SUBRESULTANTS REVISITED

JOACHIM VON ZUR GATHEN and THOMAS LÜCKING

Fachbereich Mathematik-Informatik, Universität Paderborn,
33095 Paderborn, Germany, {gathen|luck}@upb.de

Abstract

Subresultants and polynomial remainder sequences are an important tool in polynomial computer algebra. In this survey, we sketch the history, discuss the various notions, and report on implementations.

1 Introduction

1.1 Historical context

The *Euclidean Algorithm* was first documented by Euclid (c. 320–275 BC). According to Knuth (1981), p. 318, *"we might call it the granddaddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day."* It executes division with remainder repeatedly until the remainder becomes zero. With inputs 13 and 9 it performs the following:

$$\begin{aligned} 13 &= 1 \cdot 9 + 4, \\ 9 &= 2 \cdot 4 + \textcircled{1}, \\ 4 &= 4 \cdot 1 + 0. \end{aligned}$$

This allows us to compute the *greatest common divisor (gcd)* of two integers as the last non-vanishing remainder. In the example, the gcd of 13 and 9 is computed as 1.

When the concept of polynomials started to evolve, researchers were interested in finding the common roots of two polynomials f and g. Simon Stevin was the first to apply the Euclidean Algorithm to polynomials, in 1585. In 1707, Newton considered this problem and showed that the method always works in $\mathbb{Q}[x]$.

$$\begin{aligned} x^3 + 2x^2 - x - 2 &= (\frac{1}{2}x + \frac{3}{2})(2x^2 - 2x - 4) + \boxed{4x + 4} \\ 2x^2 - 2x - 4 &= (\frac{1}{2}x - 1)(4x + 4) + 0. \end{aligned}$$

In this example $f = x^3 + 2x^2 - x - 2$ and $g = 2x^2 - 2x - 4$ have a greatest common divisor $4x+4$, and therefore the only common root is -1. In a certain sense the Euclidean Algorithm computes all common roots (in an algebraically closed extension such as $\mathbb{C}$). If we only want to know whether f and g have at least *one* common root, then still the whole Euclidean Algorithm has to be executed. Thus a goal was to find an indicator for common roots without using any division with remainder.

The key to success was found in 1748 by Euler, and later by Bézout. They were looking for a *resultant* of f and g as a polynomial in the coefficients of f and g that vanishes if and only if f and g have a common root. In his 1764 paper, Bézout coined the word *équation résultante* and was the first to find a matrix whose determinant is the resultant. The entries of this *Bézout matrix* are bilinear functions of the coefficients of f and g. Today one often uses the matrix discovered by Sylvester in 1840, known as the *Sylvester matrix*. Its entries are simply coefficients of the polynomials f and g. Sylvester generalized his definition and introduced what we now call *subresultants* as determinants of certain submatrices of the Sylvester matrix. They are nonzero if and only if the corresponding degree appears as a degree of a remainder of the Euclidean Algorithm.

These indicators, in particular the resultant, also work for polynomials in $\mathbb{Z}[x]$. But it is in general not possible to apply the Euclidean Algorithm to f and g in $\mathbb{Z}[x]$ without leaving $\mathbb{Z}[x]$, as illustrated in the example above, since division with remainder is not always defined in $\mathbb{Z}[x]$, although the gcd exists. In the example it is $x + 1$.

However, in 1836 Jacobi found a way out. He introduced *pseudo-division*: he multiplied f with a certain power of the leading coefficient of g before performing the division with remainder. This is always possible in $\mathbb{Z}[x]$. So using pseudo-division instead of division with remainder in every step in the Euclidean Algorithm yields an algorithm with all intermediate results in $\mathbb{Z}[x]$.

About 40 years later Kronecker did research on the *Laurent series* in x^{-1} of g/f for two polynomials f and g. He considered the determinants of a matrix whose entries are the coefficients of the Laurent series of g/f. He obtained

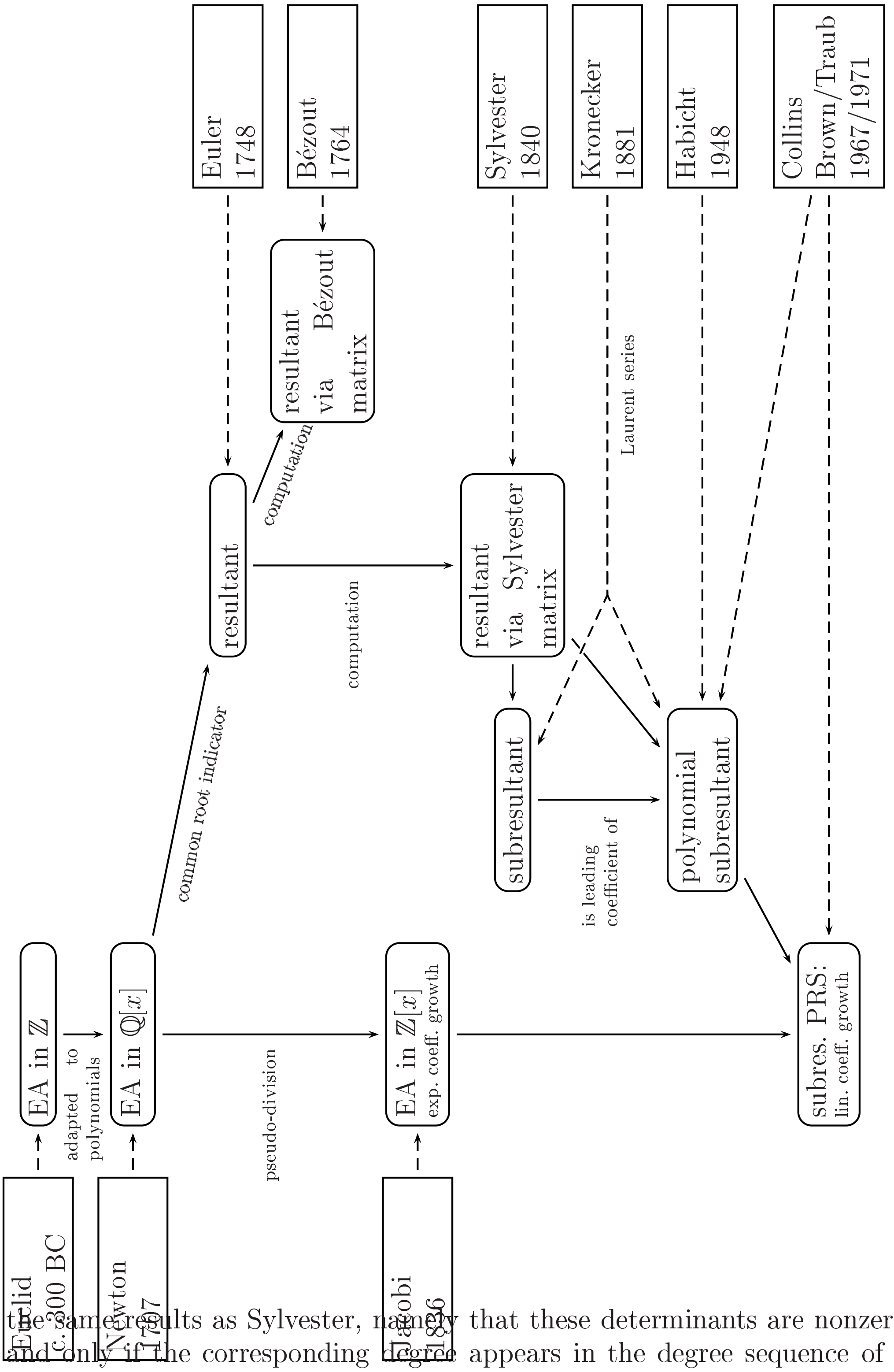

Figure 1: Historical evolution

the same results as Sylvester, namely that these determinants are nonzero if and only if the corresponding degree appears in the degree sequence of the Euclidean Algorithm. Furthermore Kronecker gave a direct way to compute low degree polynomials s, t and r with $sf + tg = r$ via determinants of matrices derived again from the Laurant series of g/f, and showed that these polynomials are essentially the only ones. He also proved that the polynomial

r, if nonzero, agrees with a remainder in the Euclidean Algorithm, up to a constant multiple. This was the first occurrence of *polynomial subresultants*.

Starting in the 1960s, people built early computer algebra systems like PM and ALPAK that made it possible to perform more and more complicated algorithms faster and faster. However, using *pseudo-division* in every step of the Euclidean Algorithm causes *exponential* coefficient growth. This was suspected in the late 1960's. Collins (1967), p. 139, explains that the ith intermediate coefficients are approximately longer by a factor of $(1+\sqrt{2})^i$ than the input coefficients, and writes: *"Thus, for the Euclidean algorithm, the lengths of the coefficients increases exponentially."* In Brown & Traub (1971) we find: *"Although the Euclidean PRS algorithm is easy to state, it is thoroughly impractical since the coefficients grow exponentially."* An exponential *upper* bound is in Knuth (1993)???, equation (27) in 4.6.1: *"Thus the upper bound* [...] *would be approximately $N^{0.5(2.414)^n}$, and experiments show that the simple algorithm does in fact have this behavior; the number of digits in the coefficients grows exponentially at each step!"*. An exponential *lower* bound is in **?**, 3.3.3, and we provide in Theorem 7.3 a more precise lower bound that essentially matches Collin's and Knuth's upper bound.

One way out of this exponential trap is to make every intermediate result *primitive*, that is, to divide the remainders by the greatest common divisors of their coefficients, the so-called *content*. However, computing the content seemed to be very expensive, especially for multivariate polynomials. So the scientists tried to find divisors of the content without using any gcd computation. Around 1970, first Collins and then Brown & Traub reinvented the *polynomial subresultants* as determinants of a certain variant of the Sylvester matrix. Habicht had also defined them independently in 1948. Collins and Brown & Traub showed that they agree with the remainders of the Euclidean Algorithm up to constant factors. They gave simple formulas to compute these factors and introduced the concept of *polynomial remainder sequences (PRS)*, generalizing the concept of Jacobi. The final result is the *subresultant PRS* that features linear coefficient growth with intermediate results in $\mathbb{Z}[x]$.

Since then two further concepts have come up. On the one hand the *fast EEA* allows to compute an arbitrary intermediate line in the Euclidean Scheme directly. Using the fast $O(n \log n \log \log n)$ multiplication algorithm of Schönhage and Strassen, we can reduce the time to compute the gcd from $O(n^2)$ to $O(n \log^2 n \log \log n)$ field operations (see Strassen (1983)). On the other hand, the *modular EEA*, also introduced by Collins, is very efficient. These two topics are not considered in this survey; for further information we refer to von zur Gathen & Gerhard (1999), Chapters 6 and 11. Figure 1 illustrates the historical evolution.

1.2 Outline

After introducing the notation and some well-known facts in Section 2, we start with an overview and comparison of various definitions of subresultants in Section 3. Mulders (1997) describes an error in software implementations of an integration algorithm which was due to the confusion caused by the these various definitions. It turns out that there are essentially two different notions: the scalar and the polynomial subresultants. We determine how they are related to each other. In the remainder of this work we will mainly consider the scalar subresultants.

In Section 4 we give a formal definition of polynomial remainder sequences and derive the most famous ones as special cases of our general notion. The relation between polynomial remainder sequences and subresultants is exhibited in the Fundamental Theorem 5.3 in Section 5. It unifies many results in the literature on various types of PRS. In Section 6 we apply it to the various types of polynomial remainder sequences. This yields a collection of results from Collins (1966, 1967, 1971, 1973), Brown (1971, 1978), Brown & Traub (1971), Lickteig & Roy (1997) and von zur Gathen & Gerhard (1999), often with simplification in the statements and proofs.

Finally we report on implementations of the various polynomial remainder sequences. We analyze the coefficient growth and the running time of the various PRS in Section 7, and compare their running times in Section 8. It turns out that computing the content is quite fast for random inputs, and that the primitive PRS behaves much better than expected.

However, this is not meant to suggest these algorithms as a practical alternative. In most situations, the modular algorithms will outperform the PRS discussed in this survey.

All examples in this paper are from $\mathbb{Z}[x]$, but the methods apply equally well to multivariate polyomials, and are even more useful there. We choose those examples because they are more concise to specify.

1.3 Acknowledgements

We thank Johannes Blömer and Erich Kaltofen for pointers to the literature. This work is part of the second author's Diplomarbeit (Lücking (2000)), and an Extended Abstract appeared at Latin '00 (von zur Gathen & Lücking (2000)).

2 Foundations

We refer to Hungerford (1990) and von zur Gathen & Gerhard (1999), Sections 2.2 and 25.5, for the notation and fundamental facts about greatest common divisors and determinants.

2.1 Polynomials

Let R be a ring. In what follows, this always means a commutative ring with 1. A basic tool in computer algebra is *division with remainder*. For given polynomials f and g in $R[x]$ the task is to find polynomials q and r in $R[x]$ with

$$f = qg + r \text{ and } \deg r < \deg g. \tag{2.1}$$

Unfortunately such q and r do not always exist.

EXAMPLE 2.2. It is not possible to divide x^2 by $2x+3$ with remainder in $\mathbb{Z}[x]$ because $x^2 = (ux + v)(2x + 3) + r$ with $u, v, r \in \mathbb{Q}$ has the unique solution $u = 1/2$, $v = 0$ and $r = -3/2$, which is not over $\mathbb{Z}$. ♢

If defined and unique we call $q = f \operatorname{quo} g$ the *quotient* and $r = f \operatorname{rem} g$ the *remainder*. A ring with a length function (like the degree of polynomials) and where division with remainder is always defined is a *Euclidean domain*. $R[x]$ is a Euclidean domain if and only if R is a field. A solution of (2.1) is not necessarily unique if the leading coefficient $\operatorname{lc}(g)$ of g is a zero divisor.

EXAMPLE 2.3. Let $R = \mathbb{Z}_8$ and consider $f = 4x^2 + 2x$ and $g = 2x + 1$. With

$$\begin{aligned} q_1 &= 2x, & r_1 &= 0, \\ q_2 &= 2x + 4, & r_2 &= 4, \end{aligned}$$

we have two distinct solutions (q_1, r_1) and (q_2, r_2) of (2.1). ♢

A way to get solutions for all commutative rings is the *general pseudo-division* which allows multiplication of f by a ring element α:

$$\alpha f = qg + r, \ \deg r < \deg g. \tag{2.4}$$

If $n = \deg f$, $m = \deg g$, and $\alpha = \operatorname{lc}(g)^{n-m+1}$, then this is the *(classical) pseudo-division* as proposed in Jacobi (1836). If $\operatorname{lc}(g)$ is not a zero divisor, then (2.4) with $\alpha = \operatorname{lc}(g)^{n-m+1}$ always has a unique solution in $R[x]$. We call $q = f \operatorname{pquo} g$ the *pseudo-quotient* and $r = f \operatorname{prem} g$ the *pseudo-remainder*.

EXAMPLE 2.2 CONTINUED. For x^2 and $2x+3$ we get the pseudo-division

$$2^2 \cdot x^2 = (2x-3)(2x+3)+9$$

A simple computation shows that we cannot choose $\alpha = 2$. $\Diamond$

LEMMA 2.5. *Let $f, g \in R[x]$ have degrees n, m, respectively, and $g \neq 0$.*

(i) Pseudo-division always yields a solution of (2.4) in $R[x]$.
(ii) If $\mathrm{lc}(g)$ is not a zero divisor, then any solution of (2.4) has $\deg q = n-m$.
(iii) The solution (q, r) of (2.4) is uniquely determined if and only if $\mathrm{lc}(g)$ is not a zero-divisor.

PROOF. (i) We prove the claim by induction on $n = \deg f$. For $n < m = \deg g$ we have the solution $q = 0$ and $r = f$. Now assume that $n \geq m$, and let $f^* = g_m f - f_n x^{n-m} g$ where f_n and g_m are the leading coefficients of f and g, respectively. Then

$$g_m^{n-m+1} f = (f_n g_m^{n-m} x^{n-m}) g + g_m^{n-m} f^*.$$

Now $\deg f^* < \deg f$, and by the induction hypothesis there exist q^* and r^* in $R[x]$ with

$$g_m^{(n-1)-m+1} f^* = q^* g + r^* \text{ and } \deg r^* < \deg g.$$

Therefore $q = f_n g_m^{n-m} x^{n-m} + q^*$ and $r = r^*$ give a solution of (2.4).

