MATHEMATICS OF COMPUTATION, VOLUME 29, NUMBER 131
JULY 1975, PAGES 935–950

Factoring Multivariate Polynomials Over the Integers

By Paul S. Wang and Linda Preiss Rothschild*

Abstract. An algorithm for the irreducible factorization of any multivariate polynomial over the integers is given. It is much faster than the classical method ascribed to Kronecker. The algorithm begins by making substitutions for all but one of the variables with selected integers, giving a polynomial in just one variable. This univariate polynomial is then factored by a known method, which uses an algorithm of Berlekamp for factoring univariate polynomials over finite fields. The multivariate factors are constructed from the univariate ones by a kind of Hensel algorithm. The procedure has been implemented in the algebraic manipulation systems MACSYMA and SCRATCHPAD. A number of machine examples with timing are included.

1. Introduction. We describe an algorithm for the irreducible factorization of any multivariate polynomial over the rational integers, **Z**. This algorithm makes use of Berlekamp's algorithm for factoring univariate polynomials modulo a prime number [1], [2]. Our algorithm is much faster than the classical method ascribed to Kronecker [10, pp. 135–136]. Experience of machine computation bears this out in both univariate and multivariate factoring with almost no exceptions.

Essentially, our algorithm first reduces a given multivariate polynomial to a polynomial in one variable by substituting integers for the other variables. The resulting univariate polynomial is then factored over **Z** by an algorithm which uses the Berlekamp algorithm with a small prime. The univariate factors over the integers are used in turn to construct the irreducible multivariate factors by using a variation of a kind of "*p*-adic" interpolation based on Hensel's lemma originally suggested by Zassenhaus [12].

The entire algorithm has been implemented in the LISP programming language [7] for the algebraic manipulation system MACSYMA [13] at Project MAC, Massachusetts Institute of Technology. It is also implemented in the SCRATCHPAD system at the Thomas J. Watson Research Center, IBM.

A list of polynomials factored by this computer program is included in an appendix. The running time for each problem is indicated. These timings are made on the MATHLAB time-sharing system at MAC which uses a PDP-10 computer with a memory-cycle time of about two microseconds.

In his thesis, Musser [9] has described an algorithm for factoring a univariate polynomial over the integers in the SAC-1 system [4]. He has also discussed briefly some

Received August 3, 1972; revised March 4, 1974.

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

*Work reported herein was supported in part by Project MAC, an M.I.T. interdepartmental laboratory sponsored by the Advanced Research Projects Agency (ARPA), Department of Defense, under Office of Naval Research Contract N00014-70-A-0362-0001.

ideas of extending his algorithm to multivariate factorization. But no specific algorithms or programs have been given.

We are grateful to Stan Brown, Joel Moses, Richard Fateman, David Yun and Rich Schroeppel for assistance and suggestions. We would also like to thank G. E. Collins and D. Musser for making available to us an outline of their program for the Berlekamp algorithm. We also wish to thank the referee whose comments and suggestions were very helpful in revising this paper.

2. Preliminaries and an Outline. Polynomials over Z, in several variables, $x, x_2, x_3, \ldots, x_n$, form a unique factorization domain $Z[x, x_2, \ldots, x_n]$. Let $U(x, x_2, x_3, \ldots, x_n)$ be a multivariate polynomial in $Z[x, x_2, \ldots, x_n]$. By choosing a main variable, say x, we may write U as a polynomial in x,

$$U(x, x_2, \ldots, x_n) = U_m x^m + U_{m-1}x^{m-1} + \ldots + U_1 x + U_0$$

with coefficients U_i in $Z[x_2, \ldots, x_n]$, $i = 0, 1, \ldots, m$. We may assume that the leading coefficient of U, $\mathrm{lc}(U) = U_m$, with respect to the main variable is not zero. Therefore, the degree of U, $\deg(U)$, is m. CONT(U), the *content* of U with respect to the main variable, is defined as

$$\mathrm{CONT}(U) = \mathrm{GCD}(U_0, U_1, \ldots, U_m),$$

where GCD stands for the common divisor of greatest degree. The *principal part* of U, $\mathrm{pp}(U)$, is equal to $U/\mathrm{CONT}(U)$. U is *primitive* if $\mathrm{CONT}(U) = 1$. U is *squarefree* if U has no repeated factors.

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

$$\{g_1 f_1 + g_2 f_2 + \ldots + g_r f_r \colon g_i \in Z[x_2, \ldots, x_n] \ \forall i\}$$

(see [10, Sectıon 16]). The set F need not be finite. For any integer, $k > 0$ and any ideal $\mathcal{S}$, $\mathcal{S}^k$ denotes the ideal generated by all products of the form $h_1 h_2 \ldots h_k$, $h_i \in \mathcal{S}$, $i = 1, 2, \ldots, k$.

If A and B are polynomials and $\mathcal{S}$ is an ideal in $Z[x, x_2, \ldots, x_n]$, we define $A \equiv B \bmod \mathcal{S}$ if $A - B \in \mathcal{S}$, i.e., if $A - B$ is divisible by an element of $\mathcal{S}$. For example, if $\mathcal{S} = (x_2 - a_2, x_3 - a_3, \ldots, x_n - a_n)$ with $a_2, \ldots, a_n$ integers, $A(x, x_2, \ldots, x_n) \equiv A(x, a_2, \ldots, a_n) \bmod \mathcal{S}$ for $A(x, a_2, \ldots, a_n)$ is the remainder of dividing A by every $x_i - a_i$, $i = 2, \ldots, n$; $\mathcal{S}^k$ is then the ideal generated by all polynomials of the form,

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

For this ideal $\mathcal{S}$ we define, for any positive integer k,

$$A = B \mod \mathcal{S}^k \quad \text{if } A \equiv B \mod \mathcal{S}^k \quad \text{and} \quad \deg(A) \text{ in } x_2, \ldots, x_n < k.$$

Similarly, $A = B \mod (q)$ for any prime power $q > 2$ if $A \equiv B \mod (q)$ and the coefficients of A are between $-q/2$ and $q/2$.

Let us outline the five major steps in our algorithm for the irreducible factorization of U over Z.

I. *Primitive and Squarefree.* If U is not primitive, $\mathrm{CONT}(U)$ and $\mathrm{pp}(U)$ may be factored separately. Thus, we may assume that U is primitive. Let

$$\mathrm{GCD}(U(x, x_2, \dots, x_n), \partial U/\partial x) = D.$$

Note that U has a repeated factor G if and only if G divides D and U/D is squarefree. The algorithm can be continued by factoring D and U/D separately.

II. *Substitution.* Find a set of integers, $\{a_2, a_3, \dots, a_n\}$, (not necessarily distinct) such that $\tilde{U}(x) = U(x, a_2, \dots, a_n)$ is squarefree and $\deg(\tilde{U}) = \deg(U)$.

