_{s,i}(x) = \binom{s}{i}(1-x)^{s-i}x^i \quad \text{for } 0 \le i \le s.$$

Then w_s is called the *Bernstein basis* of $\mathcal{K}_s[x]$. A polynomial written as a linear combination of $w_{s,i}(x)$'s is called a *Bernstein polynomial.*
The *Sylvester matrix* of F and G in Bernstein basis is defined as a matrix $\mathrm{Syl}^{(b)}(F,G)$ such that

$$\mathrm{Syl}^{(b)}(F,G) \cdot w_{m+n-1} = \begin{bmatrix} w_{n-1}F & w_{m-1}G\end{bmatrix}^T$$

We call $\det \mathrm{Syl}^{(\mathrm{b})}(F,G)$ the *Sylvester resultant* of F and G in Bernstein basis. When F and G are given in Bernstein basis, one can easily write down their Sylvester matrix in Bernstein basis from the above definition and then compute their resultant in Bernstein basis. More explicitly, when F and G are given in their Bernstein form below:

$$F(x) = \sum_{i=0}^{m} a_i w_{m,i}(x), \quad G(x) = \sum_{i=0}^{n} b_i w_{n,i}(x).$$

their *Sylvester matrix* $\mathit{Syl}^{(b)}(F,G)$ is

$$\mathrm{Syl}^{(b)}(F,G) = M^{(b)}D$$

where

- $$M^{(b)} = \left[\begin{array}{cccccc} a_m\binom{m}{m} & a_{m-1}\binom{m}{m-1} & \cdots & a_0\binom{m}{0} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & a_m\binom{m}{m} & a_{m-1}\binom{m}{m-1} & \cdots & a_0\binom{m}{0} \\ b_n\binom{n}{n} & b_{n-1}\binom{n}{n-1} & \cdots & b_0\binom{n}{0} & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & b_n\binom{n}{n} & b_{n-1}\binom{n}{n-1} & \cdots & b_0\binom{n}{0} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} n \text{ rows} \\ \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} m \text{ rows} \end{array},$$
- $$D = \mathrm{diag}\left[\frac{1}{\binom{m+n-1}{m+n-1}}, \frac{1}{\binom{m+n-1}{m+n-2}}, \ldots, \frac{1}{\binom{m+n-1}{0}}\right].$$

3.2 Sylvester subresultants in Bernstein basis

When it comes to subresultants, one may naturally ask the following question:

In: F and G in Bernstein polynomials, and an integer $k < \min(\deg F, \deg G)$
Out: $S_k(F, G)$ which is expressed in Bernstein basis.

A naive way for solving the above problem is: (1) converting F and G to polynomials in power basis, (2) computing $S_k(F, G)$, and (3) changing the basis back. We call this approach the basis-transformation-based method. It often involves intensive computation and also causes numerical instability. Another natural extension is to truncate the resultant matrices in Bernstein polynomials as done in power basis and compute the determinant polynomial (but in Bernstein basis). However, the resulting subresultants are not equal to those in power basis. Thus we have to explore other approaches to formulate subresultants in Bernstein basis.
We note that subresultants in power basis can also be expressed as determinants of matrices which contain subresultant matrices as submatrices [19]. Following this approach, Tan presents an analogy of subresultant formulation in Bernstein basis in [22]. More explicitly, the k-th *subresultant* of F and G w.r.t. x is

$$S_k(F, G) = \operatorname{detp}\begin{bmatrix}\mathrm{Syl}_k^{(b)}(F, G)\\ X_kP_k\end{bmatrix}$$

where $\mathrm{Syl}^{(b)}(F, G)$ is the submatrix of $M^{(b)}$ obtained by deleting the last k rows from each block consisting the coefficients of F and G respectively and deleting the last k columns, and

$$X_k = \begin{bmatrix}-(1-x) & x & & \\ & \ddots & \ddots & \\ & & -(1-x) & x\end{bmatrix}_{k\times(k+1)}$$

$$P_k = \begin{bmatrix}\binom{m+n-k-1}{m+n-k-1} & \cdots & \binom{m+n-k-1}{0} & & \\ & \ddots & & \ddots & \\ & & \binom{m+n-k-1}{m+n-k-1} & \cdots & \binom{m+n-k-1}{0}\end{bmatrix}_{(k+1)\times(m+n-k)}$$

Obviously, $S_0(F, G)$ is the Sylvester resultant of F and G w.r.t. x.
In order to expand $S_k = \det \mathrm{Syl}_k^{(b)}(F, G)$ into an expression in Bernstein basis, we suggest to replace $1 - x$ with y during the determinant expansion and change y back to $1 - x$ later.

