6. NEWTON'S ITERATION AND THE HENSEL CONSTRUCTION

by

K.O. Geddes

Research Report CS-83-36

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

December 1983

6. NEWTON'S ITERATION AND THE HENSEL CONSTRUCTION

In this chapter we continue our discussion of techniques for inverting modular and evaluation homomorphisms defined on the domain $\mathbf{Z}[x_1, \ldots, x_v]$. The particular methods developed in this chapter are based on Newton's iteration for solving a polynomial equation. Unlike the integer and polynomial Chinese remainder algorithms of the preceding chapter, algorithms based on Newton's iteration generally require only *one* image of the solution in a domain of the form $\mathbf{Z}_p[x_1]$ from which to reconstruct the desired solution in the larger domain $\mathbf{Z}[x_1, \ldots, x_v]$. A particularly important case of Newton's iteration to be discussed here is the *Hensel construction*. It will be seen in succeeding chapters that multivariate polynomial computations (such as GCD computation and factorization) can be performed much more efficiently (in most cases) by methods based on the Hensel construction than by methods based on the Chinese remainder algorithms of the preceding chapter.

6.1. P-ADIC AND IDEAL-ADIC REPRESENTATIONS

The reason for introducing homomorphism techniques is that a gain in efficiency can be realized for many practical problems by solving several *image problems* in simpler domains (such as $\mathbf{Z}_{p_i}[x_1]$ or $\mathbf{Z}_{p_i}$) rather than directly solving a given problem in a more complicated domain (such as $\mathbf{Z}[x_1, \ldots, x_v]$) (cf. Figure 5.1). However the particular homomorphism techniques discussed in chapter 5 have the potentially serious drawback that the *number* of image problems that have to be solved grows exponentially with the size of the solution (where by the 'size' of the solution we mean some measure that takes into account the magnitudes of the integer coefficients, the number of indeterminates, and the degree of the solution in each indeterminate). This fact can be seen by referring to the homomorphism diagram of Figure 5.1 in which we see that the number of image problems that have to be solved is $(n+1)\mathrm{N}$, where $n+1$ is the number of moduli required for the residue representation of the integer coefficients that can appear in the solution and where N is the number of multivariate evaluation homomorphisms required. Noting that if the degree of the solution in each of v indeterminates is d then $\mathrm{N} = (d+1)^{v-1}$ (because $d+1$ evaluation points are required for each of the $v-1$ indeterminates being eliminated), we see that the number of image problems is given by

$$(n + 1)(d + 1)^{v-1}$$

which grows exponentially as the number of indeterminates increases.

There is a homomorphism technique of a rather different nature in which only *one* image problem is constructed and solved and then this image solution is 'lifted' to the solution in the original domain by solving some nonlinear equations associated with the problem. Of course nothing really comes for free and this new method can be viewed as trading off a sharp decrease in the computational cost of solving image problems with a sharp increase the computational cost of 'lifting' the image solution to the larger domain. In other words, the new algorithm will be rather more complicated than the interpolation and Chinese remainder algorithms which perform the corresponding 'lifting' process in the diagram of Figure 5.1. However this new approach has been found to be significantly more efficient for many practical problems. (Actually the efficiency of the new approach lies mainly in its ability to take advantage of sparseness in the polynomial solution, and as we noted in chapter 3 polynomials which arise in practical problems involving several indeterminates will invariably be sparse).

P-adic Representation and Approximation

Consider the problem of inverting the modular homomorphism $\phi_p : \mathbf{Z}[\mathbf{x}] \to \mathbf{Z}_p[\mathbf{x}]$. The starting point in the development of the new algorithm is to consider yet another representation for integers and polynomials. Recall that in applying Garner's algorithm to solve the Chinese remainder problem, the integer solution u is developed (in step 2 of Algorithm 5.1) in its mixed radix representation:

$$u = v_0 + v_1(m_0) + v_2(m_0m_1) + \cdots + v_n(\prod_{i=0}^{n-1} m_i)$$

where m_i $(0 \le i \le n)$ are odd positive moduli such that $\prod_{i=0}^{n} m_i > 2|u|$ and $v_k \in \mathbf{Z}_{m_k}$ $(0 \le k \le n)$. The new approach is based on developing an integer solution u in its *p-adic representation:*

$$(1) \quad u = u_0 + u_1p + u_2p^2 + \cdots + u_np^n$$

where p is an odd positive prime integer, n is such that $p^{n+1} > 2|u|$, and $u_i \in \mathbf{Z}_p$ $(0 \le i \le n)$. As in the case of the mixed radix representation, the p-adic representation can be developed using either the positive or the symmetric representation of $\mathbf{Z}_p$. Obviously if the positive representation is used then (1) is simply the familiar *radix p representation* of the nonnegative integer u (and it is sufficient for n to be such that $p^{n+1} > u$). However as we have seen, the symmetric representation is more useful in practice because then the integer u is allowed to be negative.

There is a simple procedure for developing the p-adic representation for a given integer u. Firstly we see from (1) that $u \equiv u_0 \pmod p$, so using the modular mapping $\phi_p(a) = \mathrm{rem}(a,p)$ we have

$$(2) \quad u_0 = \phi_p(u).$$

For the next p-adic coefficient u_1, note that $u - u_0$ must be divisible by p and from (1) it follows that

$$\frac{u-u_0}{p} = u_1 + u_2p + \cdots + u_np^{n-1}.$$

Hence as before, we have

$$u_1 = \phi_p(\frac{u-u_0}{p}).$$

Continuing in this manner, we get

$$(3) \quad u_i = \phi_p(\frac{u - [u_0 + u_1p + \cdots + u_{i-1}p^{i-1}]}{p^i}), \; i = 1,2, \ldots, n$$

where the division by p^i is guaranteed to be an exact integer division. In formula (3) it is important to note that the calculation is to be performed in the domain $\mathbf{Z}$ and then finally the modular mapping ϕ_p is applied (unlike the 'algorithmic specification' of the ϕ_p notation previously used).

Example 6.1.

Let $u = -272300$ be the integer which arose as the solution in Example 5.15 where a mixed radix representation of u was developed. Let us develop the p-adic representation of u choosing p to be the largest two-digit prime, namely $p = 97$. The p-adic coefficients are

$$u_0 = \phi_p(u) = -21;$$

$$u_1 = \phi_p(\frac{u - u_0}{p}) = 6;$$

$$u_2 = \phi_p(\frac{u - [u_0 + u_1 p]}{p^2}) = -29.$$

(If we try to compute another coefficient u_3 we find that $u - [u_0 + u_1 p + u_2 p^2] = 0$ so we are finished). Thus the p-adic representation of $u = -272300$ when $p = 97$ is:

$$-272300 = -21 + 6(97) - 29(97)^2. \quad \square$$

As in the case of a mixed radix representation, the concept of a p-adic representation can be readily extended to polynomials. Consider the polynomial

$$\mathrm{u}(\mathbf{x}) = \sum_e u_e \mathbf{x}^e \in \mathbf{Z}[\mathbf{x}]$$

and let p and n be chosen such that $p^{n+1} > 2u_{\max}$, where $u_{\max} = \max_e | u_e |$. If each integer coefficient u_e is expressed in its p-adic representation

$$u_e = \sum_{i=0}^{n} u_{e,i} p^i \text{ with } u_{e,i} \in \mathbf{Z}_p$$

then the polynomial $\mathrm{u}(\mathbf{x})$ can be expressed as

$$\mathrm{u}(\mathbf{x}) = \sum_e (\sum_{i=0}^{n} u_{e,i} p^i) \mathbf{x}^e = \sum_{i=0}^{n} (\sum_e u_{e,i} \mathbf{x}^e) p^i.$$

The latter expression for the polynomial $\mathrm{u}(\mathbf{x})$ is called a *polynomial p-adic representation* and its general form is

$$(4) \quad \mathrm{u}(\mathbf{x}) = u_0(\mathbf{x}) + u_1(\mathbf{x})p + u_2(\mathbf{x})p^2 + \cdots + u_n(\mathbf{x})p^n$$

where $u_i(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ for $i = 0, 1, \ldots, n$. Formulas (2) - (3) remain valid when u and u_i $(0 \le i \le n)$ are polynomials.

Example 6.2.

Let $\mathrm{u}(x) = 14x^2 - 11x - 15 \in \mathbf{Z}[x]$ be the polynomial which arose as the solution in Example 5.17 where a polynomial mixed radix representation of $\mathrm{u}(x)$ was developed. Let us develop the polynomial p-adic representation of $\mathrm{u}(x)$ choosing $p = 5$. The polynomial p-adic coefficients are:

$$u_0(x) = \phi_p(\mathrm{u}(x)) = -x^2 - x;$$

$$u_1(x) = \phi_p(\frac{\mathrm{u}(x) - u_0(x)}{p}) = -2x^2 - 2x + 2;$$

$$u_2(x) = \phi_p(\frac{\mathrm{u}(x) - [u_0(x) + u_1(x)p]}{p^2}) = x^2 - 1.$$

(If we try to compute another coefficient $u_3(x)$ we find that $\mathrm{u}(x) - [u_0(x) + u_1(x)p + u_2(x)p^2] = 0$ so we are finished). Thus the polynomial p-adic representation of the given polynomial $\mathrm{u}(x) \in \mathbf{Z}[x]$ when $p = 5$ is:

$$u(x) = (-x^2 - x) + (-2x^2 - 2x + 2)5 + (x^2 - 1)5^2. \quad \square$$

It is useful to introduce a concept of *approximation* which is associated with a polynomial p-adic representation. Recall that the congruence relation

$$a(\mathbf{x}) \equiv b(\mathbf{x}) \pmod q$$

defined on the domain $\mathbf{Z}[\mathbf{x}]$ with respect to a principal ideal $<q>$ in $\mathbf{Z}[\mathbf{x}]$ has the meaning:

$$a(\mathbf{x}) - b(\mathbf{x}) \in <q>$$

(i.e. $a(\mathbf{x}) - b(\mathbf{x})$ is a multiple of q). Using this congruence notation, it is readily seen that the following relations hold for the polynomials appearing in the polynomial p-adic representation (4):

$$u(\mathbf{x}) \equiv u_0(\mathbf{x}) \pmod p$$

and more generally

$$u(\mathbf{x}) \equiv u_0(\mathbf{x}) + u_1(\mathbf{x})p + \cdots + u_{k-1}(\mathbf{x})p^{k-1} \pmod{p^k}, \; 1 \le k \le n+1.$$

We thus have a (finite) sequence of approximations to the polynomial $u(\mathbf{x})$ in the sense of the following definition.

Definition 6.1.

Let $a(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$ be a given polynomial. A polynomial $b(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$ is called an *order n p-adic approximation* to $a(\mathbf{x})$ if

$$a(\mathbf{x}) \equiv b(\mathbf{x}) \pmod{p^n}.$$

The *error* in approximating $a(\mathbf{x})$ by $b(\mathbf{x})$ is $a(\mathbf{x}) - b(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$. $\square$

Multivariate Taylor Series Representation

We now consider a generalization of the p-adic representation which will lead to a new technique for inverting a multivariate evaluation homomorphism

$$(5) \quad \phi_I : \mathbf{Z}_p[x_1, \ldots, x_v] \to \mathbf{Z}_p[x_1]$$

with kernel $I = <x_2 - \alpha_2, \ldots, x_v - \alpha_v>$ for some specified values $\alpha_i \in \mathbf{Z}_p$ $(2 \le i \le v)$. As before, the key to the development of the new algorithm is to choose an appropriate representation for the solution. In this case the 'solution' is a multivariate polynomial $\hat{u} = u(x_1, \ldots, x_v) \in \mathbf{Z}_p[x_1, \ldots, x_v]$ and the 'first term' of $\hat{u}$ is a univariate polynomial $u^{(1)} \in \mathbf{Z}_p[x_1]$, where

$$(6) \quad u^{(1)} = \phi_I(\hat{u}).$$

Note that

$$u^{(1)} = u(x_1, \alpha_2, \ldots, \alpha_v).$$

Corresponding to the previous representation, suppose that we choose a representation for the solution $\hat{u}$ of the form

$$(7) \quad \hat{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \Delta u^{(3)} + \cdots$$

with the first term given by (6). In order to determine the remaining terms, consider the 'error' $e^{(1)} = \hat{u} - u^{(1)}$ and note that from (6) we clearly have $\phi_I(e^{(1)}) = 0$ whence

$$(8) \quad e^{(1)} \in I.$$

Now any element of the ideal I can be expressed as a linear combination of the basis elements of I, so (8) can be expressed as

$$(9)\quad e^{(1)} = \sum_{i=2}^{v} c_i(x_i - \alpha_i), \;\text{ where }\; c_i \in \mathbf{Z}_p[x_1, \ldots, x_v].$$