III. *Univariate Factorization.* Apply the factoring algorithm for a univariate polynomial to find a factorization

$$\tilde{U}(x) = \tilde{p}_1(x) \dots \tilde{p}_r(x), \qquad r \geqslant 1, \tag{1}$$

over the integers. If $\tilde{U}(x)$ is irreducible, so is U and the algorithm terminates. This process also produces a prime q such that $\tilde{U}(x)$ is squarefree modulo q (see Section 5).

IV. *Construction of Multivariate Factors.*

(i) *Coefficient Bound.* Let $V = U(x, y_2 + a_2, \dots, y_n + a_n)$. Find a number B such that for any integer coefficient c of any divisor of $\mathrm{lc}(V)V$, $B > |c|$. Let j be the smallest integer such that $2B < q^{2^j} = b$. The prime power b is used as a modulus in step (ii).

(ii) *Construction of Factors.* Equation (1) can be written as

$$\tilde{P}_1(x)\tilde{P}_2(x) \dots \tilde{P}_r(x) \equiv U(x, \dots, x_n) \mod \mathcal{S},$$

where $\mathcal{S}$ is the ideal $(x_2 - a_2, x_3 - a_3, \dots, x_n - a_n)$. An extended Zassenhaus algorithm based on a variant of Hensel's lemma (see Section 8) is used to compute, from the above congruence, relatively prime polynomials $P_i(x, x_2, \dots, x_n)$, $i = 1, \dots, r$, such that each $P_i \equiv \tilde{P}_i \mod \mathcal{S}$ and

$$P_1(x, \dots, x_n) \dots P_r(x, \dots, x_n) \equiv U(x, \dots, x_n) \mod (b, \mathcal{S}^h) \tag{2}$$

where $h = 1 +$ degree of U in $x_2, \dots, x_n$.

V. *Finding Actual Factors.* If $U(x, \dots, x_n)$ is monic (with respect to the main variable x), then any irreducible factor $G(x, \dots, x_n)$ of U over $\mathbf{Z}$ either is equal to some P_i for $1 \leqslant i \leqslant r$ or is equal to the product of two or more P_i's $\mod (b, \mathcal{S}^h)$. If U is not monic then these equivalences are up to the units in the coefficient domain of U (see Section 9). In any event, the irreducible factors of U are found from these P_i's by trial divisions.

3. A Complete Example. Now let us illustrate the whole algorithm by applying it to a specific multivariate polynomial, following the steps outlined in the previous section. The polynomial to be factored is

$$\begin{aligned} U(x, y, z) = {} & x^4 + (-z + 3)x^3 + (z^3 + (y - 3)z - y^2 - 13)x^2 \\ & + (-z^4 + (y^2 + 3y + 15)z + 6)x + yz^4 + 2z^3 \\ & + (-y^3 - 15y)z - 2y^2 - 30. \end{aligned} \tag{3}$$

I. U is primitive since $\mathrm{lc}(U) = 1$ (choosing x as the main variable). U is squarefree for $\mathrm{GCD}(U, \partial U/\partial x) = 1$.

II. The substitution $y = 0$ and $z = 0$ is used because $\deg(U) = \deg(U(x, 0, 0))$ and $U(x, 0, 0)$ has no repeated factors. Thus, the ideal $\mathfrak{s} = (y, z)$ and

$$\widetilde{U}(x) = U(x, 0, 0) = x^4 + 3x^3 - 13x^2 + 6x - 30.$$

III. Factoring $\widetilde{U}(x)$ over $\mathbf{Z}$ (see Section 5), we have

$$\widetilde{U}(x) = (x^2 + 2)(x^2 + 3x - 15).$$

IV. Starting from $\widetilde{U}(x) \equiv U(x, y, z) \mod \mathfrak{s}$, we use the extended Zassenhaus algorithm to construct (see example in Section 8)

$$U \equiv (x^2 - zx + 2)(x^2 + 3x - 15) \qquad \mod \mathfrak{s}^2,$$

$$U \equiv (x^2 - zx + yz + 2)(x^2 + 3x - y^2 - 15) \quad \mod \mathfrak{s}^3,$$

and

$$U \equiv (x^2 - zx + yz + 2)(x^2 + 3x - y^2 + z^3 - 15) \quad \mod \mathfrak{s}^4.$$

There is no need to go to a higher power of $\mathfrak{s}$, because the last congruence is actually an equality over the integers.

V. Trial division immediately gives the irreducible factorization of U over $\mathbf{Z}$

$$U(x, y, z) = (x^2 - zx + yz + 2)(x^2 + 3x - y^2 + z^3 - 15). \tag{4}$$

4. Selecting Integers for Substitution. The integers needed for substitution in step II exist. That is, one can show that if U is squarefree, integers $a_2, a_3, \ldots, a_n$ can be chosen so that $\widetilde{U}(x) = U(x, a_2, \ldots, a_n)$ is still squarefree and $\deg(\widetilde{U}) = \deg(U)$. The proof, which uses the discriminant, is omitted here. Suitable a_i's may be found by trial and error. The first choices for these a_i's should be 0, 1, and -1 for they usually make coefficients of $\widetilde{U}(x)$ small in size. It is desirable to use as many zeros as possible for the substitution, because each a_i which is not zero can cause some intermediate expression growth when the extended Zassenhaus algorithm is applied in step IV.

For different substitutions, the number of factors, r, in (1) may be different. Since the larger r is, the longer the entire algorithm takes, it is sometimes advantageous to try several substitutions and work with the one which gives minimum r. Our program does not attempt to do this not only because the number r requires a fair amount of time to compute, but because a different substitution does not guarantee a smaller r.

The leading coefficient plays an important role in the factoring process. Factorization is easier if the leading coefficient is 1; for if U is monic, then any factor of U is monic. But if U is not monic, then additional computation is required to determine the leading coefficient of each factor. Therefore, the main variable of U is chosen so that $\mathrm{lc}(U)$ is 1 or small, in order to avoid or simplify later computations related to the leading coefficient. If several variables have a monic leading coefficient, it is best to choose the variable giving the smallest $\deg(\widetilde{U})$, thus limiting the number of possible factors.

5. Factorization of Univariate Polynomials Over Z. In step III, an arbitrary squarefree polynomial $\tilde{U}(x) \in \mathbf{Z}[x]$ is to be factored over Z. The basic steps of this process are outlined here.

(a) Choose a prime q such that $\tilde{U}(x) \mod(q)$ is squarefree and has the same degree as $\tilde{U}(x)$. Let $u(x) = \tilde{U}(x) \mod(q)$.