3.3 Resultant matrices and resultants in Newton basis

Let $\lambda = (\lambda_1, \ldots, \lambda_t) \in \mathcal{K}^t$ and $N_\lambda(x) = \begin{bmatrix}N_t & \cdots & N_1 & N_0\end{bmatrix}^T$ where $N_i(x) = N_{i-1}(x)(x - \lambda_i)$ with the convention $N_0 := 1$. We call $N_\lambda(x)$[1] the *Newton basis* of $\mathcal{K}_t[x]$ w.r.t. λ. Moreover, a polynomial written as a linear combination of N_i's is called a *Newton polynomial.*
Bézout type. Assume $m \geq n$. The *Bézout matrix* of F and G in the basis N_λ is defined as a matrix $\mathrm{Bez}^\lambda(F, G)$ of order $m \times m$ satisfying

$$\frac{F(x)G(y) - F(y)G(x)}{x - y} = \widetilde{N_\lambda}(x)^T \cdot \mathrm{Bez}^\lambda(F, G) \cdot \widetilde{N_\lambda}(y)$$

where $\widetilde{N_\lambda}(x) = \begin{bmatrix}N_{m-1}(x) & \cdots & N_1(x) & N_0(x)\end{bmatrix}^T$ and $\widetilde{N_\lambda}(y) = \begin{bmatrix}N_{m-1}(y) & \cdots & N_1(y) & N_0(y)\end{bmatrix}^T$.
Barnett type. The formulation of the Barnett-type resultant matrix in Newton basis needs a concept called confederate matrix, which is an extension of the companion matrix in power basis to other bases. The *confederate matrix* of a polynomial P with degree t in $\widetilde{N_\lambda}(x)$ (where $\lambda \in \mathcal{K}^t$) is a matrix

[1] $B^\lambda(x)$ can be abbreviated as B^λ if no ambiguity occurs.

C_P^λ such that $x \cdot \widetilde{N_\lambda}(x) \equiv_P C_P^\lambda \cdot \widetilde{N_\lambda}(x)$. For given polynomials $F, G \in \mathcal{K}[x]$, the *Barnett resultant matrix* of F and G associated to $\widetilde{N_\lambda}(x)$ where $\lambda \in \mathcal{K}^m$ is defined as $\mathrm{Bar}^\lambda(F,G) := G(C_F^\lambda)$, whose determinant is called the *Barnett resultant.*
When the input polynomials are formulated in Newton basis, the authors present a basis-preserving method for computing $\mathrm{Bez}^\lambda(F,G)$ effectively [27], which relies on the reveal of an internal connection among the entries of $\mathrm{Bez}^\lambda(F,G)$. Then one can utilize the connection between $\mathrm{Bez}^\lambda(F,G)$ and $\mathrm{Bar}^\lambda(F,G)$ discovered in [29] to compute $\mathrm{Bar}^\lambda(F,G)$. Both of the methods are implemented in the Maple package SresToolsNPB.

3.4 Subresultant polynomials in Newton basis

Similar to Bernstein polynomials, one may consider the following problem:

In: F and G in Newton basis, and an integer $k < \min(\deg F, \deg G)$
Out: $S_k(F,G)$ which is expressed in the given Newton basis.

For solving the posed question, Wang and Yang extended the concept of determinant polynomial from power basis to Newton basis in [24]. Given a matrix $M \in \mathcal{K}^{p\times q}$ where $p \le q$ and a Newton basis $N_\lambda(x) = \begin{bmatrix} N_m & \dots & N_1 & N_0 \end{bmatrix}^T$, the determinant polynomial of M in $N_\lambda(x)$ is defined as

$$\mathrm{detp}_{N_\lambda} M := \sum_{0\le j\le q-p} c_j N_j(x)$$

where $c_j = \det \begin{bmatrix} M_1 & \cdots & M_{p-1} & M_{q-j} \end{bmatrix}$ and M_k stands for the k-th column of M. Then we provide the following answers to the above problem:

- Let Bez_k^λ be the submatrix of $\mathrm{Bez}^\lambda(F,G)$ obtained by deleting its last k rows. Then we have $S_k = c \cdot \mathrm{detp}\,\mathrm{Bez}_k^\lambda$ for some constant c.
- Let Bar_k^λ be the submatrix of Bar^λ obtained by deleting its first k rows. Then we have $S_k = c \cdot \mathrm{detp}\,\mathrm{Bar}_k^\lambda$ for some constant c.