(ii) Let (q, r) be a solution of (2.4). Since $\deg r < \deg g$ and $\mathrm{lc}(g)$ is not a zero-divisor, we have

$$n = \deg f = \deg qg = \deg q + \deg g = \deg q + m.$$

(iii) "$\Rightarrow$": Suppose $\mathrm{lc}(g) = g_m$ is not a zero divisor, and that $q_1, r_1, q_2, r_2 \in R[x]$ are such that

$$\alpha f = q_1 g + r_1 = q_2 g + r_2.$$

We claim that $(q_1, r_1) = (q_2, r_2)$. Now

$$(q_1 - q_2) g = r_2 - r_1. \tag{2.6}$$

Since $q_1 = q_2$ implies $r_1 = r_2$, we may assume that $q_1 \neq q_2$. Now we write $g = g_m x^m + g^*$ and $q_1 - q_2 = \gamma x^\ell + q^*$ where $\deg g^* < m$, $\ell \geq 0$ and $\gamma = \mathrm{lc}(q_1 - q_2) \neq 0$, and note that $g_m \gamma \neq 0$. Therefore $\deg((q_1 - q_2)g) = m + \ell$ and

$$\deg((q_1 - q_2)g) \geq m > \deg(r_2 - r_1).$$

This contradiction to (2.6) proves our claim.

"$\Leftarrow$": We assume $\mathrm{lc}(g) = g_m$ to be a zero divisor, and $\gamma \in R$ to be nonzero with $g_m \gamma = 0$, and let (q_1, r_1) be a solution of (2.4). Then $q = q_1 + \gamma$ and $r = r_1 - \gamma g$ yield

$$qg + r = (q_1 + \gamma)g - \gamma g + r_1 = q_1 g + r_1 = \alpha f$$

with $\deg r < \deg g$. Thus (q, r) is another solution of (2.4). □

2.2 Extended Euclidean Algorithm (EEA)

We use the notation for the Extended Euclidean Algorithm (EEA) from von zur Gathen & Gerhard (1999), Chapter 3, with remainders r_i, quotients q_i and Bézout coefficients s_i and t_i, for $0 \leq i \leq \ell$.

EXAMPLE 2.7. The Extended Euclidean Scheme of the two polynomials $f = x^3 + 6x^2 + 11x + 6$ and $g = x^2 - 3x + 2 \in \mathbb{Q}[x]$ is:

i	r_i	q_i	s_i	t_i
0	$x^3 + 6x^2 + 11x + 6$		1	0
1	$x^2 - 3x + 2$	$x + 9$	0	1
2	$36x - 12$	$\frac{1}{36}x - \frac{2}{27}$	1	$-x - 9$
3	$\frac{10}{9}$	$\frac{162}{5}x - \frac{54}{5}$	$-\frac{1}{36}x + \frac{2}{27}$	$\frac{1}{36}x^2 + \frac{19}{108}x + \frac{1}{3}$
4	0		$\frac{9}{10}x^2 - \frac{27}{10}x + \frac{9}{5}$	$-\frac{9}{10}x^3 - \frac{27}{5}x^2 - \frac{99}{10}x - \frac{27}{5}$

So the Euclidean length of (f, g) is $\ell = 3$. Since $r_3 = \frac{10}{9} \in \mathbb{Q}$ is a unit, the gcd of f and g is 1. ◊

In general, $(\deg r_0, \dots, \deg r_\ell)$ is the *degree sequence*; in the example it is (6,4,2,1,0).

We have $\deg_{r_i} + \deg_{t_i} < \deg f$, and $r_i = s_i f + t_i g$ is a "small" linear combination of f and g with "small" coefficients. The following theorem, essentially due to **?**, says that the entries of the EEA are essentially the only way to get such a small linear combination; see Lemma 5.15 from von zur Gathen & Gerhard (1999).

UNIQUE REPRESENTATION THEOREM 2.8. *Let F be a field, $f, g, r, s, t \in F[x]$ with $r = sf + tg$ and $t \neq 0$, and suppose that*

$$\deg r + \deg t < n = \deg f.$$

Moreover, let r_i, s_i, t_i for $0 \leq i \leq \ell + 1$ be the rows of the Extended Euclidean Algorithm for the pair (f, g). If we define $1 \leq j \leq \ell + 1$ by

$$\deg r_j \leq \deg r < \deg r_{j-1},$$

then there exists a nonzero $\alpha \in F[x]$ such that

$$r = \alpha r_j,\ s = \alpha s_j,\ t = \alpha t_j.$$

3 Various notions of subresultants

Throughout the following we have a commutative ring R and two polynomials

$$f = \sum_{0 \leq j \leq n} f_j x^j, \quad g = \sum_{0 \leq j \leq m} g_j x^j \in R[x]$$

of degrees n, m, respectively.

3.1 The Sylvester matrix

The various definitions of the subresultant are based on the *Sylvester matrix*. We first take a look at the historical motivation for this special matrix. Our goal is to decide whether two polynomials f and g have a nontrivial common factor. To find an answer to this question, Euler (1748) and Bézout (1764) introduced the *(classical) resultant* that vanishes if (and only if) this is true. Bézout also succeeded in finding a matrix whose determinant is equal to the resultant, today called the *Bézout matrix*, but we will follow the elegant derivation in Sylvester (1840). The two linear equations

$$\begin{aligned} f_n x_n \ + \ f_{n-1} x_{n-1} \ + \cdots + \ f_1 x_1 \ + \ f_0 x_0 &= 0, \\ g_m x_m \ + \ g_{m-1} x_{m-1} \ + \cdots + \ g_1 x_1 \ + \ g_0 x_0 &= 0 \end{aligned}$$

in the indeterminates $x_0, \ldots, x_n$ are satisfied if $x_j = \alpha^j$ for all j, where α is a common root of f and g. For $n > 1$ there are many more solutions of these two linear equations in many variables, but Sylvester eliminates them by adding the $(m-1)+(n-1)$ linear equations that correspond to the following additional conditions:

$$\begin{aligned} xf(x) = 0 \ , \ldots , \ x^{m-1} f(x) &= 0, \\ xg(x) = 0 \ , \ldots , \ x^{n-1} g(x) &= 0. \end{aligned}$$

These equations give a total of $n+m$ linear relations among the variables $x_{m+n-1}, \cdots, x_0$:

$$\begin{array}{rcccccccl}
f_n x_{m+n-1} & + & \cdots & + & f_0 x_{m-1} & & & & = 0, \\
 & & \vdots & & & & & & \\
 & & f_n x_n & + & f_{n-1} x_{n-1} & + & \cdots & + f_0 x_0 & = 0, \\
g_m x_{m+n-1} & + & \cdots & + & g_0 x_{n-1} & & & & = 0, \\
 & & \vdots & & & & & & \\
 & & g_m x_m & + & g_{m-1} x_{m-1} & + & \cdots & + g_0 x_0 & = 0.
\end{array}$$

Clearly $x_j = \alpha^j$ gives a solution for any common root α of f and g, but the point is that (essentially) the converse also holds: a solution of the linear equations gives a common root (or factor). The $(n+m) \times (n+m)$ matrix, consisting of coefficients of f and g, that belongs to this system of linear equations is often called *Sylvester matrix.* We follow von zur Gathen & Gerhard (1999), Section 6.3, p. 144, and take its transpose.

Definition 3.1. *The* $(n+m) \times (n+m)$ *matrix*

$$\mathrm{Syl}(f,g) = \left(\begin{array}{cccccccccc}
f_n & & & & g_m & & & & & \\
f_{n-1} & f_n & & & g_{m-1} & g_m & & & & \\
\vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & & & \\
\vdots & \vdots & & f_n & g_1 & \vdots & & \ddots & & \\
\vdots & \vdots & & f_{n-1} & g_0 & \vdots & & & \ddots & \\
\vdots & \vdots & & \vdots & & g_0 & & & & g_m \\
f_0 & \vdots & & \vdots & & & \ddots & & & \vdots \\
 & f_0 & & \vdots & & & & \ddots & & \vdots \\
 & & \ddots & \vdots & & & & & \ddots & \vdots \\
 & & & f_0 & & & & & & g_0
\end{array}\right)$$
$$\underbrace{\hphantom{f_{n-1}\quad f_n\quad \ddots\quad f_{n-1}}}_{m}\underbrace{\hphantom{g_{m-1}\quad g_m\quad \ddots\quad \ddots\quad \ddots\quad g_m}}_{n}$$

is the Sylvester matrix of f and g.

Remark 3.2. *Multiplying the* $(n+m-j)$*th row by* x^j *and adding it to the* *last row for $1 \leq j < n+m$, we get the $(n+m)\times(n+m)$ matrix*

$$\mathrm{Syl}^*(f,g) = \begin{array}{c}\begin{pmatrix}
f_n & & & & g_m & & & & & \\
f_{n-1} & f_n & & & g_{m-1} & g_m & & & & \\
\vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & & & \\
\vdots & \vdots & & f_n & g_1 & \vdots & & \ddots & & \\
\vdots & \vdots & & f_{n-1} & g_0 & \vdots & & & \ddots & \\
\vdots & \vdots & & \vdots & & g_0 & & & & g_m \\
f_0 & \vdots & & \vdots & & & \ddots & & & \vdots \\
 & f_0 & & \vdots & & & & \ddots & & \vdots \\
 & & \ddots & f_1 & & & & & \ddots & g_1 \\
x^{m-1}f(x) & \cdots & \cdots & f(x) & x^{n-1}g(x) & \cdots & \cdots & \cdots & \cdots & g(x)
\end{pmatrix} \\ \underbrace{\hphantom{x^{m-1}f(x)\cdots\cdots f(x)}}_{m}\underbrace{\hphantom{x^{n-1}g(x)\cdots\cdots\cdots\cdots g(x)}}_{n}\end{array}.$$

Thus $\det(\mathrm{Syl}(f,g)) = \det(\mathrm{Syl}^*(f,g))$.

More details on resultants can be found in Biermann (1891), Gordan (1885) and Haskell (1892). Computations for both the univariate and multivariate case are discussed in Collins (1971).

There is also considerable recent literature on the subject: ?

Landau and Zippel on algebraic decomposition, ? on multivariate and algebraic generalizations.

3.2 The scalar subresultant

We are interested in determining which degrees appear in the degree sequence of the Extended Euclidean Algorithm. Scalar subresultants provide a solution.

Definition 3.3. *The determinant $\sigma_k(f,g) \in R$ of the $(m+n-2k)\times(m+$*

$n - 2k$) matrix

$$S_k(f,g) = \begin{array}{c} \left(\begin{array}{cccccccc} f_n & & & & g_m & & & \\ f_{n-1} & f_n & & & g_{m-1} & g_m & & \\ \vdots & & \ddots & & \vdots & & \ddots & \\ f_{n-m+k+1} & \cdots & \cdots & f_n & g_{k+1} & \cdots & g_m & \\ \vdots & & & \vdots & \vdots & & & \ddots \\ f_{k+1} & \cdots & \cdots & f_m & g_{m-n+k+1} & \cdots & \cdots & g_m \\ \vdots & & & \vdots & \vdots & & & \vdots \\ \vdots & & & \vdots & \vdots & & & \vdots \\ f_{2k-m+1} & \cdots & \cdots & f_k & g_{2k-n+1} & \cdots & \cdots & g_k \end{array} \right) \\ \underbrace{\qquad\qquad\qquad\qquad}_{m-k}\underbrace{\qquad\qquad\qquad\qquad\qquad}_{n-k} \end{array}$$

is called the k**th (scalar) subresultant of** f **and** g. *By convention an f_j or g_j with $j < 0$ is zero. If f and g are clear from the context, then we write S_k and σ_k instead of $S_k(f,g)$ and $\sigma_k(f,g)$.*

Sylvester (1840) already contains an explicit description of the (scalar) subresultants. In Habicht (1948), p. 104, σ_k is called *Nebenresultante (minor resultant)* for polynomials f and g of degrees n and $n-1$. The definition is also in von zur Gathen (1984) and is used in von zur Gathen & Gerhard (1999), Section 6.10.

Remark 3.4.

(i) $S_0 = \mathrm{Syl}(f,g)$ *and therefore* $\sigma_0 = \det(S_0)$ *is the* resultant.
(ii) $\sigma_m = g_m^{n-m}$.
(iii) *S_k is the matrix obtained from the Sylvester matrix by deleting the last $2k$ rows and the last k columns with coefficients of f, and the last k columns with coefficients of g.*
(iv) *S_k is a submatrix of S_i if $k \geq i$.*

3.3 The polynomial subresultant

Two slightly different descriptions of polynomial subresultants are in the literature. The first one is from Collins (1967), p. 129, and the second one is from Brown & Traub (1971), p. 507 and also in Zippel (1993), Chapter 9.3, p. 150. They yield polynomials that are related to the intermediate results in the Extended Euclidean Algorithm. We compare the two definitions and show their relation to scalar subresultants. In the remainder of this text we then focus on scalar subresultants.

Definition 3.5. *Let $M_{ik} = M_{ik}(f, g)$ be the $(n + m - 2k) \times (n + m - 2k)$ submatrix of* $\mathrm{Syl}(f, g)$ *obtained by deleting the last k of the m columns of coefficients of f, the last k of the n columns of coefficients of g and the last $2k + 1$ rows except row $(n + m - i - k)$, for $0 \leq k \leq m$ and $0 \leq i \leq n$:*

$$M_{ik} = \left(\begin{array}{ccccccccc} f_n & & & & g_m & & & & \\ f_{n-1} & f_n & & & g_{m-1} & g_m & & & \\ \vdots & & \ddots & & \vdots & & \ddots & & \\ \vdots & & & f_n & \vdots & & & \ddots & \\ \vdots & & & \vdots & \vdots & & & & g_m \\ \vdots & & & \vdots & \vdots & & & & \vdots \\ f_{2k-m+2} & & & f_{k+1} & g_{2k-n+2} & & & & g_{k+1} \\ f_{i+k-m+1} & \cdots & \cdots & f_i & g_{i+k-n+1} & \cdots & \cdots & \cdots & g_i \end{array} \right).$$

*The polynomial $R_k(f, g) = \sum_{0 \leq i \leq n} \det(M_{ik}) x^i \in R[x]$ is called the k***th polynomial subresultant of** f **and** g.

In fact, Collins (1967) considered the transposed matrices. If f and g are clear from the context, then we write R_k instead of $R_k(f, g)$. Note that $\det(M_{ik}) = 0$ if $i > k$, since then the last row of M_{ik} is identical to the $(n + m - i - k)$th row. Thus $R_k = \sum_{0 \leq i \leq k} \det(M_{ik}) x^i$.

Remark 3.6.

(i) $M_{00} = \mathrm{Syl}(f, g)$ *and therefore* $R_0 = \det(M_{00})$ *is the resultant.*

(ii) Remark 3.4(i) implies that $\sigma_0 = R_0$.

Definition 3.7. *We consider the determinant* $Z_k(f, g) = \det(M_k^*) \in R[x]$ *of* *the* $(n+m-2k)\times(n+m-2k)$ *matrix*

$$M_k^* = \begin{pmatrix} f_n & & & g_m & & & \\ f_{n-1} & f_n & & g_{m-1} & g_m & & \\ \vdots & \ddots & & \vdots & \ddots & & \\ \vdots & & f_n & \vdots & & \ddots & \\ \vdots & & \vdots & \vdots & & & g_m \\ \vdots & & \vdots & \vdots & & & \vdots \\ f_{2k-m+2} & & f_{k+1} & g_{2k-n+2} & & & g_{k+1} \\ x^{m-k-1}f(x) & \cdots \; \cdots & f(x) & x^{n-k-1}g(x) & \cdots & \cdots \; \cdots & g(x) \end{pmatrix}.$$

If f and g are clear from the context, then we write Z_k for short instead of $Z_k(f,g)$. We note that M_k^* is a submatrix of $\mathrm{Syl}^*(f,g)$.

Table 1 gives an overview of the literature concerning these notions. Of course, there is a much larger body of work about the special case of the resultant, which we do not quote here.

3.4 Comparison of the various definitions

As in Brown & Traub (1971), p. 508, and Geddes *et al.* (1992), Section 7.3, p. 290, we first prove the following theorem which shows that the definitions in Collins (1967) and Brown & Traub (1971) describe the same polynomial.

Theorem 3.8.