(b) Factor $u(x)$ using Berlekamp's algorithm [1], [6] to obtain

$$u(x) \equiv p_1(x)p_2(x) \ldots p_t(x) \mod(q) \tag{5}$$

with the p_i's distinct and irreducible over $\mathbf{Z}_q$. If $t = 1$, $\tilde{U}$ is irreducible mod(q), so U is clearly irreducible over Z, and the algorithm ends.

(c) Find an upper bound $\tilde{B}$ on the magnitude of the coefficients of any possible factor of $\tilde{U}(x)$, i.e. $\tilde{B}$ is a number such that $\tilde{B} > |c|$ for any coefficient c of any factor of $\tilde{U}$. Also find the least integer d such that $q^{2^d} \geqslant 2|\mathrm{lc}(\tilde{U})|\tilde{B}$.

(d) Use a "p-adic" algorithm by Zassenhaus [12] to construct $\hat{p}_1(x), \hat{p}_2(x), \ldots, \hat{p}_t(x)$ from (5) such that each $\hat{p}_i \equiv p_i \mod(q)$ and

$$\tilde{U}(x) \equiv \hat{p}_1(x)\hat{p}_2(x) \ldots \hat{p}_t(x) \mod(q^{2^d}). \tag{6}$$

(e) Use the algorithm TRUEFACTORS for the ideal (q^{2^d}) to get a factorization of $\tilde{U}(x)$ over Z (see Section 9).

$$\tilde{U}(x) = \tilde{P}_1(x) \ldots \tilde{P}_r(x), \qquad 1 \leqslant r \leqslant t.$$

If $r = 1$, U is irreducible over Z. For example, we can carry out the above steps for the polynomial,

$$\tilde{U}(x) = x^4 + 3x^3 - 13x^2 + 6x - 30.$$

(a) $\tilde{U}(x)$ is nonsquarefree mod (3). Choosing the next larger prime 5, we have

$$u(x) = x^4 - 2x^3 + 2x^2 + x \equiv \tilde{U}(x) \mod(5)$$

which is squarefree.

(b) Factoring $u(x)$ over $\mathbf{Z}_5$ gives

$$u(x) \equiv x(x - 2)(x^2 + 2) \mod(5).$$

(c) If we take, for simplicity, $\tilde{B} = 300$, then $d = 2$ since $5^4 = 625 > 2 \cdot 300$.

(d) Zassenhaus' algorithm described in Section 7 is applied to yield

$$\tilde{U}(x) \equiv (x + 45)(x - 42)(x^2 + 2) \mod(625).$$

(e) Of the three factors above, only $(x^2 + 2)$ divides $\tilde{U}(x)$ over Z. Therefore, $\tilde{U}(x)$ has only two irreducible factors. The other factor is

$$x^2 + 3x - 15 \equiv (x + 45)(x - 42) \mod(625).$$

An irreducible polynomial in $\mathbf{Z}[x]$ may have nontrivial factors modulo q. In fact, the number of factors, t, in step (b) depends on the prime q chosen in step (a). Let $m = \deg(\tilde{U})$ and r be the number of irreducible factors of $\tilde{U}(x)$ over Z. We see that

$m \geqslant t \geqslant r$. In cases where m is small, we may choose q as the smallest prime that satisfies (a). If m is rather large, a particular choice of q may cause t to be much greater than r. An extreme case is $r = 1$ and $t = m$. To guard against such instances, one may want to try two or more primes, depending on the size of m, and use the one which gives minimum t.

According to Knuth [6], steps (a) and (b) take $O(m^3)$ units of time. Step (d) takes no more than $O(qtm^2)$ units of time. For step (e) we use algorithm TRUEFACTORS which requires no more than $O(2^t m^2)$ units of time. In view of this exponential characteristic, it pays to make $(t - r)$ small. However, for each additional prime tried, the cost in time is approximately $O(m^3)$.

The referee informs us that a method due to R. Graham allows one to conclude irreducibility over **Z** from the way a polynomial factors modulo several primes. Currently, this method is not in our factoring algorithm.

6. Coefficient Bounds. There are two methods given in Knuth [6] for finding upper bounds on the magnitude of the coefficients of every possible factor of a given univariate polynomial $\widetilde{U}(x)$. One of these methods involves bounding the absolute value of the roots of $\widetilde{U}(x)$. Another uses matrix computations.

A coefficient bound which works in both the univariate and the multivariate cases is given by Gel′fond [5, p. 135]. Let the maximum coefficient magnitude and degrees of $U(x, x_2, \ldots, x_n)$ be $U_{\max}$ and $m, m_2, \ldots, m_n$, in the variables $x, x_2, \ldots, x_n$, respectively, then $U_{\max}e^M$, $M = m + m_2 + \ldots + m_n$, bounds the magnitude of the coefficients of any divisor of U. We use this method to compute the upper bounds $\widetilde{B}$ (step (c)) and B (step IV(i)). This is suggested to us by P. Weinberger.

The bounds computed are often much larger than they ought to be. If $\widetilde{B}$ is too large, then d, the number of iterations in the Zassenhaus algorithm, is too large. It is desirable to find a more accurate coefficient bound than the one we use here. MACSYMA enables the user to save computation time by electing to use a heuristic bound in the factoring process:

$$\widetilde{B} = m \cdot \mathrm{MAX}(m, |\widetilde{U}_0|, |\widetilde{U}_1|, \ldots, |\widetilde{U}_m|) \cdot 2^{\{m/2\}}, \qquad m = \deg(\widetilde{U}),$$

where the $\widetilde{U}_i$'s are the coefficients of $\widetilde{U}(x)$ and $\{m/2\}$ stands for the least integer $\geqslant m/2$. This heuristic bound is sometimes still rather large. Arithmetic involving integers more than single precision (35 bits on a PDP-10) is handled by MACSYMA's arbitrary precision integer arithmetic routines.

It is conceivable, although not probable, for the heuristic $\widetilde{B}$ to be too small. In such a case, factors found by the algorithm may be reducible. Thus, one should be careful in electing to use the heuristic coefficient bound.

7. Zassenhaus' Algorithm for Factoring mod q^{2^j}. In this section, we describe how (6) is constructed from (5) in step (d) of Section 5. The algorithm is due to Zassenhaus [12] and is based on Hensel's lemma. Let us write (5) as

$$\widetilde{U}(x) \equiv p_{10}p_{20} \cdots p_{t0} \mod(q)$$

with p_{i0}'s distinct and irreducible over the field Z_q. From these factors, the algorithm constructs p_{ij}, proceeding from j to $j + 1$, starting from $j = 0$, such that

$$\widetilde{U}(x) \equiv p_{1j}p_{2j} \cdots p_{tj} \mod(q^{2^j}) \quad \text{and} \quad p_{ij} \equiv p_{i0} \mod(q). \tag{7}$$

It suffices to have such an algorithm for two relatively prime factors of $\widetilde{U}$, which can then be used recursively when $t > 2$.

