On Euclid's Algorithm and the Theory of Subresultants

W. S. BROWN

Bell Telephone Laboratories, Incorporated Murray Hill, New Jersey

AND

J. F. TRAUB

Carnegie-Mellon University, Pittsburgh, Pennsylvania

ABSTRACT. This paper presents an elementary treatment of the theory of subresultants, and examines the relationship of the subresultants of a given pair of polynomials to their polynomial remainder sequence as determined by Euclid's algorithm. Two important versions of Euclid's algorithm are discussed. The results are essentially the same as those of Collins, but the presentation is briefer, simpler, and somewhat more general.

KEY WORDS AND PHRASES: algebra, coefficient growth, Euclid's algorithm, greatest common divisors, intermediate expression swell, polynomial remainder sequences, polynomials, subresultants

CR CATEGORIES: 5.0, 5.9

1. Introduction

According to Knuth [1, p. 294], "Euclid's algorithm, which is found in Book 7, Propositions 1 and 2 of his Elements (c. 300 B.C.), and which many scholars conjecture was actually Euclid's rendition of an algorithm due to Eudoxus (c. 375 B.C.),... is the oldest nontrivial algorithm which has survived to the present day."

The algorithm, as presented by Euclid, computes the positive greatest common divisor of two given positive integers. However it is readily generalized to apply to polynomials in one variable over a field, and further to polynomials in any number of variables over any unique factorization domain in which greatest common divisors can be computed. Since a polynomial in several variables may be viewed as a polynomial in one variable with polynomial coefficients, we may confine our attention to polynomials in one variable, with no loss of generality.

In [2] Brown examines the computation of polynomial greatest common divisors by various generalizations of Euclid's algorithm including the recently developed modular algorithm. The present paper is a theoretical companion to [2]. The results are essentially the same as those of Collins [3]; however our presentation is briefer, simpler, and somewhat more general.
Euclid's algorithm for computing greatest common divisors (GCD's) of integers is presented in Section 2, and is generalized in Section 3 to apply to polynomials. Section 4 introduces the problem of coefficient growth, which has motivated the search for improvements to the algorithm. The theory of subresultants in Section 5 is essential to the understanding and verification of Collins' two algorithms, presented in Sections 6 and 7, and is also essential to the proof that the newer "modular algorithm," presented in [2], terminates. Finally, Section 8 summarizes the results.

2. Integer Remainder Sequences

Let \( a_1, a_2 \) be positive integers with \( a_1 \geq a_2 \), and let \( \gcd(a_1, a_2) \) denote their (positive) greatest common divisor. To compute this GCD, Euclid's algorithm constructs the integer remainder sequence \( a_1, a_2, \ldots, a_k \), where \( a_i \) is the positive remainder from the division of \( a_{i-1} \) by \( a_i \), for \( i = 3, \ldots, k \), and where \( a_k \) divides \( a_{k-1} \) exactly. That is,

\[
a_i = a_{i-1} - q_i a_{i-2}, \quad 0 < a_i < a_{i-1}, \quad i = 3, \ldots, k,
\]

and \( a_k \mid a_{k-1} \). From this it is easy to see that \( \gcd(a_1, a_2) = \gcd(a_2, a_3) = \cdots = \gcd(a_{k-1}, a_k) = a_k \), and therefore \( a_k \) is the desired GCD.

3. Polynomial Remainder Sequences

Let \( \mathfrak{s} \) be a unique factorization domain \([4, \text{pp. 74-77}]\), and let \( \mathfrak{s}[x] \) denote the domain of polynomials in \( x \) with coefficients in \( \mathfrak{s} \). The goal of this section is to generalize the preceding algorithm to apply to polynomials in \( \mathfrak{s}[x] \).

