MATHEMATICS OF COMPUTATION, VOLUME 32, NUMBER 144
OCTOBER 1978, PAGES 1215–1231

An Improved Multivariate Polynomial Factoring Algorithm

By Paul S. Wang*

Abstract. A new algorithm for factoring multivariate polynomials over the integers based on an algorithm by Wang and Rothschild is described. The new algorithm has improved strategies for dealing with the known problems of the original algorithm, namely, the leading coefficient problem, the bad-zero problem and the occurrence of extraneous factors. It has an algorithm for correctly predetermining leading coefficients of the factors. A new and efficient p-adic algorithm named EEZ is described. Basically it is a linearly convergent variable-by-variable parallel construction. The improved algorithm is generally faster and requires less store then the original algorithm. Machine examples with comparative timing are included.

1. Introduction. A much improved algorithm for factoring multivariate polynomials over the rational integers, **Z**, has been developed. It is generally faster and requires less storage than the original algorithm as described by Wang and Rothschild [13]. The original algorithm has been implemented for the algebraic manipulation systems MACSYMA [7] of MIT and SCRATCHPAD [4] of IBM. It is believed to be the only such algorithm fully implemented on computers. The new algorithm is significantly faster for larger multivariate polynomials.

The overall scheme of the improved algorithm is the same as before. Namely, the given polynomial is first reduced to a polynomial of only one variable by substituting properly selected integers for the other variables. The resulting univariate polynomial is factored over **Z** by an algorithm which uses Berlekamp's algorithm [2] with a small prime. The univariate factors over **Z** are then used to construct the multivariate factors by a p-adic procedure based on Hensel's lemma. The old p-adic procedure is sometimes referred to as the EZ algorithm. There are several problems with the original factoring algorithm [14]. Specific methods have been devised for the new algorithm to deal with these problems.

The old algorithm gets terms as $(x + a)^r(y + b)^s(z + c)^t$ from substitution and expands all such terms in the p-adic construction phase. This causes large intermediate

Received February 14, 1977; revised October 31, 1977.

AMS (MOS) subject classifications (1970). Primary 10M05, 12–04, 12C05; Secondary 10D05, 12E05.

*Work reported herein was supported in part by the Laboratory for Computer Science (formerly Project MAC), an M.I.T. Interdepartmental Laboratory, sponsored by the United States Department of Energy under contract E(11-1)-3070, and by the National Aeronautics and Space Administration under Grant NSG 1323.

The work reported herein was also supported in part by Kent State University and some of the materials incorporated in this work were developed with the financial support of National Science Foundation Grant NCS78-02234.

expression growth if the selected integers a, b and c are not zero. The difficulty is referred to as the *bad-zero problem.* If the leading coefficient with respect to the chosen main variable of the given polynomial is not an integer, then in the old algorithm the factors during p-adic construction are only correct up to units in the underlying ring of truncated p-adic multivariate polynomials. This is known as the *leading coefficient problem.* Another problem is the occurrence of *extraneous factors.* An extraneous factor is a factor that does not lead to a true factor after p-adic construction. The presence of such factors complicates the computation significantly. (See [14] for details.)

Dealing with the leading coefficient problem for two factors in the context of multivariate polynomial greatest common divisor computation, Yun [15] suggested that the leading coefficient of the given polynomial or an easily computable divisor of it be "imposed" on the univariate factors during p-adic construction. The improved algorithm features an algorithm by which the correct leading coefficients of the factors can be *predetermined.* Sometimes some or all other coefficients of some or all factors can also be determined. As a result, computation during p-adic construction is greatly reduced. A complete description together with an example is given in Sections 3 and 4.

To avoid the bad-zero problem, a new p-adic algorithm is devised. Essentially, the new algorithm is a linearly convergent variable-by-variable parallel p-adic construction. It is a much improved algorithm over the EZ algorithm as described by Wang and Rothschild [13]. Let us name the new p-adic procedure the EEZ (Enhanced EZ) algorithm. The EEZ algorithm lifts all factors at once. A differentiation and evaluation process eliminates the need for substitution and expansion. The variable-by-variable approach is used to reduce the number of differentiations which can be very large if all variables are lifted simultaneously. EEZ is described in Sections 5 and 6. Yun described in detail in his thesis [15] a "semiparallel" Hensel construction for multivariate factors. Whereas EEZ adopts a completely parallel approach obtaining corrections for all factors by constructing them simultaneously using another Hensel type procedure described in Section 6.

Since there is no bad-zero problem in the new algorithm, using zeros as substitution integers is no longer a restricting consideration. Therefore, extraneous factors can be largely avoided by using *several* different sets of integers for the *univariate* factorization over $\mathbf{Z}$. If these sets are randomly generated, then the probability of getting any extraneous factor is exceedingly small.

In Section 7, we discuss the importance of preserving sparseness in the manipulation of multivariate polynomials and how this is carefully done in the improved factoring algorithm. A comparison with Musser's factoring algorithm [11] is presented.

Being interested in heuristic factoring programs, Claybrook [3] has given ten polynomials, many causing both the leading coefficient problem and the bad-zero problem, for symbolic manipulation systems to do. The old factoring algorithm did very poorly on these examples. The new algorithm, which is implemented on the MACSYMA system gives a much better performance. A table of comparative timing is included in the appendix.

An algorithm for factoring multivariate polynomials over algebraic number fields [12] has been developed based on the old algorithm. It should be interesting to consider making similar improvements there. Also of interest is to consider adopting EEZ for the computation of multivariate gcd's [10].

2. Preliminaries and Notation. Let the given multivariate polynomial be $U(x, x_2, \ldots, x_t) \in \mathbf{Z}[x, x_2, \ldots, x_t]$ whose irreducible factors over $\mathbf{Z}$ are to be obtained. By choosing a main variable, say x, one can write U in the form

$$U(x, \ldots, x_t) = V_n x^n + \cdots + V_1 x + V_0$$

with $V_i \in \mathbf{Z}[x_2, \ldots, x_t]$ for $i = 0, \ldots, n$. $V_n \neq 0$ is the *leading coefficient* of U, denoted as $\mathrm{lc}(U)$. The degree of U, $\deg(U)$, with respect to the main variable x, is n. The *content* of U with respect to x, $\mathrm{CONT}(U)$, is $\mathrm{GCD}(V_0, \ldots, V_n)$ where GCD stands for the common divisor of greatest degree; and the *principal part* of U, $\mathrm{pp}(U)$, is $U/\mathrm{CONT}(U)$. U is *primitive* if $\mathrm{CONT}(U) = 1$, and U is *squarefree* if U has no repeated factors. Any content of U, or repeated factors of U can be removed by relatively simple GCD computations (see [5]). Thus U will be assumed primitive and squarefree.