Let F_0 and G_0 be relatively prime polynomials in $Z_q[x]$ such that $\deg(F_0) = f > 0$, $\deg(G_0) = g > 0$, $\deg(\widetilde{U}) = f + g$ and $\widetilde{U}(x) \equiv F_0(x)G_0(x) \mod(q)$.

LEMMA. *Polynomials F_j, G_j, α_j and β_j, $j \geqslant 0$, can be found such that* $\deg(F_j) = f$, $\deg(G_j) = g$,

$$\widetilde{U}(x) \equiv F_jG_j \mod(q^{2^j}), \qquad F_j \equiv F_0, \qquad G_j \equiv G_0 \mod(q) \tag{8}$$

and

$$\alpha_jF_j + \beta_jG_j \equiv 1 \mod(q^{2^j}) \tag{9}$$

with $\deg(\alpha_j) < g$ *and* $\deg(\beta_j) \leqslant f$. *Furthermore, F_j and G_j are unique up to units. (Note that $\widetilde{U}(x)$ need not be monic.)*

Proof. The lemma is true for $j = 0$, because α_0 and β_0 can be found uniquely by a polynomial remainder sequence method given in [6]. Let us assume that the lemma is true for $j = k - 1$. We compute $C(x)$ by

$$C(x) = (F_{k-1}G_{k-1} - \widetilde{U})/q^{2^{k-1}} \mod(q^{2^{k-1}}). \tag{10}$$

Let $R_1(x)$ be the remainder of $\alpha_{k-1}C(x)$ divided by G_{k-1}, i.e., compute R_1 by

$$R_1 = \alpha_{k-1}C - G_{k-1}P \mod(q^{2^{k-1}})$$

with $P(x)$ the quotient. Compute $R_2(x)$ by

$$R_2 = F_{k-1}P + \beta_{k-1}C \mod(q^{2^{k-1}}).$$

It is easy to deduce that $\deg(R_1) < g$, $\deg(R_2) \leqslant f$ and, from (9),

$$R_1F_{k-1} + R_2G_{k-1} \equiv C(x) \mod(q^{2^{k-1}}). \tag{11}$$

Thus, if we compute $F_k(x)$ and $G_k(x)$ by

$$F_k = F_{k-1} - q^{2^{k-1}}R_2 \quad \text{and} \quad G_k = G_{k-1} - q^{2^{k-1}}R_1, \tag{12}$$

It follows from (10) and (11) that F_k and G_k satisfy (8) for $j = k$. Now compute $D(x)$ by

$$D(x) = (\alpha_{k-1}F_k + \beta_{k-1}G_k - 1)/q^{2^{k-1}} \mod(q^{2^k}).$$

As before, we can find $S_1(x)$ and $S_2(x)$ such that $\deg(S_1) < g$, $\deg(S_2) \leqslant f$ and $S_1F_{k-1} + S_2G_{k-1} \equiv D(x) \mod(q^{2^{k-1}})$. Therefore, if we compute α_k and β_k by

$$\alpha_k = \alpha_{k-1} - q^{2^{k-1}}S_1 \quad \text{and} \quad \beta_k = \beta_{k-1} - q^{2^{k-1}}S_2 \mod(q^{2^k}),$$

then (9) is satisfied with $j = k$.

Now we show that F_k and G_k are unique up to units. Let us assume

$$\tilde{U}(x) \equiv F_k(x)G_k(x) \equiv A(x)B(x) \mod(q^{2^k}); \tag{13}$$

and, by induction,

$$F_k \equiv F_{k-1} \equiv aA \quad \text{and} \quad G_k \equiv G_{k-1} \equiv a^{-1}B \mod(q^{2^{k-1}})$$

for some unit a in $Z/(q^{2^{k-1}})$. That is,

$$F_k \equiv aA + q^{2^{k-1}}C_1(x),$$

$$G_k \equiv a^{-1}B + q^{2^{k-1}}C_2(x) \mod(q^{2^k})$$

for some C_1 and C_2, $\deg(C_1) \leqslant \deg(A)$ and $\deg(C_2) \leqslant \deg(B)$. From (13) we have

$$aAC_2 + a^{-1}BC_1 \equiv 0 \mod(q^{2^{k-1}}).$$

A and B being relatively prime implies

$$C_1 = e_1A \quad \text{and} \quad C_2 = e_2B \mod(q^{2^{k-1}})$$

for e_1 and e_2 units in $Z/(q^{2^{k-1}})$. Hence, A and B are unit multiples of F_k and G_k, respectively. □

Therefore, (7) can be constructed and it is unique up to units in the ring $Z/(q^{2^j})$.

Example. Let

$$\tilde{U}(x) = x^4 + 3x^3 - 13x^2 + 6x - 30.$$

Extend the following congruence to modulo 25:

$$\tilde{U}(x) \equiv (x^2 - 2x)(x^2 + 2) \mod(5).$$

Let $F_0 = x^2 - 2x$, $G_0 = x^2 + 2$; we have

$$F_0G_0 - \tilde{U}(x) = 5(-x^3 + 3x^2 - 2x + 6) = 5C(x).$$

$\alpha_0(x) = 0$ and $\beta_0(x) = -(x + 2)$ give

$$\alpha_0F_0 + \beta_0G_0 \equiv C(x) \mod(5).$$

Therefore, according to Eqs. (11) and (12),

$$F_1(x) = F_0(x) - 5(-x - 2) = x^2 + 3x + 10,$$

$$G_1(x) = G_0(x) \quad \text{and} \quad F_1G_1 \equiv \tilde{U}(x) \mod(25).$$

8. An Extended Zassenhaus Algorithm. This algorithm computes (2) in step IV from (1) in step III. For the same reason as given in Section 7, we shall give the algorithm for two relatively prime factors. Let $F(x)$ and $G(x)$ be two relatively prime polynomials such that $F(x)G(x) = \tilde{U}(x)$ over **Z**. It follows that

$$U(x, x_2, \ldots, x_n) \equiv F(x)G(x) \mod \mathcal{S} \tag{14}$$

where $\mathscr{I}$ is the ideal $(x_2 - a_2, x_3 - a_3, \ldots, x_n - a_n)$. From F and G this algorithm constructs, proceeding from k to $k + 1$ starting from $k = 1$, multivariate polynomials F_k and G_k, with $F_1 = F$ and $G_1 = G$, such that

$$F_k(x, x_2, \ldots, x_n) \equiv F(x) \mod \mathscr{I}, \qquad G_k(x, x_2, \ldots, x_n) \equiv G(x) \mod \mathscr{I},$$

and

$$U(x, x_2, \ldots, x_n) \equiv F_k G_k \mod \mathscr{I}^k.$$

We shall first show how to get to $k = 2$. Then, the general case is given by induction. The algorithm ends when k reaches h which equals $1 +$ degree of U in $x_2, \ldots, x_n$.