Therefore, we call $\mathrm{detp}\,\mathrm{Bez}_k^\lambda$ the k-th *Bézout subresultant* and $\det \mathrm{Bar}_k^\lambda$ the k-th *Barnett subresultant* of F and G in the Newton basis $N_\lambda(x)$, respectively.

In summary, the problems which can be solved by the Maple package SresToolsNPB are summarized below.

- Given two Bernstein polynomials F and G with degrees m and n in Bernstein basis, compute the Sylvester resultant matrix in Bernstein basis.
- Given two Bernstein polynomials F and G with degrees m and n in Bernstein basis and $0 \le k < \min(m,n)$, compute the k-th Sylvester subresultant of F and G in Bernstein basis.
- Given $\lambda \in \mathbb{F}^n$ and two Newton polynomials F and G degrees m and n in $N_\lambda(x)$, compute the Bézout matrix of F and G in Newton basis.
- Given $\lambda \in \mathbb{F}^n$, two Newton polynomials F and G degrees m and n in $N_\lambda(x)$ and $0 \le k < \min(m,n)$, compute the k-th Bézout subresultant of F and G in Newton basis.
- Given $\lambda \in \mathbb{F}^n$ and two Newton polynomials F and G degrees m and n in $N_\lambda(x)$, compute the Barnett resultant matrix of F and G in Newton basis.
- Given $\lambda \in \mathbb{F}^n$, two Newton polynomials F and G degrees m and n in $N_\lambda(x)$ and $0 \le k < \min(m,n)$, compute the k-th Barnett subresultant of F and G in Newton basis.

4 Public Interface

This section presents the public functions of the SresToolsNPB package and their usage. It serves as a quick reference manual for the users of the package. We assume F and G are represented by their coefficient vectors f and g in the increasing degree.

BernsteinSylvesterMatrix. The calling sequence is

$$\text{BernsteinSylvesterMatrix}(f, g)$$

It computes the Sylvester resultant matrix of F and G in Bernstein basis.

BernsteinSylvesterResultant. The calling sequence is

$$\text{BernsteinSylvesterResultant}(f, g)$$

It computes the Sylvester resultant of F and G in Bernstein basis.

BernsteinSylvesterSres. The calling sequence is

$$\text{BernsteinSylvesterSres}(f, g, k)$$

where k is an integer such that $0 \le k < \min(\text{Dim}(f) - 1, \text{Dim}(g) - 1)$. The function computes the k-th Sylvester subresultant of F and G in Bernstein basis.

NewtonBezoutMatrix. The calling sequence is

$$\text{NewtonBezoutMatrix}(f, g, \lambda)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ. It computes the Bézout resultant matrix of F and G in the Newton basis N_λ.

NewtonBezoutResultant. The calling sequence is

$$\text{NewtonBezoutResultant}(f, g, \lambda)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ. It computes the Bézout resultant of f and g in the Newton basis N_λ.

NewtonBezoutSres. The calling sequence is

$$\text{NewtonBezoutSres}(f, g, \lambda, k)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ and k is an integer such that $0 \le k < \min(\text{Dim}(f) - 1, \text{Dim}(g) - 1)$. It computes the k-th Bézout subresultant of f and g in the Newton basis N_λ.

NewtonBarnettMatrix. The calling sequence is

$$\text{NewtonBarnettMatrix}(f, g, \lambda)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ. It computes the Barnett resultant matrix of f and g in the Newton basis N_λ.

NewtonBarnettResultant. The calling sequence is

$$\text{NewtonBarnettResultant}(f, g, \lambda)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ. It computes the Barnett resultant of F and G in the Newton basis N_λ.

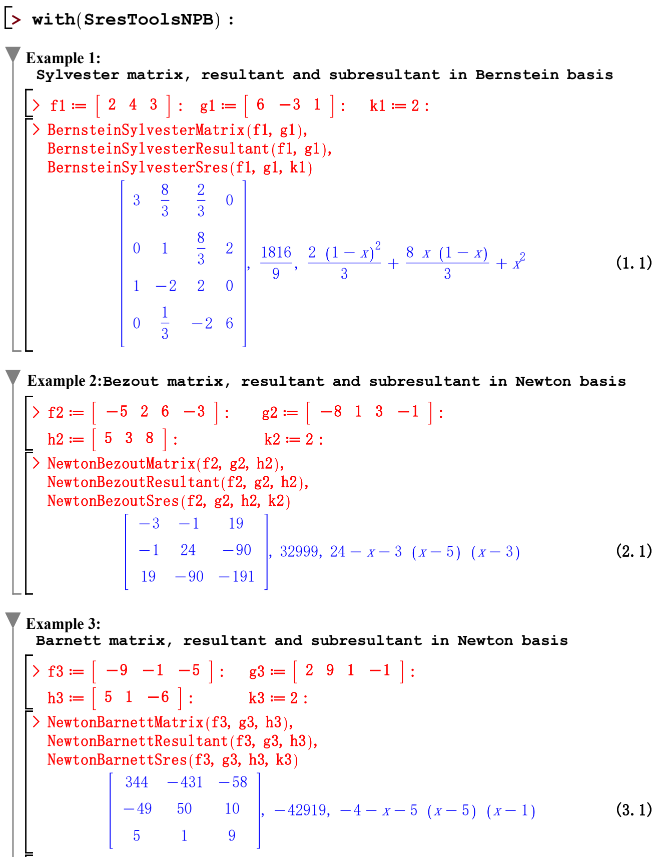
NewtonBarnettSres. The calling sequence is

