STUDIA UNIV. "BABEŞ-BOLYAI", INFORMATICA, Volume XLII, Number 2, 1997

On Some Polynomial GCD Algorithms

DRAGOŞ POP

Abstract. This paper presents an algorithm for computing the greatest common divisor of two polynomials, which does not use the traditional method of polynomial remainders sequence. Instead, it uses addition/subtraction of pairs of polynomials and divisions of a polynomial with powers of X.

This algorithm is then generalized for the computation of the greatest common divisor for an arbitrary number of polynomials. With succesive changes, we obtain an algorithm that computes solutions for Hermite Pade problems and a base for the null space of m polynomials.

1. Introduction

The computation of polynomial greatest common divisor is required in nearly all computations with polynomial or rational functions and the performance of a symbolic computation system largely depends on the quality of the gcd algorithm used.

The classical algorithm are based on a generalization of the Euclidean algorithm to polynomials. It uses the successive reduction of the problem to a similar one for polynomials whose degrees are less than the degrees of the previous polynomials. The decreasing degrees of the polynomials guarantee that after a finite number of steps, one of the polynomials becomes 0 and the other represents the GCD of the initial polynomials.

Let F and G be polynomials with rational coefficients. The Euclidean GCD algorithm is based on the following property: $\gcd(F, G) = \gcd(G, F \mod G)$. We can suppose that $\deg(F) \geq \deg(G)$. At each step, the degrees of the polynomials involved are reduced, since $\deg(G) \leq \deg(F)$ and $\deg(F \mod G) < \deg(G)$ and the first inequality becomes strict after the first step.

Received by the editors: October 20, 1997.

1991 *Mathematics Subject Classification.* 68Q40.

1991 *CR Categories and Descriptors.* I.1.2 [**Algebraic Manipulation**]: Algorithms – *Algebraic algorithms, Analysis of algorithms* .

2. The polynomial binary GCD algorithm

We can obtain alternative GCD algorithms if we use other reduction rules. For example, we can consider the following properties:

$$\gcd(F, G) = \gcd(F + kG, G), \text{ for } k \text{ a nonzero rational number },$$

and

$$gcd(X^i F, G) = gcd(F, G), \text{ if } G(0) \neq 0.$$

The first relation can be used to vanish $F(0)$ and thus, to guarantee that $i = \max\{j \in \mathbf{N} | X^j \text{ divides } F\} > 0$. The degree of F remains unchanged or may decrease, since $\deg(G) \leq \deg(F)$. In these conditions, using the second relation, we will decrease the degree of F. If after this operation $\deg(F) < \deg(G)$, the polynomials F and G will be interchanged. We obtained the following algorithm:

1. Compute $i = \max\{l \in \mathbf{N} | X^l \text{ divides } F\}$ and divide F by X^i.
2. Compute $j = max\{l \in \mathbf{N} | X^l \text{ divides } G\}$ and divide G by X^j.
3. $k \leftarrow \min(i, j)$ and interchange F and G, if $\deg(F) < \deg(G)$.
4. **while** $G \neq 1$ **do begin**
   $F \leftarrow \text{prim}(F - \frac{F(0)}{G(0)} G)$ (prim(F) represents the primitive part of F)
   Compute $i = \max\{l \in N | X^l \text{ divides } F\}$ and divide F by X^i.
   if $\deg(F) < \deg(G)$ **then** interchange F and G.
   end;
5. Returns $X^k F$.

We observe that this algorithm uses only additions/subtractions of polynomials, multiplications with a constant value and coefficient shifting (the divisions with powers of X). The polynomial divisions, which are at the base of the classical GCD algorithms but are rather costly operations, are completely avoided. Moreover, it is easy to see that if at step 4 we replace $F \leftarrow prim(F - \frac{F(0)}{G(0)} G)$ $F \leftarrow G(0)F - F(0)G$, we can use this algorithm for polynomials with coefficients in a ring.

As in the case of the classical algorithms, this algorithm stops when one of the polynomials becomes 0. The decreasing degrees of the polynomials, guarantee that the algorithms stops after a finite number of steps.

Another advantage of this algorithm is that it could be easily generalized for an arbitrary number of polynomials. Let n be an integer, $n > 1$ and F_1, F_2, ...,$F_n \in K[X]$ where K is a ring. We will use the following properties:

$$\gcd(F_1, F_2, \ldots, F_n) = \gcd(F_1 + k_1 F_i, \ldots, F_i, \ldots, F_n + k_n F_i), \text{ for any } k_i \in K$$

$$\gcd(X^{i_1} F_1, X^{i_2} F_2, \ldots, F_i, \ldots, X^{i_n} F_n) = \gcd(F_1, F_2, \ldots, F_n), \text{ if } F_i(0) \neq 0.$$

As in the case of two polynomials, we will alternatively apply these relations. First, we use F_i to vanish the free terms of the other polynomials. F_i is chosen such that it has the minimum degree among the given polynomials. Using the second relation, the degrees of the other polynomials are decreased. After each stage, the

polynomials that become 0 are eliminated and n is adjusted. The algorithm stops when $n = 1$.