(i) If $\sigma_k \neq 0$*, then* σ_k *is the leading coefficient of* R_k*. Otherwise,* $\deg R_k < k$*.*
(ii) $R_k = Z_k$*.*

Proof. (i) Since the coefficient of x^k in R_k is $\det(M_{kk}) = \det(S_k) = \sigma_k$, the first claim follows.
(ii) By linearity of the determinant, the claim follows from

$$\sum_{0\le i\le n} x^i (f_{i+k-m+1}, \ldots, f_i, g_{i+k-n+1}, \ldots, g_i)^T$$
$$= (x^{m-k-1}f(x), \ldots, f(x), x^{n-k-1}g(x), \ldots, g(x))^T. \qquad \Box$$

Definition	**Authors**
$\sigma_k(f,g) = \det(S_k) \in R$	Sylvester (1840) Habicht (1948) von zur Gathen (1984) Uteshev & Cherkasov (1998) von zur Gathen & Gerhard (1999)
$R_k(f,g) = \sum_{0 \le i \le n} \det(M_{ik})x^i$ $\parallel$ $Z_k(f,g) = \det(M_k^*) \in R[x]$	Collins (1967) Loos (1982) Geddes *et al.* (1992) Winkler (1996) Brown & Traub (1971) Zippel (1993) Lickteig & Roy (1997) Reischert (1997)

Table 1
The various subresultants

Remark 3.9. *Laplace expansion of Z_k along the last column of M_k^* yields two polynomials $s,t \in R[x]$ with $\deg s < m - k$, $\deg t < n - k$ and $sf + tg = Z_k = R_k$. This observation is due to Brown & Traub (1971), p. 507/508, see also Zippel (1993), Chapter 9.3, p. 150.*

The essential property of the subresultants is that they characterize the degree sequence; for a proof, see e.g. von zur Gathen & Gerhard (1999), Section 6.10.

Theorem 3.10. *Let f and g be polynomials over a field F of degrees $n_0 \ge n_1 > 0$, respectively, let $n_i = \deg r_i$ for $0 \le i \le \ell$ be the degrees in the Euclidean Scheme, and let $0 \le k < n_1$. Then*

$$\sigma_k \ne 0 \iff \exists i \le \ell \quad k = n_i.$$

Proposition 3.11. *Let F be a field, f and g in $F[x]$ be polynomials of degree $n \ge m > 0$, respectively, and let r_i, s_i and t_i be the entries in the ith row of the Extended Euclidean Scheme, for $0 \le i \le \ell$. Moreover, let $\rho_i = \mathrm{lc}(r_i)$ and $n_i = \deg r_i$ for all i. Then*

$$\frac{\sigma_{n_i}}{\rho_i} \cdot r_i = R_{n_i} \text{ for } 2 \le i \le \ell.$$

Proof. Let $2 \le i \le \ell$. Remark 3.9 shows that there exist polynomials s and t of degrees less than $m - n_i$ and $n - n_i$, respectively, with

$$sf + tg = R_{n_i}.$$

Thus

$$\deg R_{n_i} + \deg t \le n_i + n - n_i - 1 < n.$$

By Theorem 3.10 we know that the leading coefficient σ_{n_i} of R_{n_i} is nonzero. Since F is a field and $\deg R_{n_i} = n_i < n = \deg f$ we have $t \ne 0$. Hence, by the Unique Representation Theorem 2.8, there exists an $\alpha \in F[x]$ with

$$s = \alpha s_i,\ t = \alpha t_i,\ R_{n_i} = \alpha r_i,\ (\alpha s_i)f + (\alpha t_i)g = \alpha r_i = R_{n_i}.$$

Furthermore, $n_i = \deg r_i = \deg R_{n_i}$. Comparing leading coefficients we find

$$\alpha = \frac{\sigma_{n_i}}{\rho_i}. \qquad \square$$

Remark 3.12. *Let f and g be polynomials over an integral domain R, let F be the field of fractions of R, and consider the Extended Euclidean Scheme of f and g in $F[x]$. Then the scalar and the polynomial subresultants are in R and $R[x]$, respectively, and Proposition 3.11 also holds:*

$$\frac{\sigma_{n_i}}{\rho_i} \cdot r_i = R_{n_i} \in R[x].$$

Note that r_i is not necessarily in $R[x]$, and ρ_i not necessarily in R.

A careful reading of the proof of Proposition 3.11 also shows the relation between s, t from Remark 3.9 and the entries of the Extended Euclidean Scheme.

Remark 3.13. *Let $2 \le i \le \ell$. Then*

$$s = \frac{\sigma_{n_i}}{\rho_i} \cdot s_i \in R[x] \qquad \text{and} \qquad t = \frac{\sigma_{n_i}}{\rho_i} \cdot t_i \in R[x],$$

in the notation of the proof of Proposition 3.11.

The conceptional advantage of the scalar subresultants is that they live in R rather than $R[x]$ and still provide enough information to build up the required theory.

4 Division rules and Polynomial Remainder Sequences (PRS)

We cannot directly apply the Euclidean Algorithm to polynomials f and g over an integral domain R since polynomial division with remainder in $R[x]$, which is used in every step of the Euclidean Algorithm, is not always defined. Hence our goal now are definitions modified in such a way that they yield a variant of the Euclidean Algorithm that works over an integral domain. We introduce a generalization of the usual pseudo-division, the concept of *division rules*, which leads to intermediate results in $R[x]$.

Definition 4.1. *Let R be an integral domain. A* one-step division rule *is a partial mapping*

$$\mathcal{R}\colon R[x]^2 \rightarrow\!\!\!\!\!+ R^2$$

such that for all $(f, g) \in \operatorname{def}(\mathcal{R})$ there exist $q, r \in R[x]$ satisfying

(i) $\mathcal{R}(f, g) = (\alpha, \beta)$,
(ii) $\alpha f = qg + \beta r$ *and* $\deg r < \deg g$.

Recall that $\operatorname{def}(\mathcal{R}) \subseteq R[x]^2$ is the *domain of definition* of $\mathcal{R}$, that is, the set of $(f, g) \in R[x]^2$ at which $\mathcal{R}$ is defined. In particular, $\mathcal{R}\colon \operatorname{def}(\mathcal{R}) \longrightarrow R^2$ is a total map. In the examples below, we will usually define one-step division rules by starting with a (total or partial) map $\mathcal{R}_0\colon R[x]^2 \rightarrow\!\!\!\!\!+ R^2$ and then taking $\mathcal{R}$ to be the maximal one-step division rule consistent with $\mathcal{R}_0$. Thus

$$\operatorname{def}(\mathcal{R}) = \left\{ (f, g) \in R[x]^2 : \begin{array}{l} \exists \alpha, \beta \in R,\ \exists q, r \in R[x] \\ (\alpha, \beta) = \mathcal{R}_0(f, g) \text{ and (ii) holds} \end{array} \right\},$$

and $\mathcal{R}$ is $\mathcal{R}_0$ restricted to $\operatorname{def}(\mathcal{R})$.

Lemma 2.5(iii) says that for all $(f, g) \in \operatorname{def}(\mathcal{R})$, q and r are unique. Furthermore $(f, 0)$ is never in $\operatorname{def}(\mathcal{R})$ ("you can't divide by zero"), so that

$$\operatorname{def}(\mathcal{R}) \subseteq \mathcal{D}_{\max} = R[x] \times (R[x] \setminus \{0\}).$$

We are particularly interested in one-step division rules $\mathcal{R}$ with $\operatorname{def}(\mathcal{R}) = \mathcal{D}_{\max}$. In our examples, $(0, g)$ will always be in $\operatorname{def}(\mathcal{R})$ if $g \neq 0$.

We may consider the usual remainder as a partial function $\operatorname{rem}\colon R[x]^2 \rightarrow\!\!\!\!\!+ R[x]$ with $\operatorname{rem}(f, g) = r$ if there exist $q, r \in R[x]$ with $f = qg + r$ and $\deg r < \deg g$, and $\operatorname{def}(\operatorname{rem})$ maximal. Recall from Section 2 the definitions of rem, prem and cont.

Example 4.2. Let f and g be polynomials over an integral domain R of degrees n and m, respectively, and let $f_n = \operatorname{lc}(f)$, $g_m = \operatorname{lc}(g) \neq 0$ be their leading coefficients. Then the three most famous types of division rules are as follows:

○ *classical division rule*: $\mathcal{R}(f, g) = (1, 1)$.
○ *monic division rule*: $\mathcal{R}(f, g) = (1, \operatorname{lc}(\operatorname{rem}(f, g)))$.

- *Sturmian division rule*: $\mathcal{R}(f, g) = (1, -1)$.

Examples are given below. When R is a field, these three division rules have the largest possible domain of definition $\operatorname{def}(\mathcal{R}) = \mathcal{D}_{\max}$, but otherwise, it may be smaller; we will illustrate this in Example 4.7. Hence they do not help us in achieving our goal of finding rules with maximal domain $\mathcal{D}_{\max}$. But there exist two division rules which, in contrast to the first examples, always yield solutions in $R[x]$:

- *pseudo-division rule*: $\mathcal{R}(f, g) = (g_m^{n-m+1}, 1)$.

In case R is a unique factorization domain, we have the

- *primitive division rule*: $\mathcal{R}(f, g) = (g_m^{n-m+1}, \operatorname{cont}(\operatorname{prem}(f, g)))$.

For algorithmic purposes, it is then useful for R to be a Euclidean domain. $\Diamond$

The disadvantage of the pseudo-division rule, however, is that in the Euclidean Algorithm it leads to exponential coefficient growth; the coefficients of the intermediate results are usually enormous, their bit length may be exponential in the bit length of the input polynomials f and g. If R is a UFD, we get the smallest intermediate results if we use the primitive division rule, but the computation of the content in every step of the Euclidean Algorithm seems to be expensive. Collins (1967) already observed this in his experiments. Thus he tries to avoid the computation of the content and to keep the intermediate results "small" at the same time by using information from *all* intermediate results in the EEA, not only the two previous remainders. Our concept of one-step division rules does not cover his method. So we now extend our previous definition, and will actually capture all the "recursive" division rules from Collins (1967, 1971, 1973), Brown & Traub (1971) and Brown (1971) under one umbrella.

DEFINITION 4.3. *Let R be an integral domain. A* division rule *is a partial mapping*

$$\mathcal{R}\colon R[x]^2 \nrightarrow (R^2)^*$$

associating to $(f, g) \in \operatorname{def}(\mathcal{R})$ a sequence $((\alpha_2, \beta_2), \ldots, (\alpha_{\ell+1}, \beta_{\ell+1}))$ of arbitrary length $\ell \geq 0$ such that for all $(f, g) \in \operatorname{def}(\mathcal{R})$ there exist $\ell \in \mathbb{N}_{\geq 0}$, $q_1, \ldots, q_\ell \in R[x]$ and $r_0, \ldots, r_{\ell+1} \in R[x]$ satisfying for $2 \leq i \leq \ell + 1$

(i) $r_0 = f, r_1 = g$,

(ii) $\mathcal{R}_i(f, g) = \mathcal{R}(f, g)_i = (\alpha_i, \beta_i)$,

(iii) $\alpha_i r_{i-2} = q_{i-1} r_{i-1} + \beta_i r_i$ *and* $\deg r_i < \deg r_{i-1}$.

A division rule where $\ell = 1$ for all values is the same as a one-step division rule, and from an arbitrary division rule we can obtain a one-step division rule by projecting to the first coordinate (α_2, β_2) if $\ell \geq 2$. Using Lemma 2.5(iii), we find that for all $(f, g) \in \operatorname{def}(\mathcal{R})$, q_{i-1} and r_i are unique for $2 \leq i \leq \ell + 1$. If we have a one-step division rule $\mathcal{R}^*$ which is defined at all (r_{i-2}, r_{i-1}) for $2 \leq i \leq \ell + 1$ (defined recursively), then we obtain a division rule $\mathcal{R}$ by using $\mathcal{R}^*$ in every step:

$$\mathcal{R}_i(f, g) = \mathcal{R}^*(r_{i-2}, r_{i-1}) = (\alpha, \beta).$$

If we truncate $\mathcal{R}$ at the first coordinate, we get $\mathcal{R}^*$ back. But the notion of division rules is strictly richer than that of one-step division rules; for example the first step in the reduced division rule below is just the pseudo-division rule, but using the pseudo-division rule repeatedly does not yield the reduced division rule.

Example 4.2 continued. Let $f = r_0, g = r_1 \in R[x]$ be polynomials of degrees $n_0 \geq n_1$, respectively, and let $\rho_0 = \operatorname{lc}(r_0)$ and $\rho_1 = \operatorname{lc}(r_1)$ be their leading coefficients. We now present three different types of recursive division rules. They are based on polynomial subresultants. It is not obvious that they have domain of definition $\mathcal{D}_{\max}$, since divisions occur in their definitions. We will show that this is indeed the case in Remarks 6.10 and 6.14.

- *reduced division rule*: $\mathcal{R}_i(f, g) = (\alpha_i, \beta_i)$ for $2 \leq i \leq \ell + 1$,
  where we set $\alpha_1 = 1$ and for $2 \leq i \leq \ell + 1$ recursively define

  $$(\alpha_i, \beta_i) = (\rho_{i-1}^{d_{i-2}+1}, \alpha_{i-1}),$$

  then r_i by Definition 4.3 (iii), $\rho_i = \operatorname{lc}(r_i)$, $n_i = \deg r_i$, and $d_{i-1} = n_{i-1} - n_i$.
- *subresultant division rule*: $\mathcal{R}_i(f, g) = (\alpha_i, \beta_i)$ for $2 \leq i \leq \ell + 1$,
  where we set $\rho_0 = 1$ and for $2 \leq i \leq \ell + 1$ recursively define

  $$(\alpha_i, \beta_i) = (\rho_{i-1}^{d_{i-2}+1}, -\rho_{i-2}\psi_i^{d_{i-2}}),$$
  $$\psi_i = \begin{cases} -1 & \text{for } i = 2 \\ (-\rho_{i-2})^{d_{i-3}}\psi_{i-1}^{1-d_{i-3}} & \text{otherwise ,} \end{cases}$$

  then r_i by Definition 4.3 (iii), $\rho_i = \operatorname{lc}(r_i)$, $n_i = \deg r_i$, and $d_{i-1} = n_{i-1} - n_i$.

The subresultant PRS can be improved if we can somehow determine divisors γ_i of the content of the intermediate results.

- *improved division rule*: $\mathcal{R}_i(f, g) = (\alpha_i, \beta_i)$ for $2 \leq i \leq \ell + 1$,
  where we set $\rho_0 = 1$, $\gamma_1 = 1$ and for $2 \leq i \leq \ell + 1$ recursively define

  $$(\alpha_i, \beta_i) = (\rho_{i-1}^{d_{i-2}+1}, -\rho_{i-2}\psi_i^{d_{i-2}}\gamma_{i-1}^{-(d_{i-2}+1)}) \cdot \gamma_i,$$
  $$\psi_i = \begin{cases} -1 & \text{for } i = 2 \\ (-\gamma_{i-2}\rho_{i-2})^{d_{i-3}}\psi_{i-1}^{1-d_{i-3}} & \text{otherwise ,} \end{cases}$$

where γ_i is chosen such that r_i given by Definition 4.3 (iii) is in $R[x]$, $\rho_i = \mathrm{lc}(r_i)$, $n_i = \deg r_i$, and $d_{i-1} = n_{i-1} - n_i$. ◊

The subresultant division rule was invented by Collins (1967), p. 130. He tried to find a rule such that the r_i's agree with the polynomial subresultants up to a small constant factor. Brown (1971), p. 486, then provided a recursive definition of the α_i and β_i as given above.

We note that the exponents in the recursive definition of the ψ_i's in the subresultant division rule and in the improved division rule may be negative. Hence it is not clear that the β_i's are in R. However, we will show this in Theorem 6.17 by proving that the ψ_i are essentially the subresultants, as also done in Brown (1971) **?**???.

Question 4.4. *"At the present time it is not known whether or not these equations imply $\psi_i, \beta_i \in R$."*

By definition, a division rule $\mathcal{R}$ defines a sequence $(r_0, \dots, r_\ell)$ of remainders; recall that they are uniquely defined. Since it is more convenient to work with these "polynomial remainder sequences", we fix this notion in the following definition, following Collins (1967), p. 128/129.

Definition 4.5. *Let $\mathcal{R}$ be a division rule. A sequence $(r_0, \dots, r_\ell)$ of nonzero polynomials $r_0, \dots, r_\ell \in R[x]\setminus\{0\}$ is called the* polynomial remainder sequence (PRS) for (f, g) according to $\mathcal{R}$ *if*

(i) $r_0 = f, r_1 = g$,
(ii) $\mathcal{R}_i(f, g) = (\alpha_i, \beta_i)$,
(iii) $\alpha_i r_{i-2} = q_{i-1} r_{i-1} + \beta_i r_i$,

for $2 \le i \le \ell + 1$, where ℓ is the length *of $\mathcal{R}(f, g)$. The PRS is* complete *if (iii) is satisfied for $i = \ell + 1$ with $r_{\ell+1} = 0$. It is called* normal *if $d_i = \deg r_i - \deg r_{i+1} = 1$ for $1 \le i \le \ell - 1$.*

In fact the remainders for PRS according to arbitrary division rules over an integral domain only differ by a nonzero constant factor.