Let $y_i = x_i - a_i$, $i = 2, 3, \ldots, n$, and

$$V(x, y_2, \ldots, y_n) = U(x, y_2 + a_2, \ldots, y_n + a_n).$$

We compute R_1 and W_1 by

$$R_1(x, y_2, \ldots, y_n) = F(x)G(x) - V(x, y_2, \ldots, y_n)$$

and

$$W_1(x, y_2, \ldots, y_n) = R_1 \mod \mathscr{I}^2.$$

Since $\mathscr{I}$ is now the ideal $(y_2, \ldots, y_n)$, computations mod $\mathscr{I}^i$ are done simply by dropping all the terms of degree greater than or equal to i in $y_2, \ldots, y_n$, e.g., $y_2 y_3 \equiv 0 \mod \mathscr{I}^2$.

Since $F(x)$ and $G(x)$ are relative prime, unique $\alpha_i(x)$ and $\beta_i(x)$ can be computed such that for all $i \leqslant m$

$$\alpha_i(x)F(x) + \beta_i(x)G(x) = x^i, \tag{15}$$

with $\deg(\alpha_i) < \deg(G)$ and $\deg(\beta_i) \leqslant \deg(F)$. These α_i and β_i are computed as the need arises and are stored for future use. This scheme is used at the suggestion of J. Moses.

For any polynomial $T(x, y_2, \ldots, y_n)$, let us denote by $\mathrm{A}(T)$ the polynomial obtained by substituting $\alpha_i(x)$ for x^i in T and by $\mathrm{B}(T)$ the polynomial obtained by substituting $\beta_i(x)$ for x^i in T. Now compuie F_2, G_2 and R_2 by

$$F_2(x, y_2, \ldots, y_n) = F(x) - \mathrm{B}(W_1), \qquad G_2(x, y_2, \ldots, y_n) = G(x) - \mathrm{A}(W_1),$$

and

$$R_2 = R_1 + \mathrm{A}(W_1)\mathrm{B}(W_1) - \mathrm{A}(W_1)F - \mathrm{B}(W_1)G.$$

It is readily verified that

$$\mathrm{A}(W_1)F + \mathrm{B}(W_1)G = W_1, \qquad \mathrm{A}(W_1)\mathrm{B}(W_1) \equiv 0 \mod \mathscr{I}^2,$$

$$V \equiv F_2 G_2 \mod \mathscr{I}^2 \quad \text{and} \quad R_2 = F_2 G_2 - V.$$

Suppose we have, by induction, F_{k-1}, G_{k-1} and R_{k-1} such that, for $k \geqslant 2$,

$$V \equiv F_{k-1} G_{k-1} \mod \mathscr{I}^{k-1}, \qquad F_{k-1} \equiv F \mod \mathscr{I}, \qquad G_{k-1} \equiv G \mod \mathscr{I}$$

and

$$R_{k-1} = F_{k-1}G_{k-1} - V.$$

Now let us compute W_{k-1}, F_k, G_k and R_k by

$$W_{k-1} = R_{k-1} \mod \mathcal{S}^k, \qquad F_k = F_{k-1} - \mathrm{B}(W_{k-1}),$$
$$G_k = G_{k-1} - \mathrm{A}(W_{k-1}) \tag{16}$$

and

$$R_k = R_{k-1} + \mathrm{A}(W_{k-1})\mathrm{B}(W_{k-1}) - \mathrm{A}(W_{k-1})F_{k-1} - \mathrm{B}(W_{k-1})G_{k-1}. \tag{17}$$

Then, it can be deduced that

$$F_k \equiv F \mod \mathcal{S}, \qquad G_k \equiv G \mod \mathcal{S}$$

and

$$F_kG_k - V \equiv R_k \equiv 0 \mod \mathcal{S}^k.$$

Consequently, the congruence

$$V(x, y_2, \ldots, y_n) \equiv F_k(x, y_2, \ldots, y_n)G_k(x, y_2, \ldots, y_n) \mod \mathcal{S}^k \tag{18}$$

can be constructed for any $k \geqslant 2$. The algorithm ends whenever R_k is zero or when k reaches h. It can be shown that F_k and G_k are unique up to units. Therefore, the P_i's in (2) are also unique up to units.

Integer arithmetic would be sufficient for this algorithm if it were not for the $\alpha_i(x)$ and $\beta_i(x)$ needed in (15) which normally have rational coefficients. To avoid the costly process of using rational arithmetic for the entire extended Zassenhaus algorithm, we can use a large enough modulus for our computation.

Let B be an upper bound on the magnitude of the coefficients of any possible factor of $V(x, y_2, \ldots, y_n)$ or $\widetilde{U}(x)$. We can use the prime q chosen in Section 5 to form a prime power $b = q^{2^j}$ such that $b > 2|\mathrm{lc}(U)|B$. It is easy to see that all computation can be done modulo b. However, a true bound is usually too large. In MACSYMA we provide an optional heuristic bound:

$$B = \mathrm{MAX}(\widetilde{B}, h \cdot \mathrm{MAX}(h, v_{\max}) \cdot 2^{\{h/2\}})$$

where $v_{\max}$ is the absolute value of the largest integer coefficient of V.

The computing time of this algorithm is dominated by the multiplication of multivariate polynomials in Eq. (17). If V has v terms, Eq. (17) takes no more than v^2 units of time. Thus, the entire algorithm takes $O(hv^2)$ units of time. And to construct the congruence (2) from (1), we need to apply this algorithm r times.

Example. Given

$$\begin{aligned} U(x, y, z) = {} & x^4 + (-z + 3)x^3 + ((y - 3)z - y^2 - 13)x^2 \\ & + ((y^2 + 3y + 15)z + 6)x \\ & + (-y^3 - 15y)z - 2y^2 - 30, \end{aligned} \tag{19}$$
$$F(x) = (x^2 + 2), \qquad G(x) = (x^2 + 3x - 15)$$

and

$$U(x, 0, 0) = F(x)G(x),$$

find F_2, G_2, F_3 and G_3.

To avoid rational arithmetic, we use 625 as a modulus; that is, all arithmetic will be done modulo 625. Let $\mathfrak{s}$ be the ideal (y, z); then $U \equiv F(x)G(x) \mod \mathfrak{s}$. Now

$$W_1(x, y, z) = z(x^3 + 3x^2 - 15x) = F \cdot G - U \mod \mathfrak{s}^2.$$

And we have

$$\alpha_0 = -171x - 232, \quad \alpha_1 = 281x - 65,$$

$$\alpha_2 = -283x - 160, \quad \alpha_3 = 64x + 130,$$

$$\beta_0 = 171x - 281, \quad \beta_1 = 281x + 283,$$

$$\beta_2 = 283x - 63, \quad \beta_3 = -63x + 59,$$

satisfying Eq. (15). All these α_i's and β_i's are derived, as indicated in Section 7, from the relation $-(x + 2)F + (x - 1)G \equiv 1 \mod(5)$ which can be obtained by a polynomial remainder sequence method.