For \( F \in \mathfrak{s}[x] \) let \( \partial(F) \) denote the degree of \( F \), and let \( \partial(0) = -\infty \). Since the familiar process of polynomial division with remainder requires exact divisibility in the coefficient domain, it is usually impossible to carry it out for nonzero \( F, G \in \mathfrak{s}[x] \). However, the process of pseudo-division \([1, \text{p. 369}]\) always yields a unique pseudo-quotient \( Q = \text{pquo}(F, G) \) and pseudo-remainder \( R = \text{prem}(F, G) \), such that \( g_0 F = QG + R \) and \( \partial(R) < \partial(G) \), where \( g_0 \) is the leading coefficient of \( G \), and \( \delta = \partial(F) - \partial(G) \).

For nonzero \( F, G \in \mathfrak{s}[x] \) we say that \( F \) is similar to \( G \) (\( F \sim G \)) if there exist nonzero \( a, b \in \mathfrak{s} \) such that \( aF = bG \). Here \( a \) and \( b \) are called coefficients of similarity.

For nonzero \( F_1, F_2 \in \mathfrak{s}[x] \) with \( \partial(F_1) \geq \partial(F_2) \), let \( F_1, F_2, \ldots, F_k \) be a sequence of nonzero polynomials such that \( F_i \sim \text{prem}(F_{i-2}, F_{i-1}) \) for \( i = 3, \ldots, k \), and \( \text{prem}(F_{k-1}, F_k) = 0 \). Such a sequence is called a polynomial remainder sequence (PRS). From the definitions, it follows that there exist nonzero \( a_i, b_i \in \mathfrak{s} \) and \( Q_i \sim \text{pquo}(F_{i-2}, F_{i-1}) \) such that

\[
\beta_i F_i = a_i F_{i-2} - Q_i F_{i-1}, \quad \partial(F_i) < \partial(F_{i-1}), \quad i = 3, \ldots, k.
\]

Because of the uniqueness of pseudo-division, the PRS beginning with \( F_1 \) and \( F_2 \) is unique up to similarity. Furthermore, it is easy to see that \( \gcd(F_1, F_2) \sim \gcd(F_2, F_3) \sim \cdots \sim \gcd(F_{k-1}, F_k) \sim F_1 \). Thus the construction of the PRS yields the desired GCD to within similarity. For a complete GCD algorithm, we need a little more apparatus.

For nonzero \( F \in \mathfrak{s}[x] \), we define the content, denoted by \( \text{cont}(F) \), to be a GCD of the coefficients; this is unique up to multiplication by a unit. We also define the
primitive part of $F$ by the relation $F = \text{cont}(F) \text{pp}(F)$. If the coefficients of $F$ are relatively prime, $F$ is said to be primitive; hence $\text{pp}(F)$ is primitive by construction.

Since the GCD of two polynomials in $\mathbb{S}[x]$ is the product of the GCD of their contents and the GCD of their primitive parts, we may confine our attention to the case of primitive polynomials, provided only that we have some way to obtain all required GCD's in $\mathbb{S}$. But if $F_1$ and $F_2$ are primitive, their GCD is also primitive, and therefore $\gcd(F_1, F_2) = \text{pp}(F_1)$.

4. Coefficient Growth

The most obvious way to construct a PRS is to choose $F_i = \text{prem}(F_{i-2}, F_{i-1})$ for $i = 3, \ldots, k$. Collins calls this the Euclidean PRS algorithm. For many domains $\mathbb{S}$, including the integers, this method is thoroughly impractical because the coefficients grow exponentially (see [2]) as we proceed through the sequence.

To minimize coefficient growth, we set $F_i = \text{pp}(\text{prem}(F_{i-2}, F_{i-1}))$ for $i = 3 \ldots, k$. Collins calls this the primitive PRS algorithm. Unfortunately this method requires many coefficient GCD computations, which may be very time consuming.

The purpose of the algorithms presented in Sections 6 and 7 is to reduce coefficient growth without computing coefficient GCD's.

5. Subresultants

From (2) it follows by induction on $i$ that there exist $\gamma_i \in \mathbb{S}$ and $A_i, B_i \in \mathbb{S}[x]$ such that

$$A_i F_1 + B_i F_2 = \gamma_i F_i, \quad \partial(A_i) = n_i - n_{i-1}, \quad \partial(B_i) = n_i - n_{i-1},$$

$$i = 3, \ldots, k, \quad (3)$$

where $n_j = \partial(F_j)$ for $j = 1, \ldots, k$. Furthermore, if $\gamma_i$ is given, it is easy to show that (3) determines $A_i$ and $B_i$ uniquely.