For the first 'correction term' $\Delta u^{(1)}$ in the representation (7), we choose the linear terms in the error expression (9) defined by

$$(10)\quad \Delta u^{(1)} = \sum_{i=2}^{v} u_i(x_1)\,(x_i - \alpha_i)$$

where the coefficients $u_i(x_1) \in \mathbf{Z}_p[x_1]$ are given by

$$(11)\quad u_i(x_1) = \phi_{\mathrm{I}}(c_i), \;\; 2 \le i \le v.$$

Note that $\Delta u^{(1)} \in \mathrm{I}$. At this point we have the 'approximation' to $\hat{u}$

$$u^{(2)} = u^{(1)} + \Delta u^{(1)}$$

defined by (6) and (10). Consider the new error term

$$e^{(2)} = \hat{u} - u^{(2)} = e^{(1)} - \Delta u^{(1)}.$$

Applying (9) and (10) we have

$$e^{(2)} = \sum_{i=2}^{v} (c_i - u_i(x_1))(x_i - \alpha_i).$$

Now

$$c_i - u_i(x_1) \in \mathrm{I}, \;\; 2 \le i \le v$$

because from (11) clearly $\phi_{\mathrm{I}}(c_i - u_i(x_1)) = 0$, which implies that

$$(12)\quad e^{(2)} \in \mathrm{I}^2.$$

In order to understand the statement (12) (and similar statements in the sequel) let us recall from chapter 5 the definition of the i-th power of an ideal I with finite basis. Specifically, I^2 is the ideal generated by all pairs of products of basis elements of I, I^3 is the ideal generated by all triples of products of basis elements of I, and so on. In our particular case since the basis elements of $\mathrm{I} = \langle x_2 - \alpha_2, \ldots, x_v - \alpha_v \rangle$ are linear terms, the basis elements of I^2 will be multivariate terms of total degree 2 and, in general, the basis elements of I^i will be multivariate terms of total degree i. As a clarification, consider the particular case where $v = 3$ in which case we have:

$$\begin{aligned}
\mathrm{I} &= \langle x_2 - \alpha_2, x_3 - \alpha_3 \rangle;\\
\mathrm{I}^2 &= \langle (x_2 - \alpha_2)^2, (x_2 - \alpha_2)(x_3 - \alpha_3), (x_3 - \alpha_3)^2 \rangle;\\
\mathrm{I}^3 &= \langle (x_2 - \alpha_2)^3, (x_2 - \alpha_2)^2(x_3 - \alpha_3), (x_2 - \alpha_2)(x_3 - \alpha_3)^2, (x_3 - \alpha_3)^3 \rangle;\\
&\;\;\vdots\\
\mathrm{I}^i &= \langle (x_2 - \alpha_2)^i, (x_2 - \alpha_2)^{i-1}(x_3 - \alpha_3), \ldots, (x_2 - \alpha_2)(x_3 - \alpha_3)^{i-1}, (x_3 - \alpha_3)^i \rangle.
\end{aligned}$$

The result (12) should now be evident. Expressing $e^{(2)} \in \mathrm{I}^2$ as a linear combination of the basis elements of I^2 yields

$$e^{(2)} = \sum_{i=2}^{v} \sum_{j=i}^{v} c_{ij}(x_i - \alpha_i)(x_j - \alpha_j), \;\text{ where }\; c_{ij} \in \mathbf{Z}_p[x_1, \ldots, x_v].$$

The next correction term in the representation (7) is the term $\Delta u^{(2)} \in \mathrm{I}^2$ defined by

$$(13)\quad \Delta u^{(2)} = \sum_{i=2}^{v} \sum_{j=i}^{v} u_{ij}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)$$

where the coefficients $u_{ij}(x_1) \in \mathbf{Z}_p[x_1]$ are given by

$$u_{ij}(x_1) = \phi_{\mathrm{I}}(c_{ij}), \;\; 2 \le i \le j \le v.$$

We then have the 'approximation' to $\tilde{u}$

$$u^{(3)} = u^{(2)} + \Delta u^{(2)} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)}$$

defined by (6), (10), and (13). Continuing in this manner, we can show that

$$e^{(3)} \in \mathrm{I}^3$$

where $e^{(3)} = \tilde{u} - u^{(3)}$ and we can proceed to define the next correction term $\Delta u^{(3)} \in \mathrm{I}^3$ in the form

$$\Delta u^{(3)} = \sum_{i=2}^{v} \sum_{j=i}^{v} \sum_{k=j}^{v} u_{ijk}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)(x_k - \alpha_k)$$

for some coefficients $u_{ijk}(x_1) \in \mathbf{Z}_p[x_1]$. This process will eventually terminate because the solution $\tilde{u}$ is a polynomial. Specifically, if d denotes the *total degree* of $\tilde{u}$ as an element of the domain $\mathbf{Z}_p[x_1][x_2, \ldots, x_v]$ (i.e. as a polynomial in the indeterminates $x_2, \ldots, x_v$) then with

$$u^{(d+1)} = u^{(1)} + \Delta u^{(1)} + \cdots + \Delta u^{(d)}$$

we will have $e^{(d+1)} = \tilde{u} - u^{(d+1)} = 0$ so that $u^{(d+1)}$ is the desired polynomial. This must be so because each correction term $\Delta u^{(k)} \in \mathrm{I}^k$ is of total degree k (with respect to $x_2, \ldots, x_v$).

The representation (7) which we have just developed for a polynomial $\tilde{u} = \mathrm{u}(x_1, \ldots, x_v) \in \mathbf{Z}_p[x_1, \ldots, x_v]$ is called the *multivariate Taylor series representation* with respect to the ideal $\mathrm{I} = <x_2 - \alpha_2, \ldots, x_v - \alpha_v>$ and its general form is

$$\begin{aligned} (14) \quad \mathrm{u}(x_1, \ldots, x_v) = \mathrm{u}(x_1, \alpha_2, \ldots, \alpha_v) &+ \sum_{i=2}^{v} u_i(x_1)(x_i - \alpha_i) \\ &+ \sum_{i=2}^{v} \sum_{j=i}^{v} u_{ij}(x_1)(x_i - \alpha_i)(x_j - \alpha_j) \\ &+ \sum_{i=2}^{v} \sum_{j=i}^{v} \sum_{k=j}^{v} u_{ijk}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)(x_k - \alpha_k) + \cdots . \end{aligned}$$

The number of terms here will be finite with the last term containing d nested summations, where d is the total degree of $\mathrm{u}(x_1, \ldots, x_v)$ with respect to the indeterminates $x_2, \ldots, x_v$.

Ideal-adic Representation and Approximation

The multivariate Taylor series representation (14) for a polynomial $\mathrm{u}(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ can be viewed as a direct generalization of a polynomial p-adic representation. Recall that the polynomial p-adic representation of a polynomial $\tilde{u} = \mathrm{u}(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$ can be expressed in the form

$$\tilde{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \cdots + \Delta u^{(n)}$$

where

$$u^{(1)} = u_0(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]/<p>;$$

$$\Delta u^{(k)} = u_k(\mathbf{x})p^k \in <p>^k, \;\; \text{for } k = 1,2, \ldots, n.$$

Note here that $\mathbf{Z}[\mathbf{x}]/<p> = \mathbf{Z}_p[\mathbf{x}]$ and that $<p>^k = <p^k>$. We also have the property that the coefficient $u_k(\mathbf{x})$ in the expression for $\Delta u^{(k)}$ as a multiple of the basis element of the ideal in which it lies satisfies

$$u_k(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]/<p>, \;\; 1 \le k \le n.$$

In the p-adic case, we may define a sequence of order $k+1$ p-adic approximations

$$u^{(k+1)} \in \mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k+1}, \quad \text{for } k = 1,2, \ldots, n$$

where

$$u^{(k+1)} = u^{(1)} + \Delta u^{(1)} + \cdots + \Delta u^{(k)}.$$

In defining the k-th element of this sequence, we have an approximation $u^{(k)} \in \mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k}$ and we define the new approximation $u^{(k+1)} \in \mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k+1}$ by adding the term $\Delta u^{(k)} \in \langle p\rangle^{k}$; the addition

$$u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$$

is an addition in the larger domain $\mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k+1}$ and is made valid by assuming the natural embedding of the domain $\mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k}$ into the larger domain $\mathbf{Z}[\mathbf{x}]/\langle p\rangle^{k+1}$. Thus the successive p-adic approximations $u^{(1)}, u^{(2)}, u^{(3)}, \cdots$ to $\tilde{u} \in \mathbf{Z}[\mathbf{x}]$ lie in a sequence of subdomains of $\mathbf{Z}[\mathbf{x}]$ of increasing size indicated by

$$\mathbf{Z}[\mathbf{x}]/\langle p\rangle \subset \mathbf{Z}[\mathbf{x}]/\langle p\rangle^{2} \subset \mathbf{Z}[\mathbf{x}]/\langle p\rangle^{3} \subset \cdots \subset \mathbf{Z}[\mathbf{x}].$$

Noting that a polynomial $\tilde{u} \in \mathbf{Z}[\mathbf{x}]$ has a finite polynomial p-adic representation, it is clear that for some $k = n$ the subdomain $\mathbf{Z}[\mathbf{x}]/\langle p\rangle^{n+1}$ will be large enough to contain the polynomial $\tilde{u}$.

The multivariate Taylor series representation (14) for a polynomial $\tilde{u} = \mathrm{u}(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ can be viewed in an abstractly equivalent manner with the ideal I taking the place of the ideal $\langle p\rangle$ above. The polynomial $\tilde{u}$ was developed in the form

$$\tilde{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \cdots + \Delta u^{(d)}$$

where

$$u^{(1)} = \mathrm{u}(x_1, \alpha_2, \ldots, \alpha_v) \in \mathbf{Z}_p[\mathbf{x}]/\mathrm{I};$$

$$\Delta u^{(k)} \in \mathrm{I}^k, \quad \text{for } k = 1,2, \ldots, d.$$

Here $\mathbf{x} = (x_1, \ldots, x_v)$, $\mathrm{I} = \langle x_2 - \alpha_2, \ldots, x_v - \alpha_v\rangle$, and note that $\mathbf{Z}_p[\mathbf{x}]/\mathrm{I} = \mathbf{Z}_p[x_1]$. Corresponding to the p-adic case, we have the additional property that for each k the coefficients in the expression for $\Delta u^{(k)}$ as a linear combination of the basis elements of the ideal I^k all lie in the domain $\mathbf{Z}_p[\mathbf{x}]/\mathrm{I}$. (For example,

$$\Delta u^{(2)} = \sum_{i=2}^{v} \sum_{j=i}^{v} u_{ij}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)$$

with $u_{ij}(x_1) \in \mathbf{Z}_p[\mathbf{x}]/\mathrm{I}$, $2 \le i \le j \le v$). It is therefore appropriate to speak of a sequence of approximations (see Definition 6.2) to $\tilde{u}$ defined by

$$u^{(k+1)} \in \mathbf{Z}_p[\mathbf{x}]/\mathrm{I}^{k+1}, \quad \text{for } k = 1,2, \ldots, d$$

where

$$u^{(k+1)} = u^{(1)} + \Delta u^{(1)} + \cdots + \Delta u^{(k)}.$$

Again we must assume a natural embedding of domains and the sequence of approximations $u^{(1)}, u^{(2)}, u^{(3)}, \cdots$ to $\tilde{u} \in \mathbf{Z}_p[\mathbf{x}]$ lie in the following sequence of subdomains of $\mathbf{Z}_p[\mathbf{x}]$ of increasing size:

$$\mathbf{Z}_p[\mathbf{x}]/\mathrm{I} \subset \mathbf{Z}_p[\mathbf{x}]/\mathrm{I}^2 \subset \mathbf{Z}_p[\mathbf{x}]/\mathrm{I}^3 \subset \cdots \subset \mathbf{Z}_p[\mathbf{x}].$$

As in the p-adic case, since the multivariate Taylor series representation for $\tilde{u}$ is finite there is an index $k = d$ such that the subdomain $\mathbf{Z}_p[\mathbf{x}]/\mathrm{I}^{d+1}$ is large enough to contain the polynomial $\tilde{u}$.

In view of this close correspondence with the p-adic representation of a polynomial, the multivariate Taylor series representation (14) of a polynomial $\mathrm{u}(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ is also called the *ideal-adic representation* of $u(\mathbf{x})$ with respect to the ideal $I = <x_2-\alpha_2, \ldots, x_v-\alpha_v>$. The concept of approximation mentioned above is made precise by the following definition which is an obvious abstraction of Definition 6.1.

Definition 6.2.

Let D be a Noetherian integral domain and let I be an ideal in D. For a given element $a \in D$, the element $b \in D$ is called an *order n ideal-adic approximation* to a with respect to the ideal I if

$$a \equiv b \pmod{I^n}.$$

The *error* in approximating a by b is the element $a - b \in D$. □

Recalling that $a \equiv b \pmod{I^n}$ means that $a - b \in I^n$, it is clear from the development of the ideal-adic representation (multivariate Taylor series representation) (14) for $u(x_1, \ldots, x_v) \in \mathbf{Z}_p[x_1, \ldots, x_v]$ that $u^{(k)}$ is an order k ideal-adic approximation to $u(x_1, \ldots, x_v)$ with respect to the ideal $I = <x_2-\alpha_2, \ldots, x_v-\alpha_v>$, where

$$u^{(1)} = u(x_1, \alpha_2, \ldots, \alpha_v);$$
$$u^{(k+1)} = u^{(k)} + \Delta u^{(k)}, \text{ for } k = 1,2, \ldots, d;$$

with $\Delta u^{(k)}$ defined to be the term in (14) of total degree k with respect to I (i.e. the term represented by k nested summations). In connection with the concept of ideal-adic approximation it is useful to note the following computational definition of the homomorphism ϕ_{I^n} defined on the domain $\mathbf{Z}_p[\mathbf{x}]$, where $I = <x_2-\alpha_2, \ldots, x_v-\alpha_v>$. Since

$$\phi_{I^n}: \mathbf{Z}_p[\mathbf{x}] \to \mathbf{Z}_p[\mathbf{x}]/I^n$$

denotes the homomorphism with kernel I^n, if the polynomial $a(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ is represented in its ideal-adic representation with respect to the ideal I then $\phi_{I^n}(a(\mathbf{x}))$ is precisely the order n ideal-adic approximation to $a(\mathbf{x})$ obtained by dropping all terms in the ideal-adic representation of $a(\mathbf{x})$ which have total degree equal to or exceeding n (with respect to I).