For any set $F = \{f_1, f_2, \ldots, f_r\} \subset \mathbf{Z}[x_2, x_3, \ldots, x_t]$, the ideal generated by F, denoted as $(f_1, f_2, \ldots, f_r)$, is defined as the set

$$\{g_1 f_1 + g_2 f_2 + \cdots + g_r f_r : g_i \in \mathbf{Z}[x_2, \ldots, x_t] \ \forall i\}.$$

The set F need not be finite. For any integer $k > 0$ and any ideal $\mathfrak{a}$, $\mathfrak{a}^k$ denotes the ideal generated by all products of the form $h_1 h_2 \cdots h_k$, $h_i \in \mathfrak{a}$, $i = 1, 2, \ldots, k$.

If A and B are polynomials and $\mathfrak{a}$ is an ideal in $\mathbf{Z}[x, x_2, \ldots, x_t]$, we define $A \equiv B \bmod \mathfrak{a}$ if $A - B \in \mathfrak{a}$. For example, if $\mathfrak{a} = (x_2 - a_2, x_3 - a_3, \ldots, x_t - a_t)$, $a_i \in \mathbf{Z}$, then $A(x, x_2, \ldots, x_t) \equiv A(x, a_2, \ldots, a_t) \bmod \mathfrak{a}$ for $A(x, a_2, \ldots, a_t)$ is the remainder of dividing A by each $x_i - a_i$, $i = 2, \ldots, t$, successively.

Let us denote by $\langle e_2, e_3, \ldots, e_j \rangle$ the ideal $((x_2 - a_2)^{e_2}, (x_3 - a_3)^{e_3}, \ldots, (x_j - a_j)^{e_j})$. We define that $A = B \bmod \langle e_2, \ldots, e_j \rangle$ if $A \equiv B \bmod \langle e_2, \ldots, e_j \rangle$ and $\deg(A)$ in $x_i < e_i$ for all $i = 2, \ldots, j$. We denote the ideal $(x_2 - a_2, \ldots, x_j - a_j)$ by $\mathfrak{a}_j$. Thus $\mathfrak{a}_j^k$ is the ideal generated by all polynomials of the form

$$\prod_{i=2}^{j} (x_i - a_i)^{e_i} \quad \text{with} \quad \sum_{i=2}^{j} e_i = k, \ e_i \geqslant 0.$$

For the ideal $\mathfrak{a}_j$ we define, for any positive integer k, $A = B \bmod \mathfrak{a}_j^k$ if $A \equiv B \bmod \mathfrak{a}_j^k$ and the total degree of A in $x_2, \ldots, x_j$ is less than k.

For any given U a *coefficient bound*, B, can be computed [8] such that for any integer coefficient c of any divisor of U, $B > 2|c|$. The number B is used later in the algorithm.

3. Determining the Leading Coefficients. The leading coefficient of U, V_n, is *factored* over $\mathbf{Z}$,

$$V_n = \Omega F_1^{e_1} F_2^{e_2} \cdots F_k^{e_k},$$

where F_i are distinct irreducible polynomials of positive degree in $\mathbf{Z}[x_2, \ldots, x_t]$ and Ω is an integer. Let us assume that V_n is not an integer, for if it is the case is trivial. A set of integers $\{a_2, \ldots, a_t\}$ will be found satisfying the following three conditions: (1) $\widetilde{V}_n = V_n(a_2, \ldots, a_t) \neq 0$, (2) for each F_i, $\widetilde{F}_i = F_i(a_2, \ldots, a_t)$ has at least one prime divisor p_i which does not divide any $\widetilde{F}_j$, $j < i$, Ω, or the content of $U_0(x) = U(x, a_2, \ldots, a_t)$, and (3) $U_0(x)$ is squarefree. Let $\delta = \mathrm{CONT}(U_0)$ and $u(x) = \mathrm{pp}(U_0)$.

It is obvious that infinite sets of integers exist satisfying the conditions (1) and (3). In fact, any set of integers a_i that is not a solution of $V_n(x_2, \ldots, x_t) = 0$ or the resultant (with respect to x) of U and $\partial U/\partial x$ satisfies (1) and (3). Among these eligible sets, an infinity also satisfy condition (2). The proof is based on the facts: (i) if $f(x)$ and $g(x)$ are relatively prime, then there are polynomials $\alpha(x)$ and $\beta(x)$ such that $\alpha f + \beta g$ is a fixed integer; (ii) for any $f(x)$, if $\deg(f) > 0$, then $f(x)$ has an infinity of different prime divisors as x varies through integer values.

In the computer program the a_i are generated randomly modulo an integer. This integer modulus is 3 to begin with and is increased slightly every time a few sets have been generated. The set of integers thus created is then tested for suitability. It is important to avoid prime factorization in checking condition (2) because factoring large integers is not easy [5]. An algorithm which uses integer gcd computations for condition (2) is given here.

ALGORITHM N (Nondivisors): $\Omega, \widetilde{F}_1, \ldots, \widetilde{F}_k, \delta$. This algorithm takes Ω, $\widetilde{F}_i$, $i = 1, 2, \ldots, k$, and δ as input. It determines whether condition (2) is met. Integers d_i, $i = 1, \ldots, k$, will be output if the condition is satisfied. Any prime divisor of d_i may serve as p_i. Therefore, d_i may be used in place of p_i.

$d_0 \leftarrow \delta \cdot \Omega$.
For $i = 1, 2, \ldots, k$ do [steps A1, A2, A3].
A1: $q \leftarrow |\widetilde{F}_i|$.
A2: For $j = i - 1, i - 2, \ldots, 0$ do [steps B1, B2, B3, B4].
B1: $r \leftarrow d_j$.
B2: $r \leftarrow \gcd(r, q)$, $q \leftarrow q/r$.
B3: If $r \neq 1$ go to step B2.
B4: If $q = 1$, exit (condition (2) not met).
A3: $d_i \leftarrow q$.
Return the answers $d_1, \ldots, d_k$.

If condition (2) is satisfied, the $d_1, \ldots, d_k$ will be used later. If it is not satisfied, a new set of integers a_i is generated.

The polynomial $u(x) = \mathrm{pp}(U_0)$ is now factored over the integers (see [13] for details), into distinct irreducible factors, $u(x) = u_1(x) \cdots u_r(x)$. To minimize r, the univariate factorization over $\mathbf{Z}$ is carried out for a number of sets $\{a_i\}$ until several different sets give the same low r value. It follows from Hilbert's irreducibility theorem [6, p. 14] that for any irreducible polynomial $U(x, \ldots, x_t)$ over $\mathbf{Z}$ the subset $\{(a_2, \ldots, a_t)\} \subset \mathbf{Z}^{t-1}$ of points such that $U(x, a_2, \ldots, a_t)$ remains

irreducible over **Z** is dense. Experiments with the program indicate that this process virtually eliminates the possibility of extraneous factors. Thus, in almost all cases, there will be no expression growth caused by extraneous factors later in the factoring process.