$$\text{NewtonBarnettSres}(f, g, \lambda, k)$$

where $\lambda \in \mathcal{K}^{\max(m,n)}$ which represents the Newton basis obtained from λ and k is an integer such that $0 \le k < \min(\text{Dim}(f) - 1, \text{Dim}(g) - 1)$. It computes the k-th Barnett subresultant of F and G in the Newton basis N_λ.

The package SresToolsNPB is available at

https://github.com/JYangMATH/SresToolsNPB for download. In Figure 1 below, we provide an illustration on how to use functions in SresToolsNPB.



```
> with(SresToolsNPB) :
```

Example 1:
Sylvester matrix, resultant and subresultant in Bernstein basis

```
> f1 := [ 2  4  3 ] :  g1 := [ 6  -3  1 ] :  k1 := 2 :
> BernsteinSylvesterMatrix(f1, g1),
  BernsteinSylvesterResultant(f1, g1),
  BernsteinSylvesterSres(f1, g1, k1)
```

$$\begin{bmatrix} 3 & \frac{8}{3} & \frac{2}{3} & 0 \\ 0 & 1 & \frac{8}{3} & 2 \\ 1 & -2 & 2 & 0 \\ 0 & \frac{1}{3} & -2 & 6 \end{bmatrix}, \frac{1816}{9}, \frac{2\,(1-x)^2}{3} + \frac{8\,x\,(1-x)}{3} + x^2 \qquad (1.1)$$

Example 2: Bezout matrix, resultant and subresultant in Newton basis

```
> f2 := [ -5  2  6  -3 ] :    g2 := [ -8  1  3  -1 ] :
  h2 := [ 5  3  8 ] :           k2 := 2 :
> NewtonBezoutMatrix(f2, g2, h2),
  NewtonBezoutResultant(f2, g2, h2),
  NewtonBezoutSres(f2, g2, h2, k2)
```

$$\begin{bmatrix} -3 & -1 & 19 \\ -1 & 24 & -90 \\ 19 & -90 & -191 \end{bmatrix}, 32999, 24 - x - 3\,(x-5)\,(x-3) \qquad (2.1)$$

Example 3:
Barnett matrix, resultant and subresultant in Newton basis

```
> f3 := [ -9  -1  -5 ] :   g3 := [ 2  9  1  -1 ] :
  h3 := [ 5  1  -6 ] :         k3 := 2 :
> NewtonBarnettMatrix(f3, g3, h3),
  NewtonBarnettResultant(f3, g3, h3),
  NewtonBarnettSres(f3, g3, h3, k3)
```

$$\begin{bmatrix} 344 & -431 & -58 \\ -49 & 50 & 10 \\ 5 & 1 & 9 \end{bmatrix}, -42919, -4 - x - 5\,(x-5)\,(x-1) \qquad (3.1)$$

Fig. 1. Computing the resultant matrix/resultant/subresultants in non-power basis

5 Conclusion

This paper presents a Maple package SresToolsNPB for computing the resultants and subresultants of polynomials in non-power bases, with a focus on the Newton and Bernstein bases as two typical representatives. It is required that both the input and output polynomials are provided in the same basis. Compared with the basis-transformation-based methods, the tools provided by the package can compute the resultants and subresultants in non-power basis effectively. A natural question is: for other bases (e.g., Chebyshev basis), are there any basis-preserving methods for computing the resultants and subresultants? This question will be investigated in the future.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 12261010 and the Natural Science Foundation of Guangxi under Grant No. 2023GXNSFBA026019.

References

[1] A. Amiraslani, D. A. Aruliah, and R. M. Corless. The Rayleigh quotient iteration for generalized companion matrix pencils. *Preprint*, 2006.