6.2. NEWTON'S ITERATION FOR $f(u) = 0$

Linear p-adic Iteration

We wish to develop a method corresponding to the Chinese remainder algorithm for inverting the modular homomorphism $\phi_p : \mathbf{Z}[\mathbf{x}] \to \mathbf{Z}_p[\mathbf{x}]$. In the new approach we assume that we use only one prime p and that we know the image $u_0(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ of the desired solution $u(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$. In the terminology of the preceding section, $u_0(\mathbf{x})$ is an order 1 (or first-order) p-adic approximation to $u(\mathbf{x})$ and it is also the first term in the polynomial p-adic representation of $u(\mathbf{x})$. We will develop a method to compute successively the order k approximation

$$u_0(\mathbf{x}) + u_1(\mathbf{x})p + \cdots + u_{k-1}(\mathbf{x})p^{k-1} \in \mathbf{Z}_{p^k}[\mathbf{x}],$$

for $k = 1,2, \ldots, n+1$. Then the order $n+1$ approximation which lies in the domain $\mathbf{Z}_{p^{n+1}}[\mathbf{x}]$ is the desired solution $u(\mathbf{x}) \in \mathbf{Z}[\mathbf{x}]$ (assuming that n was chosen large enough). This general process is called *lifting* the image $u_0(\mathbf{x}) \in \mathbf{Z}_p[\mathbf{x}]$ to the solution $u(\mathbf{x})$ in the larger domain $\mathbf{Z}[\mathbf{x}]$.

The lifting process clearly requires more information about the solution $u(\mathbf{x})$ than simply the single image $u_0(\mathbf{x})$. We will assume that the additional information can be specified in the form of one or more equations (usually nonlinear) which $u(\mathbf{x})$ must satisfy. For now, let us assume that the solution $u = u(x)$ is known to satisfy

$$(15) \quad f(u) = 0$$

where $f(u) \in Z[x][u]$ – i.e. $f(u)$ is some polynomial expression in u with coefficients lying in the domain $Z[x]$. The basic idea of the new approach is to have an iterative method which will improve the given first-order p-adic approximation $u_0(x)$ into successively higher-order p-adic approximations to the solution $u(x)$ of (15). The iterative process will be finite if (15) has a polynomial solution $u(x)$ since, in the above notation, the order $n + 1$ p-adic approximation to $u(x)$ will be $u(x)$ itself.

Let us recall the classical Newton's iteration for solving a nonlinear equation of the form (15) in the traditional analytic setting where $f(u)$ is a differentiable real-valued function of a real variable u. Letting $u^{(k)}$ denote an approximation to a solution $\hat{u}$ and expanding the function $f(u)$ in a Taylor series about the point $u^{(k)}$, we have

$$f(u) = f(u^{(k)}) + f'(u^{(k)})(u - u^{(k)}) + ½f''(u^{(k)})(u - u^{(k)})^2 + \cdots .$$

Setting $u = \hat{u}$, the left hand side becomes zero and retaining only linear terms in the Taylor series we have the approximate equality:

$$0 \approx f(u^{(k)}) + f'(u^{(k)})(\hat{u} - u^{(k)}).$$

Solving for $\hat{u}$ and calling it the new approximation $u^{(k+1)}$, we have Newton's iterative formula:

$$u^{(k+1)} = u^{(k)} - \frac{f(u^{(k)})}{f'(u^{(k)})}$$

(where we need the assumption that $f'(u^{(k)}) \neq 0$). The iteration must be started with an initial guess $u^{(1)}$ and using techniques of real analysis it can be proved that if $u^{(1)}$ is 'close enough' to a solution $\hat{u}$ of $f(u) = 0$ and if $f'(\hat{u}) \neq 0$ then the infinite iteration specified above will converge (quadratically) to the solution $\hat{u}$. We will develop a similar iterative formula for our polynomial setting and it will have two significant computational advantages over the traditional analytic case: (i) the first-order p-adic approximation will be sufficient to give *guaranteed convergence*, and (ii) the iteration will be *finite*.

We wish to solve the polynomial equation *assuming that it has a polynomial solution* $\hat{u} = u(x) \in Z[x]$, given the first-order p-adic approximation $u_0(x) \in Z_p[x]$ to $\hat{u}$. (Note that an arbitrary polynomial equation of the form (15) would not in general have a polynomial solution but we are assuming a context in which a polynomial solution is known to exist). Writing the solution in its polynomial p-adic representation

$$(16) \quad \hat{u} = u_0(x) + u_1(x)p + \cdots + u_n(x)p^n$$

we wish to determine the polynomial p-adic coefficients $u_i(x) \in Z_p[x]$ for $i = 1,2, \ldots, n$ ($u_0(x)$ is given). Let us denote by $u^{(k)}$ the order k p-adic approximation to $\hat{u}$ given by the first k terms of (16); thus $u^{(1)} = u_0(x)$ and in general

$$u^{(k)} = u_0(x) + u_1(x)p + \cdots + u_{k-1}(x)p^{k-1}, 1 \leq k \leq n + 1.$$

We would like an iteration formula which at step k is given the order k approximation $u^{(k)}$ and which computes the polynomial p-adic coefficient $u_k(x) \in Z_p[x]$ yielding the order $k + 1$ approximation

$$(17) \quad u^{(k+1)} = u^{(k)} + u_k(x)p^k, 1 \leq k \leq n.$$

By Theorem 2.8 of chapter 2 applied to the polynomial $f(u) \in D[u]$ where $D = Z[x]$, we have the following 'Taylor series expansion':

$$(18) \quad f(u^{(k)} + u_k(x)p^k) = f(u^{(k)}) + f'(u^{(k)})u_k(x)p^k + g(u^{(k)},u_k(x)p^k)[u_k(x)]^2p^{2k}$$

for some polynomial $g(u, w) \in D[u, w]$.

At this point we need to use a property of congruences. Recall the congruence properties developed in chapter 5 which show that congruences can be added, subtracted, and multiplied. As a direct consequence of these properties, it follows that if I is any ideal in a commutative ring R and if $h(x) \in R[x]$ is any polynomial expression over R then for $a, b \in R$:

$$(19) \quad a \equiv b \pmod{I} \implies h(a) \equiv h(b) \pmod{I}.$$

Now since $u^{(k)} \equiv \hat{u} \pmod{p^k}$, applying property (19) and the fact that $f(\ddot{u}) = 0$ yields

$$f(u^{(k)}) \equiv 0 \pmod{p^k}.$$

Similarly,

$$f(u^{(k)} + u_k(x)p^k) \equiv 0 \pmod{p^{k+1}}$$

if (17) is to define the order $k + 1$ approximation $u^{(k+1)}$. Therefore we can divide by p^k in (18) yielding

$$\frac{f(u^{(k)} + u_k(x)p^k)}{p^k} = \frac{f(u^{(k)})}{p^k} + f'(u^{(k)})u_k(x) + g(u^{(k)}, u_k(x)p^k)[u_k(x)]^2 p^k.$$

Now applying the modular homomorphism ϕ_p and noting that the left hand side is still a multiple of p, we find that the desired polynomial p-adic coefficient $u_k(x) \in Z_p[x]$ must satisfy

$$0 = \phi_p\left(\frac{f(u^{(k)})}{p^k}\right) + \phi_p(f'(u^{(k)}))u_k(x) \in Z_p[x].$$

Finally since $u^{(k)} \equiv u^{(1)} \pmod{p}$ for all $k \geq 1$, it follows from property (19) that

$$f'(u^{(k)}) \equiv f'(u^{(1)}) \pmod{p}.$$

Therefore if the given first-order approximation $u^{(1)}$ satisfies the condition

$$f'(u^{(1)}) \not\equiv 0 \pmod{p}$$

then the desired polynomial p-adic coefficient is given by

$$(20) \quad u_k(x) = -\frac{\phi_p\left(\frac{f(u^{(k)})}{p^k}\right)}{\phi_p(f'(u^{(1)}))} \in Z_p[x].$$

The division appearing in (20) must be an exact division in the polynomial domain $Z_p[x]$ if equation (15) has a polynomial solution. The iteration formula (17) together with the *linear update formula* (20) is known as the *linear p-adic Newton's iteration*. Note that in formula (20) the calculation of $f(u^{(k)})$ must be performed in the domain $Z[x]$, followed by an exact division by p^k in $Z[x]$, before the modular homomorphism ϕ_p is applied.

Example 6.3.

Consider the problem of determining a polynomial $u(x) \in Z[x]$ which is a square root of the polynomial

$$a(x) = 36x^4 - 180x^3 + 93x^2 + 330x + 121 \in Z[x]$$

(assuming that $a(x)$ is a perfect square). Then $u(x)$ can be expressed as the solution of the polynomial equation

$$f(u) = a(x) - u^2 = 0.$$

Choosing $p = 5$, the first-order p-adic approximation $u^{(1)} = u_0(x) \in Z_5[x]$ must be a square root of $\phi_5(a(x))$ in $Z_5[x]$. Now

$$\phi_5(a(x)) = x^4 - 2x^2 + 1$$

which clearly has the square root

$$u^{(1)} = u_0(x) = x^2 - 1 \in \mathbf{Z}_5[x].$$

Now to apply the linear p-adic Newton's iteration, first note that

$$\phi_5(\mathrm{f}'(u^{(1)})) = \phi_5(-2u^{(1)}) = -2x^2 + 2.$$

Then

$$u_1(x) = -\frac{\phi_5(\frac{\mathrm{f}(u^{(1)})}{5})}{(-2x^2 + 2)} = -\frac{\phi_5(\frac{35x^4 - 180x^3 + 95x^2 + 330x + 120}{5})}{(-2x^2 + 2)}$$

$$= -\frac{(2x^4 - x^3 - x^2 + x - 1)}{(-2x^2 + 2)} = x^2 + 2x - 1 \in \mathbf{Z}_5[x]$$

yielding

$$u^{(2)} = (x^2 - 1) + (x^2 + 2x - 2)5 \in \mathbf{Z}_{25}[x].$$

Similarly we get

$$u_2(x) = -\frac{(-2x^3 + 2x)}{(-2x^2 + 2)} = -x \in \mathbf{Z}_5[x]$$

yielding

$$u^{(3)} = (x^2 - 1) + (x^2 + 2x - 2)5 + (-x)5^2 \in \mathbf{Z}_{125}[x].$$

If we proceed to calculate another polynomial p-adic coefficient $u_3(x)$ we find that $\mathrm{f}(u^{(3)}) = 0$ (in the domain $\mathbf{Z}[x]$) so we are finished. The desired square root of $\mathrm{a}(x)$ is therefore

$$\mathrm{u}(x) = u^{(3)} = 6x^2 - 15x - 11 \in \mathbf{Z}[x]. \quad \square$$

Quadratic p-adic Iteration

Newton's iteration as specified by (17) and (20) increases the order of approximation by one per iteration. However it is possible to develop Newton's iteration in such a way that the order of approximation *doubles* per iteration and this corresponds to the concept of *quadratic* convergence familiar in the analytic applications of Newton's iteration. In the quadratic version, at step k we have the order $n_k = 2^{k-1}$ approximation

$$u^{(k)} = u_0(\mathbf{x}) + u_1(\mathbf{x})p + \cdots + u_{n_k-1}(\mathbf{x})p^{n_k-1}$$

to a solution $\tilde{\mathrm{u}}$ of $\mathrm{f(u)} = 0$ and we compute an update $\Delta u^{(k)}$ such that

$$\textbf{(21)} \quad u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$$

is an order $2n_k = 2^k$ approximation; namely,

$$\Delta u^{(k)} = u_{n_k}(\mathbf{x})p^{n_k} + \cdots + u_{2n_k-1}(\mathbf{x})p^{2n_k-1}$$

$$= p^{n_k}\left[u_{n_k}(\mathbf{x}) + \cdots + u_{2n_k-1}(\mathbf{x})p^{n_k-1}\right]$$

Corresponding to formula (18) we have from Theorem 2.8 of chapter 2:

$$\mathrm{f}(u^{(k)} + \Delta u^{(k)}) = \mathrm{f}(u^{(k)}) + \mathrm{f}'(u^{(k)})\Delta u^{(k)} + \mathrm{g}(u^{(k)},\Delta u^{(k)})[\Delta u^{(k)}]^2$$

for some polynomial $\mathrm{g(u,\ w)}$. Noting from above that $\Delta u^{(k)}$ can be divided by p^{n_k} and using arguments similar to the linear case, we get the following formula which must be satisfied by the update $\Delta u^{(k)}$:

$$(22) \quad 0 = \phi_{p^{n_k}}\left(\frac{f(u^{(k)})}{p^{n_k}}\right) + \phi_{p^{n_k}}(f'(u^{(k)}))\frac{\Delta u^{(k)}}{p^{n_k}} \in \mathbf{Z}_{p^{n_k}}[x]$$

where $n_k = 2^{k-1}$. As before we have the result that for all $k \geq 1$,

$$f'(u^{(k)}) \equiv f'(u^{(1)}) \pmod{p}.$$

This time this result does not yield a simplification of the derivative since the modular homomorphism being applied to the derivative is now $\phi_{p^{n_k}}$ rather than ϕ_p. However we again wish to divide by the derivative term in (22) and the condition needed to guarantee that it is nonzero is precisely as in the linear update formula:

$$f'(u^{(1)}) \not\equiv 0 \pmod{p},$$

since from above this guarantees that $f'(u^{(k)}) \not\equiv 0 \pmod{p}$ whence $f'(u^{(k)}) \not\equiv 0 \pmod{p^{n_k}}$. (In other words, if the derivative term in (22) is nonzero for $k = 1$ then it is nonzero for all $k \geq 1$). Finally, solving for the update term in (22) yields the *quadratic update formula*

$$(23) \quad \frac{\Delta u^{(k)}}{p^{n_k}} = -\frac{\phi_{p^{n_k}}\left(\frac{f(u^{(k)})}{p^{n_k}}\right)}{\phi_{p^{n_k}}(f'(u^{(k)}))} \in \mathbf{Z}_{p^{n_k}}[x].$$

As in the case of the linear update formula, the division in (23) must be an exact division in the polynomial domain $\mathbf{Z}_{p^{n_k}}[x]$ if there exists a polynomial solution to the original problem.