If none of $u_1, \ldots, u_r$ is extraneous, then U factors into r distinct irreducible polynomials, $U = \Pi_{i=1}^{r} G_i(x, \ldots, x_t)$. Let $C_i(x_2, \ldots, x_t) = \mathrm{lc}(G_i)$, $\widetilde{C}_i = C_i(a_2, \ldots, a_t)$ and $G_i(x, a_2, \ldots, a_t) = \delta_i u_i(x)$ where δ_i is some divisor of δ.

LEMMA. *If there are no extraneous factors, then, for all i and m, F_k^m divides C_i if and only if $\widetilde{F}_k^m$ divides* $\mathrm{lc}(u_i)\delta$.

Proof. If $F_k^m \mid C_i$ then $\widetilde{F}_k^m$ divides $\widetilde{C}_i = \mathrm{lc}(u_i)\delta_i$. On the other hand, if F_k^m does not divide C_i, then $\widetilde{C}_i = \widetilde{F}_1^{s_1} \cdots \widetilde{F}_k^{s_k}\omega$ where $\omega \mid \Omega$ $s_i \geqslant 0$ and $s_k < m$. Thus, p_k^m does not divide $\widetilde{C}_i$, which implies that $\widetilde{F}_k^m$ does not divide $\mathrm{lc}(u_i)\delta$. □

This lemma enables one to distribute all F_k first, then all F_{k-1}, etc. Thus, all $D_i(x_2, \ldots, x_t) = \mathrm{pp}(C_i)$ can be determined as products of powers of F_i. Now let $\widetilde{D}_i = D_i(a_2, \ldots, a_t)$. If $\delta = 1$, then $C_i = (\mathrm{lc}(u_i)/\widetilde{D}_i)D_i$. Otherwise, if $\delta \neq 1$, the following steps are carried out for all $i = 1, \ldots, r$ to correctly distribute the factors of δ.

(1) Let $d = \mathrm{GCD}(\mathrm{lc}(u_i), \widetilde{D}_i)$ and $C_i = (\mathrm{lc}(u_i)/d)D_i$,

(2) Let $u_i = (\widetilde{D}_i/d)u_i$,

(3) Let $\delta = \delta/(\widetilde{D}_i/d)$.

Now if $\delta = 1$ the process ends. Otherwise, let $u_i = \delta u_i$, $C_i = \delta C_i$ and $U = \delta^{r-1}U$. In this case, when the true factors over **Z** of U are found they may have integer contents which should be removed.

In the above process, if any factor of V_n is not distributed, then there are extraneous factors and the program goes back for different substitutions that lower r.

For example, consider factoring the polynomial

$$\begin{aligned}
U(x, y, z) = {} & (4z^2y^4 + 4z^3y^3 - 4z^4y^2 - 4z^5y)x^6 \\
& + (z^3y^4 + 12z^3 + (-z^5 + 12z^2)y^2 - 12z^3y - 12z^4)x^5 \\
& + (8y^4 + (6z^2 + 8z)y^3 + (-4z^4 + 4z^3 - 8z^2)y^2 \\
& \qquad + (-4z^5 - 2z^4 - 8z^3)y)x^4 \\
& + (2zy^4 + z^3y^3 + (-z^5 - 2z^3 + 9z)y^2 \\
& \qquad + (-12z^3 + 12z^2)y - 12z^4 + 3z^3)x^3 \\
& + (6y^3 + (-6z^2 + 8z)y^2 + (-2z^4 - 8z^3 + 2z^2)y)x^2 \\
& + (2zy^3 - 2z^3y^2 - 3zy + 3z^3)x - 2y^2 + 2z^2y.
\end{aligned}$$

Factoring lc(U) over **Z** gives

$$V_6 = 4yz^2(y+z)^2(y-z).$$

Therefore, we have, say, $\Omega = 4$, $F_1 = y$, $F_2 = z$, $F_3 = y+z$ and $F_4 = y-z$. The sets of integers $\{-14, 3\}$, $\{5, -12\}$ and $\{-23, 3\}$ satisfy the three conditions and all give $r = 3$. Let us use $y = -14$ and $z = 3$. Thus,

$$\tilde{F}_1 = -14, \quad \tilde{F}_2 = 3, \quad \tilde{F}_3 = -11, \quad \tilde{F}_4 = -17$$

and

$$U_0(x) = U(x, -14, 3) = 1036728x^6 + 915552x^5 + 55748x^4 + 105621x^3 - 17304x^2 - 26841x - 644.$$

Since $U_0(x)$ is primitive, $u(x) = U_0(x)$ and $\delta = 1$. Upon factorization over **Z** one obtains $u(x) = u_1u_2u_3$ where

$$(1) \qquad u_1 = 187x^2 - 23, \quad u_2 = 44x^2 + 42x + 1, \quad u_3 = 126x^2 - 9x + 28.$$

Now $187 = \tilde{F}_3\tilde{F}_4$ gives $C_1 = F_3F_4 = (y^2 - z^2)$. Likewise $C_2 = -4(y+z)$ and $C_3 = -yz^2$. These are the correct leading coefficients of the irreducible factors of $U(x, y, z)$.

The algorithm can obviously be applied to determine trailing coefficients as well. To do this a fourth condition similar to (2) is required. It complicates the choice of $\{a_i\}$ and is not currently implemented.

4. Determining Other Coefficients. Sometimes it is possible to correctly determine certain other coefficients in the factors once their leading (trailing) coefficients are known. Using the same notation, let $U_0(x) = u_1u_2$ where either or both of u_1 and u_2 are sparse. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the exponent vectors, d_1 and d_2 the degrees and C_1 and C_2 the predicted leading coefficients of $u_1(x)$ and $u_2(x)$, respectively. Consider a term V_ix^i of $U(x, \ldots, x_t)$. If the set $(i - \mathbf{v}_1) \cap \mathbf{v}_2$ has one and only one element, then V_i may lead to the determination of a coefficient in u_1 or u_2 provided that either $(i - d_1) \in \mathbf{v}_2$ or $(i - d_2) \in \mathbf{v}_1$. Let us assume the former is the case. Then if C_1 divides V_i, the coefficient for x^{i-d_1} in u_2 is predicted as V_i/C_1. However, there is a slight chance that V_i/C_1 is only congruent to the true coefficient of the term modulo $\mathfrak{g}_t$. If C_1 does not divide V_i, then no coefficient prediction is made for the term x^{i-d_1}. The other case is obviously similar. This process is carried out for all i such that $i - d_1 \geqslant 0$ or $i - d_2 \geqslant 0$. At the end of this straightforward procedure some coefficients may or may not be determined. If there are new coefficients determined, the same process is applied again, using these newly found terms instead of the leading terms, in an attempt to determine still more coefficients.