Equating the coefficients of like powers of $x$ in (3), we obtain a system of $n_1 + n_2 - n_{i-1} - n_i$ homogeneous linear equations and $n_i + 1$ inhomogeneous linear equations in the $n_1 + n_2 - 2n_{i-1} + 2$ unknown coefficients of $A_i$ and $B_i$. Now the first $n_1 + n_2 - 2n_{i-1} + 1$ homogeneous equations and the first inhomogeneous one constitute a square system, whose determinant is nonzero since a unique solution exists. After arriving at the solution by another route, we shall return briefly to this linear system.

We now define

$$S_i(F, G) = \det \begin{bmatrix} f_0 & g_0 \\ \vdots & \vdots \\ f_{\gamma_1-1} & f_0 \\ \vdots & \vdots \\ f_{\gamma_2-2} & g_{\gamma_2-2} \\ f_{\gamma_1-1} & g_{\gamma_1-1} \\ x^{\gamma_1-1} F & x^{\gamma_1-1} G \end{bmatrix}, \quad (4)$$

Journal of the Association for Computing Machinery, Vol. 18, No. 4, October 1971
for $0 \leq j < \min (\varphi, \gamma)$, where $\varphi = \partial(F)$, $\gamma = \partial(G)$, $F = f_0 x^\varphi + \cdots + f_\varphi$, and $G = g_0 x^\gamma + \cdots + g_\gamma$. (Both here and subsequently, any coefficient with a subscript out of range is defined to be zero.) Expanding (4) in minors of the last row, we obtain

$$S_j(F, G) = U_j F + V_j G, \quad \partial(U_j) \leq \gamma - j - 1, \quad \partial(V_j) \leq \varphi - j - 1,$$

for $0 \leq j < \min (\varphi, \gamma)$. On the other, if we expand each occurrence of $F$ and $G$ in the last row, we find that

$$S_j(F, G) = \sum s_j f_i^2,$$

where

$$s_j = \det \begin{bmatrix} f_0 & g_0 \\ \vdots & \vdots \\ f_{\varphi - j - 1} & f_0 \\ g_{\varphi - j - 1} & g_0 \\ \vdots & \vdots \\ f_{\varphi + \gamma - 2j - 2} & f_{\varphi - j - 1} \\ g_{\varphi + \gamma - 2j - 2} & g_{\varphi - j - 1} \\ f_{\varphi + \gamma - 2j - 1} & f_{\varphi - j} \\ g_{\varphi + \gamma - 2j - 1} & g_{\varphi - j} \end{bmatrix},$$

and where the sum goes from $I = - (\varphi + \gamma - 2j - 1)$ through $I = j$. However, for each negative value of $I$ the last row of the determinant is identical to one of the earlier rows, and so $s_j = 0$. It follows that $S_j(F, G) = s_j f_i^2$, for $0 \leq j < \min (\varphi, \gamma)$, and in particular

$$S_0(F, G) = \det \begin{bmatrix} f_0 & g_0 \\ \vdots & \vdots \\ f_{\varphi - 1} & f_0 \\ g_{\varphi - 1} & g_0 \\ \vdots & \vdots \end{bmatrix},$$

which is the classical resultant [5, Ch. 12] of $F$ and $G$. Since all of the coefficients of $S_j$ are determinants of submatrices of this resultant, $S_j$ is called the $j$th subresultant of $F$ and $G$. (This terminology was introduced by Collins. Bocher [6] uses the word to refer to the determinants of certain submatrices of the resultant, while Householder [7] calls our $S_j$ polynomial bigradients.)

Now if we replace $F$ by $F_1$, $G$ by $F_2$, and $j$ by $n_{i-1} - 1$ in (5), we see that $U_j$ and $V_j$ provide a solution (to within similarity) of the linear system derived from (3), and therefore

$$F_i \sim S_{n_{i-1} - 1}(F_1, F_2)$$

for $i = 3, \cdots, k$. Of course, the same result can be obtained by using Cramer's
On Euclid's Algorithm and the Theory of Subresultants

rule [4, p. 286] to solve that linear system, and in doing so one easily discovers the definition (4).