[2] D. A. Aruliah, R. M. Corless, L. Gonzalez-Vega, and A. Shakoori. Geometric applications of the Bezout matrix in the Lagrange basis. In *Proceedings of the 2007 International Workshop on Symbolic-Numeric Computation*, SNC '07, pages 55–64, New York, NY, USA, 2007. Association for Computing Machinery.

[3] S. Barnett. Greatest common divisor of several polynomials. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 70, pages 263–268. Cambridge University Press, 1971.

[4] D. A Bini and L. Gemignani. Bernstein-Bezoutian matrices. *Theoretical Computer Science*, 315(2-3):319–333, 2004.

[5] A. Bostan, C. D'Andrea, T. Krick, A. Szanto, and M. Valdettaro. Subresultants in multiple roots: an extremal case. *Linear Algebra and its Applications*, 529:185–198, 2017.

[6] J. M. Carnicer and J. M. Pena. Shape preserving representations and optimality of the Bernstein basis. *Advances in Computational Mathematics*, 1(2):173–196, 1993.

[7] A. D. Chtcherba and D. Kapur. Constructing Sylvester-type resultant matrices using the Dixon formulation. *Journal of Symbolic Computation*, 38(1):777–814, 2004.

[8] G. E. Collins. Subresultants and reduced polynomial remainder sequences. *Journal of the ACM (JACM)*, 14(1):128–142, 1967.

[9] D. A. Cox and C. D'Andrea. Subresultants and the Shape Lemma. *Mathematics of Computation*, 92:2355–2379, 2021.

[10] J. Delgado and J. M. Peña. A shape preserving representation with an evaluation algorithm of linear complexity. *Computer Aided Geometric Design*, 20(1):1–10, 2003.

[11] G. M. Diaz-Toca and L. González-Vega. Various new expressions for subresultants and their applications. *Applicable Algebra in Engineering, Communication and Computing*, 15(3):233–266, 2004.

[12] R. T Farouki and V. T. Rajan. On the numerical condition of polynomials in Bernstein form. *Computer Aided Geometric Design*, 4(3):191–216, 1987.

[13] T. N. T. Goodman and H. B. Said. Shape preserving properties of the generalised Ball basis. *Computer Aided Geometric Design*, 8(2):115–121, 1991.

[14] C.-J. Ho. *Topics in Algebraic Computing: Subresultants, GCD, Factoring and Primary Ideal Decomposition.* PhD thesis, USA, 1989.

[15] H. Hong and J. Yang. A condition for multiplicity structure of univariate polynomials. *Journal of Symbolic Computation*, 104:523–538, 2021.

[16] H. Hong and J. Yang. Subresultant of several univariate polynomials. *arXiv:2112.15370*, 2021.

[17] H. Hong and J. Yang. Computing greatest common divisor of several parametric univariate polynomials via generalized subresultant polynomials. *arXiv:2401.00408*, 2023.

[18] A. Lascoux and P. Pragacz. Double Sylvester sums for subresultants and multi-Schur functions. *Journal of Symbolic Computation*, 35(6):689–710, 2003.

[19] Y. B. Li. A new approach for constructing subresultants. *Applied Mathematics and Computation*, 183(1):471–476, 2006.

[20] A. Marco and J.-J. Martínez. Bernstein–Bezoutian matrices and curve implicitization. *Theoretical Computer Science*, 377(1-3):65–72, 2007.

[21] J. Sylvester. On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraic common measure. *Phil. Trans*, 143:407–548, 1853.

[22] M. Tan. Computation and Application of generalized subresultants for Bernstein polynomials. Master's thesis, GuangXi University for Nationalities, 2024.

[23] A. Terui. Recursive polynomial remainder sequence and its subresultants. *Journal of Algebra*, 320(2):633–659, 2008.

[24] W. D. Wang and J. Yang. Generalized companion subresultants of several univariate polynomials in Newton basis. *arXiv:2212.03422*, 2022.

[25] J. R. Winkler. A resultant matrix for scaled Bernstein polynomials. *Linear Algebra and its Applications*, 319(1):179–191, 2000.

[26] J. R. Winkler. A companion matrix resultant for Bernstein polynomials. *Linear Algebra and its Applications*, 362:153–175, 2003.

[27] J. Yang and W. Yang. A Basis-preserving Algorithm for Computing the Bezout matrix of Newton polynomials. *arXiv:2404.18117.*

[28] J. Yang and W. Yang. Bézout subresultants for univariate polynomials in general basis. *arXiv:2305.03906*, 2023.

[29] Z.H. Yang. Polynomial Bezoutian matrix with respect to a general basis. *Linear Algebra and its Applications*, 331(1):165–179, 2001.