The scheme is best illustrated by an example. Consider factoring the polynomial

$$U(x, y) = (24y^3 + 48y^2)x^8 + (24y^5 - 72y^2)x^7$$
$$+ (25y^4 + 2y^3 + 4y + 8)x^6 + (y^6 + y^3 - 12)x^5$$
$$+ (y^5 - y^4 - 2y^3 + 292y^2)x^4 + (-y^6 + 3y^3)x^3$$
$$+ (-y^5 + 12y^3 + 48)x^2 - 12y^3.$$

By the method described in Section 3, it is found that

$$u_1 = 5x^4 + 24x^3 + 9x^2 + 12,$$
$$u_2 = 216x^4 + 31x^2 - 27,$$
$$U_0(x) = U(x, 3) = u_1 u_2$$

and

$$C_1 = y + 2, \quad C_2 = 24y^2.$$

The exponent vectors are $\mathbf{v}_1 = (4\ 3\ 2\ 0)$ and $\mathbf{v}_2 = (4\ 2\ 0)$. Since $(7 - \mathbf{v}_2) \cap \mathbf{v}_1 = (3)$, the correct coefficient for the cubic term in u_1 is, therefore, given by $V_7/(24y^2) = y^3 - 3$. The determination of this cubic term leads, by V_5 and V_3, to the determination of the quadratic and the constant terms of u_2 giving a factor

$$24y^2x^4 + (y^3 + 4)x^2 - y^3.$$

Consequently, the other factor is also completely determined by trial division.

In describing the procedure, two factors are used. But it should be pointed out that it can be applied recursively if there are more than two factors. One way of splitting the factors into two groups may lead to more determined coefficients than another. Since we are only interested in obtaining as many coefficients as can be conveniently determined, a fixed splitting pattern is employed. It also seems possible to generalize the process for two factors to one for any number of factors. The predetermined coefficients are used later in the p-adic procedure cutting down, sometimes considerably, the amount of work needed there. If some of the additional coefficients predicted are not exactly correct but are only congruent to the actual coefficients mod $\hat{s}_t$, then they will get corrected in the p-adic construction procedure. A predicted coefficient will be discarded as soon as it changes during p-adic construction.

The EEZ algorithm produces true factors of $U(x, x_2, a_3, \ldots, a_t)$, $U(x, x_2, x_3, a_4, \ldots, a_t)$, etc. These multivariate true factors can be used to predetermine coefficients in the same manner as factors of $U_0(x)$. For instance, from the factors of $U(x, x_2, a_3, \ldots, a_t)$ coefficients as polynomials in $x_3, \ldots, x_t$ for terms of various powers in x and x_2 can be determined. This procedure can be very important in practical computations because as the number of variables increases in the true factors the probability of the correct determination of coefficients increases significantly if $U(x, x_2, \ldots, x_t)$ is sparse.

5. The EEZ Algorithm. In this section the EEZ algorithm is presented in detail. Essentially it is a linearly convergent variable-by-variable parallel p-adic construction algorithm. It is parallel because all the factors are lifted at once as opposed to building only two factors at a time. Although the EEZ is a multivariate algorithm, the parallel feature should be very useful for univariate p-adic procedures such as the Hensel algorithm and the Zassenhaus algorithm [16].

Let $u_1(x), u_2(x), \ldots, u_r(x)$ be the distinct irreducible factors computed by the method given in Section 3 such that

$$U(x, \ldots, x_t) \equiv u_1 \cdots u_r \mod \mathfrak{a}_t, \quad \text{over } \mathbf{Z}.$$

Recall that $\mathfrak{a}_j = (x_2 - a_2, \ldots, x_j - a_j)$ for $2 \leqslant j \leqslant t$. Let

$$U_2 = U(x, x_2, a_3, \ldots, a_t), \quad U_3 = U(x, x_2, x_3, a_4, \ldots, a_t),$$

etc. Thus, $U_t = U$. An algorithm is given here which constructs from u_i a series of polynomials $P_{i,j}(x, x_2, \ldots, x_j)$, for $j = 1, \ldots, t$ and i from 1 to a value less than or equal to r, such that

$$P_{i,1} = u_i \quad \text{and} \quad U_j = \prod_i P_{i,j} \tag{2}$$

over $\mathbf{Z}$. The polynomials $P_{i,2}$ are constructed from u_i, $P_{i,3}$ from $P_{i,2}$ and so forth. Thus, the factors are constructed variable-by-variable.

Let b be a large prime or a suitable prime power which is greater than the coefficient bound B. The number b is used as a modulus for all computations described in this and the next section in order to avoid arithmetic with rational numbers.

By induction, suppose $P_{i,k-1}$, $i = 1, \ldots, r'$, $r' \leqslant r$, satisfy Eq. (2) for some $k \geqslant 2$. The process for constructing polynomials $P_{i,k}$ from $P_{i,k-1}$ will be described. Let $\mathfrak{R}_{i,1}(x, \ldots, x_k)$ be $P_{i,k-1}$ with possibly some of its coefficients replaced by predetermined coefficients. Of course, these predetermined coefficients, which in general involve all the variables $x_2, \ldots, x_t$, first have to be reduced mod $(x_{k+1} - a_{k+1}, \ldots, x_t - a_t)$. Let n_i be the degree of U_k in x_i, $i = 2, \ldots, k$. A sequence of polynomials $\mathfrak{R}_{i,j}(x, \ldots, x_k)$, $j = 1, \ldots, n_k + 1$, will be constructed such that $\mathfrak{R}_{i,j} \equiv \mathfrak{R}_{i,j+1} \bmod (x_k - a_k)^j$ and

$$U_k \equiv \prod_{i=1}^{r'} \mathfrak{R}_{i,j} \mod ((x_k - a_k)^j, \Delta_k), \qquad 0 < j \leqslant n_k + 1,$$

where Δ_k *is the ideal* $\langle n_2 + 1, n_3 + 1, \ldots, n_{k-1} + 1\rangle$. Thus, j is increased by one each time to a maximum $n_k + 1$. Often many or all of $\mathfrak{R}_{i,j}$ will stop changing once j is greater than n_k/r', the average degree of x_k in the r' factors. When $j > n_k/r'$, a checking procedure which uses division is activated so that factors no longer changing can be removed. This means that one will have a reduced U_k and fewer factors to work with. The savings in computation can be significant if the factors are quite different in size and complexity.

If every $\mathfrak{R}_{i,n_k+1}$ divides U_k over $\mathbf{Z}$, then we have $P_{i,k} = \mathfrak{R}_{i,n_k+1}$ for all i. If

there are extraneous factors, which seldom happen in this new algorithm, a true factor of U_k may be the product, modulo Δ_{k+1}, of two or more $\mathfrak{R}_{i,n_k+1}$. A combinatorial search [13] is then performed to obtain $P_{i,k}$. In this case, the subscript i has a smaller range for $P_{i,k}$ than for $P_{i,k-1}$. This is the reason why r' is used as the number of factors instead of r. If there are no extraneous factors, $r' = r$.