Theorem 6.1 formally proves the quadratic convergence property of the p-adic Newton's iteration (21) with the quadratic update formula (23). In most practical problems requiring a p-adic Newton's iteration, the linear iteration is used rather than the quadratic iteration. Note that the quadratic iteration would require fewer iteration steps but the cost of each iteration step beyond the first is significantly higher than in the linear iteration because the domain $\mathbf{Z}_{p^{n_k}}[x]$ in which the update formula (23) must be computed becomes larger as k increases. Moreover, the derivative appearing in the divisor in (23) must be recomputed at each iteration step while the divisor in the linear update formula (20) is fixed for all iteration steps. For these reasons the linear iteration is often preferable to the quadratic iteration in terms of overall efficiency.

Theorem 6.1.

Let $f(u) \in \mathbf{Z}[x][u]$ be such that the polynomial equation $f(u) = 0$ has a polynomial solution $\tilde{u} = u(x) \in \mathbf{Z}[x]$. Let $u_0(x) \in \mathbf{Z}_p[x]$ be a first-order p-adic approximation to the solution $\tilde{u}$ so that

$$f(u_0(x)) \equiv 0 \pmod{p}.$$

Further suppose that $u_0(x)$ satisfies

$$f'(u_0(x)) \not\equiv 0 \pmod{p}.$$

Then the sequence of iterates defined by

$$u^{(1)} = u_0(x);$$
$$u^{(k+1)} = u^{(k)} + \Delta u^{(k)}, k = 1,2,3, \cdots$$

where $\Delta u^{(k)}$ is defined by the quadratic update formula (23), is such that $u^{(k+1)}$ is an order 2^k p-adic approximation to the solution $\tilde{u}$.

Proof: The proof is by induction on k. The basis holds trivially: $u^{(1)}$ is an order 1 p-adic approximation to $\tilde{u}$.

For the induction step, assume for $k \geq 1$ that $u^{(k)}$ is an order $n_k = 2^{k-1}$ p-adic approximation to $\tilde{u}$. This means that

$$\tilde{u} \equiv u^{(k)} \pmod{p^{n_k}}$$

or, defining the error $e^{(k)} = \tilde{u} - u^{(k)}$ we have

$$e^{(k)} \equiv 0 \pmod{p^{n_k}}.$$

Applying Theorem 2.8 of chapter 2 yields

$$\mathrm{f}(u^{(k)} + e^{(k)}) = \mathrm{f}(u^{(k)}) + \mathrm{f}'(u^{(k)})e^{(k)} + \mathrm{g}(u^{(k)},e^{(k)})[e^{(k)}]^2$$

for some polynomial g(u, w). Now $u^{(k)} + e^{(k)} = \tilde{u}$ so the left hand side becomes zero and, since $\mathrm{f}(u^{(k)})$ and $e^{(k)}$ are multiples of p^{n_k}, we have

$$0 = \frac{\mathrm{f}(u^{(k)})}{p^{n_k}} + \mathrm{f}'(u^{(k)})\frac{e^{(k)}}{p^{n_k}} + \mathrm{g}(u^{(k)},e^{(k)})\frac{e^{(k)}}{p^{n_k}}e^{(k)}.$$

Applying the modular homomorphism $\phi_{p^{n_k}}$ then yields

$$0 = \phi_{p^{n_k}}\left(\frac{\mathrm{f}(u^{(k)})}{p^{n_k}}\right) + \phi_{p^{n_k}}(\mathrm{f}'(u^{(k)}))\phi_{p^{n_k}}\left(\frac{e^{(k)}}{p^{n_k}}\right)$$

where we note that the last term vanishes because $\phi_{p^{n_k}}(e^{(k)}) = 0$. Now applying the definition of the quadratic update formula (23), this becomes

$$0 = -\frac{\Delta u^{(k)}}{p^{n_k}} + \phi_{p^{n_k}}\left(\frac{e^{(k)}}{p^{n_k}}\right) \in \mathbf{Z}_{p^{n_k}}[\mathbf{x}].$$

Hence

$$\frac{e^{(k)} - \Delta u^{(k)}}{p^{n_k}} \equiv 0 \left(\mathrm{mod}\, p^{n_k}\right)$$

or

$$e^{(k)} - \Delta u^{(k)} \equiv 0 \left(\mathrm{mod}\, p^{2n_k}\right).$$

Finally, since $e^{(k)} = \tilde{u} - u^{(k)}$ we have

$$\tilde{u} - (u^{(k)} + \Delta u^{(k)}) \equiv 0 \left(\mathrm{mod}\, p^{2n_k}\right)$$

or

$$\tilde{u} \equiv u^{(k+1)} \left(\mathrm{mod}\, p^{2n_k}\right)$$

which proves that $u^{(k+1)}$ is an order $2n_k = 2^k$ p-adic approximation to $\tilde{u}$. □

Ideal-adic Iteration

We now turn to the problem of inverting a multivariate evaluation homomorphism

$$\phi_{\mathrm{I}} : \mathbf{Z}_p[x_1, \ldots, x_v] \rightarrow \mathbf{Z}_p[x_1]$$

with kernel $\mathrm{I} = \langle x_2 - \alpha_2, \ldots, x_v - \alpha_v \rangle$ for some specified values $\alpha_i \in \mathbf{Z}_p$ $(2 \leq i \leq v)$. The inversion process will be accomplished by an ideal-adic version of Newton's iteration. We are given the order 1 ideal-adic approximation

$$u^{(1)} = \phi_{\mathrm{I}}(\tilde{u}) \in \mathbf{Z}_p[x_1] = \mathbf{Z}_p[\mathbf{x}] / \mathrm{I}$$

to the solution $\tilde{u} \in \mathbf{Z}_p[\mathbf{x}]$ and, as before, let us assume for now that the additional information about the solution $\tilde{u}$ is that it satisfies a polynomial equation

$$f(u) = 0$$

where $f(u) \in \mathbf{Z}_p[\mathbf{x}][u]$. We wish to define an iteration formula such that at step k the order k ideal-adic approximation $u^{(k)}$ is updated to the order $k+1$ ideal-adic approximation $u^{(k+1)}$ by the addition of the correction term $\Delta u^{(k)} \in \mathrm{I}^k$. By Theorem 2.8 of chapter 2 applied to the polynomial $f(u) \in \mathbf{Z}_p[\mathbf{x}][u]$, we have the following 'Taylor series expansion':

$$\textbf{(24)} \quad f(u^{(k)} + \Delta u^{(k)}) = f(u^{(k)}) + f'(u^{(k)})\Delta u^{(k)} + g(u^{(k)}, \Delta u^{(k)})[\Delta u^{(k)}]^2$$

for some polynomial $g(u, w)$. Now if $u^{(k)} + \Delta u^{(k)}$ is to be the order $k + 1$ ideal-adic approximation $u^{(k+1)}$ then using property (19) we deduce that

$$f(u^{(k)} + \Delta u^{(k)}) \in \mathrm{I}^{k+1}.$$

Also since $\Delta u^{(k)} \in \mathrm{I}^k$ it follows that

$$[\Delta u^{(k)}]^2 \in \mathrm{I}^{2k}.$$

Hence applying the homomorphism $\phi_{\mathrm{I}^{k+1}}$ to (24) yields the equation

$$\textbf{(25)} \quad 0 = \phi_{\mathrm{I}^{k+1}}(f(u^{(k)})) + \phi_{\mathrm{I}^{k+1}}(f'(u^{(k)}))\Delta u^{(k)} \in \mathbf{Z}_p[\mathbf{x}] \,/\, \mathrm{I}^{k+1}$$

which must be satisfied by the correction term $\Delta u^{(k)} \in \mathrm{I}^k$.

Consider iteration step $k=1$. In this case the correction term $\Delta u^{(1)} \in \mathrm{I}$ takes the form

$$\textbf{(26)} \quad \Delta u^{(1)} = \sum_{i=2}^{v} u_i(x_1)(x_i - \alpha_i)$$

where the coefficients $u_i(x_1) \in \mathbf{Z}_p[x_1]$ are to be determined. Using property (19) and the fact that $u^{(1)} \equiv \tilde{u} \pmod{\mathrm{I}}$, we deduce that $f(u^{(1)}) \in \mathrm{I}$ and therefore we can write

$$\textbf{(27)} \quad f(u^{(1)}) = \sum_{i=2}^{v} c_i(x_i - \alpha_i)$$

for some coefficients $c_i \in \mathbf{Z}_p[\mathbf{x}]$, $2 \le i \le v$. Now the homomorphism being applied in equation (25) is ϕ_{I^2} when $k = 1$ and since the effect of ϕ_{I^2} is to drop the ideal-adic terms of total degree equal to or exceeding 2, it follows from (27) that

$$\phi_{\mathrm{I}^2}(f(u^{(1)})) = \sum_{i=2}^{v} c_i(x_1)(x_i - \alpha_i)$$

where the coefficients $c_i(x_1) \in \mathbf{Z}_p[x_1]$ are defined from the coefficients $c_i \in \mathbf{Z}_p[\mathbf{x}]$ appearing in (27) by

$$c_i(x_1) = \phi_{\mathrm{I}}(c_i), \; 2 \le i \le v.$$

Equation (25) is now

$$\textbf{(28)} \quad 0 = \sum_{i=2}^{v} c_i(x_1)(x_i - \alpha_i) + \phi_{\mathrm{I}^2}(f'(u^{(1)}))\left(\sum_{i=2}^{v} u_i(x_1)(x_i - \alpha_i)\right) \in \mathbf{Z}_p[\mathbf{x}] \,/\, \mathrm{I}^2.$$

Now the ideal-adic representation of $\phi_{\mathrm{I}^2}(f'(u^{(1)}))$ can be written in the form

$$\phi_{\mathrm{I}^2}(f'(u^{(1)})) = \phi_{\mathrm{I}}(f'(u^{(1)})) + \sum_{i=2}^{v} d_i(x_1)(x_i - \alpha_i)$$

for some coefficients $d_i(x_1) \in \mathbf{Z}_p[x_1]$, $2 \le i \le v$. Putting this form into equation (28) yields

$$\textbf{(29)} \quad 0 = \sum_{i=2}^{v} c_i(x_1)(x_i - \alpha_i) + \phi_{\mathrm{I}}(f'(u^{(1)}))\left(\sum_{i=2}^{v} u_i(x_1)(x_i - \alpha_i)\right) \in \mathbf{Z}_p[\mathbf{x}] \,/\, \mathrm{I}^2$$

where we have noted that

$$\left[\sum_{i=2}^{v} d_i(x_1)(x_i - \alpha_i)\right]\left[\sum_{i=2}^{v} u_i(x_1)(x_i - \alpha_i)\right] \in I^2.$$

Equating coefficients on the left and right in equation (29) yields finally

$$\text{(30)} \quad u_i(x_1) = -\frac{c_i(x_1)}{\phi_I(f'(u^{(1)}))} \in \mathbf{Z}_p[x_1],\ 2 \le i \le v.$$

Equation (30) is the desired update formula which defines the correction term (26) and the division appearing in (30) must be an exact division in the univariate polynomial domain $\mathbf{Z}_p[x_1]$ if the given equation $f(u) = 0$ has a polynomial solution. Note that the coefficients $c_i(x_1)$ appearing in (30) are simply the coefficients of the linear terms in the ideal-adic representation of $f(u^{(1)})$.