We now can compute $\mathrm{A}(W_1) = 0$ and $\mathrm{B}(W_1) = zx$ which give $F_2 = x^2 - zx + 2$, $G_2 = G$. Similarly, to find F_3 and G_3, we compute

$$W_2 = -(yz - y^2)x^2 - 3yzx + 15yz + 2y^2 = R_2 \mod \mathfrak{s}^3,$$

$$\mathrm{A}(W_2) = y^2 \quad \text{and} \quad \mathrm{B}(W_2) = -yz.$$

This means that

$$F_3 = x^2 - zx + yz + 2 \quad \text{and} \quad G_3 = x^2 + 3x - y^2 - 15.$$

One can show that $F_2G_2 \equiv U \mod \mathfrak{s}^2$ and $F_3G_3 \equiv U \mod \mathfrak{s}^3$.

9. **Obtaining True Factors.** In this section, we describe the algorithm for obtaining actual factors of $U(x, x_2, \ldots, x_n)$ over the integers from the factorization

$$U(x, \ldots, x_n) \equiv P_1(x, \ldots, x_n) \ldots P_r(x, \ldots, x_n) \mod(b, \mathfrak{s}^h) \tag{2}$$

with $r \geq 2$, the P_i's distinct and irreducible.

The factorization (2) is unique up to units in the quotient ring $R = \mathbf{Z}[x_2, \ldots, x_n]/(b, \mathfrak{s}^h)$, though R is not a unique factorization domain.

If U is monic, then the P_i's are all monic and any irreducible factor $G(x, \ldots, x_n)$ of U either is equal to some P_i, $1 \leq i \leq r$, or is equal to the product of two or more P_i's $\mathrm{mod}(b, \mathfrak{s}^h)$. If U is not monic, then these equivalences are up to units in R. Suppose an irreducible factor G, of U over $\mathbf{Z}$, is equal to H up to units in R. Then

$$H^* = \mathrm{lc}(U)\mathrm{lc}(H)^{-1}H \equiv \mathrm{lc}(U)\mathrm{lc}(G)^{-1}G \mod(b, \mathfrak{s}^h) \tag{20}$$

and $G = \mathrm{pp}(H^*)$ taken over the integers. The quantity $\mathrm{lc}(U)\mathrm{lc}(H)^{-1}$ is easily computed. For example, for any $1 \leq i \leq r$,

$$\mathrm{lc}(U)\mathrm{lc}(P_i)^{-1} = \prod_{j=1;j\neq i}^{r} \mathrm{lc}(P_j) \quad \mathrm{mod}(b, \mathfrak{s}^h).$$

Evidently, if H^* divides $\mathrm{lc}(U)U$, $\mathrm{pp}(H^*)$ is a factor of U over **Z**.

Because U is squarefree, different irreducible factors of U over the integers cannot involve the same P_i for any i. Exactly how these P_i's should be grouped to give rise to the irreducible factors of U requires a combinatorial search.

With U, h, $\mathfrak{s}$, P_i, $i = 1, 2, \ldots, r$, as input, the following algorithm returns a list of irreducible factors of U over **Z**. This algorithm finds the true factors by trial division over **Z**. The divisors are formed by taking different products, $\mathrm{mod}(b, \mathfrak{s}^h)$, of $1, 2, \ldots, r-2$ elements, in that order, from the set $\{P_1, P_2, \ldots, P_r\}$. Of all such products, only those of degree not exceeding $m/2$ are formed. If U is not monic, necessary steps for correcting the leading coefficients of possible factors as indicated in (20) will be taken. Each successful division produces an irreducible factor of U and replaces U by the quotient. This algorithm ends when $U = 1$ or when all combinations are taken. In the latter case, the remaining quotient U is an irreducible factor.

```
Algorithm TRUEFACTORS:
(1) For $i = 1$ through $r$, execute the following:
    Set $U^* = \mathrm{lc}(U)U$, $Y = \mathrm{lc}(U)\mathrm{lc}(P_i)^{-1}P_i \ \mathrm{mod}(b, \mathfrak{s}^h)$. If $Y$
    divides $U^*$ over Z, put $\mathrm{pp}(Y)$ on the list FAC and set
    $U = U/\mathrm{pp}(Y)$; otherwise put $P_i$ on the list $L$.
(2) If $L$ is an empty list, return the answer FAC and exit. If $L$
    contains less than four elements, put $U$ on the list FAC return
    FAC and exit (for $L$ cannot contain two disjoint subsets of two
    or more elements). Otherwise, set $M = 1$, $\alpha$ = number of elements on $L$,
    $r = r -$ number of elements on FAC, $u = \deg(U)/2$, $U^* = \mathrm{lc}(U)U$.
(3) Set $M = M + 1$.
(4) If $U = 1$, return FAC and exit. If $r = 1, M \geqslant r - 1$ or $M > u$, put $U$ on the
    list FAC, return FAC and exit.
(5) Select a combination, $E$, of $M$ elements from $L$ with the sum of their
    degrees not to exceed $u$. If no such combination can be found, put $U$
    on the list FAC, return FAC and exit. If all such combinations of
    $M$ elements from $L$ have been taken, go to step 3.
(6) Set $Y$ = product of elements in $E$ $\mathrm{mod}(b, \mathfrak{s}^h)$.
    Set $Y = \mathrm{lc}(U)\mathrm{lc}(Y)^{-1}Y \ \mathrm{mod}(b, \mathfrak{s}^h)$. If $Y$ divides $U^*$ over Z, do the
    following:
      (i) Put $\mathrm{pp}(Y)$ on the list FAC and set $L = L - E, r = r - 1$;
      (ii) set $U = U/\mathrm{pp}(Y)$, $u = \deg(U)/2$, $U = \mathrm{lc}(\mathrm{pp}(Y))U$;
      (iii) delete from $L$ any element with degree greater than $u$; set $\alpha$ = number of
            elements on $L$.
      (iv) go to step 4.
    If $Y$ does not divide $U^*$ over Z, go to step 5.
```

In the above algorithm, the computing of U^* and $\mathrm{pp}(Y)$ is unnecessary if the original polynomial U is monic. Trial division of U^* by Y is preceded by test divisions of the leading and trailing coefficients.

Note that only one multiplication is needed to form a product of M elements from that of $M - 1$ elements. In the worst case, where U is irreducible and $r = \deg(U) = m$, this algorithm goes through 2^{m-1} iterations and each iteration involves essentially one multiplication and one division of multivariate polynomials.