Note that combining extraneous factors is the only place in the entire p-adic construction described in this and the next section where explicit modulo operation by a polynomial, namely division by a polynomial and taking the remainder, is used.

Now let us consider the construction of the $\mathfrak{R}_{i,j}$. Again suppose, by induction, $\mathfrak{R}_{i,m}(x, \ldots, x_k)$, $i = 1, \ldots, r'$, are obtained for some $1 \leqslant m < n_k$ such that

$$U_k \equiv \prod_{i=1}^{r'} \mathfrak{R}_{i,m} \mod ((x_k - a_k)^m, \Delta_k)$$

and

$$\mathfrak{R}_{i,m-1} \equiv \mathfrak{R}_{i,m} \mod (x_k - a_k)^{m-1}.$$

Now $\mathfrak{R}_{i,m+1}$ will be constructed from $\mathfrak{R}_{i,m}$. One first computes the difference

$$R_m = \prod_{i=1}^{r'} \mathfrak{R}_{i,m} - U_k \mod \Delta_k,$$

which is zero mod $(x_k - a_k)^m$. If $R_m \equiv 0 \mod (x_k - a_k)^{m+1}$, then $\mathfrak{R}_{i,m+1} = \mathfrak{R}_{i,m}$. If $R_m \not\equiv 0 \mod (x_k - a_k)^{m+1}$, there exist $C \not\equiv 0 \mod (x_k - a_k)$ and $D \equiv 0 \mod (x_k - a_k)^{m+1}$ such that

$$R_m = C(x, \ldots, x_{k-1})(x_k - a_k)^m + D(x, \ldots, x_k). \tag{3}$$

The degree of R_m in x is less than n, the degree of U in x, due to the correct distribution of the leading coefficients. The polynomial C is computed by the formula

$$C(x, \ldots, x_{k-1}) = \frac{1}{m!} \left(\frac{\partial^m}{\partial x_k^m} R_m \right)_{x_k = a_k}$$

It involves one differentiation and one evaluation. The polynomial C is smaller than R_m because it is lower in degree and has one less variable. In particular, $\deg(C) < n$. Note that no expansion of any sort is used.

Now r' unique polynomials $\alpha_1(x, \ldots, x_{k-1}), \ldots, \alpha_{r'}(x, \ldots, x_{k-1})$, called *correction coefficients*, will be computed such that

$$\begin{aligned} &\alpha_1 P_{2,k-1} P_{3,k-1} \cdots P_{r',k-1} + \alpha_2 P_{1,k-1} P_{3,k-1} \cdots P_{r',k-1} \\ &\quad + \cdots + \alpha_i P_{1,k-1} \cdots P_{i-1,k-1} P_{i+1,k-1} \cdots P_{r',k-1} + \cdots \\ &\quad + \alpha_{r'} P_{1,k-1} \cdots P_{r'-1,k-1} \equiv C \mod \Delta_k \end{aligned} \tag{4}$$

and

$$\deg(\alpha_i) < \deg(\mathfrak{R}_{i,m}) = \deg(P_{i,k-1}).$$

These correction coefficients are obtained by a p-adic procedure which will be explained in the next section. Now one sets

$$\mathfrak{R}_{i,m+1} = \mathfrak{R}_{i,m} - \alpha_i(x_k - a_k)^m, \quad i = 1, \ldots, r'. \tag{5}$$

Thus, $\mathfrak{R}_{i,m} \equiv \mathfrak{R}_{i,m+1} \bmod (x_k - a_k)^m$ for all i and it can be deduced from (3), (4) and (5) that

$$U_k \equiv \prod_{i=1}^{r'} \mathfrak{R}_{i,m+1} \mod ((x_k - a_k)^{m+1}, \Delta_k).$$

Given that the leading coefficients are correctly distributed, then all α_i exist in $\mathbf{Z}[x_1, \ldots, x_{k-1}]$ provided that there are no extraneous factors. This means that, under these conditions, Eq. (4) can be satisfied over $\mathbf{Z}$ without modulus. This situation is very desirable because considerable expression growth results from introducing inverses of polynomials modulo Δ_k. In fact, this growth problem may be enough to warrant a procedure for early detection of extraneous factors. One such procedure is to do the construction from one variable to two variables several times using a different second variable each time. If extra factors are found for a certain second variable, then these factors are combined by the true factors procedure mentioned before. Otherwise, all the $t - 1$ alternatives shall be exhausted. In either case, the possibility of extraneous factors showing up later in the p-adic construction is largely eliminated.

As an example, let us carry out the p-adic construction for the polynomial $U(x, y, z)$ given in Section 3. To begin with, one has

$$U(x, y, z) \equiv P_{1,1}(x)P_{2,1}(x)P_{3,1}(x) \mod (y + 14, z - 3)$$

with

$$P_{1,1} = 187x^2 - 23, \quad P_{2,1} = 44x^2 + 42x + 1$$

and

$$P_{3,1} = 126x^2 - 9x + 28.$$

The leading coefficients are $C_1 = y^2 - z^2$, $C_2 = -4(y + z)$ and $C_3 = -yz^2$. Let the modulus be $11^8 = 214358881$. The immediate goal is to construct $P_{i,2}$, $i = 1, 2, 3$. Using C_1, C_2 and C_3, one sets

$$\mathfrak{R}_{1,1} = (y^2 - 9)x^2 - 23, \quad \mathfrak{R}_{2,1} = -4(y + 3)x^2 + 42x + 1$$

and

$$\mathfrak{R}_{3,1} = -9yx^2 - 9x + 28.$$

The difference R_1 is then computed as

$$R_1 = \mathfrak{R}_{1,1}\mathfrak{R}_{2,1}\mathfrak{R}_{3,1} - U(x, y, 3),$$

which is congruent to zero modulo $(y + 14)$. The polynomial $C(x)$ which is the

coefficient of $y + 14$ in R_1 is calculated by

$$C(x) = (\partial/\partial y R_1)_{y=-14}$$

$$= 70686x^5 + 5863x^4 + 17826x^3 - 2009x^2 - 5031x - 74.$$

The correction coefficients are found to be $\alpha_1 = -1$, $\alpha_2 = 3x$ and $\alpha_3 = 2$. Thus,

$$\mathfrak{R}_{1,2} = \mathfrak{R}_{1,1} + (y + 14) = (y^2 - 9)x^2 + y - 9,$$

$$\mathfrak{R}_{2,2} = \mathfrak{R}_{2,1} - 3x(y + 14) = -4(y + 3)x^2 - 3xy + 1$$