The primary goal of this section is to rederive (8) in such a way that the coefficients of similarity appear explicitly in a useful form. Our fundamental theorem shows further that each of the polynomials $S_j(F_1, F_2)$ (for $0 \leq j < n_2$) is either similar to one of the $F_i$ (for $i = 3, \ldots, k$), or is zero.

In order to prove the fundamental theorem, we shall use two lemmas. The first corresponds to a single step in the construction of a PRS (2), provided we make the substitutions

$$
F = \alpha_i F_{i-2},
G = F_{i-1},
H = \beta_i F_i,
B = -Q_i,
$$

as is done in the proof of the second.

**Lemma 1.** Let $F, G, H,$ and $B$ be nonzero polynomials in $\mathbb{R}[x]$, of degrees $\varphi, \gamma, \eta,$ and $\beta$, respectively, such that

$$
F + BG = H,
\varphi \geq \gamma > \eta,
\beta = \varphi - \gamma,
$$

and let

$$
F = f_0 x^\varphi + \cdots + f_\varphi,
G = g_\gamma x^\gamma + \cdots + g_\gamma,
H = h_\eta x^\eta + \cdots + h_\eta,
B = b_\beta x^\beta + \cdots + b_\beta.
$$

Then

$$
S_j(F, G) = (-1)^{\varphi-\gamma-\eta} g_\gamma S_j(G, H), \quad 0 \leq j < \eta,
$$

$$
S_\gamma(F, G) = (-1)^{\varphi-\gamma-\eta} g_\gamma h_\eta S_\gamma(G, H),
$$

$$
S_j(F, G) = 0, \quad \eta < j < \gamma - 1,
$$

$$
S_{\gamma-1}(F, G) = (-1)^{\varphi-\gamma-1} g_\gamma h_\eta H.
$$

**Proof.** Equating the coefficients of like powers of $x$ in (10), we find

$$
\begin{bmatrix}
  f_0 & g_0 \\
  \vdots & \vdots \\
  g_\gamma & g_\gamma \\
  f_\gamma & g_\gamma \\
\end{bmatrix}
\begin{bmatrix}
  1 \\
  b_\gamma \\
  \vdots \\
  b_\gamma \\
\end{bmatrix}
= \begin{bmatrix}
  0 \\
  \vdots \\
  h_\gamma \\
  h_\gamma \\
\end{bmatrix}
= \begin{bmatrix}
  h_{\varphi+\gamma} \\
  \vdots \\
  h_\gamma \\
  h_\gamma \\
\end{bmatrix}.
$$

It follows immediately that
for all $j$ and $l$ with $0 \leq j < \gamma$ and $1 \leq l \leq \gamma - j$, where the bottom row is obtained directly from (10). (Here $0_m$ denotes a column of $m$ zeros, while $0_{mn}$ denotes an $m$ by $n$ array of zeros.)

Since the left side of (17) represents the sum of the $l$th column of the $F$ portion of (4) and a linear combination of columns $l$ through $l + \beta$ of the $G$ portion, we may replace the $Z$th column of the $F$ portion of (4) by the right side of (17) without altering the value of the determinant. Doing this for all $l = 1, \cdots, \gamma - j$, and then interchanging the two portions, we obtain

$$S_j(F, G) = (-1)^{(\gamma - j)(\gamma - j - 1)} \det \begin{bmatrix} y_0 & h_{-\beta+q} \\ \vdots & \vdots \\ y_{\gamma-j-1} & h_{-\beta+q} \\ g_{\gamma-j} & h_{-\beta+q} \\ h_{\gamma-j} & \cdots \\ G & H \end{bmatrix}. \quad (18)$$

If $j \geq \eta$, all elements above the main diagonal are zero, so

$$S_j(F, G) = (-1)^{(\gamma - j)(\gamma - j - 1)} g_0 h_{-\beta+q} H_{\gamma-j-1}, \quad \eta \leq j \leq \gamma - 1, \quad (19)$$

which proves (13)-(15). On the other hand, if $j < \eta$, the determinant has the form

$$\begin{bmatrix} T & 0 \\ X & S_j(G, H) \end{bmatrix}, \quad (20)$$

where $T$ is triangular with all $\phi - \gamma$ elements on its main diagonal equal to $g_0$, and (12) follows immediately.