Turning now to the general iteration step, the k-th correction term $\Delta u^{(k)} \in I^k$ is the term of total degree k in the ideal-adic representation of the solution $\tilde{u} = u(x_1, \ldots, x_v)$ and its general form consists of k nested summations as follows:

$$\text{(31)} \quad \Delta u^{(k)} = \sum_{i_1=2}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} u_{\mathbf{i}}(x_1)\prod_{j=1}^{k}(x_{i_j} - \alpha_{i_j})$$

where the subscript $\mathbf{i}$ denotes the vector of indices $\mathbf{i} = (i_1, \ldots, i_k)$. The coefficients $u_{\mathbf{i}}(x_1) \in \mathbf{Z}_p[x_1]$ are to be determined. We are given the order k ideal-adic approximation $u^{(k)}$ and the correction term $\Delta u^{(k)}$ must satisfy equation (25). As before, we deduce that $f(u^{(k)}) \in I^k$ from which it follows that

$$\phi_{I^{k+1}}(f(u^{(k)})) = \sum_{i_1=2}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} c_{\mathbf{i}}(x_1)\prod_{j=1}^{k}(x_{i_j} - \alpha_{i_j})$$

for some coefficients $c_{\mathbf{i}}(x_1) \in \mathbf{Z}_p[x_1]$. Also, the term $\phi_{I^{k+1}}(f'(u^{(k)}))$ in equation (25) can be replaced by $\phi_I(f'(u^{(k)}))$ because just as in the case $k = 1$, the terms of order greater than 1 in the ideal-adic representation of $\phi_{I^{k+1}}(f'(u^{(k)}))$ disappear when multiplied by $\Delta u^{(k)} \in I^k$ (since the multiplication is in the domain $\mathbf{Z}_p[\mathbf{x}] / I^{k+1}$). But for all $k \ge 1$, $u^{(k)} \equiv u^{(1)} \pmod I$ which implies by property (19) that $f'(u^{(k)}) \equiv f'(u^{(1)}) \pmod I$; i.e.

$$\phi_I(f'(u^{(k)})) = \phi_I(f'(u^{(1)})) \text{ for all } k \ge 1.$$

Equation (25) therefore becomes

$$\text{(32)} \quad 0 = \sum_{i_1=2}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} c_{\mathbf{i}}(x_1)\prod_{j=1}^{k}(x_{i_j} - \alpha_{i_j}) + \phi_I(f'(u^{(1)}))\left[\sum_{i_1=2}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} u_{\mathbf{i}}(x_1)\prod_{j=1}^{k}(x_{i_j} - \alpha_{i_j})\right] \in \mathbf{Z}_p[\mathbf{x}] / I^{k+1}.$$

Finally, if the given first-order approximation $u^{(1)}$ satisfies the condition

$$\text{(33)} \quad f'(u^{(1)}) \not\equiv 0 \pmod I$$

then by equating coefficients on the left and right in equation (32) we get the *linear ideal-adic Newton's iteration:*

$$\text{(34)} \quad u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$$

where $\Delta u^{(k)}$ is the correction term (31) with coefficients defined by

$$\text{(35)} \quad u_{\mathbf{i}}(x_1) = -\frac{c_{\mathbf{i}}(x_1)}{\phi_I(f'(u^{(1)}))} \in \mathbf{Z}_p[x_1].$$

Once again, the division appearing in (35) must be an exact division in the univariate polynomial domain $\mathbf{Z}_p[x_1]$ if the given equation $f(u) = 0$ has a polynomial solution. Note that the coefficients $c_i(x_1)$ appearing in (35) are simply the coefficients of the terms of total degree k in the ideal-adic representation of $f(u^{(k)})$ and note further that $f(u^{(k)})$ has no terms of total degree less than k (with respect to I).

The linear ideal-adic Newton's iteration (34)-(35) proceeds by computing in iteration step k *all* ideal-adic terms in the solution $\tilde{u}$ which have total degree k (with respect to I). It is possible to define a *quadratic* ideal-adic Newton's iteration just as in the p-adic case. Such an iteration would produce an order 2^k ideal-adic approximation $u^{(k+1)}$ in iteration step k. In other words, the quadratic iteration would compute in iteration step k all ideal-adic terms in the solution $\tilde{u}$ which have total degrees $2^{k-1}, 2^{k-1}+1, \ldots, 2^k - 1$. However as was noted in the p-adic case, the quadratic iteration entails a cost per iteration which is so much higher than the linear iteration that in terms of overall efficiency the linear iteration has been found to been generally superior.

Example 6.4.

Consider the problem of determining a polynomial $u(x,y,z) \in \mathbf{Z}_5[x,y,z]$ which is a square root of the polynomial

$$a(x,y,z) = x^4 + x^3y^2 - x^2y^4 + x^2yz + 2x^2z - 2x^2 - 2xy^3z + xy^2z - xy^2$$
$$- y^2z^2 + yz^2 - yz + z^2 - 2z + 1 \in \mathbf{Z}_5[x,y,z]$$

(assuming that $a(x,y,z)$ is a perfect square). Then $u(x,y,z)$ can be expressed as the solution of the polynomial equation

$$f(u) = a(x,y,z) - u^2 = 0.$$

Choosing the ideal $I = \langle y,z \rangle$ (i.e. choosing the evaluation points $y = 0$ and $z = 0$), the first-order ideal-adic approximation $u^{(1)} = u(x,0,0) \in \mathbf{Z}_5[x]$ must be a square root of $a(x,0,0)$ in $\mathbf{Z}_5[x]$. Now

$$a(x,0,0) = x^4 - 2x^2 + 1$$

which clearly has the square root

$$u^{(1)} = u(x,0,0) = x^2 - 1 \in \mathbf{Z}_5[x].$$

Now to apply the linear ideal-adic Newton's iteration, first note that

$$\phi_I(f'(u^{(1)})) = \phi_I(-2u^{(1)}) = -2x^2 + 2.$$

It is convenient to express $a(x,y,z)$ in its ideal-adic representation with respect to I, which is:

$$a(x,y,z) = [(x^4 - 2x^2 + 1)] + [(2x^2 - 2)z] + [(x^3 - x)y^2 + (x^2 - 1)yz + (1)z^2]$$
$$+ [(x)y^2z + (1)yz^2] + [(-x^2)y^4 + (-2x)y^3z + (-1)y^2z^2].$$

Now

$$\phi_{I^2}(f(u^{(1)})) = \phi_{I^2}(a(x,y,z) - (x^2 - 1)^2) = (2x^2 - 2)z \in \mathbf{Z}_5[x,y,z] / I^2.$$

The first correction term is

$$\Delta u^{(1)} = u_2(x)y + u_3(x)z$$

where $u_2(x) = 0$ (because the corresponding term in $\phi_{I^2}(f(u^{(1)}))$ is zero) and where

$$u_3(x) = -\frac{c_3(x)}{(-2x^2 + 2)} = -\frac{(2x^2 - 2)}{(-2x^2 + 2)} = 1 \in \mathbf{Z}_5[x].$$

Hence

$$u^{(2)} = u^{(1)} + \Delta u^{(1)} = (x^2 - 1) + z \in \mathbf{Z}_5[x,y,z] / I^2.$$

For the next iteration, we have

$$\phi_{I^3}(f(u^{(2)})) = \phi_{I^3}(a(x,y,z) - [(x^2-1) + z]^2) = (x^3 - x)y^2 + (x^2 - 1)yz \in \mathbf{Z}_5[x,y,z] / I^3.$$

The new correction term is

$$\Delta u^{(2)} = u_{22}(x)y^2 + u_{23}(x)yz + u_{33}(x)z^2$$

where $u_{33}(x) = 0$ (because the corresponding term in $\phi_{I^3}(f(u^{(2)}))$ is zero) and where

$$u_{22}(x) = -\frac{c_{22}(x)}{(-2x^2+2)} = -\frac{(x^3-x)}{(-2x^2+2)} = -2x \in \mathbf{Z}_5[x];$$

$$u_{23}(x) = -\frac{c_{23}(x)}{(-2x^2+2)} = -\frac{(x^2-1)}{(-2x^2+2)} = -2 \in \mathbf{Z}_5[x].$$

Hence

$$u^{(3)} = u^{(2)} + \Delta u^{(2)} = (x^2 - 1) + z + (-2x)y^2 + (-2)yz \in \mathbf{Z}_5[x,y,z] / I^3.$$

If we proceed to the next iteration we find that $f(u^{(3)}) = 0$ (in the domain $\mathbf{Z}_5[x,y,z]$) so we are finished. The desired square root of $a(x,y,z)$ is therefore

$$u(x,y,z) = u^{(3)} = x^2 - 2xy^2 - 2yz + z - 1 \in \mathbf{Z}_5[x,y,z]. \quad \square$$

A Homomorphism Diagram

Finally in this section, Figure 6.1 shows a homomorphism diagram for the case of solving a multivariate polynomial problem using the *p*-adic and ideal-adic Newton's iterations. This diagram should be compared with the diagram of Figure 5.1 where many image problems had to be constructed and solved rather than just one image problem. Note that in order to apply Newton's iteration it is assumed that the desired polynomial can be expressed as a solution of a polynomial equation f(u) = 0.

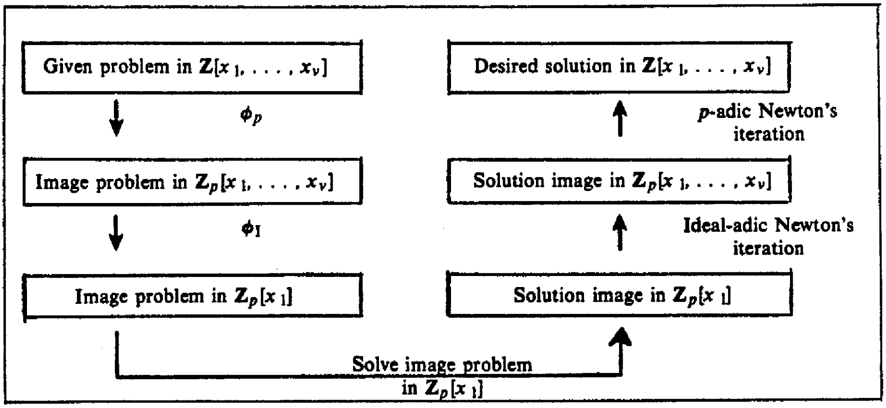
Given problem in $\mathbf{Z}[x_1, \ldots, x_\nu]$
ϕ_p
Image problem in $\mathbf{Z}_p[x_1, \ldots, x_\nu]$
ϕ_I
Image problem in $\mathbf{Z}_p[x_1]$
Desired solution in $\mathbf{Z}[x_1, \ldots, x_\nu]$
p-adic Newton's iteration
Solution image in $\mathbf{Z}_p[x_1, \ldots, x_\nu]$
Ideal-adic Newton's iteration
Solution image in $\mathbf{Z}_p[x_1]$
Solve image problem in $\mathbf{Z}_p[x_1]$

Figure 6.1: Homomorphism diagram for p-adic and ideal-adic Newton's iterations.

6.3. HENSEL'S LEMMA

Bivariate Newton's Iteration

In the preceding discussion of Newton's iteration for lifting an image polynomial $\phi_{<I,p>}(u) \in \mathbf{Z}_p[x_1]$ up to a desired polynomial $u \in \mathbf{Z}[x_1, \ldots, x_v]$ – i.e. for inverting a composite homomorphism

(36) $\quad \phi_{<I,p>} : \mathbf{Z}[x_1, \ldots, x_v] \to \mathbf{Z}_p[x_1]$,

it was assumed that the polynomial u could be expressed as the solution of a polynomial equation

(37) $\quad f(u) = 0$

for some $f(u) \in \mathbf{Z}[x_1, \ldots, x_v][u]$. However the most common applications of Newton's iteration for such a lifting process involve problems which cannot generally be expressed in the form (37), but rather can be expressed in the form

(38) $\quad F(u,w) = 0$

for some bivariate polynomial $F(u,w) \in \mathbf{Z}[x_1, \ldots, x_v][u,w]$. An equation such as (38) will have a pair of solutions u and w so we will in fact be lifting two polynomials, not just one.

The fundamental problem which can be expressed in the form (38) is the polynomial factorization problem. Suppose we wish to find factors in the domain $\mathbf{Z}[x_1, \ldots, x_v]$ of a polynomial $a(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$. By applying a composite homomorphism of the form (36), the factorization problem is reduced to a problem of factoring a univariate polynomial over the field $\mathbf{Z}_p$ (which as we see in chapter 7 is a comparatively simple problem). Let $a_0(x_1)$ denote the image of $a(x_1, \ldots, x_v)$ in $\mathbf{Z}_p[x_1]$ and suppose we discover that $u_0(x_1)$ is a factor of $a_0(x_1)$ in the domain $\mathbf{Z}_p[x_1]$. Then we have the following relationship in the domain $\mathbf{Z}_p[x_1]$:

$$a_0(x_1) = u_0(x_1)w_0(x_1) \text{ where } w_0(x_1) = \frac{a_0(x_1)}{u_0(x_1)} \in \mathbf{Z}_p[x_1].$$

We therefore pose the problem of finding multivariate polynomials $u(x_1, \ldots, x_v)$, $w(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$ which satisfy the bivariate polynomial equation

(39) $\quad F(u,w) = a(x_1, \ldots, x_v) - uw = 0$

such that

$$(40) \quad \begin{cases} u(x_1, \ldots, x_v) \equiv u_0(x_1) \pmod{<I,p>}; \\ w(x_1, \ldots, x_v) \equiv w_0(x_1) \pmod{<I,p>}. \end{cases}$$

In other words, we wish to lift the factors $u_0(x_1), w_0(x_1) \in \mathbf{Z}_p[x_1]$ to factors $u(x_1, \ldots, x_v), w(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$ by applying a form of Newton's iteration to the nonlinear equation (39). (Note that this process could be applied recursively to further factor the polynomials $u(x_1, \ldots, x_v)$ and $w(x_1, \ldots, x_v)$ in order to ultimately obtain the complete factorization of $a(x_1, \ldots, x_v)$ in the domain $\mathbf{Z}[x_1, \ldots, x_v]$). Sufficient conditions for such a lifting process to be possible will be determined shortly. A detailed discussion of the polynomial factorization problem is given in chapter 7.