1. Compute $j_i = \max\{l \in \mathbf{N} | X^l \text{ divides } F_i\}$ and divide F by X^{j_i}, $i = 1, \ldots, n$.
2. $k \leftarrow \min\{j_1, j_2, \ldots, j_n\}$.
3. **while** $n > 0$ **do begin**
   Compute j such that $\deg(F_j) = \min\{\deg(F_i) | i = 1, \ldots, n\}$
   Compute $m_i = \max\{l \in \mathbf{N} | X^l \text{ divides } F_i\}$ for each $i = 1, \ldots, n, i \neq j$.
   Eliminate polynomials $F_i = 0$, renumber polynomials F_i and adjust n.
   end;
4. Returns $X^k F_1$, where F_1 is the last polynomial that remains nonzero.

3. Applications

Our goal is to prove that a modified version of this algorithm can be used to determine other values which are useful in the polynomial computation. Let $FI = F = (F_1, F_2, \ldots, F_m)$ the initial vector of polynomials and let $n = (n_1, n_2, \ldots, n_m)$ be the vector of their degree. If P is the m^{th} order unity matrix, then $P \cdot FI = F$.

In the previous section, all the polynomials, excepting a pivot polynomial, are divided by X^{m_1} after vanishing the coefficient of $X^0, \ldots, X^{m_1 - 1}$, with $m_i \geq 1$. We will change this step as follows: after the elimination of the terms in X_0, the pivot polynomial is multiplied by X. At the next steps, the terms in $X^1, X^2, \ldots$ will be eliminate using different pivot polynomials.

Meantime, we will try to modify the polynomial matrix P such that the relation $P \cdot FI = F$ remains true. If at step l, F_j is the pivot polynomial, the line P_j will be considered as pivot line and the operations:

$$F_i \leftarrow F_i - \frac{coef(F_i, l)}{coef(F_j, l)} F_j, \text{ for } i = 1, \ldots, n, i \neq j, \text{ and } F_j \leftarrow F_j \cdot X$$

will be followed by:

$$P_i \leftarrow P_i - \frac{coef(F_i, l)}{coef(F_j, l)} P_j, \text{ for } i = 1, \ldots, n, i \neq j, \text{ and } P_j \leftarrow P_j \cdot X$$

After step l, the polynomials F_i will have nonzero terms only for powers of X greater than l. We remark that as in the previous case, the polynomials that vanishes and the corresponding lines of the matrix P will be ignored in the next steps. However, these lines will provide a basis for the null space of the polynomials $\{F_1, F_2, \ldots, F_m\}$, which represents one of the applications of this algorithm. We also remark that using only this kind of transformation on the lines P_i of matrix P, they remain linear independent with respect to polynomial coefficients, after each step.

In the previous section, the polynomial with the minimum degree is selected as pivot polynomial, because $F_i(0) \neq 0$ for all $i = 1, \ldots, n$. Now, we will select as pivot polynomial at step l, the polynomial with the minimum degree among the polynomials with a nonzero coefficient of X^l.

Another application of the modified algorithm which will be presented below is solving Hermite-Padé approximation problems. We define this problem as follows.

Definition 3.1. *Let $F = (F_1, F_2, \ldots, F_m)^T$ be an m-tuple of power series with coefficients from a field K and $n = (n_1, n_2, \ldots, n_m)$ an m-tuple of integers, $n_i \geq -1$. A Hermite-Padé approximant for F of type n is a nontrivial tuple $P = (P_1, P_2, \ldots, P_m)$ of polynomials over K, having degrees bounded by n_i, such that:*

$$P(z) \cdot F(z) = P_1(z) \cdot F_1(z) + P_2(z) \cdot F_2(z) + \ldots + P_m(z) \cdot F_m(z) = c_N z^N + c_{N+1} z^{N+1} + \ldots$$

with $N = n_1 + \ldots + n_m + m - 1 = |n|$.

Definition 3.2. *The defect of $P = (P_1, P_2, \ldots, P_m) \in K^m[z]$ with respect to $n = (n_1, n_2, \ldots, n_m)$ is $dct(P) = \min\{n_l + 1 - \deg(P_l) | l = 1, \ldots, m\}$, where the zero polynomial has degree $-\infty$. The order of P with respect to F is defined by $ord(P) = \sup\{\sigma \in N_0 | P(z) \cdot F(z) = z^\sigma \cdot R(z)$, with $R \in K[[z]]\}$.*

A Hermite-Padé problem has a set of solutions, which will be denoted by:

$$L_\delta^s igma = \{P \in K^m[z] | \text{dct} > -\delta \text{ and ord}(P) \geq \sigma\}$$

for $\sigma \in \mathbf{N}_0$ and $\delta \in Z \cup \{+\infty\}$.