**Lemma 2.** Let $F_1, F_2, \cdots, F_k$ be a PRS in $s[x]$. Let $c_i$ and $n_i$ be the leading coefficient and the degree, respectively, of $F_i$ (for $i = 1, \cdots, k$), and let $\delta_i = n_i - n_{i+1}$ (for $i = 1, \cdots, k-1$). Then for $i = 3, \cdots, k$,

$$S_j(F_{i-2}, F_{i-1}) \alpha_{i-1} = S_j(F_{i-1}, F_i) \alpha_{i-1} \beta_i \alpha_{i-1} \beta_i \alpha_{i-1} \beta_i (-1)^{(\delta_i-1)\delta_i}, \quad 0 \leq j < n_i, \quad (21)$$

$$S_{n_i}(F_{i-2}, F_{i-1}) \alpha_{i-1} = F_i \beta_i \alpha_{i-1} \beta_i \alpha_{i-1} \beta_i (-1)^{(\delta_i+1)\delta_i}, \quad (22)$$
On Euclid's Algorithm and the Theory of Subresultants

\[ S_j(F_{i-2}, F_{i-1}) = 0, \quad n_i < j < n_{i-1} - 1, \]  
\[ S_{n_{i-1}-1}(F_{i-2}, F_{i-1})a_i = F_i\beta_i\delta_i^{i-1+i}(-1)^{i-1+i}. \]  

PROOF. Substituting (9) into (12)–(15) and using the identity
\[ S_j(aF, bG) = a^r b^r S_j(F, G), \]
which follows directly from (4), we obtain (21)–(24), respectively.

**Fundamental Theorem.** Let \( F_1, F_2, \cdots, F_k \) be a PRS in \( \mathfrak{g}[x] \). Then in the notation of Lemma 2,
\[ S_j(F_1, F_2) = 0, \quad 0 \leq j < n_k, \]  
\[ S_n(F_1, F_2) \prod_{i=3}^k \alpha_i^{n_i-n_i-1} = F_{i-1}^{i-1-i} \prod_{i=3}^k \beta_i^{i-1-i} \delta_i^{i-1+i}(-1)^{i-1+i}(n_i-1-n_i-i), \]  
\[ S_n(F_1, F_2) = 0, \quad n_i < j < n_{i-1}-1, \]
\[ S_{n_{i-1}-1}(F_1, F_2) \prod_{i=3}^k \alpha_i^{n_i-n_i-1} = F_{i-1}^{i-1-i} \prod_{i=3}^k \beta_i^{i-1-i} \delta_i^{i-1+i}(-1)^{i-1+i}(n_i-1-n_i-i), \]

for \( i = 3, \cdots, k. \)

**Remark.** This theorem accounts for all of the \( S_j(F_1, F_2), 0 \leq j < n_k. \) It should be noted that (28) is vacuous when \( \delta_i \leq 2, \) and furthermore (27) and (29) are identical when \( \delta_i = 1. \)

**Proof.** From (21) we see that
\[ S_j(F_1, F_2) \prod_{i=3}^k \alpha_i^{n_i-n_i-1-j} = S_j(F_{i-1}, F_{i-1}) \prod_{i=3}^k \beta_i^{i-1-i} \delta_i^{i-1+i}(-1)^{i-1+i}(n_i-1-n_i-i), \]

for \( 0 \leq j < n_{i-1} \) and \( 3 \leq i \leq k+1. \) When \( i = k + 1, \) this yields
\[ S_j(F_1, F_2) \sim S_j(F_{k-1}, F_k) \]

for \( 0 \leq j < n_k. \) Since \( \text{prem}(F_{k-1}, F_k) = 0 \) by the definition of a PRS, it follows by an argument similar to that used in the proof of Lemma 1 that the right side of (31) is zero, and this proves (26). Otherwise let \( i \) be fixed in the interval \( 3 \leq i \leq k. \) When \( j = n_i, \) eqs. (22) and (30) yield (27). When \( n_i < j < n_{i+1}, \) eqs. (23) and (30) yield (28). Finally, when \( j = n_{i+1}, \) eqs. (24) and (30) yield (29).