Proposition 4.6. *Let R be an integral domain, $f, g \in R[x]$ and let $r = (r_0, \dots, r_\ell)$ and $r^* = (r_0^*, \dots, r_{\ell^*}^*)$ be PRS for (f, g) according to two division rules $\mathcal{R}$ and $\mathcal{R}^*$, respectively, none of whose results $\alpha_i, \beta_i, \alpha_i^*, \beta_i^*$ is zero. Then $r_i^* = \gamma_i r_i$ with*

$$\gamma_i = \prod_{0 \le k \le i/2-1} \frac{\alpha_{i-2k}^* \beta_{i-2k}}{\alpha_{i-2k} \beta_{i-2k}^*} \in F \setminus \{0\}$$

for $0 \le i \le \min\{\ell, \ell^*\}$, where F is the field of fractions of R.

Proof. We show the proposition by induction on i. It is clear for $i \leq 1$, and we assume that $i \geq 2$. Then with $\mathcal{R}_i(f,g) = (\alpha_i, \beta_i)$ and $\mathcal{R}_i^*(f,g) = (\alpha_i^*, \beta_i^*)$ we have

$$\begin{aligned} \alpha_i r_{i-2} &= q_{i-1} r_{i-1} + \beta_i r_i, \\ \alpha_i^* r_{i-2}^* &= q_{i-1}^* r_{i-1}^* + \beta_i^* r_i^*. \end{aligned}$$

The induction hypothesis plugged into the second equation and multiplication by α_i yields

$$(\alpha_i \alpha_i^* \gamma_{i-2}) \cdot r_{i-2} = (\alpha_i \gamma_{i-1} q_{i-1}^*) \cdot r_{i-1} + (\alpha_i \beta_i^*) \cdot r_i^*.$$

Multiplying the first equation above by $\alpha_i^* \gamma_{i-2}$ we obtain

$$(\alpha_i \alpha_i^* \gamma_{i-2}) \cdot r_{i-2} = (\alpha_i^* \gamma_{i-2} q_{i-1}) \cdot r_{i-1} + (\alpha_i^* \gamma_{i-2} \beta_i) \cdot r_i$$

From Lemma 2.5(iii) we obtain $(\alpha_i \beta_i^*) \cdot r_i^* = (\alpha_i^* \gamma_{i-2} \beta_i) \cdot r_i$ and $r_i^* = \gamma_i r_i$ with

$$\gamma_i = \frac{\alpha_i^* \beta_i}{\alpha_i \beta_i^*} \cdot \gamma_{i-2} \in F \setminus \{0\}.$$

By induction this completes the proof of the proposition. □

The proposition yields a direct way to compute the PRS for (f,g) according to $\mathcal{R}^*$ from the PRS for (f,g) according to $\mathcal{R}$ and the $\alpha_i, \beta_i, \alpha_i^*, \beta_i^*$. In particular, the degrees of the remainders in any two PRS are identical.

In Example 4.2 we have seen eight different division rules. Now we consider the different polynomial remainder sequences according to these rules. Each PRS will be illustrated by the following example.

Example 4.7. We perform the computations on the polynomials

$$\begin{aligned} f &= r_0 = 9x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45 \quad \text{and} \\ g &= r_1 = 3x^4 - 4x^2 - 9x + 21 \end{aligned}$$

over $R = \mathbb{Q}$ and, wherever possible, also over $R = \mathbb{Z}$. In order to illustrate the coefficient growth of the various PRS, we first present the subresultants of f and g. They are given in reverse order to make it easier to compare them with the intermediate results of the different PRS.

We choose the integers as our ground domain because we then have a reasonably concise presentation of our polynomials.

i	$\sigma_i(f,g)$	factorization of $\sigma_i(f,g)$
$4 = \deg r_1$	9	3^2
3	0	0
$2 = \deg r_2$	9801	$3^4 \cdot 11^2$
$1 = \deg r_3$	13 355 280	$2^4 \cdot 3^6 \cdot 5 \cdot 229$
$0 = \deg r_4$	9 657 273 681	$3^8 \cdot 11 \cdot 133811$

Furthermore we give the factorizations of the α_i, β_i and the leading coefficients of the r_i below the corresponding entries. ◊

4.1 Classical PRS

The most familiar PRS for (f,g) is obtained according to the *classical division rule*. Collins (1973), p. 736, calls this the *natural Euclidean PRS (algorithm)*. The intermediate results of the classical PRS and of the Euclidean Algorithm coincide.

Example 4.7 continued.

i	α_i	β_i	r_i
0			$\underset{3^2}{9}\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$
1			$\underset{3}{3}\,x^4 - 4x^2 - 9x + 21$
2	1	1	$\underset{-11}{-\,11}\,x^2 - 27x + 60$
3	1	1	$\underset{-2^4\cdot 3^2\cdot 5\cdot 229/11^3}{-\,\frac{164\,880}{1331}}\;x + \frac{248\,931}{1331}$
4	1	1	$\underset{-11^4\cdot 133811/2^8\cdot 5^2\cdot 229^2}{-\,\frac{1\,959\,126\,851}{335\,622\,400}}$

◊

The first division works over $\mathbb{Z}$, but not the subsequent ones. In our formalism, this means the following. If we take $\mathcal{R}_0\colon\ R[x]^2 \longrightarrow \mathbb{Z}^2$ with $\mathcal{R}_0(h,k) = (1,1)$ for all $(h,k) \in \mathbb{Z}[x]^2$, then we obtain the division rule $\mathcal{R}$ on $\mathbb{Z}[x]^2$ with $\mathcal{R}(f,g) = ((1,1))$ of length $\ell = 1$.

4.2 Monic PRS

In Collins (1973), p. 736, the PRS for (f, g) according to the *monic division rule* is called *monic PRS (algorithm).* The r_i are monic for $2 \leq i \leq \ell$, and we get the same intermediate results as in the *monic Euclidean Algorithm* in von zur Gathen & Gerhard (1999), Section 3.2.

EXAMPLE 4.7 CONTINUED.

i	α_i	β_i	r_i
0			$9\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$ 3^2
1			$3\,x^4 - 4x^2 - 9x + 21$ 3
2	1	$-\,11$ -11	$x^2 + \frac{27}{11}x - \frac{60}{11}$ 1
3	1	$-\,\frac{164\,880}{1331}$ $-2^4 \cdot 3^2 \cdot 5 \cdot 229/11^3$	$x - \frac{27\,659}{18\,320}$ 1
4	1	$\frac{178\,102\,441}{335\,622\,400}$ $11^3 \cdot 133811/2^8 \cdot 5^2 \cdot 229^2$	1 1

$\diamondsuit$

4.3 Sturmian PRS

We choose the PRS for (f, g) according to the *Sturmian division rule* as introduced in Sturm (1835). Kronecker (1873), p. 117, Habicht (1948), p. 102, and Loos (1982), p. 119, deal with this *generalized Sturmian PRS (algorithm).* Kronecker (1873) calls it *Sturmsche Reihe (Sturmian sequence)*, and in Habicht (1948) it is the *verallgemeinerte Sturmsche Kette (generalized Sturmian chain).* If $g = \partial f/\partial x$ as in Habicht (1948), p. 99, then this is the *classical Sturmian PRS (algorithm).* Note that the Sturmian PRS agrees with the classical PRS up to sign.

EXAMPLE 4.7 CONTINUED.

i	α_i	β_i	r_i
0			$9\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$ 3^2
1			$3\,x^4 - 4x^2 - 9x + 21$ 3
2	1	-1	$11\,x^2 - 27x + 60$ 11
3	1	-1	$\frac{164\,880}{1331}\quad x + \frac{248\,931}{1331}$ $2^4 \cdot 3^2 \cdot 5 \cdot 229/11^3$
4	1	-1	$-\,\frac{1\,959\,126\,851}{335\,622\,400}$ $-11^4 \cdot 133811/2^8 \cdot 5^2 \cdot 229^2$

◊

If we assume that R is an integral domain but not a field, the example shows that the first three types of PRS do not have $\mathcal{D}_{\max}$ as their domain of definition. In the example they are only of length 1. But fortunately there are division rules that have this property.

4.4 Pseudo PRS

If we choose the PRS according to the *pseudo-division rule*, then we get the *pseudo PRS*. Collins (1967), p. 138, calls this the *Euclidean PRS (algorithm)* because it is the most obvious generalization of the Euclidean Algorithm to polynomials over an integral domain R that is not a field. In Collins (1973), p. 737, it is called the *pseudo-remainder PRS*.

Example 4.7 continued.

i	α_i	β_i	r_i
0			$\underset{3^2}{9}\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$
1			$\underset{3}{3}\,x^4 - 4x^2 - 9x + 21$
2	$\underset{3^3}{27}$	1	$\underset{-3^3\cdot 11}{-\,297}\,x^2 - 729x + 1620$
3	$\underset{(-3^3\cdot 11)^3}{-\,26\,198\,073}$	1	$\underset{2^4\cdot 3^{11}\cdot 5\cdot 229}{3\,245\,333\,040}\,x - 4\,899\,708\,873$
4	$\underset{(2^4\cdot 3^{11}\cdot 5\cdot 229)^2}{10\,532\,186\,540\,515\,641\,600}$	1	$\underset{-3^{25}\cdot 11^4\cdot 133811}{-\,1\,659\,945\,865\,306\,233\,453\,993}$

◊

4.5 Primitive PRS

To obtain a PRS over R with minimal coefficient growth, we choose the PRS according to the *primitive division rule* which yields primitive intermediate results. Brown (1971), p. 484, calls this the *primitive PRS (algorithm)*.

Example 4.7 continued.

i	α_i	β_i	r_i
0			$9\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$ 3^2
1			$3\,x^4 - 4x^2 - 9x + 21$ 3
2	27 3^3	3 3	$-\,11\,x^2 - 27x + 60$ -11
3	$-\,1331$ $(-11)^3$	9 3^2	$18\,320\,x - 27\,659$ $2^4 \cdot 5 \cdot 229$
4	$335\,622\,400$ $(2^4 \cdot 5 \cdot 229)^2$	$1\,959\,126\,851$ $11^4 \cdot 133811$	$-\,1$ -1

$\diamondsuit$

4.6 Reduced PRS

A perceived drawback of the primitive PRS is the (seemingly) costly computation of the content. With probabilistic methods, this can in fact be done with an expected number of about one pairwise gcd calculation for multivariate polynomials (see von zur Gathen & Gerhard (1999), ?) and less than two pairwise gcd's for integers Cooperman *et al.* (1999). In fact, in our experiments in Section 8, the primitive PRS sometimes turns out to be most efficient among those discussed here. But Collins (1967) introduced his *reduced PRS (algorithm)* in order to avoid the computation of the content completely. His algorithm uses the *reduced division rule* and keeps the intermediate coefficients reasonably small but not necessarily as small as with the primitive PRS.

Example 4.7 continued.

i	α_i	β_i	r_i
0			$9\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$ 3^2
1			$3\,x^4 - 4x^2 - 9x + 21$ 3
2	27 3^3	1 1	$-\,297\,x^2 - 729x + 1620$ $-3^3 \cdot 11$
3	$-\,26\,198\,073$ $(-3^3 \cdot 11)^3$	27 3^3	$120\,197\,520\,x - 181\,470\,699$ $2^4 \cdot 3^8 \cdot 5 \cdot 229$
4	$14\,447\,443\,814\,150\,400$ $(2^4 \cdot 3^8 \cdot 5 \cdot 229)^2$	$-\,26\,198\,073$ $-3^9 \cdot 11^3$	$86\,915\,463\,129$ $3^{10} \cdot 11 \cdot 133811$

$\diamondsuit$

4.7 Subresultant PRS

The reduced PRS is not the only way to keep the coefficients small without computing contents. We can also use the *subresultant division rule*. According to Collins (1967), p. 130, this is the *subresultant PRS (algorithm)*.

EXAMPLE 4.7 CONTINUED.

i	α_i	β_i	r_i
0			$9\,x^6 - 27x^4 - 27x^3 + 72x^2 + 18x - 45$ 3^2
1			$3\,x^4 - 4x^2 - 9x + 21$ 3
2	27 3^3	-1 -1	$297\,x^2 + 729x - 1620$ $3^3 \cdot 11$
3	$26\,198\,073$ $(3^3 \cdot 11)^3$	-243 -3^5	$13\,355\,280\,x - 20\,163\,411$ $2^4 \cdot 3^6 \cdot 5 \cdot 229$
4	$178\,363\,503\,878\,400$ $(2^4 \cdot 3^6 \cdot 5 \cdot 229)^2$	$2\,910\,897$ $3^7 \cdot 11^3$	$9\,657\,273\,681$ $3^8 \cdot 11 \cdot 133811$

$\diamondsuit$

4.8 Improved PRS

It is possible to improve the subresultant PRS (algorithm) if we can determine divisors γ_i of the content of the intermediate results. Then we are allowed to use the PRS according to the *improved division rule*. In Brown (1971), p. 487, and Brown (1978), p. 243–245, this is called *improved PRS (algorithm)*. So obviously $r_i \in R[x]$ for $2 \leq i \leq \ell$. It is not clear to us how to find such γ_i in a manner that essentially avoids the content computation.

5 Fundamental Theorem on subresultants

The Fundamental Theorem on subresultants was discovered independently in 1968 by Brown and by (Collins, footnote on page 519). It expresses an arbitrary subresultant as a power product of certain data in the PRS, namely the multipliers α and β and the leading coefficients of the remainders in the Euclidean Algorithm. In this section our first goal is to prove the Fundamental Theorem on subresultants for polynomial remainder sequences according to an arbitrary division rule $\mathcal{R}$. From this theorem we then derive results for the various PRS according to the division rules in Example 4.2. We start with two technical lemmas. The first one gives a relation between the subresultants of (f,g) and (g,r) when $r = f \operatorname{rem} g$. Proofs can be found in Geddes *et al.* (1992), Chapter 7.3, p. 292/293, Lemma 7.1; von zur Gathen & Gerhard (1999), Lemma 11.12; and Brown & Traub (1971), p. 509, Lemma 1, for polynomial subresultants.

LEMMA 5.1. *Let f and $g \in R[x]$ be polynomials of degrees $n \geq m > 0$, respectively, over an integral domain R, and let $q,r \in R[x]$ with $f = qg + r$ and $\deg r = k < m$. Then*

$$\sigma_j(f,g) = \begin{cases} (-1)^{(n-j)(m-j)}\mathrm{lc}(g)^{n-k}\sigma_j(g,r) & \text{for } 0 \leq j \leq k, \\ 0 & \text{for } k < j < m. \end{cases}$$

We apply Lemma 5.1 to polynomial remainder sequences. For polynomial subresultants this result is in Brown & Traub (1971), p. 510, Lemma 2, and for reduced PRS in Collins (1967), p. 131, Lemma 1.

LEMMA 5.2. *Let f and $g \in R[x]$ be polynomials of degrees $n \geq m > 0$, respectively, over an integral domain R, let $\mathcal{R}$ be a division rule, $(f,g) \in \mathrm{def}(\mathcal{R})$ and $(r_0, \ldots, r_\ell)$ be the PRS for (f,g) according to $\mathcal{R}$, $(\alpha_i, \beta_i) = \mathcal{R}_i(f,g)$ the constant multipliers, $n_i = \deg r_i$ and $\rho_i = \mathrm{lc}(r_i)$ for $0 \leq i \leq \ell$. Then*

$$\sigma_j(r_{i-2}, r_{i-1}) = (-1)^{(n_{i-2}-j)(n_{i-1}-j)} \left(\frac{\beta_i}{\alpha_i}\right)^{n_{i-1}-j} \rho_{i-1}^{n_{i-2}-n_i}\sigma_j(r_{i-1}, r_i)$$

if $0 \leq j \leq n_i$, and $\sigma_j(r_{i-2}, r_{i-1}) = 0$ if $n_i < j < n_{i-1}$.

In particular, this implies that $\alpha_i^{n_{i-1}-j}$ divides in R the numerator of the right hand side.

Now we are ready to give a proof of the following result which is shown for PRS in Brown & Traub (1971), p. 511, Fundamental theorem, and for reduced PRS in Collins (1967), p. 132, Lemma 2, and p. 133, Theorem 1.