Another problem which can be posed in the form (39) is the problem of computing the GCD of multivariate polynomials $a(x_1, \ldots, x_v), b(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$. Applying a composite homomorphism of the form (36) the problem is reduced to computing $\mathrm{GCD}(a_0(x_1), b_0(x_1))$ in the Euclidean domain $\mathbf{Z}_p[x_1]$, which can be easily accomplished by the basic Euclidean algorithm (Algorithm 2.1). Then if $u_0(x_1) = \mathrm{GCD}(a_0(x_1), b_0(x_1))$, we define the *cofactor* $w_0(x_1) = \dfrac{a_0(x_1)}{u_0(x_1)}$ and pose the problem of lifting the image polynomials $u_0(x_1), w_0(x_1) \in \mathbf{Z}_p[x_1]$ to multivariate polynomials

$$u(x_1, \ldots, x_v), w(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$$

which satisfy (39)-(40). (Note that the polynomial $b(x_1, \ldots, x_v)$ could as well play the role of $a(x_1, \ldots, x_v)$ in this lifting process). The problem of computing the GCD of polynomials by this method (and other methods) is discussed in more detail in chapter 7).

In this section we discuss how, and under what conditions, Newton's iteration can be applied to solve the problem (39)-(40). Noting that (39) is a single nonlinear equation in two unknowns, we would expect from general mathematical principles that it would not have a unique solution without imposing additional conditions. Rather than imposing the additional conditions explicitly as a second equation of the form $G(u,w) = 0$, the additional conditions will appear more indirectly in the following development.

The general form of Newton's iteration for the bivariate polynomial equation

$$F(u,w) = 0$$

can be determined by applying Theorem 2.9. Suppose that we have a pair of approximations $u^{(k)}, w^{(k)}$ to the solution pair $\bar{u}, \bar{w}$ and that we wish to compute a pair of correction terms $\Delta u^{(k)}, \Delta w^{(k)}$. Theorem 2.9 yields the equation

$$F(u^{(k)} + \Delta u^{(k)}, w^{(k)} + \Delta w^{(k)}) = F(u^{(k)}, w^{(k)}) + F_u(u^{(k)}, w^{(k)})\Delta u^{(k)} + F_w(u^{(k)}, w^{(k)})\Delta w^{(k)} + E$$

where the term E involves higher-order expressions with respect to $\Delta u^{(k)}, \Delta w^{(k)}$. By arguments which can be formalized as before (or loosely speaking, setting the left hand side to zero and ignoring the higher-order term E), we find that the correction terms should be chosen to satisfy the following equation (modulo some ideal):

$$(41) \quad F_u(u^{(k)}, w^{(k)})\Delta u^{(k)} + F_w(u^{(k)}, w^{(k)})\Delta w^{(k)} = -F(u^{(k)}, w^{(k)}).$$

Thus we see that the basic computation to be performed in applying a step of Newton's iteration will be to solve the polynomial diophantine equation (41) which takes the form

$$A^{(k)}\Delta u^{(k)} + B^{(k)}\Delta w^{(k)} = C^{(k)}$$

where $A^{(k)}, B^{(k)}, C^{(k)}$ are given polynomials and $\Delta u^{(k)}, \Delta w^{(k)}$ are the unknown polynomials to be determined. Equation (41) will in general have either no solution or else a whole family of solutions. However Theorem 2.6 of chapter 2 shows that under certain conditions the polynomial diophantine equation (41) has a unique solution.

From now on, we will specialize the development of Newton's iteration to the particular bivariate polynomial equation (39). As we have seen, the problem of polynomial factorization and also the problem of polynomial GCD computation can be posed in the particular form (39). Other problems may lead to different bivariate polynomial equations $F(u,w) = 0$ but the validity of Newton's iteration will depend on the particular problem. This is because of the need to introduce additional conditions which ensure the existence and uniqueness of a solution to the polynomial diophantine equation (41) which must be solved in each step of Newton's iteration.

Hensel's Lemma

Let us consider the univariate case of solving the problem (39)-(40). Thus we are given a polynomial $a(x) \in \mathbf{Z}[x]$ and a pair of factors $u_0(x), w_0(x) \in \mathbb{Z}_p[x]$ such that

$$a(x) \equiv u_0(x)w_0(x) \pmod{p}$$

and we wish to lift the factors $u_0(x), w_0(x)$ from the image domain $\mathbf{Z}_p[x]$ up to a pair of factors $\mathrm{u}(x), \mathrm{w}(x) \in \mathbf{Z}[x]$. In other words, we wish to invert the modular homomorphism

$$\phi_p : \mathbf{Z}[x] \to \mathbf{Z}_p[x]$$

by applying Newton's iteration to compute the solution $\bar{\mathrm{u}} = \mathrm{u}(x), \bar{\mathrm{w}} = \mathrm{w}(x)$ in $\mathbf{Z}[x]$ of the nonlinear equation

$$(42) \quad \mathrm{F}(\mathrm{u}, \mathrm{w}) = \mathrm{a}(x) - \mathrm{uw} = 0$$

such that

$$(43) \quad \begin{cases} \mathrm{u}(x) \equiv \mathrm{u}_0(x) \pmod{p}; \\ \mathrm{w}(x) \equiv \mathrm{w}_0(x) \pmod{p}. \end{cases}$$

Writing the solution polynomials $\bar{\mathrm{u}}$ and $\bar{\mathrm{w}}$ in their polynomial p-adic representations

$$(44) \quad \begin{cases} \bar{\mathrm{u}} = \mathrm{u}_0(x) + \mathrm{u}_1(x)p + \cdots + \mathrm{u}_n(x)p^n; \\ \bar{\mathrm{w}} = \mathrm{w}_0(x) + \mathrm{w}_1(x)p + \cdots + \mathrm{w}_n(x)p^n \end{cases}$$

(where n must be large enough so that $\frac{1}{2}p^{n+1}$ bounds the magnitudes of all integer coefficients appearing in $\mathrm{a}(x)$ and its factors $\bar{\mathrm{u}}$ and $\bar{\mathrm{w}}$), we wish to determine the polynomial p-adic coefficients $\mathrm{u}_i(x), \mathrm{w}_i(x) \in \mathbf{Z}_p[x]$ for $i = 1, 2, \ldots, n$. Let $\mathrm{u}^{(k)}, \mathrm{w}^{(k)}$ denote the order k p-adic approximations to $\bar{\mathrm{u}}, \bar{\mathrm{w}}$ given by the first k terms in (44) and let $\Delta\mathrm{u}^{(k)} = \mathrm{u}_k(x)p^k, \Delta\mathrm{w}^{(k)} = \mathrm{w}_k(x)p^k$. Note that $\mathrm{u}^{(1)} = \mathrm{u}_0(x)$ and $\mathrm{w}^{(1)} = \mathrm{w}_0(x)$. We find that the correction terms must satisfy the polynomial diophantine equation (41) modulo p^{k+1}, which for the particular nonlinear equation (42) takes the form

$$-\mathrm{w}^{(k)}\Delta\mathrm{u}^{(k)} - \mathrm{u}^{(k)}\Delta\mathrm{w}^{(k)} \equiv -\left(\mathrm{a}(x) - \mathrm{u}^{(k)}\mathrm{w}^{(k)}\right) \pmod{p^{k+1}}.$$

Since $\mathrm{u}^{(k)}\mathrm{w}^{(k)}$ must be an order k p-adic approximation to $\mathrm{a}(x)$ we can divide through by p^k, and also removing the negative signs we get

$$\mathrm{w}^{(k)}\mathrm{u}_k(x) + \mathrm{u}^{(k)}\mathrm{w}_k(x) \equiv \frac{\mathrm{a}(x) - \mathrm{u}^{(k)}\mathrm{w}^{(k)}}{p^k} \pmod{p}.$$

Now we may apply the modular homomorphism ϕ_p to the left and right (because this is a congruence modulo p) and, noting that $\phi_p(\mathrm{w}^{(k)}) = \mathrm{w}_0(x)$ and $\phi_p(\mathrm{u}^{(k)}) = \mathrm{u}_0(x)$, we get the following polynomial diophantine equation to solve in the domain $\mathbf{Z}_p[x]$:

$$\mathrm{w}_0(x)\mathrm{u}_k(x) + \mathrm{u}_0(x)\mathrm{w}_k(x) = \phi_p\left(\frac{\mathrm{a}(x) - \mathrm{u}^{(k)}\mathrm{w}^{(k)}}{p^k}\right).$$

Now since $\mathbf{Z}_p[x]$ is a Euclidean domain (we choose p to be a prime integer), Theorem 2.6 shows that if $\mathrm{u}_0(x), \mathrm{w}_0(x) \in \mathbf{Z}_p[x]$ are relatively prime then we can find unique polynomials $\sigma(x), \tau(x) \in \mathbf{Z}_p[x]$ such that

$$\sigma(x)\mathrm{u}_0(x) + \tau(x)\mathrm{w}_0(x) = \phi_p\left(\frac{\mathrm{a}(x) - \mathrm{u}^{(k)}\mathrm{w}^{(k)}}{p^k}\right)$$

and

$$\deg[\sigma(x)] < \deg[\mathrm{w}_0(x)].$$

We then define $\mathrm{u}^{(k+1)} = \mathrm{u}^{(k)} + \tau(x)p^k, \mathrm{w}^{(k+1)} = \mathrm{w}^{(k)} + \sigma(x)p^k$ and we claim that these are order $k+1$ p-adic approximations to the solutions $\bar{\mathrm{u}}, \bar{\mathrm{w}}$ respectively.

The following theorem formally proves the validity of the above method. This theorem is a standard result in algebra known as Hensel's Lemma and it dates back to the early

1900's. The proof of Hensel's Lemma is a constructive proof which follows naturally from the above development and this process is referred to as the *Hensel construction.*

Theorem 6.2. *Hensel's Lemma.*

Let p be a prime in $\mathbf{Z}$ and let $\mathrm{a}(x) \in \mathbf{Z}[x]$ be a given polynomial over the integers. Let $\mathrm{u}^{(1)}(x)$, $\mathrm{w}^{(1)}(x) \in \mathbf{Z}_p[x]$ be two relatively prime polynomials over the field $\mathbf{Z}_p$ such that

$$\mathrm{a}(x) \equiv \mathrm{u}^{(1)}(x)\mathrm{w}^{(1)}(x) \pmod{p}.$$

Then for any integer $k \geq 1$ there exist polynomials $\mathrm{u}^{(k)}(x), \mathrm{w}^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$ such that

$$(45) \quad \mathrm{a}(x) \equiv \mathrm{u}^{(k)}(x)\mathrm{w}^{(k)}(x) \pmod{p^k}$$

and

$$(46) \quad \begin{cases} \mathrm{u}^{(k)}(x) \equiv \mathrm{u}^{(1)}(x) \pmod{p}; \\ \mathrm{w}^{(k)}(x) \equiv \mathrm{w}^{(1)}(x) \pmod{p}. \end{cases}$$

Proof: The proof is by induction on k. The case $k = 1$ is given. Assume for $k \geq 1$ that we have $\mathrm{u}^{(k)}(x)$, $\mathrm{w}^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$ satisfying (45) and (46). Define

$$(47) \quad c^{(k)}(x) = \phi_p\left(\frac{\mathrm{a}(x) - \mathrm{u}^{(k)}(x)\mathrm{w}^{(k)}(x)}{p^k}\right)$$

where all operations are performed in the domain $\mathbf{Z}_{p^{k+1}}[x]$ before applying ϕ_p. Since $\mathrm{u}^{(1)}(x), \mathrm{w}^{(1)}(x) \in \mathbf{Z}_p[x]$ are relatively prime, by Theorem 2.6 we can find unique polynomials $\sigma^{(k)}(x), \tau^{(k)}(x) \in \mathbf{Z}_p[x]$ such that

$$(48) \quad \sigma^{(k)}(x)\mathrm{u}^{(1)}(x) + \tau^{(k)}(x)\mathrm{w}^{(1)}(x) \equiv c^{(k)}(x) \pmod{p}$$

and

$$(49) \quad \deg[\sigma^{(k)}(x)] < \deg[\mathrm{w}^{(1)}(x)].$$

Then by defining

$$(50) \quad \begin{cases} \mathrm{u}^{(k+1)}(x) = \mathrm{u}^{(k)}(x) + \tau^{(k)}(x)p^k; \\ \mathrm{w}^{(k+1)}(x) = \mathrm{w}^{(k)}(x) + \sigma^{(k)}(x)p^k \end{cases}$$

we have by performing multiplication modulo p^{k+1}:

$$\begin{aligned} \mathrm{u}^{(k+1)}(x)\mathrm{w}^{(k+1)}(x) &\equiv \mathrm{u}^{(k)}(x)\mathrm{w}^{(k)}(x) + (\sigma^{(k)}(x)\mathrm{u}^{(1)}(x) + \tau^{(k)}(x)\mathrm{w}^{(1)}(x))p^k \pmod{p^{k+1}} \\ &\equiv \mathrm{u}^{(k)}(x)\mathrm{w}^{(k)}(x) + c^{(k)}(x)p^k \pmod{p^{k+1}}, \text{ by (48)} \\ &\equiv \mathrm{a}(x) \pmod{p^{k+1}}, \text{by (47)}. \end{aligned}$$

Thus (45) holds for $k+1$. Also, from (50) it is clear that

$$\mathrm{u}^{(k+1)}(x) \equiv \mathrm{u}^{(k)}(x) \pmod{p};$$
$$\mathrm{w}^{(k+1)}(x) \equiv \mathrm{w}^{(k)}(x) \pmod{p}$$

and therefore since (46) holds for k it also holds for $k+1$. □

Corollary to Theorem 6.2. *Uniqueness of the Hensel Construction.*

In Theorem 6.2, if the given polynomial $\mathrm{a}(x) \in \mathbf{Z}[x]$ is monic and correspondingly if the relatively prime factors $\mathrm{u}^{(1)}(x), \mathrm{w}^{(1)}(x) \in \mathbf{Z}_p[x]$ are chosen to be monic, then for any integer $k \geq 1$ conditions (45)-(46) uniquely determine the monic polynomial factors $\mathrm{u}^{(k)}(x), \mathrm{w}^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$.