10. Special Cases. The factoring process for polynomials of some particular forms can be greatly speeded up by the use of special techniques. For instance, the factors of $x^n \pm 1$ can be obtained by computing the factors of the relevant cyclotomic polynomials [10]. For other univariate polynomials, Eisenstein's irreducibility criterion [10] can be applied first. In addition to these, we have in MACSYMA the following special cases that have been implemented by R. Fateman:

A. *Linear Case:* If U is a polynomial linear in the variable x, i.e., $U = ax + b$, then U is irreducible if and only if $\mathrm{GCD}(a, b) = 1$.

B. *Quadratic Case.* If U is quadratic in some variable x, that is, $U = ax^2 + bx + c$, and U is primitive with respect to x. Then, either U is irreducible over the integers or factors into two linear factors computed by the quadratic formula.

11. Practical Difficulties. In early tests with the factoring program, it was discovered that taking GCD of multivariate polynomials was very slow. This affected the very first step in our factoring algorithm. Namely, content and squarefree operations were taking too long. At that time, MACSYMA was using the modular GCD algorithm [3]. The need for a better method for multivariate GCD motivated the work on the EZ-GCD algorithm [8], [11]. The EZ-GCD algorithm uses the extended Zassenhaus algorithm. It is very efficient for multivariate polynomials that are not completely dense. It also makes content taking extremely fast.

There are still two major sources of difficulty. One is the coefficient bound which we have mentioned in Section 6. The other has to do with being able to use zeros for the a_i's. An important step in the extended Zassenhaus algorithm is the change of variable

$$V(x, y_2, \ldots, y_n) = U(x, y_2 + a_2, \ldots, y_n + a_n).$$

If not enough of the a_i's are zero, one can expect V to have many more terms than U. Just compare the size of $x^r y^s z^t$ with $(x + 1)^r (y + 2)^s (z + 3)^t$. The former has only one term and the latter has $(r + 1)(s + 1)(t + 1)$ terms when expanded. This intermediate expression growth is typical of many problems in symbolic manipulation. One may try avoiding this change of variable. But this is not always easily done. We have yet no general solution for this growth problem.

Appendix. This appendix contains 18 examples of factoring done by the MACSYMA system (version 252) at Project MAC, M.I.T. The running times are indicated in milliseconds. We have chosen the examples with the following complexity measurements of polynomials relative to factorization in mind: degree, number of variables, density, number of irreducible factors, number of terms in the polynomial, size and form of the coefficients, especially the leading coefficient. For each example, the order of the variables is *X*, *Y*, *Z*, *W*, with *X* the most main variable and *W* the least main. In MACSYMA, labels (Ci) and (Di) are used for the *i*th command and display lines respectively. The symbol % stands for the previous expression.

```
                            14
(D1)                       X   - 1
(C2) FACTORS (%);
TIME = 239 MSEC.

                            6    5    4    3    2            6    5    4    3    2
(D2)      (X - 1) (X + 1) (X  + X  + X  + X  + X  + X + 1) (X  - X  + X  - X  + X  - X + 1)

                            8
(D3)                       X  + X + 1
(C4) FACTOR (%);
TIME = 1005 MSEC.

                       2            6    5    3    2
(D4)                 (X  + X + 1) (X  - X  + X  - X  + 1)

                  6               5              4              3              2
(D5)   228533760 X  + 1921081160 X  + 233096077 X  - 204462708 X  + 170301571 X  - 291338682 X + 7552512
(C6) FACTOR(%);
TIME = 3326 MSEC.

                   2                      2                      2
(D6)         (455 X  + 3750 X - 99) (576 X  + 131 X - 256) (872 X  - 55 X + 298)

                            3    3
(D7)                       X  - Y
(C8) FACTOR(%);
TIME = 526 MSEC.

                               2          2
(D8)                 (X - Y) (Y  + X Y + X )

                            4    3    5
(D9)                       X  + Y  + Z
(C10) FACTOR(%);
TIME = 421 MSEC.

                            5    3    4
(D10)                      Z  + Y  + X

                      4     2       2   2      4    2  2       4
(D11)              2 X  + (Y  + 10 Z ) X  - 6 Y  - Z  Y  + 12 Z
(C12) FACTOR(%);
TIME = 1227 MSEC.

                        2      2    2      2      2      2
(D12)               (3 Z  + 2 Y  + X ) (4 Z  - 3 Y  + 2 X )

              3  3         2         2       3    2            2  2
(D13)        Y  X  + (2 Z Y  + 9 Y) X  + (Z Y  + Z Y+9 Z) X + Z  Y  + 9 Z Y
(C14) FACTOR(%);
TIME = 908 MSEC.

                              2                  2
(D14)               (Y Z + X Y  + 9) (Y Z + X Z + X  Y)

              3              2       2            2       3      2    2      3
(D15)        X  + (- Y + Z) X  + (- Y  - 2 Z Y - Z ) X + Y  + Z Y  - Z  Y - Z
(C16) FACTOR(%);
TIME = 2309 MSEC.

(D16)               (X + Y + Z) ( Z - Y + X) (- Z - Y + X)

        4              3       2          3              2       2            4                    3      2
(D17)  X  + (- Z + 3) X  + (- Y  + Z Y + Z  - 3 Z - 13) X  + (Z Y  + 3 Z Y - Z  + 15 Z + 6) X - Z Y  - 2 Y
                                                                                    4                3
                                                                                + (Z  - 15 Z) Y + 2 Z  - 30

(C18) FACTOR(%);
TIME = 1520 MSEC.

                                  2        3    2    2
(D18)               (Y Z - X Z + X  + 2) (Z  - Y  + X  + 3 X - 15)

                            6    6
(D19)                      X  + Y
(C20) FACTOR(%);
TIME = 1025 MSEC.

                      2    2    4    2  2    4
(D20)               (Y  + X ) (Y  - X  Y  + X )

                  4        2 3       2   3           3          3                         2 3       2   2
(D21)     2115 Y X  + (45 W Z  - 45 W ) X  + (- 470 Y  + (141 Z  + 94 W Z) Y) X + (- 10 W Z  + 10 W ) Y
                                                                                     2 6      3 4      2 3      3
                                                                        + 3 W Z  + 2 W Z  - 3 W Z  - 2 W Z

(C22) FACTOR(%)
TIME = 1952 MSEC.

                      3               2        3     2 3             2
(D22)             (3 Z  + 2 W Z - 10 Y  + 45 X ) (W Z  + 47 X Y - W )

              6       2       4                3            3        2                        2
(D23)        X  + (6 Y  + 1) X  + (11 Y + 24) X  + (60 Y  + 54 Y  + 10 Y + 9) X + 10 Y  + 159 Y + 135
(C24) FACTOR(%);
TIME = 1566 MSEC.
```

$$(10Y + X^3 + 9)\ (X(6Y^2 + 1) + Y + X^3 + 15) \tag{D24}$$

$$\begin{aligned}&((Z - W)\,Y^2 + (-Z^2 + W^2)\,Y + W Z^2 - W^2 Z)\,X^3 + ((-Z + W)\,Y^3 + (Z^3 - W^3)\,Y - W Z^3 + W^3 Z)\,X^2\\ &+ ((Z^2 - W^2)\,Y^3 + (-Z^3 + W^3)\,Y^2 + W^2 Z^3 - W^3 Z^2)\,X + (-W Z^2 + W^2 Z)\,Y^3 + (W Z^3 - W^3 Z)\,Y^2\\ &+ (-W^2 Z^3 + W^3 Z^2)\,Y\end{aligned} \tag{D25}$$