FUNDAMENTAL THEOREM 5.3. *Let f and $g \in R[x]$ be polynomials of degrees $n \geq m > 0$, respectively, over an integral domain R, let $\mathcal{R}$ be a division rule and $(r_0, \ldots, r_\ell)$ be the PRS for (f,g) according to $\mathcal{R}$, $(\alpha_i, \beta_i) = \mathcal{R}_i(f,g)$ the constant multipliers, $n_i = \deg r_i$ and $\rho_i = \mathrm{lc}(r_i)$ for $0 \leq i \leq \ell$, and $d_i = n_i - n_{i+1}$ for $0 \leq i \leq \ell - 1$.*

(i) For $0 \leq j \leq n_1$, the jth subresultant of (f,g) is

$$\sigma_j(f,g) = (-1)^{b_i}\rho_i^{n_{i-1}-n_i} \prod_{2 \leq k \leq i} \left(\frac{\beta_k}{\alpha_k}\right)^{n_{k-1}-n_i} \rho_{k-1}^{n_{k-2}-n_k}$$

if $j = n_i$ for some $1 \leq i \leq \ell$, otherwise 0*, where $b_i = \sum_{2 \leq k \leq i}(n_{k-2} - n_i)(n_{k-1} - n_i)$.*

(ii) *The subresultants satisfy for $1 \leq i < \ell$ the recursive formulas*

$$\sigma_{n_1}(f,g) = \rho_1^{d_0} \text{ and}$$

$$\sigma_{n_{i+1}}(f,g) = \sigma_{n_i}(f,g) \cdot (-1)^{d_i(n_0 - n_{i+1} + i + 1)} (\rho_{i+1}\rho_i)^{d_i} \prod_{2 \leq k \leq i+1} \left(\frac{\beta_k}{\alpha_k}\right)^{d_i}.$$

Proof. (i) We define i by the conditions that $1 \leq i \leq \ell$ and $n_{i+1} < j \leq n_i$. By induction on i, we find from Lemma 5.2

$$\sigma_j(f,g) = \sigma_j(r_{i-1}, r_i) \prod_{2 \leq k \leq i} (-1)^{(n_{k-2}-j)(n_{k-1}-j)} \left(\frac{\beta_k}{\alpha_k}\right)^{n_{k-1}-j} \rho_{k-1}^{n_{k-2}-n_k}$$

if $j = n_i$, and $\sigma_j(f,g) = 0$ if $n_{i+1} < j < n_i$. Furthermore, if $j = n_i$, then

$$\sigma_{n_i}(r_{i-1}, r_i) = \det \begin{pmatrix} \rho_i & & \\ \vdots & \ddots & \\ \vdots & & \rho_i \end{pmatrix} = \rho_i^{n_{i-1}-n_i}.$$

(ii) Firstly, (i) implies that $\sigma_{n_1}(f,g) = \rho_1^{d_0}$. Now assume $i \geq 1$. Then from (i) we obtain

$$\begin{aligned}
&\sigma_{n_{i+1}}(f,g) \\
&= \rho_{i+1}^{d_i} \prod_{2 \leq k \leq i+1} (-1)^{(n_{k-2}-n_{i+1})(n_{k-1}-n_{i+1})} \left(\frac{\beta_k}{\alpha_k}\right)^{n_{k-1}-n_{i+1}} \rho_{k-1}^{n_{k-2}-n_k} \\
&= \rho_{i+1}^{d_i} \prod_{2 \leq k \leq i} (-1)^{(n_{k-2}-n_i)(n_{k-1}-n_i)} \left(\frac{\beta_k}{\alpha_k}\right)^{n_{k-1}-n_i} \rho_{k-1}^{n_{k-2}-n_k} \cdot \\
&\qquad \prod_{2 \leq k \leq i} (-1)^{d_i(n_{k-2}+n_{k-1}+1)} \left(\frac{\beta_k}{\alpha_k}\right)^{n_i - n_{i+1}} \cdot \\
&\qquad (-1)^{(n_{i-1}-n_i)(n_i-n_{i+1})} \left(\frac{\beta_{i+1}}{\alpha_{i+1}}\right)^{n_i-n_{i+1}} \rho_i^{n_{i-1}-n_{i+1}} \\
&= \rho_{i+1}^{d_i} \rho_i^{-d_{i-1}+n_{i-1}-n_i} \cdot \sigma_{n_i}(f,g) \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)} \prod_{2 \leq k \leq i+1} \left(\frac{\beta_k}{\alpha_k}\right)^{d_i}.
\end{aligned}$$

This completes the proof of the fundamental theorem. □

We now have the following generalization of Theorem 3.10.

Corollary 5.4. *Let $\mathcal{R}$ be a division rule and $(r_0, \ldots, r_\ell)$ be the PRS for (f, g) according to $\mathcal{R}$, let $n_i = \deg r_i$ for $0 \le i \le \ell$ be the degrees in the PRS, and let $0 \le k \le n_1$. Then*

$$\sigma_k \neq 0 \Longleftrightarrow \exists i \le \ell \quad k = n_i.$$

6 Applications of the Fundamental Theorem

We now derive results for the various PRS for polynomials $f, g \in R[x]$ of degrees $n \ge m \ge 0$, respectively, over an integral domain R, according to the division rules in Example 4.2. The first type of result expresses the subresultants $\sigma_k = \sigma_k(f, g)$ in terms of the quantities $\rho_i = \mathrm{lc}(r_i)$, $n_i = \deg r_i$, and $d_i = n_i - n_{i+1}$, and others in the PRS. The second type gives a recursive equation expressing $\rho_{n_{i+1}}$ as a multiple of ρ_{n_i}. Both types of formula simplify considerably in the normal case. Finally, we can also reverse these equations in the normal case and express the ρ_i in terms of the other quantities. We start with a technical lemma.

Lemma 6.1. *Let $b_i = \sum_{2 \le k \le i}(n_{k-2} - n_i)(n_{k-1} - n_i)$ be as in Fundamental Theorem 5.3. If the PRS is normal, then*

$$b_i \equiv (d_0 + 1)(i + 1) \bmod 2 \quad \textit{for } 2 \le i \le \ell.$$

Proof. Since the PRS is normal, we have $d_j = 1$ for $1 \le j \le \ell$, and get

$$\begin{aligned} b_i &= \sum_{2 \le k \le i} (n_{k-2} - n_i)(n_{k-1} - n_i) \\ &= (d_0 + i - 1)(i - 1) + \sum_{3 \le k \le i} (i - k + 2)(i - k + 1) \\ &\equiv (d_0 + 1)(i + 1) \bmod 2. \end{aligned}$$

□

6.1 Classical PRS

The following claims for the classical PRS are proved by substituting $(\alpha_i, \beta_i) = (1, 1)$ for $2 \le i \le \ell$ in the Fundamental Theorem 5.3.

Corollary 6.2. *Let $(r_0, \ldots, r_\ell)$ be a classical PRS and $1 \le i \le \ell$. Then*

(i)
$$\sigma_{n_i} = (-1)^{b_i} \rho_i^{d_{i-1}} \prod_{2 \le k \le i} \rho_{k-1}^{n_{k-2} - n_k}.$$

(ii) The subresultants satisfy the recursive formulas

$$\begin{aligned} \sigma_{n_1} &= \rho_1^{d_0}, \text{ and} \\ \sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)}(\rho_{i+1}\rho_i)^{d_i}. \end{aligned}$$

If the PRS is normal, then this simplifies to:

(iii)
$$\sigma_{n_i} = (-1)^{(d_0+1)(i+1)}\rho_i\rho_1^{d_0+1} \prod_{3\leq k\leq i} \rho_{k-1}^2 \text{ for } i \geq 2.$$

(iv) The subresultants satisfy the recursive formulas

$$\begin{aligned} \sigma_{n_1} &= \rho_1^{d_0}, \text{ and} \\ \sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_0+1}\rho_{i+1}\rho_i. \end{aligned}$$

6.2 Monic PRS

For the monic PRS, the Fundamental Theorem 5.3 yields the following corollary which is the Fundamental Theorem 11.13 in von zur Gathen & Gerhard (1999).

Corollary 6.3. *Let $(r_0, \ldots, r_\ell)$ be a monic PRS, and $2 \leq i \leq \ell$. Then*

(i)
$$\begin{aligned} \sigma_{n_1} &= \rho_1^{d_0}, \text{ and} \\ \sigma_{n_i} &= (-1)^{b_i}\rho_1^{n_0-n_2} \prod_{2\leq k\leq i} \beta_k^{n_{k-1}-n_i}. \end{aligned}$$

(ii) The subresultants satisfy the recursive formulas

$$\begin{aligned} \sigma_{n_1} &= \rho_1^{d_0}, \\ \sigma_{n_2} &= \sigma_{n_1} \cdot (-1)^{d_1(n_0-n_2+2)}(\rho_1\beta_2)^{d_1}, \text{ and} \\ \sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)} \prod_{2\leq k\leq i+1} \beta_k^{d_i}. \end{aligned}$$

If the PRS is normal, then this simplifies to:

(iii)
$$\begin{aligned} \sigma_{n_1} &= \rho_1^{d_0}, \text{ and} \\ \sigma_{n_i} &= (-1)^{(d_0+1)(i+1)}\rho_1^{d_0+1} \prod_{2\leq k\leq i} \beta_k^{i-(k-1)}. \end{aligned}$$

(iv) The subresultants satisfy the recursive formulas

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0}\\ \sigma_{n_2} &= \sigma_{n_1}\cdot(-1)^{(d_0+1)2}\rho_1\beta_2, \text{ and}\\ \sigma_{n_{i+1}} &= \sigma_{n_i}\cdot(-1)^{(d_0+1)(i+1)}\prod_{2\le k\le i+1}\beta_k.\end{aligned}$$

6.3 Sturmian PRS

For the Sturmian PRS, the results read as follows.

Corollary 6.4. *Let $(r_0,\dots,r_\ell)$ be a Sturmian PRS, and $1\le i\le \ell$. Then*

(i) $$\sigma_{n_i} = (-1)^{b_i+\sum_{2\le k\le i}(n_{k-1}-n_i)}\rho_i^{d_{i-1}}\prod_{2\le k\le i}\rho_{k-1}^{n_{k-2}-n_k}.$$

(ii) The subresultants satisfy the recursive formulas

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\ \sigma_{n_{i+1}} &= \sigma_{n_i}\cdot(-1)^{d_i(n_0-n_{i+1}+1)}(\rho_{i+1}\rho_i)^{d_i}.\end{aligned}$$

If the PRS is normal, then this simplifies to:

(iii) $$\sigma_{n_i} = (-1)^{(d_0+1)(i+1)}\rho_1^{d_0+1}\rho_i\prod_{3\le k\le i}\rho_{k-1}^2 \text{ for } i\ge 2.$$

(iv) The subresultants satisfy the recursive formulas

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\ \sigma_{n_{i+1}} &= \sigma_{n_i}\cdot(-1)^{d_0+i+1}\rho_{i+1}\rho_i.\end{aligned}$$

6.4 Pseudo PRS

Again the Fundamental Theorem 5.3, after substituting $(\alpha_i,\beta_i) = (\rho_{i-1}^{d_{i-2}+1},1)$ for $2\le i\le\ell$, provides the following corollary for the pseudo PRS. It can also be found in Collins (1966), p. 710, Theorem 1, for polynomial subresultants.

Corollary 6.5. *Let $(r_0,\dots,r_\ell)$ be a pseudo PRS, and $1\le i\le\ell$. Then*

(i) $$\sigma_{n_i} = (-1)^{b_i}\rho_i^{d_{i-1}}\prod_{2\le k\le i}\rho_{k-1}^{n_{k-2}-n_k-(n_{k-1}-n_i)(d_{k-2}+1)}.$$

(ii) The subresultants satisfy the recursive formulas

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\ \sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)} (\rho_{i+1}\rho_i)^{d_i} \prod_{2\le k\le i+1} \rho_{k-1}^{-(d_{k-2}+1)d_i}.\end{aligned}$$

If the PRS is normal, then this simplifies to:

(iii) $$\sigma_{n_i} = (-1)^{(d_0+1)(i+1)} \rho_1^{(d_0+1)(2-i)} \rho_i \prod_{3\le k\le i-1} \rho_{k-1}^{2(k-i)} \text{ for } i \ge 2.$$

(iv) The subresultants satisfy the recursive formulas

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\ \sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_0+1} \rho_1^{-(d_0+1)} \rho_{i+1}\rho_i \prod_{3\le k\le i+1} \rho_{k-1}^{-2}.\end{aligned}$$

Remark 6.6. *If the PRS is normal, then Corollary 6.5(iii) implies that*

$$\rho_i = \sigma_{n_i} (-1)^{(d_0+1)(i+1)} \rho_1^{(d_0+1)(i-2)} \prod_{3\le k\le i-1} \rho_{k-1}^{2(i-k)}.$$

Thus σ_{n_i} divides ρ_i. This result is also shown for polynomial subresultants in Collins (1966), p. 711, Corollary 1.

6.5 Primitive PRS

Since the content of two polynomials cannot be expressed in terms of our parameters ρ_i and n_i, we do not consider the Fundamental Theorem for this type of PRS. We only make the following remark.

Remark 6.7. *Let $(r_0, \dots, r_\ell)$ be a primitive PRS. Then ρ_i divides σ_{n_i} for $2 \le i \le \ell$ since $\sigma_{n_i} \cdot r_i\rho_i \in R[x]$ according to Proposition 3.11 and r_i is primitive.*

If $R = \mathbb{Z}$, then the required gcd calculations can become quite expensive, but see Cooperman *et al.* (1999) for an efficient proposal.

6.6 Reduced PRS

For reduced PRS the Fundamental Theorem 5.3 yields the following corollary. The non-normal parts are shown for polynomial subresultants in Collins (1967), p. 135, Corollary 1.2, and Collins (1967), p. 135, Corollary 1.4, respectively.

Corollary 6.8. *Let $(r_0, \dots, r_\ell)$ be a reduced PRS, and $1 \leq i \leq \ell$. Then*

(i)
$$\sigma_{n_i} = (-1)^{b_i} \rho_i^{d_{i-1}} \prod_{2 \leq k \leq i} \rho_{k-1}^{d_{k-2}(1-d_{k-1})}.$$

(ii) The subresultants satisfy for the recursive formulas

$$\sigma_{n_1} = \rho_1^{d_0}, \text{ and}$$
$$\sigma_{n_{i+1}} = \sigma_{n_i} \cdot (-1)^{d_i(n_0 - n_{i+1} + i + 1)} \rho_{i+1}^{d_i} \rho_i^{-d_{i-1}d_i}.$$

If the PRS is normal, then this simplifies to:

(iii)
$$\sigma_{n_i} = (-1)^{(d_0+1)(i+1)} \rho_i \text{ for } i \geq 2.$$

(iv) The subresultants satisfy the recursive formulas

$$\sigma_{n_1} = \rho_1^{d_0}, \text{ and}$$
$$\sigma_{n_{i+1}} = \sigma_{n_i} \cdot (-1)^{d_0+1} \rho_{i+1} \rho_i^{-1}.$$

Proof. Since $(\alpha_2, \beta_2) = (\rho_1^{d_0+1}, 1)$ and $(\alpha_i, \beta_i) = (\rho_{i-1}^{d_{i-2}+1}, \alpha_{i-1})$ we get

$$\begin{aligned}\prod_{2 \leq k \leq i} \left(\frac{\beta_k}{\alpha_k}\right)^{n_{k-1} - n_i} &= \prod_{3 \leq k \leq i} \alpha_{k-1}^{n_{k-1} - n_i} \prod_{2 \leq k \leq i} \alpha_k^{-(n_{k-1} - n_i)} \\ &= \alpha_i^{-(n_{i-1} - n_i)} \prod_{2 \leq k \leq i-1} \alpha_k^{n_{k-1} - n_k} \\ &= \prod_{2 \leq k \leq i} \alpha_k^{-d_{k-1}} = \prod_{2 \leq k \leq i} \rho_{k-1}^{-(d_{k-2}+1)d_{k-1}}.\end{aligned}$$

Together with Fundamental Theorem 5.3 this yields the claims. □

Remark 6.9. *We obtain from Corollary 6.8(i)*

$$\rho_i^{d_{i-1}} = \sigma_{n_i} \prod_{2 \leq k \leq i} (-1)^{(n_{k-2} - n_i)(n_{k-1} - n_i)} \rho_{k-1}^{d_{k-2}(d_{k-1}-1)}.$$

Thus σ_{n_i} divides $\rho_i^{d_{i-1}}$. This result can also be found in Collins (1967), p.135, Corollary 1.2.

Remark 6.10. *For every reduced PRS, r_i is in $R[x]$ for $2 \leq i \leq \ell$. Note that Corollary 6.8(iii) implies $r_i = (-1)^{(d_0+1)(i+1)} R_i(f, g)$. So the normal case is clear. A proof for the general case based on polynomial subresultants is in Collins (1967), p. 134, Corollary 1.1, and Brown (1971), p. 485/486.*

We now derive some results for subresultant PRS with the help of the Fundamental Theorem 5.3. To simplify our formulas we use

$$e_{i,j} = d_{j-1} \prod_{j \le k \le i} (1 - d_k).$$

Our first goal is to solve the recurrence for the β_i and eliminate the ψ_i. This is done in the following two technical lemmas.