and

$$\mathfrak{R}_{3,2} = \mathfrak{R}_{3,1} - 2(y + 14) = -9yx^2 - 9x - 2y.$$

It turns out that $P_{i,2} = \mathfrak{R}_{i,2}$ for $i = 1, 2, 3$ and $U(x, y, 3) = P_{1,2}P_{2,2}P_{3,2}$ over $\mathbf{Z}$. Now, $P_{i,3}$ are to be computed. One resets

$$\mathfrak{R}_{1,1} = (y^2 - z^2)x^2 + y - 9, \quad \mathfrak{R}_{2,1} = -4(y + z)x^2 - 3xy + 1$$

and

$$\mathfrak{R}_{3,1} = -yz^2x^2 - 9x - 2y.$$

And one resets $R_1 = \mathfrak{R}_{1,1}\mathfrak{R}_{2,1}\mathfrak{R}_{3,1} - U(x, y, z)$ which is congruent to zero modulo $(z - 3)$. The polynomial $C(x, y)$ is again computed as

$$\begin{aligned} C(x, y) = (\partial/\partial z R_1)_{z=3} &= (-9y^4 - 12y^3 + 45y^2 + 108y + 324)x^5 \\ &+ (-18y^3 + 216y^2 + 810y)x^4 \\ &+ (-2y^4 - 9y^3 + 252y^2 + 288y + 945)x^3 \\ &+ (30y^2 + 414y)x^2 + (-2y^3 + 54y^2 + 3y - 81)x - 12y. \end{aligned} \tag{6}$$

The correction coefficients are then found to be $\alpha_1 = 6$, $\alpha_2 = xy$ and $\alpha_3 = 3x$ (see Section 6 for details). Therefore,

$$\mathfrak{R}_{1,2} = (y^2 - z^2)x^2 + y - 6z + 9,$$

$$\mathfrak{R}_{2,2} = -4(y + z)x^2 - xyz + 1$$

and

$$\mathfrak{R}_{3,2} = -yz^2x^2 - 3zx - 2y.$$

For this example, one more step of correction yields the actual factors over $\mathbf{Z}$,

$$U(x, y, z) = ((y^2 - z^2)x^2 + y - z^2)(4(y + z)x^2 + xyz - 1) \cdot (yz^2x^2 + 3xz + 2y).$$

6. Computing the Correction Coefficients. In this section a p-adic procedure for parallelly constructing the correction coefficients which satisfy Eq. (4) will be

presented. If one writes P_i for $P_{i,k-1}(x, \ldots, x_{k-1})$ and Q_i for the product $P_{i+1} \cdots P_{r'}$ then the left-hand side of Eq. (4) becomes

$$f(\alpha_1, \ldots, \alpha_{r'}) = \alpha_1 Q_1 + \alpha_2 P_1 Q_2 + \alpha_3 P_1 P_2 Q_3 + \cdots + \alpha_{r'} P_1 \cdots P_{r'-1}.$$

The objective is to find $\alpha_i(x, \ldots, x_{k-1})$ for given $P_i(x, \ldots, x_{k-1})$, $Q_i(x, \ldots, x_{k-1})$, $C(x, \ldots, x_{k-1})$ and Δ_k such that $\deg(\alpha_i) < \deg(P_i)$ in x and

$$f(\alpha_1, \ldots, \alpha_{r'}) \equiv C \mod \Delta_k.$$

Note that $\deg(C) < \Sigma \deg(P_i)$ in x. The p-adic construction process computes for all i a sequence of unique polynomials $A_{i,j}$, $i = 1, \ldots, r'$, such that $A_{i,j} = A_{i,j+1} \mod \mathfrak{s}_{k-1}^{j+1}$ and

$$f(A_{1,j}, \ldots, A_{r',j}) \equiv C \mod \mathfrak{s}_{k-1}^{j+1}, \qquad j = 0, 1, \ldots, h,$$

where h is the degree of U_k in $x_2, \ldots, x_{k-1}$. The process begins by computing $A_{i,0}$ for all i then builds these $A_{i,0}$ up to form $A_{i,1}$, then $A_{i,2}$, etc. The process ends whenever $f(A_{1,j}, \ldots, A_{r',j}) = C$ over $\mathbf{Z}$ or when j reaches h. The last set of $A_{i,j}$ will be the desired α_i. As indicated before, if there are no extraneous factors and if the leading coefficients are correctly determined, then equality over $\mathbf{Z}$ will always take place and it happens well before j reaches h.

Let $P_{i0}(x) = P_i \mod \mathfrak{s}_{k-1}$ and $Q_{i0}(x) = Q_i \mod \mathfrak{s}_{k-1}$. Unique polynomials $\alpha_1(x), \ldots, \alpha_{r'}(x)$ can be found such that $\deg(\alpha_i) < \deg(P_{i0})$ for all i and

$$f_0(\alpha_1, \ldots, \alpha_{r'}) = 1 \mod (b),$$

where b is the coefficient bound modulus mentioned earlier and $f_0 = f \mod \mathfrak{s}_{k-1}$. To compute the $\alpha_i(x)$, one computes $\beta_i(x)P_{i0}(x) + \alpha_i(x)Q_{i0}(x) = \beta_{i-1}(x)$, where $\beta_0(x) = 1$ and $\beta_{r'-1}(x) = \alpha_{r'}(x)$ by the Euclidean algorithm. The α_i are stored for further reference until no longer needed.

For any polynomial $g(x)$ with $\deg(g) < \Sigma \deg(P_{i0})$ it can be shown that

$$f_0(\beta_1(x), \ldots, \beta_{r'}(x)) = g(x) \mod (b)$$

if $\beta_i(x) = \alpha_i(x)g(x) \mod P_{i0}(x)$. Thus, if $C_0(x) = C(x, \ldots, x_{k-1}) \mod \mathfrak{s}_{k-1}$ and $A_{i,0}(x) = \alpha_i(x)C_0(x) \mod P_{i0}(x)$, then $f(A_{1,0}, \ldots, A_{r',0}) \equiv C \mod \mathfrak{s}_{k-1}$. By induction, suppose $A_{i,m}$ are already obtained for some $m \geqslant 0$ such that $D_m = f(A_{1,m}, \ldots, A_{r',m}) - C \equiv 0 \mod \mathfrak{s}_{k-1}^{m+1}$. Considering $W = D_m \mod \mathfrak{s}_{k-1}^{m+2}$, one sees that W can be written in such a form that it consists only of terms of degree $m + 1$ in $(x_2 - a_2), \ldots, (x_{k-1} - a_{k-1})$ with coefficients polynomials in x. A typical term of W looks like

$$g(x)(x_2 - a_2)^{e_2} \cdots (x_{k-1} - a_{k-1})^{e_{k-1}}, \tag{7}$$

with $e_2 + \cdots + e_{k-1} = m + 1$ and $\deg(g) < \Sigma \deg(P_{i0})$. To obtain $g(x)$, the following differentiation formula is used

$$g(x) = \frac{1}{e_2! \cdots e_{k-1}!} \left(\frac{\partial}{\partial x_2}\right)^{e_2} \cdots \left(\frac{\partial}{\partial x_{k-1}}\right)^{e_{k-1}} D_m \Big|_{x_i = a_i}. \tag{8}$$