Beckermann and Labahn introduced the so-called σ-bases of a Hermite-Padé problem and proved some properties of the

defect and the order, with respect to arithmetic operations between two solutions:

Definition 3.3. *Let $\sigma \in N_0$. The system $P_1, P_2, \ldots, P_m \in K^m[z]$ is called a σ-bases if and only if:*

i): $P_1, P_2, \ldots, P_m \in L_{+\infty}^\sigma$, *i.e.* $ord(P_i) \geq \sigma$

ii): *For each $\delta \in Z \cup \{+\infty\}$ and for each $Q \in L_\delta^\sigma$ there exists one and only one tuple of polynomials $(\alpha_1, \ldots, \alpha_m)$, $\deg(\alpha_i) < dct(P_i) + \delta$ such that $Q = \alpha_1 P_1 + \cdots + \alpha_m P_m$.*

Note that as a consequence, the polynomial vectors $P_1, P_2, \ldots, P_m$ must be linearly independents with respect to polynomial coefficients.

Lemma 3.4. *If $P, Q \in K^m[z]$, $c \in K - \{0\}$ then:*

i): $dct(c \cdot P) = dct(P)$, $dct(P+Q) \geq \min\{dct(P), dct(Q)\}$, $dct(z \cdot P) = dct(P) - 1$.

ii): $ord(c \cdot P) = ord(P)$, $ord(P+Q) \geq \min\{ord(P), ord(Q)\}$, $ord(z \cdot P) = ord(P) + 1$

We are interested about the dimension of L_δ^σ. It is clear that $L_\delta^{\sigma+1} \subseteq L_\delta^\sigma$. If $P \in L_\delta^\sigma - L_\delta^{\sigma+1}$, then $\text{ord}(P) = \sigma$ and from the above properties, it is easy to see that for each $Q \in L_\delta^\sigma$ – there exists a $c \in K$ such that $Q - c \cdot P \in L_\delta^{\sigma+1}$. This implies $\dim L_\delta^{\sigma+1} \geq \dim L_\delta^\sigma - 1$.

Considering the coefficients of the polynomials from P_i as unknowns, one can observe that the relation $P_i(z) - F(z) = z^l - R(z)$ is a homogenous linear system of l equations (the first l coefficient of the polynomial in the right member are 0). It follows that a Hermite-Padé problem of order σ has at least $|n| - \sigma$ linearly independent solutions over K. This means that the number of linearly independent solutions is decreasing to 1 when l is increasing. This provide us the stop condition for the algorithm.

1. Initialize P as the m^{th} order unity matrix
   $l \leftarrow 0$; $k \leftarrow m$;

2. **repeat**
   $T \leftarrow \{1 \leq i \leq n | \text{coef}(F_i, l) \neq 0\}$;
   if $T = \emptyset$ **then** $l \leftarrow l + 1$ and **goto** 2;
   let $j \in T$ such that $\deg(F_j) = \min\{\deg(F_i) | i \in T\}$;
   $F_i \leftarrow F_i - \frac{\text{coef}(F_i,l)}{\text{coef}(F_j,l)} F_j$ and $P_i \leftarrow P_i - \frac{\text{coef}(F_i,l)}{\text{coef}(F_j,l)} P_j$ for all $i \in T$, $i \neq j$;
   $F_j \leftarrow F_j \cdot X$ and $P_j \leftarrow P_j \cdot X$;
   decrease m with the number of polynomials F_i that became 0 at this step;
   $l \leftarrow l + 1$;
   until $m = 1$;

After step σ, the lines of the matrix P that correspond to nonzero entries in the vector of polynomials F, represent a σ-base for the Hermite-Padé problem of order σ. When the algorithm stops, the last nonzero polynomial entry of F is $X^l \cdot D(X)$, where $D(X)$ is a factor of $\gcd(F_1, F_2, \ldots, F_m)$. The other factor is a power of X, that could be determined at the beginning of the algorithm, as in the first version of the binary gcd algorithm.

References

[1] Beckermann B., Labahn G., *A uniform approach for the fast computation of matrix-type Padé approximants*, in SIAM Journal of Matrix Analysis Appl, vol 15, no. 3/1994.
[2] Knuth DE, *The Art of Computer Programming*, Addison-Wesley, 1973
[3] Zippel R., Effective polynomial computation, Kluwer Academic Publishers, 1993.
[4] Van Barel M., Bultheel A., *A new approach to the rational interpolation problem*, J. Comput. Appl. Math., 32/1990.

"Babeş-Bolyai" University, Faculty of Mathematics and Computer Science, RO-3400 Cluj-Napoca, Romania
E-mail address: dragos@cs.ubbcluj.ro