Lemma 6.11. *Let ψ_i be defined recursively as in Example 4.2 by $\psi_2 = -1$ and $\psi_i = (-\rho_{i-2})^{d_{i-3}} \psi_{i-1}^{1-d_{i-3}}$ for $3 \le i \le \ell$. Then*

$$\psi_i = - \prod_{1 \le j \le i-2} \rho_j^{e_{i-3,j}} \text{ for } 2 \le i \le \ell.$$

Proof. For a proof by induction, we first verify the claim for $i = 2$:

$$\psi_2 = - \prod_{1 \le j \le 0} \rho_j^{e_{-1,j}}.$$

Now we assume that $i \ge 2$. Then

$$\begin{aligned}\psi_{i+1} &= (-1)^{d_{i-2}} \rho_{i-1}^{d_{i-2}} \psi_i^{1-d_{i-2}} = (-1)^{d_{i-2}} \rho_{i-1}^{d_{i-2}} \left(- \prod_{1 \le j \le i-2} \rho_j^{e_{i-3,j}} \right)^{1-d_{i-2}} \\ &= -\rho_{i-1}^{d_{i-2}} \prod_{1 \le j \le i-2} \rho_j^{(1-d_{i-2}) e_{i-3,j}} = - \prod_{1 \le j \le (i+1)-2} \rho_j^{e_{(i+1)-3,j}}.\end{aligned}$$

By induction, this completes the proof of the lemma. □

Lemma 6.12. *Let $\alpha_i = \rho_{i-1}^{d_{i-2}+1}$ for $2 \le i \le \ell$, and let $\beta_2 = (-1)^{d_0+1}$ and $\beta_i = -\rho_{i-2} \psi_i^{d_{i-2}}$ for $3 \le i \le \ell$. Then*

$$\prod_{2 \le k \le i} \frac{\beta_k}{\alpha_k} = (-1)^{n_0 - n_{i-1} + i - 1} \rho_{i-1}^{-(d_{i-2}+1)} \prod_{1 \le k \le i-2} \rho_k^{-e_{i-2,k}} \text{ for } 2 \le i \le \ell.$$

Proof. Since

$$\frac{\beta_2}{\alpha_2} = (-1)^{d_0+1} \left(\rho_1^{d_0+1} \right)^{(-1)} = (-1)^{n_0 - n_1 + 1} \rho_1^{-(d_0+1)} \prod_{1 \le k \le 0} \rho_k^{-e_{0,k}},$$

the claim is true for $i = 2$. Now assume that the claim holds for $i - 1 \ge 2$ and consider

$$\prod_{2\leq k\leq i}\frac{\beta_k}{\alpha_k} = \frac{\beta_i}{\alpha_i}\prod_{2\leq k\leq i-1}\frac{\beta_k}{\alpha_k} = -\rho_{i-2}\psi_i^{d_{i-2}}\rho_{i-1}^{-(d_{i-2}+1)}\prod_{2\leq k\leq i-1}\frac{\beta_k}{\alpha_k}.$$

From Lemma 6.11 we get

$$\begin{aligned}
\prod_{2\leq k\leq i}\frac{\beta_k}{\alpha_k} &= -\rho_{i-2}\Bigg(-\prod_{1\leq k\leq i-2}\rho_k^{e_{i-3,k}}\Bigg)^{d_{i-2}}\rho_{i-1}^{-(d_{i-2}+1)}\prod_{2\leq k\leq i-1}\frac{\beta_k}{\alpha_k}\\
&= (-1)^{d_{i-2}+1}\rho_{i-2}\Bigg(\prod_{1\leq k\leq i-2}\rho_k^{e_{i-3,k}}\cdot\Bigg)^{d_{i-2}}\rho_{i-1}^{-(d_{i-2}+1)}\\
&\quad (-1)^{n_0-n_{i-2}+i-2}\rho_{i-2}^{-(d_{i-3}+1)}\prod_{1\leq k\leq i-3}\rho_k^{-e_{i-3,k}}\\
&= (-1)^{n_0-n_{i-1}+i-1}\rho_{i-1}^{-(d_{i-2}+1)}\rho_{i-2}^{d_{i-3}(d_{i-2}-1)}\Bigg(\prod_{1\leq k\leq i-3}\rho_k^{e_{i-3,k}}\Bigg)^{d_{i-2}-1}\\
&= (-1)^{n_0-n_{i-1}+i-1}\rho_{i-1}^{-(d_{i-2}+1)}\Bigg(\prod_{1\leq k\leq i-2}\rho_k^{e_{i-2,k}}\Bigg)^{d_{i-2}-1}.
\end{aligned}$$

By induction, this completes the proof of the lemma. □

Corollary 6.13. *Let $(r_0,\ldots,r_\ell)$ be a subresultant PRS, and $1\leq i\leq\ell$. Then*

(i)
$$\sigma_{n_i} = \prod_{1\leq k\leq i}\rho_k^{e_{i-1,k}}.$$

(ii) The subresultants satisfy the recursive formulas

$$\begin{aligned}
\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\
\sigma_{n_{i+1}} &= \sigma_{n_i}\cdot\rho_{i+1}^{d_i}\prod_{1\leq k\leq i}\rho_k^{-d_ie_{i-1,k}}.
\end{aligned}$$

If the PRS is normal, then this simplifies to:

(iii)
$$\sigma_{n_i} = \rho_i \text{ for } i\geq 2.$$

(iv) The subresultants satisfy the recursive formulas

$$\begin{aligned}
\sigma_{n_1} &= \rho_1^{d_0}, \text{ and}\\
\sigma_{n_{i+1}} &= \sigma_{n_i}\cdot\rho_{i+1}\rho_i^{-1}.
\end{aligned}$$

Proof. We first prove (ii) and use it to show (i).

(ii) From the Fundamental Theorem 5.3(ii) and Lemma 6.12 we find

$$\begin{aligned}\sigma_{n_{i+1}} &= \sigma_{n_i} \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)}(\rho_{i+1}\rho_i)^{d_i} \prod_{2\leq k\leq i+1} \left(\frac{\beta_k}{\alpha_k}\right)^{d_i} \\ &= \sigma_{n_i} \cdot (-1)^{d_i(n_0-n_{i+1}+i+1)}(\rho_{i+1}\rho_i)^{d_i} \\ &\quad (-1)^{d_i(i+n_0-n_i)}\rho_i^{-(d_{i-1}+1)d_i} \prod_{1\leq k\leq i-1} \rho_k^{-d_i e_{i-1,k}} \\ &= \sigma_{n_i} \cdot (-1)^{d_i(d_i+1)}\rho_{i+1}^{d_i} \prod_{1\leq k\leq i} \rho_k^{-d_i e_{i-1,k}}.\end{aligned}$$

The claim now follows since $d_i(d_i+1)$ is even.

(i) The claim for $i = 1$ is clear from Fundamental Theorem 5.3(i). Now assume that the claim holds for some $i \in \mathbb{N}$. Then (ii) yields

$$\sigma_{n_{i+1}} = \rho_{i+1}^{d_i} \prod_{1\leq j\leq i} \rho_j^{-d_i e_{i-1,j}} \sigma_{n_i},$$

and by induction we have

$$\sigma_{n_{i+1}} = \rho_{i+1}^{d_i} \prod_{1\leq k\leq i} \rho_k^{-d_i e_{i-1,k}} \rho_i^{d_{i-1}} \prod_{1\leq k\leq i-1} \rho_k^{e_{i-1,k}} = \prod_{1\leq k\leq i+1} \rho_k^{e_{i,k}}. \qquad \square$$

Remark 6.14. *For every subresultant PRS the polynomials r_i are in $R[x]$ for $2 \leq i \leq \ell$. Note that Corollary 6.13(iii) implies $r_i = R_i(f,g)$. So the normal case is clear. Proofs for the general case based on polynomial subresultants are in Collins (1967), p. 130, and Brown (1971), p. 486.*

Corollary 6.13 does not provide the only recursive formula for subresultants. Another one is based on an idea in Lickteig & Roy (1997), p. 12, and Reischert (1997), p. 238, where the following formula has been proven for polynomial subresultants. It follows from Corollary 6.13.

Corollary 6.15. *Let $(r_0, \ldots, r_\ell)$ be a subresultant PRS. Then the subresultants satisfy for $1 \leq i < \ell$ the recursive formulas*

$$\begin{aligned}\sigma_{n_1} &= \rho_1^{d_0} \text{ and} \\ \sigma_{n_{i+1}} &= \sigma_{n_i}^{1-d_i} \cdot \rho_{i+1}^{d_i}.\end{aligned}$$

These results also show that the subresultant PRS does take place in $R[x]$, as proven by Brown (1978).

Corollary 6.16. *Let $\psi_2 = -1$ and $\psi_i = (-\rho_{i-2})^{d_{i-3}} \psi_{i-1}^{1-d_{i-3}}$ for $3 \le i \le \ell$.*

(i) $\psi_i = -\sigma_{n_{i-2}}$ for $3 \le i \le \ell$.
(ii) The coefficients ψ_i and β_i of the subresultant PRS are always in R.

Proof. By Lemma 6.11 and Corollary 6.15, we have

$$\psi_3 = -\rho_1^{d_0} = -\sigma_{n_1}.$$

This proves the corollary for $i = 3$. Now assume $i > 3$. Then again Lemma 6.11, Corollary 6.15, and the induction hypothesis yield

$$\psi_i = (-\rho_{i-2})^{d_{i-3}} \psi_{i-1}^{1-d_{i-3}} = -\sigma_{n_{i-2}} \cdot \sigma_{n_{i-3}}^{d_{i-3}} \cdot \sigma_{n_{i-3}}^{1-d_{i-3}} = -\sigma_{n_{i-2}}. \qquad \square$$

Theorem 6.17.

6.8 Comparison of reduced PRS and subresultant PRS

We conclude this section with a comparison of the reduced PRS and the subresultant PRS. To this end we first prove a formula for $\rho_i^{d_{i-1}}$ in the reduced PRS only depending on subresultants, thus solving the recursion in Remark 6.9.

Theorem 6.18. *Let $(r_0, \ldots, r_\ell)$ be a reduced PRS. Then*

$$\rho_i^{d_{i-1}} = \sigma_{n_i} \cdot (-1)^{a_i} \prod_{1 \le k \le i-1} \sigma_{n_k}^{(d_k - 1)\prod_{k \le j \le i-1} d_j},$$

where $a_i = \sum_{2 \le k \le i} (n_0 - n_k + k) \cdot \prod_{k-1 \le j \le i-1} d_j$.

Proof. Corollary 6.8(ii) implies that

$$\rho_2^{d_1} = \sigma_{n_2} \cdot \sigma_{n_1}^{-1} \cdot (-1)^{d_1(n_0 - n_2 + 2)} \rho_1^{d_0 d_1} = \sigma_{n_2} \cdot (-1)^{d_1(n_0 - n_2 + 2)} \sigma_{n_1}^{d_1 - 1},$$

and this proves the claim for $i = 2$. Now assume $i \ge 2$. Then Corollary 6.8(ii) and the induction hypothesis yield

$$\begin{aligned} \rho_{i+1}^{d_i} &= \sigma_{n_{i+1}} \cdot \sigma_{n_i}^{-1} \cdot (-1)^{d_1(n_0 - n_{i+1} + i + 1)} \rho_i^{d_{i-1} d_i} \\ &= \sigma_{n_{i+1}} \cdot \sigma_{n_i}^{-1} \cdot (-1)^{d_i(n_0 - n_{i+1} + i + 1)} \\ &\quad \cdot \left(\sigma_{n_i} \cdot (-1)^{a_i} \prod_{1 \le k \le i-1} \sigma_{n_k}^{(d_k - 1) \cdot \prod_{k \le j \le i-1} d_j} \right)^{d_i} \\ &= \sigma_{n_{i+1}} \cdot (-1)^{a_{i+1}} \prod_{1 \le k \le i} \sigma_{n_k}^{(d_k - 1) \cdot \prod_{k \le j \le i} d_j}. \end{aligned} \qquad \square$$

We can now prove the relation between reduced and subresultant PRS. The normal case can be found in Collins (1967), p. 135, Corollary 1.3, and Collins (1973), p. 738. Since we how deal with two different PRS, we use $\mathrm{lc}(r_i), \mathrm{lc}(r_i^*)$ instead of the unspecific notation ρ_i here.

COROLLARY 6.19. *Let $(r_0, \ldots, r_\ell)$ be a reduced PRS and $(r_0^*, \ldots, r_\ell^*)$ a subresultant PRS for the polynomials $r_0 = r_0^* = f$ and $r_1 = r_1^* = g$. Then the following holds for $2 \leq i \leq \ell$:*

$$\mathrm{lc}(r_i)^{d_i} = (-1)^{a_i} \prod_{1 \leq k \leq i-2} \sigma_{n_k}^{(d_k-1)\cdot\prod_{k\leq j\leq i-1} d_j} \cdot \mathrm{lc}(r_i^*)^{d_i},$$

where $a_i = \sum_{2\leq k\leq i}(n_0 - n_k + k) \cdot \prod_{k-1\leq j\leq i-1} d_j$. If the PRS are normal, this simplifies to

$$\mathrm{lc}(r_i) = (-1)^{(n_0-n_i)(n_1-n_i)} \cdot \mathrm{lc}(r_i^*).$$

PROOF. Follows immediately from Theorem 6.18 and Corollary 6.15. □

Since the exponent of σ_{n_k} is nonnegative, this means that the entries in the reduced PRS are at least as large in absolute value as those in the subresultant PRS.

7 Analysis of coefficient growth and running time

This section presents two types of results. We first show an exponential lower bound on the size of the entries of the pseudo PRS that matches the upper bound from Knuth (1981), 4.6.1. A slightly different lower bound is in **?**, 3.3.3. On the other hand, we show polynomial upper bounds for all other PRSs.

LEMMA 7.1. *Let $e_2 = 0$, $e_3 = 1$, and $e_{i+1} = 2e_i + e_{i-1}$ for $i \geq 3$. Then*

(i) $\sum_{2\leq k\leq i-1} 2e_k = e_i + e_{i-1} - 1.$
(ii) $e_i = \cdots$

PROOF. Since $2e_2 = 0 = e_3 + e_2 - e_3$, the claim holds for $i = 3$. Now assume $i \geq 3$. By induction hypothesis we get

$$\begin{aligned}\sum_{2\leq k\leq (i+1)-1} 2e_k &= 2e_i + \sum_{2\leq k\leq i-1} 2e_k \\ &= 2e_i + e_i + e_{i-1} - e_3 = e_{i+1} + e_i - e_3.\end{aligned}$$ □

Lemma 7.2. *Suppose that $(f, g) \in \mathbb{Z}[x]^2$ have a normal pseudo PRS. Then*

$$\rho_i = \sigma_{n_i} \cdot (-1)^{(d_0+1)(i+1)} (\rho_1 \sigma_{n_1})^{e_i} \prod_{2 \le j \le i-2} \sigma_{n_j}^{2e_{i-j+1}}$$

with $e_2 = 0$, $e_3 = 1$ and $e_{i+1} = 2e_i + e_{i-1}$ for $3 \le i \le \ell - 1$.