Proof: The proof is by induction on k. For the case $k=1$, the given polynomials

$u^{(1)}(x), w^{(1)}(x)$ are clearly the unique monic polynomials in $\mathbf{Z}_p[x]$ which satisfy conditions (45)-(46). For the induction assumption, assume for some $k \geq 1$ that the uniqueness of the monic polynomials $u^{(k)}(x), w^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$ satisfying (45)-(46) has been determined. Then we must prove the uniqueness of the monic polynomials $u^{(k+1)}(x), w^{(k+1)}(x) \in \mathbf{Z}_{p^{k+1}}[x]$ satisfying the conditions

$$\text{(51)} \quad a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \pmod{p^{k+1}}$$

and

$$\text{(52)} \quad \begin{cases} u^{(k+1)}(x) \equiv u^{(1)}(x) \pmod{p}; \\ w^{(k+1)}(x) \equiv w^{(1)}(x) \pmod{p}. \end{cases}$$

Condition (51) implies, in particular, that

$$a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \pmod{p^k}$$

which together with (52) yields, by the induction assumption,

$$u^{(k+1)}(x) \equiv u^{(k)}(x) \pmod{p^k};$$
$$w^{(k+1)}(x) \equiv w^{(k)}(x) \pmod{p^k}.$$

We may therefore write

$$\text{(53)} \quad \begin{cases} u^{(k+1)}(x) = u^{(k)}(x) + \tau(x)p^k; \\ w^{(k+1)}(x) = w^{(k)}(x) + \sigma(x)p^k \end{cases}$$

for some polynomials $\sigma(x), \tau(x) \in \mathbf{Z}_p[x]$ and it remains to prove the uniqueness of $\sigma(x)$ and $\tau(x)$.

Since $a(x), u^{(1)}(x)$, and $w^{(1)}(x)$ are given to be monic, it follows that for any $k \geq 1$ the polynomials $\sigma(x)$ and $\tau(x)$ appearing in (53) must satisfy

$$\text{(54)} \quad \deg[\sigma(x)] < \deg[w^{(1)}(x)] \text{ and } \deg[\tau(x)] < \deg[u^{(1)}(x)]$$

(i.e. $u^{(k+1)}(x)$ and $w^{(k+1)}(x)$ must always have the same leading terms as $u^{(1)}(x)$ and $w^{(1)}(x)$, respectively). Now by multiplying out the two polynomials expressed in (53) and using (51), we get (performing the multiplication modulo p^{k+1}):

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) + (\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x))p^k \pmod{p^{k+1}}$$

which can be expressed in the form

$$\text{(55)} \quad \sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x) \equiv \frac{a(x) - u^{(k)}(x)w^{(k)}(x)}{p^k} \pmod{p}.$$

By theorem 2.6, the polynomials $\sigma(x), \tau(x) \in \mathbf{Z}_p[x]$ satisfying (55) and (54) are unique. □

6.4. THE UNIVARIATE HENSEL LIFTING ALGORITHM

Description of the Algorithm

The Hensel construction of Theorem 6.2 is based on a linear p-adic Newton's iteration. Zassenhaus [Zas69] was the first to propose the application of Hensel's Lemma to the problem of polynomial factorization over the integers and he proposed the use of a quadratic p-adic Newton's iteration. This quadratic iteration is usually referred to as the *Zassenhaus construction* and it computes a sequence of factors modulo p^{2^k}, for $k = 1, 2, 3, \cdots$. However as we noted in section 6.2, a quadratic iteration is not necessarily more efficient than a linear iteration because the added complexity of each iteration step in the quadratic iteration may outweigh the advantage of fewer iteration steps. For example, in each iteration step of the quadratic Zassenhaus construction one must solve a polynomial diophantine equation of the form

$$\textbf{(56)}\quad \sigma^{(k)}(x)\mathrm{u}^{(k)}(x) + \tau^{(k)}(x)\mathrm{w}^{(k)}(x) \equiv c^{(k)}(x) \pmod{p^{2^{k-1}}}$$

for $\sigma^{(k)}(x), \tau^{(k)}(x) \in \mathbf{Z}_{p^{2^{k-1}}}[x]$. The corresponding computation in the linear Hensel construction is to solve the same polynomial diophantine equation modulo p for $\sigma^{(k)}(x), \tau^{(k)}(x) \in \mathbf{Z}_p[x]$. The latter computation is simpler because it is performed in the smaller domain $\mathbf{Z}_p[x]$ and another level of efficiency arises because the $\mathrm{u}^{(k)}(x)$ and $\mathrm{w}^{(k)}(x)$ in (56) can be replaced by the fixed polynomials $\mathrm{u}^{(1)}(x)$ and $\mathrm{w}^{(1)}(x)$ in the linear Hensel case. A detailed comparison of these two p-adic constructions was carried out by Miola and Yun [M&Y74] and their analysis showed that the computational cost of the quadratic Zassenhaus construction is higher than that of the linear Hensel construction for achieving the same p-adic order of approximation. Therefore we will not present the details of the quadratic Zassenhaus construction.

The basic algorithm for lifting a factorization in $\mathbf{Z}_p[x]$ up to a factorization in $\mathbf{Z}[x]$ is presented as Algorithm 6.1. In the monic case Algorithm 6.1 corresponds precisely to the Hensel construction presented in the proof of Hensel's lemma, since by the Corollary to Theorem 6.2 the factors at each step of the lifting process are uniquely determined in the monic case. However in the non-monic case the nonuniqueness of the factors modulo p^k leads to the 'leading coefficient problem' to be discussed shortly, and as we shall see this accounts for the additional conditions and the additional operations appearing in Algorithm 6.1. For the moment, Algorithm 6.1 may be understood for the monic case if we simply ignore the stated conditions (other than the conditions appearing in Hensel's lemma), ignore step 1, ignore the 'replaceLc' operation in step 3 (using instead the initialization $\mathrm{u}(x) \leftarrow \mathrm{u}^{(1)}(x)$; $\mathrm{w}(x) \leftarrow \mathrm{w}^{(1)}(x)$), and note that no adjustment of $\mathrm{u}(x)$ and $\mathrm{w}(x)$ is required in step 5 for the monic case.

Algorithm 6.1. Univariate Hensel Lifting Algorithm.

```
begin
    comment INPUT:
        (1) A primitive polynomial $a(x) \in \mathbf{Z}[x]$.
        (2) A prime integer p which does not divide $Lc[a(x)]$.
        (3) Two relatively prime polynomials $u^{(1)}(x), w^{(1)}(x) \in \mathbf{Z}_p[x]$ such that
                $a(x) \equiv u^{(1)}(x)\, w^{(1)}(x) \pmod{p}$.
        (4) An integer B which bounds the magnitudes of all integer coefficients appearing
            in a(x) and in any of its possible factors with degrees not exceeding
                $\max\{deg[u^{(1)}(x)], deg[w^{(1)}(x)]\}$.
        (5) Optionally, an integer $\gamma \in \mathbf{Z}$ which is known to be a multiple of $Lc[u(x)]$, where
            u(x) (see OUTPUT below) is one of the factors of a(x) in Z[x] to be computed;

    comment OUTPUT:
        (1) If there exist polynomials $u(x), w(x) \in \mathbf{Z}[x]$ such that
                $a(x) = u(x)\, w(x) \in \mathbf{Z}[x]$
            and
                $\mathbf{n}[u(x)] \equiv \mathbf{n}[u^{(1)}(x)] \pmod{p}$, $\mathbf{n}[w(x)] \equiv \mathbf{n}[w^{(1)}(x)] \pmod{p}$
            where n denotes the normalization 'make the polynomial monic as an
            element of the domain $\mathbf{Z}_p[x]$', then u(x) and w(x) will be computed.
        (2) Otherwise, the nonexistence of such a factorization will be determined.;
```

Algorithm 6.1. (continued). Univariate Hensel Lifting Algorithm.

```
    comment 1. Define new polynomial and its modulo p factors;

    $\alpha \leftarrow \mathrm{Lc}[a(x)]$;
    if $\gamma$ is undefined then $\gamma \leftarrow \alpha$;
    $a(x) \leftarrow \gamma \times a(x)$;
    $u^{(1)}(x) \leftarrow \phi_p(\gamma \times n[u^{(1)}(x)])$;  $w^{(1)}(x) \leftarrow \phi_p(\alpha \times n[w^{(1)}(x)])$;

    comment 2. Apply extended Euclidean algorithm to $u^{(1)}(x)$, $w^{(1)}(x) \in Z_p[x]$;

    $s(x)$, $t(x)$ ← polynomials in $Z_p[x]$ computed by Algorithm 2.2 such that
                $s(x)\,u^{(1)}(x) + t(x)\,w^{(1)}(x) \equiv 1 \pmod p$;

    comment 3. Initialization for the iteration;

    $u(x) \leftarrow \mathrm{replaceLc}(u^{(1)}(x), \gamma)$;  $w(x) \leftarrow \mathrm{replaceLc}(w^{(1)}(x), \alpha)$;
    $e(x) \leftarrow a(x) - u(x) \times w(x)$;
    modulus ← p;

    comment 4. Iterate until either the factorization in Z[x] is obtained or else the
        bound on modulus is reached;

    while $e(x) \neq 0$ and modulus $< 2 \times B \times \gamma$ do
        begin
            comment 4.1. Solve in the domain $Z_p[x]$ the polynomial diophantine equation
                $\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x) \equiv c(x) \pmod p$
                where $c(x) = e(x)/$modulus;

            $c(x) \leftarrow e(x)/$modulus;
            $\bar{\sigma}(x) \leftarrow \phi_p(s(x) \times c(x))$;  $\bar{\tau}(x) \leftarrow \phi_p(t(x) \times c(x))$;
            $q(x)$, $r(x)$ ← polynomials in $Z_p[x]$ computed by Euclidean division such that
                $\bar{\sigma}(x) = w^{(1)}(x)\,q(x) + r(x) \in Z_p[x]$;
            $\sigma(x) \leftarrow r(x)$;  $\tau(x) \leftarrow \phi_p(\bar{\tau}(x) + q(x) \times u^{(1)}(x))$;

            comment 4.2. Update the factors and compute the error in the new factorization;
            $u(x) \leftarrow u(x) + \tau(x) \times$ modulus;
            $w(x) \leftarrow w(x) + \sigma(x) \times$ modulus;
            $e(x) \leftarrow a(x) - u(x) \times w(x)$;
            modulus ← modulus × p
        end;

    comment 5. Check termination status;
    if $e(x) = 0$ then
        begin comment Factorization obtained – remove contents;
            $\delta \leftarrow \mathrm{cont}[u(x)]$;
            $u(x) \leftarrow u(x)/\delta$;  $w(x) \leftarrow w(x)/(\gamma/\delta)$;
            comment Note that $a(x) \leftarrow a(x)/\gamma$ would restore $a(x)$ to its input value
        end
    else comment No such factorization exists;
        termination status is 'there exists no such factorization'

  end.
```

Example 6.5.

Consider the problem of factoring the following monic polynomial over the integers:

$$a(x) = x^3 + 10x^2 - 432x + 5040 \in \mathbf{Z}[x].$$

Choosing $p = 5$ and applying the modular homomorphism ϕ_5 to $a(x)$ yields

$$\phi_5(a(x)) = x^3 - 2x \in \mathbf{Z}_5[x].$$

The unique unit normal (i.e. monic) factorization in $\mathbf{Z}_5[x]$ of this polynomial is

$$\phi_5(a(x)) = x(x^2 - 2) \in \mathbf{Z}_5[x].$$

We therefore define

$$u^{(1)}(x) = x; \;\; w^{(1)}(x) = x^2 - 2$$

and since $u^{(1)}(x)$ and $w^{(1)}(x)$ are relatively prime in $\mathbf{Z}_5[x]$, the Hensel construction may be applied.

Applying Algorithm 6.1 in the form noted above for the monic case, we first apply in step 2 the extended Euclidean algorithm which yields

$$s(x) = -2x; \;\; t(x) = 2.$$

The initializations in step 3 yield

$$u(x) = x; \;\; w(x) = x^2 - 2;$$
$$e(x) = 10x^2 - 430x + 5040;$$
$$\mathit{modulus} = 5.$$

Step 4 then applies the Hensel construction precisely as outlined in the proof of Hensel's lemma. (For now we are ignoring the second termination condition of the **while**-loop). The sequence of values computed for $\sigma(x), \tau(x), u(x), w(x)$, and $e(x)$ in step 4 is as follows.

center; End of iteration no.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
0	–	–	x	$x^2 - 2$	$10x^2 - 430x + 5040$
1	$x - 1$	1	$x + 5$	$x^2 + 5x - 7$	$-450x + 5075$
2	$-x + 2$	1	$x + 30$	$x^2 - 20x + 43$	$125x + 3750$
3	1	0	$x + 30$	$x^2 - 20x + 168$	0

Note that at the end of each iteration step k, $e(x)$ is exactly divisible by $\mathit{modulus} = 5^{k+1}$ as required at the beginning of the next iteration. The iteration terminates with $u(x) = x + 30$ and $w(x) = x^2 - 20x + 168$; we therefore have the following factorization over the integers:

$$x^3 + 10x^2 - 432x + 5040 = (x + 30)(x^2 - 20x + 168). \quad \square$$

Example 6.6.