By using this formula one avoids the costly change of variable $(y_i + a_i)$ for x_i which is used in [13] to obtain $g(x)$. For each $g(x)$ we compute a set of $\alpha_i(x)$ such that $f_0(\alpha_1, \ldots, \alpha_{r'}) \equiv g(x) \bmod (b)$. Now, each $A_{i,m}$ is modified by subtracting the product $\alpha_i(x)(x_2 - a_2)^{e_2} \cdots (x_{k-1} - a_{k-1})^{e_{k-1}}$ from it, $i = 1, \ldots, r'$. If this correction process is done for all possible exponents e_v satisfying $e_v \leqslant n_v$ and $e_2 + \cdots + e_{k-1} = m + 1$, then $A_{i,m}$ are transformed into $A_{i,m+1}$ which satisfy

$$f(A_{1,m+1}, \ldots, A_{r',m+1}) \equiv C \mod \mathfrak{B}_{k-1}^{m+2}.$$

For example, if $P_1 = (y^2 - 9)x^2 + y - 9$, $P_2 = -4(y + 3)x^2 - 3xy + 1$, $P_3 = -9yx^2 - 9x - 2y$ and $C(x, y)$ is given by (6), then $P_{i0} = u_i$ as given by (1). It is found that $A_{1,0} = 6$, $A_{2,0} = -14x$ and $A_{3,0} = 3x$. One more step in the p-adic procedure yields $A_{1,1} = 6$, $A_{2,1} = xy$ and $A_{3,1} = 3x$ which turn out to be the desired correction coefficients, i.e., $\alpha_1 = 6$, $\alpha_2 = xy$ and $\alpha_3 = 3x$.

The number of possible terms in the form (7) can be large. In the worst case, the polynomial C may have $t - 1$ variables. It means $k = t$ and there are $t - 2$ variables in (7). If d is the total degree of C in $x_2, \ldots, x_{t-1}$, then the number of such terms with $e_2 + \cdots + e_{t-1} = d$ is given by the binomial coefficient $\binom{d+t-3}{t-3}$ which is of order $O(d^{t-3})$ if d is much larger than t. This is the worst number of possible differentiations and evaluations as given by formula (8). Some of these calculations may produce $g(x) = 0$ because those terms are missing. Normally the actual number of applications of (8) is much less than the bound due to the absence of extraneous factors and the correct determination of coefficients.

To lessen this potential exponential behavior, one may consider a variable-by-variable approach to the p-adic construction of the multivariate correction coefficients α_i. Such an approach seems to be quite promising though it is highly recursive.

The algorithm described in this section can be applied to multivariate partial fraction expansion. It can also be generalized to solve any equation $\alpha_1 f_1 + \cdots + \alpha_n f_n = h$ if f_i and h are multivariate and $\gcd(f_1, \ldots, f_n) = 1$.

7. Discussion. It can be said in summary that the factoring algorithm is much improved because the leading coefficient problem is eliminated by predetermining correct leading coefficients for the factors; the bad-zero problem is significantly reduced in EEZ and extraneous factors are largely avoided. These three problems have one thing in common. That is, they all cause the intermediate polynomials to degrade into much denser polynomials than necessary. For any multivariate polynomial algorithm, it is important to preserve sparseness if the input polynomials are sparse. Two different algorithms may be equally efficient (inefficient) asymptotically for completely dense multivariate polynomials while one is much better than the other in practical (average) use because it preserves sparseness.

The new factoring algorithm also takes advantage of sparseness by trying to determine many or all other coefficients of the factors. A few more determined coefficients may simplify the subsequent p-adic construction considerably. Sometimes factorization may even be completed without any p-adic construction.

Musser [11] reported a factoring algorithm which is more efficient for

univariate polynomials than for multivariate polynomials. There are several very serious problems with this algorithm. Firstly, it recursively applies a quadratic p-adic construction for two factors. This means that the construction is carried out as many times as the number of factors being lifted. Hence, it is about that many times as expensive as a parallel construction. Secondly, the variables are lifted modulo a certain prime which is lifted last. Hence, there is no control over the frequent occurrence of extraneous factors. The presence of extra factors complicates significantly each p-adic construction and increases the number of such constructions at the same time. We use a clearly better approach of obtaining true univariate factors over Z before multivariate p-adic construction. True factors are also obtained at each stage of the variable-by-variable construction as explained in Section 5. Thirdly, Musser's algorithm frequently divides multivariate polynomials over a ring of such polynomials which means, among other things, multiplying by multivariate inverses in the ring. Neither our old algorithm nor the improved algorithm uses any such division. Fourthly, Musser's algorithm does not address the leading coefficient problem at all. The polynomial $U(x, a_2, \ldots, a_t)$ may have large coefficients if most $a_i \neq 0$ or ± 1. Large coefficients cause univariate factoring over Z to be more time consuming. This problem is common to the old and the new algorithm. Although the univariate factoring time is a minor part of the total multivariate factoring time, it is desirable to improve the univariate procedure for increased efficiency when coefficients are large. A recent paper by R. Moenck [9] proposes to do this by factoring modulo large primes of the form $M.\ 2^m + 1$ where $M \simeq m$. Another possible approach is to speed up the one-variable Hensel or Zassenhaus lifting scheme.

Acknowledgement. The author wishes to thank Joel Moses, D. Spear and B. Trager for helpful discussions and the referee for comments.

Appendix

Contained here are fifteen factoring examples done by MACSYMA using the original algorithm (A) and the improved algorithm (B). To conserve space these polynomials are given in factored form below. Polynomials #6 through #15 correspond to the ten polynomials proposed by Claybrook [3]. The timing was done on a DEC KL-10. Times listed in Table 1 are in milliseconds. A * indicates running out of storage.