Proof. Since Remark 6.6 shows the claim for $i \le 3$, we assume $i \ge 3$. From Corollary 6.5(iv) and the induction hypothesis we get

$$\begin{aligned}
&\rho_{i+1} \\
&= \sigma_{n_{i+1}} \sigma_{n_i}^{-1} \cdot (-1)^{d_0+1} (\rho_1 \sigma_{n_1}) \rho_i \prod_{2 \le k \le i-1} \rho_k^2 \\
&= \sigma_{n_{i+1}} \sigma_{n_i}^{-1} \cdot (-1)^{d_0+1} (\rho_1 \sigma_{n_1}) \sigma_{n_i} \cdot (-1)^{(d_0+1)(i+1)} (\rho_1 \sigma_{n_1})^{e_i} \\
&\qquad \cdot \prod_{2 \le j \le i-2} \sigma_{n_j}^{2e_{i-j+1}} \cdot \prod_{2 \le k \le i-1} \left(\sigma_{n_k} (\rho_1 \sigma_{n_1})^{e_k} \prod_{2 \le j \le k-2} \sigma_{n_j}^{2e_{k-j+1}} \right)^2 \\
&= \sigma_{n_{i+1}} \cdot (-1)^{(d_0+1)(i+2)} (\rho_1 \sigma_{n_1})^{1+e_i+\sum_{2 \le k \le i-1} 2e_k} \prod_{2 \le j \le i-2} \sigma_{n_j}^{2e_{i-j+1}} \cdot \\
&\quad \prod_{2 \le k \le i-1} \sigma_{n_k}^2 \cdot \prod_{\substack{j+2 \le k \le i-1 \\ 2 \le j \le i-3}} \sigma_{n_j}^{4e_{k-j+1}} \\
&= \sigma_{n_{i+1}} \cdot (-1)^{(d_0+1)(i+2)} (\rho_1 \sigma_{n_1})^{1+e_i+\sum_{2 \le k \le i-1} 2e_k} \prod_{2 \le j \le i-2} \sigma_{n_j}^{2e_{i-j+1}} \\
&\qquad \cdot \prod_{2 \le j \le i-1} \sigma_{n_j}^{2+2\sum_{j+2 \le k \le i-1} 2e_{k-j+1}}.
\end{aligned}$$

With Lemma 7.1 we get

$$\begin{aligned}
\rho_{i+1} &= \sigma_{n_{i+1}} \cdot (-1)^{(d_0+1)(i+2)} (\rho_1 \sigma_{n_1})^{e_{i+1}} \prod_{2 \le j \le i-2} \sigma_{n_j}^{2e_{i-j+1}} \prod_{2 \le j \le i-1} \sigma_{n_j}^{2e_{i-j+1}+2e_{i-j}} \\
&= \sigma_{n_{i+1}} \cdot (-1)^{(d_0+1)(i+2)} (\rho_1 \sigma_{n_1})^{e_{i+1}} \prod_{2 \le j \le i-1} \sigma_{n_j}^{2e_{i-j+2}}.
\end{aligned}$$

By induction, this proves the lemma. □

Theorem 7.3. *The final remainder ρ_ℓ in the pseudo PRS is at least 2^{2^n} in some cases with input polynomials of degrees at most n and coefficients of constant size.*

Proof. Let e_i be as in Lemma 7.2 for $2 \leq i \leq \ell$. Then we have

$$\begin{pmatrix} e_i \\ e_{i-1} \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} e_{i-1} \\ e_{i-2} \end{pmatrix} = \ldots = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^k \begin{pmatrix} e_{i-k} \\ e_{i-(k+1)} \end{pmatrix}.$$

Since the eigenvalues of the matrix are $1 \pm \sqrt{2}$, we get

$$\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^k = \begin{pmatrix} 1 & -1+\sqrt{2} \\ 1 & -1-\sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} (1+\sqrt{2})^k & 0 \\ 0 & (1-\sqrt{2})^k \end{pmatrix} \cdot \begin{pmatrix} 1 & -1+\sqrt{2} \\ 1 & -1-\sqrt{2} \end{pmatrix}^{-1},$$

and this shows

$$e_\ell = \cdots \in \Omega\left((1+\sqrt{2})^{\ell-3}\right).$$

Now let $f, g \in \mathbb{Z}[x]$ have degrees n and $n-1$, respectively, and have a normal degree sequence and $|\mathrm{lc}(g)| \geq 65536 = 2^{16}$. Then $d_0 = 1$, $\ell = n-1$ and by Lemma 7.2

$$|\rho_\ell| \geq |\rho_1|^{e_\ell} \geq 2^{2^4 \cdot 2^{\ell-3}} = 2^{2^n}$$

for large n. □

The algorithm writes down the final result ρ_ℓ, and takes at least as much time as the bit length of $|\rho_\ell|$, which is at least 2^n.

After this "negative" result, saying that the pseudo PRS is decided by impractical, we turn to "positive" upper bounds for the other PRS. We assume $f = \sum_{0 \leq j \leq n} f_j x^j$ and $g = \sum_{0 \leq j \leq m} g_j x^j \in \mathbb{Z}[x]$ to be polynomials of degrees $n \geq m \geq 0$, respectively. For the estimates we will use the *max-norm* of f which is defined as

$$\|f\|_\infty = \max\{|f_j| : 0 \leq j \leq n\},$$

and the following famous result:

Hadamard's inequality 7.4. *Let $A \in \mathbb{Z}^{n \times n}$, with row vectors $f_1, \ldots, f_n \in \mathbb{Z}^n$, and $B \in \mathbb{Z}$ such that all entries of A are at most B in absolute value. Then*

$$|\det A| \leq n^{n/2} B^n$$

(see von zur Gathen & Gerhard (1999), Theorem 16.6).

We now seek an upper bound for the running time of both the reduced PRS and the subresultant PRS in the normal case. Therefore we first show estimations for the coefficients of q and r in the pseudo-division.

LEMMA 7.5. *Let $\|f\|_\infty \leq A$, $\|g\|_\infty \leq B$ and $|g_m| = C$. Furthermore let $q = \sum_{0\leq j\leq n-m} q_j x^j$, $r = \sum_{0\leq j\leq k} r_j x^j$ be such that $g_m^{n-m+1} f = qg + r$ and $\deg r = k < m = \deg g$. Then*

(i) $|q_{n-m-i}| \leq A(B+C)^i C^{n-m-i}$ *for* $0 \leq i \leq n-m$*,*

(ii) $\|r\|_\infty \leq A(B+C)^{n-m+1}$.

PROOF. (i) Since $\deg r < m$ we find

$$g_m^{n-m+1} f_{n-i} = q_{n-m-i} g_m + \sum_{\substack{a+b=n-i \\ a\neq n-m-i}} q_a g_b + 0. \tag{7.6}$$

Hence

$$|q_{n-m}| = |g_m^{n-m} f_n| \leq C^{n-m} A,$$

and this proves the claim for $i = 0$. Now assume $0 < i \leq n-m$. Then 7.6, $B \geq C$ and the induction hypothesis imply

$$\begin{aligned}|q_{n-m-i} g_m| &\leq |g_m^{n-m+1}| \cdot \|f\|_\infty + A(B+C)^{i-1} C^{n-m-(i-1)} B \\ &\leq A \cdot C^{n-m+1} + A(B+C)^{i-1} C^{n-m-(i-1)} B \\ &\leq A(B+C)^{i-1} C^{n-m-(i-1)+1} + A(B+C)^{i-1} C^{n-m-(i-1)} B \\ &= A(B+C)^i C^{n-m-(i-1)}.\end{aligned}$$

By induction this proves the first claim.

(ii) With Lemma 7.5(i) we get

$$\begin{aligned}\|r\|_\infty &\leq |g_m^{n-m+1}| \cdot \|f\| + \|q\|_\infty \cdot \|g\|_\infty \\ &\leq A \cdot C^{n-m+1} + A(B+C)^{n-m} B \\ &\leq A(B+C)^{n-m} C + A(B+C)^{n-m} B \\ &= A(B+C)^{n-m+1}.\end{aligned}$$

□

With Lemma 7.5 we now prove the following running time of the normal reduced PRS algorithm.

THEOREM 7.7. *Let $\|f\|_\infty, \|g\|_\infty \leq A$, $B = (n+1)^n A^{n+m}$, and let $(r_0, \ldots, r_\ell)$ be the normal reduced PRS for f, g. Then the max-norm of the r_i is at most $4B^3$, and the algorithm uses $O(n^3 m \log^2(nA))$ word operations.*

PROOF. Consider one step in the computation of the reduced PRS:

$$\alpha_i r_{i-2} = q_{i-1} r_{i-1} + \alpha_{i-1} r_i.$$

For $2 \leq i \leq \ell$ we get with Corollary 6.8 Corollary 6.8(iii) that $\sigma_{n_i}(f, g)$ is the leading coefficient of r_i. Thus Remark 3.12 and Hadamard's inequality 7.4 yield

$$\|r_i\|_\infty = \|R_{n_i}(f, g)\|_\infty \leq B.$$

Since the PRS is normal, it follows that $\alpha_i = \rho_{i-1}^2$ for $3 \leq i \leq \ell$. Hence

$$\|\alpha_i r_i\|_\infty = |\sigma_{n_{i-1}}(f, g)^2| \cdot \|R_{n_i}(f, g)\| \leq B^3.$$

Furthermore Lemma 7.5 implies

$$\begin{aligned} \|\alpha_{i-1} r_i\|_\infty &\leq B(2B)^2 = 4B^3 \\ \|q_{n-m-i}\|_\infty &\leq B(2B)^i B^{k-i} \leq 2^k B^{k+1} = 2B^2. \end{aligned}$$

So the max-norm of all intermediate results is at most $4B^3$. The number of operations in R is $O(nm)$, and the estimate follows from $\log B \in O(n \log 2(nA))$. □

Since Corollary 6.19 shows that normal reduced PRS and normal subresultant PRS agree up to sign, the estimates in Theorem 7.7 are also true for normal subresultant PRS.

PRS	**time**	
classical/Sturmian/monic	$O^{\sim}(n^8)$	
	$O^{\sim}(n^6)$	
pseudo	$\Theta((1 = \sqrt{2})^n)$	Theorem 7.3
primitive	$O^{\sim}(n^6)$	
reduced/subresultant	$O^{\sim}(n^6)$	Theorem 7.7

Table 2
Comparison of various normal PRS. The time (= word operations) is for polynomials of degree at most n in x and with coefficients of length at most n and ignores logarithmic factors.

We conclude the theoretical part of our comparison with an overview of all worst-case running times for the various normal PRS in Table 2. The length of the coefficients of f and g are assumed to be at most n. The estimates that are not proven here can be found in von zur Gathen & Gerhard (1999).

8 Experiments

We have implemented six of the PRS for polynomials with integral coefficients in C++, using Victor Shoup's "Number Theory Library" NTL 3.5a for integer

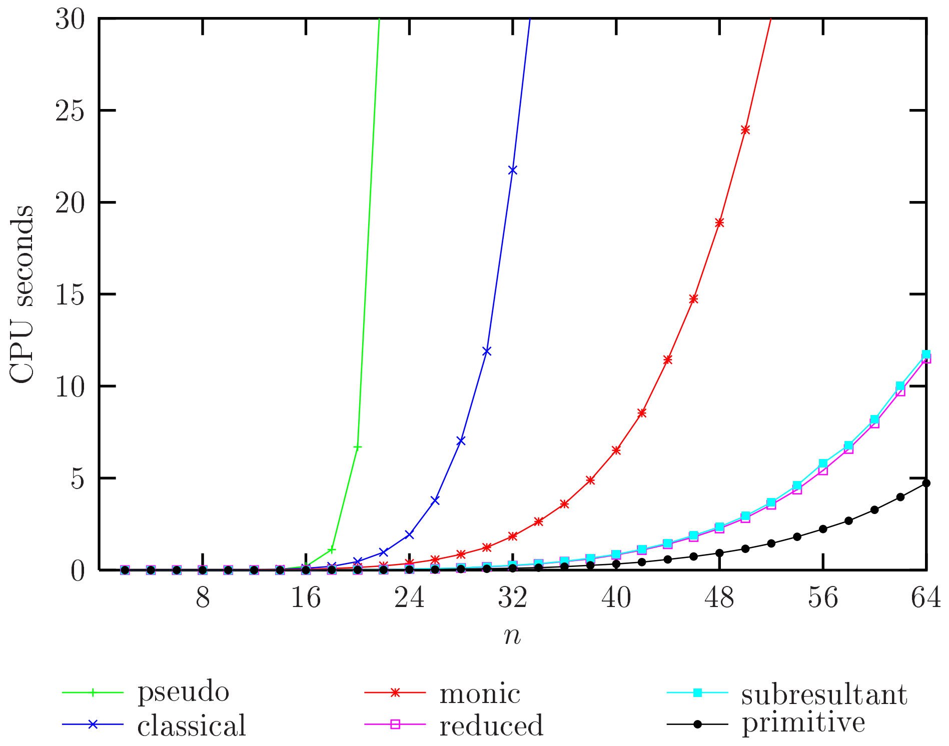

Fig. 2. Computation of polynomial remainder sequences for polynomials of degree $n-2$ with coefficients of bit length less than n for $2 \leq n \leq 64$.

and polynomial arithmetic. Since the Sturmian PRS agrees with the classical PRS up to sign, it is not mentioned here. The contents of the intermediate results in the primitive PRS are simply computed by successive gcd computations. Cooperman *et al.* (1999) propose a new algorithm that uses only an expected number of two gcd computations, but on random inputs it is slower than the naïve approach. All timings are the average over 10 pseudorandom inputs. The software ran on a Sun Sparc Ultra 1 clocked at 167MHz.

In the first experiment we pseudorandomly and independently chose three polynomials $f, g, h \in \mathbb{Z}[x]$ of degree $(n-2)/2$ with nonnegative coefficients of length less than $n/2$, for various values of n. Then we used the various PRS algorithms to compute the gcd of fh and gh. Thus the degree of the gcd was at least $(n-2)/2$; in fact, it was equal to $(n-2)/2$ in all cases when $n \geq 6$. The running times are shown in Figures 2 and 3.

As seen in Table 2 the pseudo PRS turns out to be the slowest algorithm. The reason is that for random inputs with coefficients of length at most n the second polynomial is almost never monic. Theorem 7.3 shows that then the running time for pseudo PRS is exponential. A surprising result is that the primitive PRS, even implemented in a straightforward manner, turns out to be the fastest PRS. Collins and Brown & Traub invented the subresultant PRS in order to avoid the primitive PRS since it seemed too expensive. Our

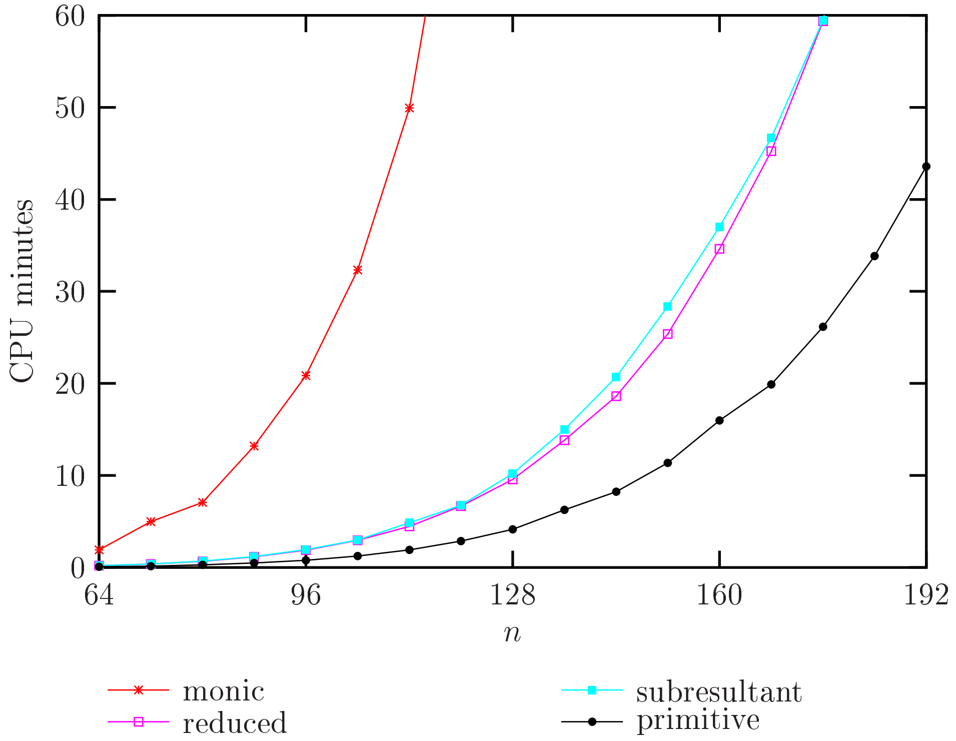

Fig. 3. Computation of polynomial remainder sequences for polynomials of degree $n - 2$ with coefficients of bit length less than n for $64 \leq n \leq 192$. Time is now measured in minutes.

tests show that this was unnecessary in case of large gcd's.

Polynomial remainder sequences of random polynomials tend to be normal. Since Corollary 6.19 shows that reduced and subresultant PRS agree up to signs in the normal case, their running times also differ by little.

We are also interested in comparing the reduced and subresultant PRS, so we construct PRS which are not normal. To this end, we pseudorandomly and independently choose six polynomials f, f_1, g, g_1, h, h_1 for various n as follows:

$$\begin{array}{rcccccccccccc}
 & F & = & (& f & \cdot & h & \cdot & x^{n/6} & + & f_1 &) & h_1 \\
 & G & = & (& g & \cdot & h & \cdot & x^{n/6} & + & g_1 &) & h_1 \\
 & & & & & & & & & & & & \\
\text{degree bound:} & n & & & \frac{n}{12} & & \frac{n}{4} & & & & \frac{n}{6} & & \frac{n}{2} \\
 & & & & & & & & & & & & \\
\text{coefficient length:} & n & & & \frac{n}{8} & & \frac{3n}{8} & & & & \frac{n}{2} & & \frac{n}{2}
\end{array}$$

So F and G have degrees less than $n - 2$ with coefficient length less than n, and every polynomial remainder sequence of F and G has a degree jump of $\frac{n}{6}$ at degree $n - \frac{n}{12}$. Then we used the various PRS algorithms to compute the gcd of F and G. The running times are illustrated in Figures 4 and 5.