**6. The Reduced PRS Algorithm**

In this section we present Collins' reduced PRS algorithm, and we show that the resulting PRS satisfies
\[ F_i = \rho_i S_{n_{i-1}-1}(F_1, F_2), \quad \rho_i \in \mathfrak{g}, \quad i = 3, \cdots, k. \]

Referring to (2), if we choose
\[ \alpha_i = c^{i-1}_i, \quad i = 3, \cdots, k, \]

*Journal of the Association for Computing Machinery, Vol. 18, No. 4, October 1971*
then

\[ \beta_i F_i = \text{prem}(F_{i-2}, F_{i-1}), \quad i = 3, \ldots, k. \]  

(34)

Surprisingly, if we choose

\[ \beta_1 = 1, \]
\[ \beta_i = \alpha_{i-1}, \quad i = 4, \ldots, k, \]  

(35)

then the right hand side of (34) is exactly divisible by \( \beta_i \), for \( i = 3, \ldots, k \).

Equations (34) and (35) define the reduced PRS algorithm. To justify it, we first observe that (34) and (35) define a PRS in \( \mathcal{F}[x] \), where \( \mathcal{F} \) is the quotient field of \( \mathcal{S} \). Next, letting \( \rho_i \in \mathcal{S} \) be defined by (32), we shall prove that \( \rho_i \in \mathcal{S} \).

By (29) and (32),

\[
\rho_i = c_{i-1}^{l_i-1} \prod_{j=3}^{l_i} \left[ \left( \frac{\alpha_j}{\beta_j} \right)^{n_{i-1} - n_{i-2} - 1} \right] \times \left( -1 \right)^{n_{i-1} - n_{i-2} - 1} \]  

(36)

Inserting (33) and (35), we find

\[
\rho_i = \left( -1 \right)^{l_i - 1} \prod_{j=3}^{l_i} \left[ c_{j-1}^{l_j-1} \left( -1 \right)^{n_{j-1} - n_{j-2} - 1} \right], \quad i = 3, \ldots, k, \]  

(37)

and it follows by induction that \( \rho_i \in \mathcal{S} \), as was to be shown.

The removal of \( \beta_i \) from \( \text{prem}(F_{i-2}, F_{i-1}) \) is very helpful in controlling coefficient growth. If the PRS is normal (that is, if \( \delta_j = 1 \) for \( j = 2, \ldots, k - 2 \)), then \( \rho_i = \pm 1 \) for \( i = 3, \ldots, k \). However, if the PRS is abnormal, the growth of \( \rho_i \) can be very serious, as discussed in [2].

7. The Subresultant PRS Algorithm

The idea of the subresultant PRS algorithm is to choose the \( \alpha_i \) and \( \beta_i \) so that the \( F_i \) are precisely the nonzero subresultants of \( F_1 \) and \( F_2 \); that is, we want to make \( \rho_i = 1 \) for \( i = 3, \ldots, k \). We shall prove that this is achieved by using (34) with

\[ \beta_i = \left( -1 \right)^{l_i + 1} \]  

(38)

and

\[ \beta_i = -c_{i-1}^{l_i-1} \psi_i^{l_i-1}, \quad i = 4, \ldots, k, \]  

(39)

where

\[ \psi_i = -1 \]  

(40)

and

\[ \psi_i = \left( -c_{i-1}^{l_i-1} \right)^{l_i - 1} \psi_i^{l_i-1}, \quad i = 4, \ldots, k. \]  

(41)

At the present time it is not known whether or not these equations imply \( \psi_i \), \( \beta_i \in \mathcal{S} \). In any event \( \psi_i \) and \( \beta_i \) belong to \( \mathcal{S} \), and they yield the desired PRS of subresultants in \( \mathcal{S}[x] \).
It is possible to eliminate $\psi_i$ from (39), and write $\beta_i$ explicitly as a product of powers of $c_1, \cdots, c_{i-2}$, and $(-1)$. The resulting formula is given in [3].