TABLE 1

Polynomial	Factoring Time A	Factoring Time B	Ratio A/B
1	1950	1210	1.61
2	2564	597	4.29
3	11579	760	15.24
4	48057	4770	10.07

TABLE 1 (*continued*)

Polynomial	Factoring Time	Factoring Time	Ratio
5	55594	5840	9.52
6	*	3303	–
7	956	954	1.00
8	*	7830	–
9	*	5121	–
10	*	9069	–
11	*	5921	–
12	274	277	1.00
13	3388	583	5.81
14	10518	2822	3.73
15	79676	582	136.90

The fifteen polynomials

(1) $$(Z + XY + 10)(XZ + Y + 30)(YZ + X + 20),$$

(2) $$(X^3(Z + Y) + Z - 11)(X^2(Z^2 + Y^2) + Y + 90),$$

(3) $$(YZ^3 + XYZ + Y^2 + X^3)(X(Z^4 + 1) + Z + X^3Y^2),$$

(4) $$(Z^2 - X^3Y + 3)(Z^2 + XY^3)(Z^2 + X^3Y^4)(Y^4Z^2 + X^2Z + 5),$$

(5) $$(Z^2 + X^3Y^4 + U^2)((Y^2 + X)Z^2 + 3U^2X^3Y^4Z + 19Y^2)(U^2Y^4Z^2 + X^2Z + 5),$$

(6) $$(W^4Z^3 - XY^2Z^2 - W^4X^5Y^6 - W^2X^3Y)(-X^5Z^3 + YZ + X^2Y^3) \cdot (W^4Z^6 + Y^2Z^3 - W^2X^2Y^2Z^2 + X^5Z - X^4Y^2 - W^3X^3Y),$$

(7) $$(Z + Y + X - 3)^3(Z + Y + X - 2)^2,$$

(8) $$(-15Y^2Z^{16} + 29W^4X^{12}Y^{12}Z^3 + 21X^3Z^2 + 3W^{15}Y^{20}) \cdot (-Z^{31} - W^{12}Z^{20} + Y^{18} - Y^{14} + X^2Y^2 + X^{21} + W^2),$$

$$U^4XZ^2(6W^2Y^3Z^2 + 18U^2W^3XZ^2 + 15UZ^2 + 10U^2WXY^3)$$

$$\cdot(-44UWXY^4Z^4 - 25U^2W^3YZ^4 + 8UWX^3Z^4 - 32U^2W^4Y^4Z^3 \tag{9}$$

$$+ 48U^2X^2Y^3Z^3 - 12Y^3Z^2 + 2U^2WX^2Y^2 - 11UW^2X^3Y - 4W^2X),$$

$$(31U^2XZ + 35W^2Y^2 + 6XY + 40WX^2)(U^2W^2XY^2Z^2 + 24U^2WXY^2Z^2$$

$$+ 12U^2XY^2Z^2 + 24U^2X^2YZ^2 + 43WXYZ^2 + 31W^2YZ^2 + 8U^2W^2Z^2$$

$$+ 44UW^2Z^2 + 37U^2Y^2Z + 41Y^2Z + 12WX^2YZ + 21U^2WXYZ + 23XYZ \tag{10}$$

$$+ 47U^2W^2Z + 13UW^2X^2Y^2 + 22XY^2 + 42U^2W^2Y^2 + 29W^2Y^2 + 27UW^2X^2Y$$

$$+ 37W^2XZ + 39UWXZ + 43UX^2Y + 24XY + 9U^2WX^2 + 22U^2W^2),$$

$$XY(-13U^3W^2XYZ^3 + W^3Z^3 + 4UXY^2 + 47XY)$$

$$\cdot(43UX^3Y^3Z^3 + 36U^2W^3XYZ^3 + 14W^3X^3Y^3Z^2 - 29W^3XY^3Z^2$$

$$-20U^2W^2X^2Y^2Z^2 + 36U^2WXY^3Z - 48UWX^3Y^2Z + 5UWX^2Y^3 \tag{11}$$

$$+ 36UW^2Y^3 - 9UWY^3 - 23UWX^3Y^2 + 46UX^3Y^2 + 8XY^2 + 31U^2W^3Y^2$$

$$- 9U^2Y^2 + 45X^3 - 46U^2WX),$$

$$(Z + Y + X - 3)^3, \tag{12}$$

$$(3Z^3 + 2WZ - 9Y^3 - Y^2 + 45X^3)(W^2Z^3 + 47XY - W^2), \tag{13}$$

$$(-18X^4Y^5 + 22Y^5 - 26X^3Y^4 - 38X^2Y^4 + 29X^2Y^3 - 41X^4Y^2 + 37X^4) \tag{14}$$

$$\cdot(33X^5Y^6 + 11Y^2 + 35X^3Y - 22X^4),$$

$$X^6Y^3Z^2(3Z^3 + 2WZ - 8XY^2 + 14W^2Y^2 - Y^2 + 18X^3Y) \tag{15}$$

$$\cdot(-12W^2XYZ^3 + W^2Z^3 + 3XY^2 + 29X - W^2).$$

Department of Mathematics
Kent State University
Kent, Ohio 44242

1. E. R. BERLEKAMP, "Factoring polynomials over large finite fields," *Math. Comp.*, v. 24, 1970, pp. 713–735. MR 43 #1948.

2. E. R. BERLEKAMP, "Factoring polynomials over finite fields," *Bell System Tech. J.*, v. 46, 1967, pp. 1853–1859. MR 36 #2314.

3. B. G. CLAYBROOK, "Factorization of multivariate polynomials over the integers," *SIGSAM Bulletin,* Feb. 1976, p. 13.

4. J. H. GRIESMER, D. R. JENKS & D. Y. Y. YUN, *SCRATCHPAD User's Manual,* IBM Research Report, RA 70, June 1975.

5. D. E. KNUTH, *The Art of Computer Programming,* Vol. 2: *Seminumerical Algorithms,* Addison-Wesley, Reading, Mass., 1969. MR 44 #3531.

6. S. LANG, *Diophantine Geometry,* Wiley, New York, 1962.

7. MACSYMA Reference Manual, *The MATHLAB Group,* Project MAC, M.I.T., Cambridge, Mass., Nov. 1975.

8. M. MIGNOTTE, "An inequality about factors of polynomials," *Math. Comp.,* v. 28, 1974, pp. 1153–1157.

9. R. T. MOENCK, "On the efficiency of algorithms for polynomial factoring," *Math. Comp.,* v. 31, 1977, pp. 235–250.

10. J. MOSES & D. Y. Y. YUN, "The EZ-GCD algorithm," *Proceedings of ACM Annual Conference,* August 1973.

11. D. R. MUSSER, "Multivariate polynomial factorization," *J. Assoc. Comput. Mach.,* v. 22, 1975, pp. 291–308.

12. P. S. WANG, "Factoring multivariate polynomials over algebraic number fields," *Math. Comp.,* v. 30, 1976, pp. 324–336.

13. P. S. WANG & L. P. ROTHSCHILD, "Factoring multivariate polynomials over the integers," *Math. Comp.,* v. 29, 1975, pp. 935–950.

14. P. S. WANG, "Preserving sparseness in multivariate polynomial factorization," *Proceedings, MACSYMA User's Conference,* University of California at Berkeley, July 27–29, 1977.

15. D. Y. Y. YUN, *The Hensel Lemma in Algebraic Manipulation,* Ph. D. Thesis, Department of Mathematics, M.I.T., Nov. 1973 (also Project MAC TR-138, Nov. 1974).

16. H. ZASSENHAUS, "On Hensel factorization. I," *J. Number Theory,* v. 1, 1969, pp. 291–311. MR 39 #4120.