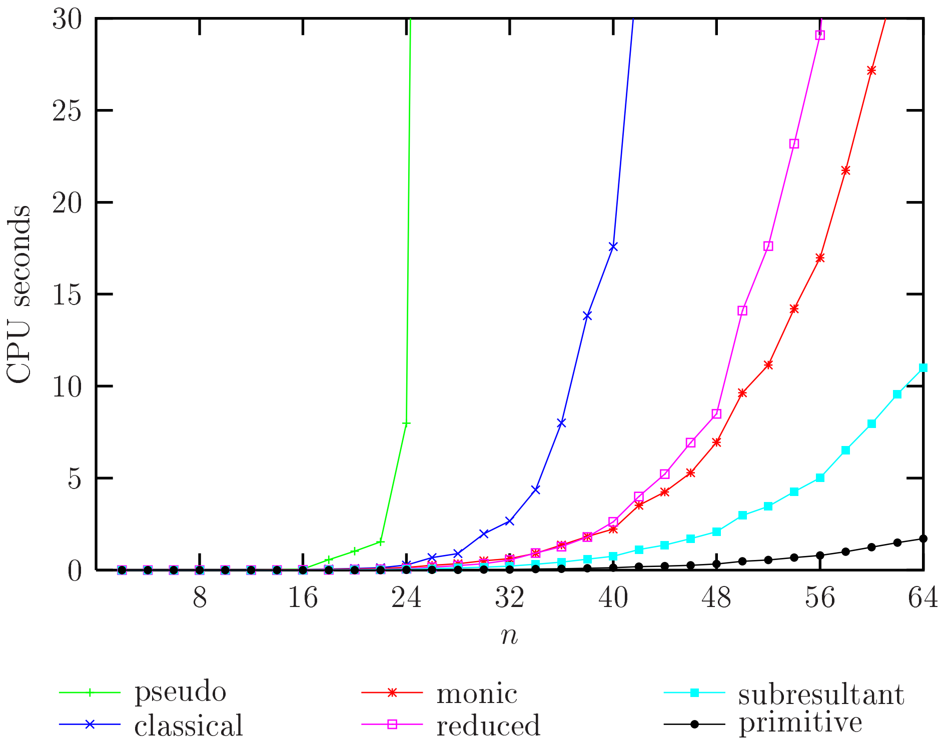

Fig. 4. Computation of non-normal polynomial remainder sequences for polynomials of degree $n-2$ with coefficient length less than n and a degree jump of $\frac{n}{6}$ at degree $n-\frac{n}{12}$, for $2 \leq n \leq 64$.

As in the first test series the pseudo PRS turns out to be the slowest, and the primitive PRS is the fastest. Here the monic PRS is faster than the reduced PRS. Since the PRS is non-normal, the coefficients become quite large, as seen in Theorem 6.18.

We already find running times for reduced and primitive PRS in Collins (1967), p. 140. He used a IBM 7094 computer to calculate the gcd of two polynomials of degrees $5k$ with random integer coefficients of two decimal digits for various k. His results are in Table 3. He found the reduced PRS to be faster than the primitive PRS. This difference is presumably due to the fact that two pseudorandom polynomials are usually coprime. Thus the PRS is longer and the coefficient growth influences the running times more than in our tests, where a half degree gcd was built in. Collins writes: "*For a nonnormal p.r.s.* $[\cdots]$ *we have no theory to indicate that the reduced p.r.s. algorithm would still be more efficient than the primitive p.r.s. algorithm*". He also reports that for larger gcd's, the primitive PRS "may even be sligthtly faster in extreme cases" than the reduced one, but that this does not seem to compensate for its relative inefficiency in the other cases.

In order to illustrate the dependency of the running times and the degree of the gcd's, we implemented one more test. We pseudorandomly and independently chose two polynomials f and g of degrees $63-k$ with bit length less than

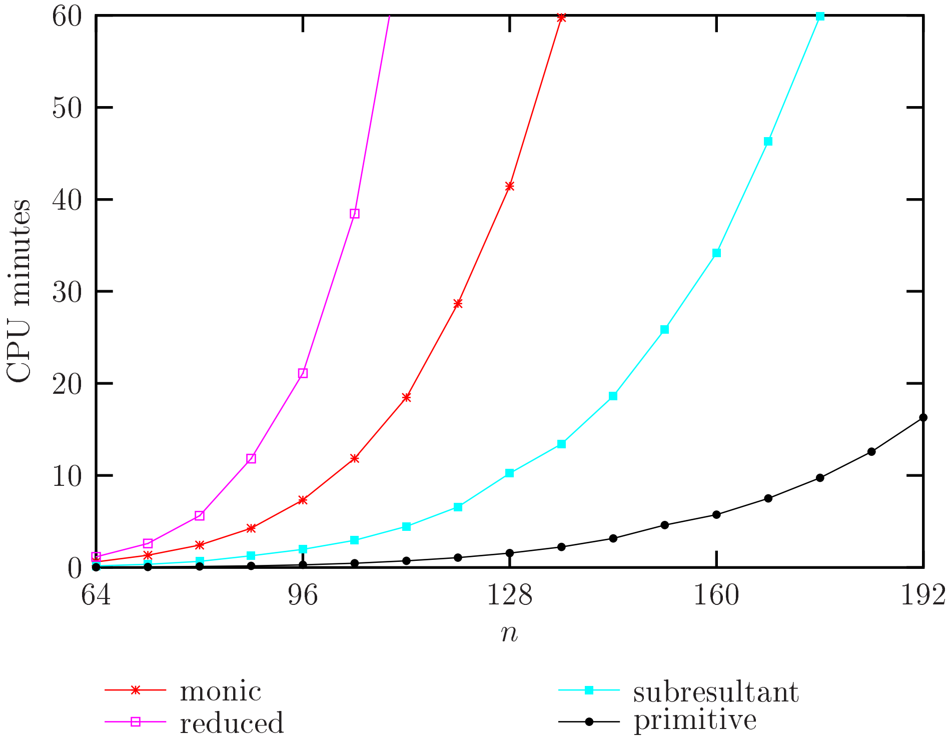

Fig. 5. Computation of non-normal polynomial remainder sequences for polynomials of degree $n-2$ with coefficient length less than n and a degree jump of $\frac{n}{6}$ at degree $n-\frac{n}{12}$, for $64 \le n \le 192$. Time is now measured in minutes.

Degree	primitive	reduced
5	0.009	0.0043
10	0.064	0.023
15	0.22	0.077
20	0.51	0.21
25	1.06	0.43
30	1.79	0.78
35	3.25	1.48

Table 3
Running times from Collins (1967), p. 140, in minutes.

$64-k$, and a polynomial h of degree k and with bit length less than k. Then we used the various PRS to compute the gcd of fh and gh. So the running times of the PRS only depended on the size of the gcd. The result is in Figure 6. For small gcd's the reduced PRS is faster than the primitive PRS, but this changes for growing gcd's. Thus the choice of the optimal PRS is output-driven: it depends on the degree of the gcd. In practice, one has to make this decision beforehand, however. For "random" inputs, the expected deg gcd is small, and one will favor the reduced PRS. If one has reason to expect deg gcd to be large, one will choose the primitive PRS; this may be the case, e.g., in recursive (primitive) PRS computations for multivariate polynomials.

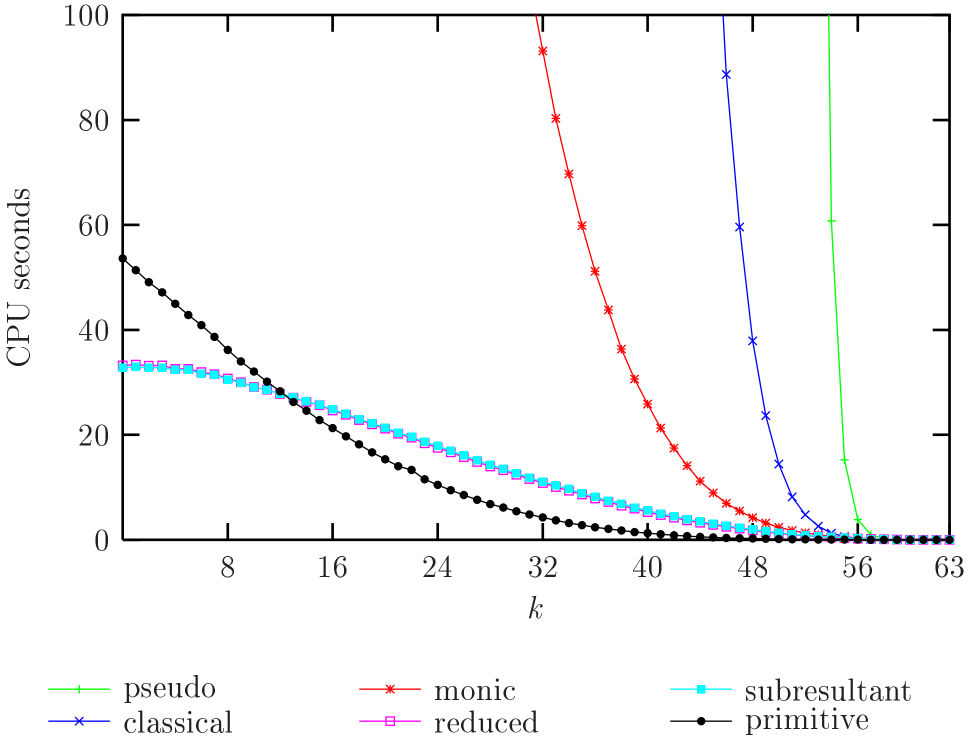

Fig. 6. Computation of polynomial remainder sequences for polynomials of degree 63 with coefficients of bit length less than 64 and gcd of degree k with coefficients of bit length less than k for $0 \leq k \leq 63$.

References

ÉTIENNE BÉZOUT (1764). Recherches sur le degré des équations résultantes de l'évanouissement des inconnues. *Histoire de l'académie royale des sciences* 288–338. Summary 88–91.

OTTO BIERMANN (1891). Über die Resultante ganzer Functionen. *Monatshefte fuer Mathematik und Physik* 143–146. II. Jahrgang.

W. S. BROWN (1971). On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *Journal of the ACM* **18**(4), 478–504.

W. S. BROWN (1978). The Subresultant PRS Algorithm. *ACM Transactions on Mathematical Software* **4**(3), 237–249.

W. S. BROWN & J. F. TRAUB (1971). On Euclid's Algorithm and the Theory of Subresultants. *Journal of the ACM* **18**(4), 505–514.

G. E. COLLINS (1966). Polynomial remainder sequences and determinants. *The American Mathematical Monthly* **73**, 708–712.

G. E. COLLINS (1973). Computer algebra of polynomials and rational functions. *The American Mathematical Monthly* **80**, 725–755.

GEORGE E. COLLINS (1967). Subresultants and Reduced Polynomial Remainder Sequences. *Journal of the ACM* **14**(1), 128–142.

GEORGE E. COLLINS (1971). The Calculation of Multivariate Polynomial Resultants. *Journal of the ACM* **18**(4), 515–532.

GENE COOPERMAN, SANDRA FEISEL, JOACHIM VON ZUR GATHEN & GEORGE HAVAS (1999). GCD of Many Integers. In *COCOON '99*, T. ASANO ET AL., editor, number 1627 in Lecture Notes in Computer Science, 310–317. Springer-Verlag.

LEONHARD EULER (1748). Démonstration sur le nombre des points où deux lignes des ordres quelconques peuvent se couper. *Mémoires de l'Académie des Sciences de Berlin* **4**, 1750, 234–248. Eneström 148. *Opera Omnia*, ser. 1, vol. 26, Orell Füssli, Zürich, 1953, 46–59.

JOACHIM VON ZUR GATHEN (1984). Parallel algorithms for algebraic problems. *SIAM Journal on Computing* **13**(4), 802–824.

JOACHIM VON ZUR GATHEN & JÜRGEN GERHARD (1999). *Modern Computer Algebra*. Cambridge University Press.

JOACHIM VON ZUR GATHEN & THOMAS LÜCKING (2000). Subresultants revisited. In *Proceedings of LATIN 2000,* Punta del Este, Uruguay, GASTÓN H. GONNET, DANIEL PANRIO & ALFREDO VIOLA, editors, number 1776 in Lecture Notes in Computer Science, 318–342. Springer-Verlag.

K. O. GEDDES, S. R. CZAPOR & G. LABAHN (1992). *Algorithms for Computer Algebra*. Kluwer Academic Publishers.

PAUL GORDAN (1885). *Vorlesungen über Invariantentheorie. Erster Band: Determinanten*. B. G. Teubner, Leipzig. Herausgegeben von GEORG KERSCHENSTEINER.

WALTER HABICHT (1948). Eine Verallgemeinerung des Sturmschen Wurzelzählverfahrens. *Commentarii Mathematici Helvetici* **21**, 99–116.

M. W. HASKELL (1892). Note on resultants. *Bulletin of the New York Mathematical Society* **1**, 223–224.

THOMAS W. HUNGERFORD (1990). *Abstract Algebra: An Introduction*. Saunders College Publishing, Philadelphia PA.

C. G. J. JACOBI (1836). De eliminatione variabilis e duabus aequationibus algebraicis. *Journal für die Reine und Angewandte Mathematik* **15**, 101–124.

Donald E. Knuth (1981). *The Art of Computer Programming, vol.2, Seminumerical Algorithms*. Addison-Wesley, Reading MA, 2nd edition.

Donald E. Knuth (1993). Johann Faulhaber and sums of powers. *Mathematics of Computation* **61**(203), 277–294.

L. Kronecker (1873). Die verschiedenen *Sturm*schen Reihen und ihre gegenseitigen Beziehungen. *Monatsberichte der Königlich Preussischen Akademie der Wissenschaften, Berlin* 117–154.

L. Kronecker (1881). Zur Theorie der Elimination einer Variabeln aus zwei algebraischen Gleichungen. *Monatsberichte der Königlich Preussischen Akademie der Wissenschaften, Berlin* 535–600. *Werke*, Zweiter Band, ed. K. Hensel, Leipzig, 1897, 113–192. Reprint by Chelsea Publishing Co., New York, 1968.

Thomas Lickteig & Marie-Françoise Roy (1997). Cauchy Index Computation. *Calcolo* **33**, 331–357.

R. Loos (1982). Generalized Polynomial Remainder Sequences. *Computing* **4**, 115–137.

Thomas Lücking (2000). Subresultants. Diplomarbeit.

Thom Mulders (1997). A note on subresultants and the Lazard/Rioboo/Trager formula in rational function integration. *Journal of Symbolic Computation* **24**(1), 45–50.

Isaac Newton (1707). *Arithmetica Universalis, sive de compositione et resolutione arithmetica liber*. J. Senex, London. English translation as *Universal Arithmetick: or, A Treatise on Arithmetical composition and Resolution*, translated by the late Mr. Raphson and revised and corrected by Mr. Cunn, London, 1728. Reprinted in: Derek T. Whiteside, *The mathematical works of Isaac Newton*, Johnson Reprint Co, New York, 1967, p. 4 ff.

Daniel Reischert (1997). Asymptotically Fast Computation of Subresultants. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation ISSAC '97*, Maui HI, Wolfgang W. Küchlin, editor, 233–240. ACM Press.

V. Strassen (1983). The computational complexity of continued fractions. *SIAM Journal on Computing* **12**(1), 1–27.

C. Sturm (1835). Mémoire sur la résolution des équations numériques. *Mémoires présentés par divers savants à l'Acadèmie des Sciences de l'Institut de France* **6**, 273–318.

J. J. Sylvester (1840). A method of determining by mere inspection the derivatives from two equations of any degree. *Philosophical Magazine* **16**, 132–135. *Mathematical Papers* **1**, Chelsea Publishing Co., New York, 1973, 54–57.

Alexei Yu. Uteshev & Timofei M. Cherkasov (1998). The Search for the Maximum of a Polynomial. *Journal of Symbolic Computation* **25**, 587–618.

Franz Winkler (1996). *Polynomial Algorithms in Computer Algebra*. Texts and Monographs in Symbolic Computation. Springer-Verlag.

Richard Zippel (1993). *Effective polynomial computation*. Kluwer Academic Publishers.