We shall now verify (38)-(41). Returning to (36), and setting $\rho_3 = 1$, we obtain (38). Next, for $i = 4, \cdots, k$, we set $\rho_i/\rho_{i-1} = 1$ to obtain (39), where

$$\psi_i = (-1)^{n_1+i} c_{i-2}^{\frac{i-1}{2}} \prod_{j=1}^{i-1} \alpha_j$$

(42)

with

$$\sigma_i = \sum_{j=1}^{i-1} (\delta_j + 1) = n_i - n_{i-2} + i - 3.$$  

(43)

Substituting (33), (38), (39), and (43) into (42), we find

$$\psi_4 = -c_2^\sigma$$

and

$$\psi_i = (-1)^{n_i-i-n_1+i} c_2^{\frac{i-1}{2}} \prod_{j=1}^{i-1} \alpha_j^2 \psi_i^{-i-1}$$

(44)

for $i = 5, \cdots, k$. Dividing $\psi_i$ by $\psi_{i-1}$, we now obtain (41) for $i = 5, \cdots, k$.

Since $\psi_3$ does not appear in (39) or (45), we are free to define it as we like. Setting $\psi_3 = -1$, in agreement with (40), we extend the validity of (41) to $i = 4$, and this completes the proof.

Since the polynomials $F_3, \cdots, F_k$ generated by this algorithm are subresultants of $F_1$ and $F_2$, their coefficients can be bounded, as discussed in [2].

### 8. Summary

The computation of polynomial greatest common divisors is intimately related to the construction of polynomial remainder sequences (Section 3), and this construction is a direct generalization of Euclid's algorithm for integers (Section 2). The most obvious methods for constructing a PRS are the Euclidean PRS algorithm (Section 4), which suffers severely from coefficient growth, and the primitive PRS algorithm (Section 4), which requires many time consuming computations of coefficients GCD's.

The search for better methods leads naturally to the theory of subresultants (Section 5), and eventually to the fundamental theorem. This theorem states that

$$F_i \sim S_{n_1}(F_1, F_2) \sim S_{n_{i-1}}(F_1, F_2), \quad i = 3, \cdots, k,$$

(46)

where “$\sim$” denotes similarity (Section 3). The coefficients of similarity are given explicitly, and it is shown that any subresultants $S_i(F_1, F_2)$ not included in (46) are zero.

From this theorem one can easily justify the reduced PRS algorithm (Section 6) and derive the subresultant PRS algorithm (Section 7). Both of these algorithms reduce the coefficient growth without requiring computations of coefficient GCD's, and furthermore they are identical (up to signs) whenever the PRS is normal. However, since the reduced PRS algorithm can suffer severely from coefficient growth when the PRS is abnormal, the subresultant PRS algorithm is preferred.

The fundamental theorem finds further applications in [2], where it is used in
the proof that the "modular algorithm" for polynomial GCD's terminates, and in
[8] and [9].

REFERENCES
   1969.
2. BROWN, W. S. On Euclid's algorithm and the computation of polynomial greatest common
   divisors. Proc. 2nd Symposium on Symbolic and Algebraic Manipulation. ACM, New York,
3. COLLINS, G. E. Subresultants and reduced polynomial remainder sequences. J. ACM
   14 (1967), 128-142.
   York, 1965.
7. HOUSEHOLDER, A. S. Bigradients and the problem of Routh and Hurwitz. SIAM Rev. 10
   (1968), 56-66.
8. COLLINS, G. E. The calculation of multivariate polynomial resultants. Proc. 2nd Symposi­
   um on Symbolic and Algebraic Manipulation. ACM, New York, 1971, pp. 212-222; J. ACM.
9. HEINDEL, L. E. Algorithms for Exact Polynomial Root Calculation, Ph.D. Diss., U. of