(C26) FACTOR(%);
TIME = 4153 MSEC.

$$(X - W)\ (Y - W)\ (X - Y)\ (Z - W)\ (X - Z)\ (Y - Z) \tag{D26}$$

$$\begin{aligned}&(Y^3 + Z Y^2 + Z^2 Y + Z^3)\,X^5 + (Y^2 + (Z + 90)\,Y + 90 Z)\,X^3 + ((Z - 11)\,Y^2 + Z^3 - 11 Z^2)\,X^2 + (Z - 11)\,Y\\ &+ 90 Z - 990\end{aligned} \tag{D27}$$

(C28) FACTOR(%);
TIME = 5072 MSEC.

$$(X^3\,(Y + Z) + Z - 11)\ (X^2\,(Z^2 + Y^2) + Y + 90) \tag{D28}$$

$$\begin{aligned}&Z Y X^3 + ((Z^2 + 1)\,Y^2 + (20 Z + 30)\,Y + Z^2 + 10 Z)\,X^2\\ &+ (Z Y^3 + (30 Z + 20)\,Y^2 + (Z^3 + 10 Z^2 + Z + 610)\,Y + 20 Z^2 + 230 Z + 300)\,X + (Z^2 + 10 Z)\,Y^2\\ &+ (30 Z^2 + 320 Z + 200)\,Y + 600 Z + 6000\end{aligned} \tag{D29}$$

(C30) FACTOR(%);
TIME = 4670 MSEC.

$$(Z + X Y + 10)\ (X Z + Y + 30)\ (Y Z + X + 20) \tag{D30}$$

$$\begin{aligned}&(54 Z^2 Y^6 - 216 W Z^2 Y^5 + 522 Z^2 Y^4)\,X^{10} + (18 W^2 Z^5 - 18 W^2 Z^2)\,Y^4 X^9\\ &+ (-24 Z^2 Y^7 + 96 W Z^2 Y^6 - 232 Z^2 Y^5)\,X^8 + ((42 W^2 - 3)\,Z^2 Y^7 + (-168 W^4 + 12 W^2)\,Z^5 Y^6\\ &+ ((-8 W^2 + 9)\,Z^5 + 6 W Z^3 + (414 W^2 - 29)\,Z^2)\,Y^5 + (-36 W^2 Z^8 - 24 W^3 Z^6)\,Y^4 + (87 Z^5 + 58 W Z^3)\,Y^3)\,X^7\\ &+ (((14 W^4 - W^2)\,Z^5 + (-14 W^4 + W^2)\,Z^2)\,Y^5 + (3 W^2 Z^8 + 2 W^3 Z^6 - 3 W^2 Z^5 - 2 W^3 Z^3)\,Y^3)\,X^6\end{aligned} \tag{D31}$$

(C32) FACTOR(%);
TIME = 11224 MSEC.

$$X^6 Y^3 Z^2\,(3 Z^3 + 2 W Z - 8 X Y^2 + (14 W^2 - 1)\,Y^2 + 18 X^3 Y)(X\,(-12 W^2 Y Z^3 + 3 Y^2 + 29) + W^2 Z^3 - W^2) \tag{D32}$$

$$\begin{aligned}&6 Y^4 X^8 + 35 Y^4 X^7 + (75 Y^4 + 53 Y^3)\,X^6 + (70 Y^4 + 226 Y^3)\,X^5 + (24 Y^4 + 314 Y^3 + 169 Y^2)\,X^4\\ &+ (142 Y^3 + 465 Y^2)\,X^3 + (311 Y^2 + 227 Y)\,X^2 + 298 Y X + 105\end{aligned} \tag{D33}$$

(C34) FACTOR(%);
TIME = 16374 MSEC.

$$(X^2 Y + X Y + 1)\ (X^2 Y + 2 X Y + 3)\ (2 X^2 Y + 3 X Y + 5)\ (3 X^2 Y + 4 X Y + 7) \tag{D34}$$

$$Y^2 X^6 + (Z Y^3 + Z^4 + 1)\,X^4 + (Y^4 + Z^3 Y^2 + Z)\,X^3 + (Z^5 + Z)\,Y X^2 + ((Z^4 + 1)\,Y^2 + Z^2 Y + Z^7 + Z^3)\,X + Z Y^2 + Z^4 \tag{D35}$$

(C36) FACTOR(%);
TIME = 41076 MSEC.

$$(Z^3 + X Y Z + Y^2 + X^3)\ (X\,(Z^4 + 1) + Z + X^3 Y^2) \tag{D36}$$

Project MAC and Department of Mathematics
Massachusets Institute of Technology
Cambridge, Massachusetts 02139

School of Mathematics
Institute for Advanced Studies
Princeton, New Jersey 08540

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

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

3. W. S. BROWN, "On Euclid's algorithm and the computation of polynomial greatest common divisors," *J. Assoc. Comput. Mach.*, v. 18, 1971, pp. 478–504. MR 46 #6570.

4. G. E. COLLINS, SAC-1 *Modular Arithmetic System*, University of Wisconsin Technical Report No. 10, June 1969.

5. A. O. GEL'FOND, *Transcendental and Algebraic Numbers*, GITTL, Moscow, 1952; English transl., Dover, New York, 1960. MR 15, 292; 22 #2598.

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

7. J. McCARTHY et al., LISP 1.5 *Programmer's Manual*, M.I.T. Press, Cambridge, Mass., 1963.

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

9. D. R. MUSSER, *Algorithms for Polynomial Factorization*, Ph. D. Thesis, Computer Science Department, The University of Wisconsin, Madison, Wis., 1971.

10. B. L. VAN DER WAERDEN, *Modern Algebra*. Vol. 1, Springer, Berlin, 1930; English transl., Ungar, New York, 1949. MR 10, 587.

11. 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, November 1974).

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

13. MACSYMA *Reference Manual*, the MATHLAB group, Project MAC, M.I.T., Cambridge, Mass., Sept. 1974.