In this example we shall see that the Hensel construction may apply even when the given polynomial cannot be factored over the integers. Consider the monic polynomial

$$a(x) = x^4 + 1 \in \mathbf{Z}[x]$$

which is irreducible over the integers. Choosing $p = 5$ and applying the modular homomorphism ϕ_5 to $a(x)$ yields $\phi_5(a(x)) = x^4 + 1$. The unique unit normal factorization in $\mathbf{Z}_5[x]$ of this polynomial is

$$x^4 + 1 = (x^2 + 2)(x^2 - 2) \in \mathbf{Z}_5[x].$$

Since the polynomials $u^{(1)}(x) = x^2 + 2$ and $w^{(1)}(x) = x^2 - 2$ are relatively prime in $\mathbf{Z}_5[x]$, the Hensel construction may be applied. In this case we get an infinite sequence of factors

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) \pmod{p^k}$$

for $k = 1,2,3, \cdots$.

If we apply Algorithm 6.1 to this monic case as in Example 6.5, the result of step 2 is

$$s(x) = -1; \ t(x) = 1$$

and the initializations in step 3 yield

$$u(x) = x^2 + 2; \ w(x) = x^2 - 2;$$
$$e(x) = 5;$$
$$\textit{modulus} = 5.$$

If we allow the **while**-loop in step 4 to proceed through four iterations (again we are ignoring the second termination condition of the **while**-loop), the sequence of values computed for $\sigma(x), \tau(x), u(x), w(x)$, and $e(x)$ is as follows.

End of iteration no.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
0	–	–	$x^2 + 2$	$x^2 - 2$	5
1	-1	1	$x^2 + 7$	$x^2 - 7$	50
2	-2	2	$x^2 + 57$	$x^2 - 57$	3250
3	-1	1	$x^2 + 182$	$x^2 - 182$	33125
4	2	-2	$x^2 - 1068$	$x^2 + 1068$	1140625

These iterations could be continued indefinitely yielding an infinite sequence of factors satisfying Hensel's lemma. Note that at the end of iteration step k we always have

$$u(x)w(x) \equiv x^4 + 1 \pmod{5^{k+1}}$$

as claimed in Hensel's lemma. However we will never obtain quadratic factors $u(x), w(x) \in \mathbf{Z}[x]$ such that

$$u(x)w(x) = x^4 + 1 \in \mathbf{Z}[x]. \quad \square$$

The Leading Coefficient Problem

The Hensel construction provides a method for lifting a factorization modulo p up to a factorization modulo p^l for any integer $l \geq 1$. Example 6.6 shows that this construction does not necessarily lead to a factorization over the integers. However if the monic polynomial $a(x) \in \mathbf{Z}[x]$ has the modulo p factorization

$$a(x) \equiv u^{(1)}(x)w^{(1)}(x) \pmod{p}$$

where $u^{(1)}(x), w^{(1)}(x) \in \mathbf{Z}_p[x]$ are relatively prime monic polynomials and if there exists a factorization over the integers

$$(57) \quad a(x) = u(x)w(x) \in \mathbf{Z}[x]$$

such that

$$(58) \quad u(x) \equiv u^{(1)}(x) \pmod{p}; \ w(x) \equiv w^{(1)}(x) \pmod{p}$$

then the Hensel construction must obtain this factorization. Specifically, let l be large enough so that $\frac{1}{2}p^l > B$ where B is an integer which bounds the magnitudes of all integer coefficients appearing in $a(x)$ and in any of its possible factors with the particular degrees $\deg[u^{(1)}(x)]$ and $\deg[w^{(1)}(x)]$. (For a discussion of techniques for computing such a bound B see [Mig82]). Then the Hensel construction may be applied to compute monic polynomials $u^{(l)}(x), w^{(l)}(x) \in \mathbf{Z}_{p^l}[x]$ satisfying

$$(59)\quad a(x) \equiv u^{(l)}(x)w^{(l)}(x) \pmod{p^l}$$

and

$$(60)\quad \begin{cases} u^{(l)}(x) \equiv u^{(1)}(x) \pmod{p}; \\ w^{(l)}(x) \equiv w^{(1)}(x) \pmod{p} \end{cases}$$

and by the Corollary to Theorem 6.2 the factors $u^{(l)}(x), w^{(l)}(x) \in \mathbf{Z}_{p^l}[x]$ are uniquely determined by conditions (59)-(60). Now if there exists a factorization (57) satisfying (58) then another such monic factorization in $\mathbf{Z}_{p^l}[x]$ is provided by $\phi_{p^l}(u(x))$ and $\phi_{p^l}(w(x))$ and hence by uniqueness

$$u^{(l)}(x) = \phi_{p^l}(u(x));\ \ w^{(l)}(x) = \phi_{p^l}(w(x)).$$

But since $\frac{1}{2}p^l > B$ we have $\phi_{p^l}(u(x)) = u(x)$ and $\phi_{p^l}(w(x)) = w(x)$, which proves that $u^{(l)}(x)$ and $w^{(l)}(x)$ are the desired factors over the integers.

The above discussion shows that in the monic case, the Hensel construction may be halted when $\frac{1}{2}p^l > B$ at which point either $u^{(l)}(x)w^{(l)}(x) = a(x)$ over the integers or else there exists no factorization satisfying (57)-(58). The second termination condition of the **while**-loop in step 4 of Algorithm 6.1 is, in the monic case, precisely this condition. Note that since the bound B given to Algorithm 6.1 will invariably be very pessimistic, the first termination condition of the **while**-loop is required to avoid extra costly iterations after a factorization has been discovered.

In the non-monic case the situation is not quite so simple. The Hensel construction requires (in step 4.1 of Algorithm 6.1) the solution $\sigma(x), \tau(x) \in \mathbf{Z}_p[x]$ of the polynomial diophantine equation

$$(61)\quad \sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x) \equiv c(x) \pmod{p}.$$

The solution of this equation is not uniquely determined but uniqueness is (somewhat artificially) imposed by requiring that the solution satisfy

$$(62)\quad \deg[\sigma(x)] < \deg[w^{(1)}(x)]$$

(see Theorem 2.6). Noting that the update formulas (in step 4.2 of Algorithm 6.1) are then

$$u(x) \leftarrow u(x) + \tau(x)p;\ \ w(x) \leftarrow w(x) + \sigma(x)p,$$

it is clear that the degree constraint (62) implies that the leading coefficient of $w(x)$ is never updated. In the monic case, this is exactly what we want. Moreover, since

$$(63)\quad c(x) = \frac{a(x) - u(x)w(x)}{modulus}$$

it follows in the monic case that

$$\deg[c(x)] < \deg[a(x)] = \deg[u^{(1)}(x)] + \deg[w^{(1)}(x)]$$

and therefore the solution of (61) also satisfies, by Theorem 2.6,

$$\deg[\tau(x)] < \deg[u^{(1)}(x)];$$

it follows that the leading coefficient of $u(x)$ also is never updated in the monic case. Turning to the non-monic case, we must first assume that the chosen prime p does not divide the leading coefficient of $a(x)$. With this assumption we are assured that $u^{(1)}(x)$ and $w^{(1)}(x)$ have 'correct' degrees in the sense that $\deg[u^{(1)}(x)] + \deg[w^{(1)}(x)] = \deg[a(x)]$. Then we have from (63) that

$$\deg[c(x)] \le \deg[a(x)] = \deg[u^{(1)}(x)] + \deg[w^{(1)}(x)]$$

from which it follows exactly as in Theorem 2.6 that

$$(64)\quad \deg[\tau(x)] \le \deg[u^{(1)}(x)].$$

The degree constraint (64) allows the leading coefficient of $u(x)$ to be updated since, unlike (62), the inequality here is not strict inequality. Now at the end of each iteration step k, we have the relationship

$$a(x) = u(x)w(x) \pmod{p^{k+1}}$$

and therefore since (62) forces the leading coefficient of $w(x)$ to remain unchanged, all of the updating required by the leading coefficient of $a(x)$ is forced onto the leading coefficient of $u(x)$. (Note that any unit in the ring $\mathbf{Z}_{p^k}$ can be multiplied into one factor and its inverse multiplied into the other factor without changing the given relationship). This is referred to as the *leading coefficient problem* and it can cause the factors in the Hensel construction never to yield a factorization over the integers even when such a factorization over the integers exists. The following example will clarify this leading coefficient problem.

Example 6.7.

Consider the problem of factoring the following polynomial over the integers:

$$a(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbf{Z}[x].$$

In order to understand the leading coefficient problem which arises we start by presenting the correct answer; namely, the complete unit normal factorization of $a(x)$ over the integers is

$$a(x) = u(x)w(x) = (2x + 5)(6x^2 - 10x + 7) \in \mathbf{Z}[x].$$

Let us attempt to solve this factorization problem by the method used in Example 6.5. Choosing $p = 5$ and applying the modular homomorphism ϕ_5 to $a(x)$ yields

$$\phi_5(a(x)) = 2x^3 - x \in \mathbf{Z}_5[x].$$

The unique unit normal factorization in $\mathbf{Z}_5[x]$ of this polynomial is

$$\phi_5(a(x)) = 2(x)(x^2 + 2) \in \mathbf{Z}_5[x]$$

where 2 is a unit in $\mathbf{Z}_5[x]$. Now in order to choose the initial factors $u^{(1)}(x), w^{(1)}(x) \in \mathbf{Z}_5[x]$ to be lifted, we must attach the unit 2 either to the factor x or else to the factor $x^2 + 2$. This is precisely the problem of non-uniqueness which exists at each stage of the Hensel construction; at this initial stage we have

$$\phi_5(a(x)) = (2x)(x^2 + 2) = (x)(2x^2 - 1) \in \mathbf{Z}_5[x].$$

Since in this problem we are given the answer, we can see that the 'correct' images under ϕ_5 of $u(x), w(x) \in \mathbf{Z}[x]$ are

$$u^{(1)}(x) = 2x;\ \ w^{(1)}(x) = x^2 + 2.$$

However it is important to note that this 'correct' attachment of units to factors is irrelevant; the other choice for $u^{(1)}(x)$ and $w^{(1)}(x)$ in this example would be equally valid and would lead to the same 'leading coefficient problem' which will arise from the above choice.

The polynomials $u^{(1)}(x)$ and $w^{(1)}(x)$ defined above are relatively prime in $\mathbf{Z}_5[x]$ so the Hensel construction may be applied. Let us apply Algorithm 6.1 in the form used for the monic case of Example 6.5 (i.e. the unmodified Hensel construction as presented in the proof of Hensel's lemma). In step 2 of Algorithm 6.1, the extended Euclidean algorithm applied to $u^{(1)}(x)$ and $w^{(1)}(x)$ yields

$$s(x) = x;\ \ t(x) = -2.$$

The initializations in step 3 yield

$$u(x) = 2x;\ \ w(x) = x^2 + 2;$$

$$e(x) = 10x^3 + 10x^2 - 40x + 35;$$

$$\mathit{modulus} = 5.$$

If we allow the **while**-loop in step 4 to proceed through four iterations (again we are ignoring the second termination condition of the **while**-loop), the sequence of values computed for $\sigma(x), \tau(x), u(x), w(x)$, and $e(x)$ is as follows.

End of iteration no.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
0	–	–	$2x$	$x^2 + 2$	$10x^3 + 10x^2 - 40x + 35$
1	$-2x - 1$	$2x + 1$	$12x + 5$	$x^2 - 10x - 3$	$125x^2 + 50x + 50$
2	$2x + 1$	1	$12x + 30$	$x^2 + 40x + 22$	$-500x^2 - 1500x - 625$
3	$-2x - 1$	0	$12x + 30$	$x^2 - 210x - 103$	$2500x^2 + 7500x + 3125$
4	$2x + 1$	0	$12x + 30$	$x^2 + 1040x + 522$	$-12500x^2 - 37500x - 15625$

These iterations could be continued indefinitely yielding an infinite sequence of factors satisfying Hensel's lemma – i.e. at the end of iteration step k we always have

$$u(x)w(x) \equiv a(x) \pmod{5^{k+1}}.$$

However these factors will never satisfy the desired relationship

$$u(x)w(x) = a(x) \in \mathbf{Z}[x]$$

because $w(x)$ is always monic and there does not exist a monic quadratic factor of $a(x)$ over the integers. □

It is clear in the above example that the leading coefficient of $a(x)$ is completely forced onto the factor $u(x)$ since $w^{(1)}(x)$, and hence each updated $w(x)$, is monic. Noting the correct factorization of $a(x)$ over the integers, we see that the leading coefficient of $a(x)$ needs to be split in the form $12 = 2 \times 6$ with the factor 2 appearing as the leading coefficient of $u(x)$ and the factor 6 appearing as the leading coefficient of $w(x)$. Algorithm 6.1 contains additional statements which will force the leading coefficients to be correct and we now turn to an explanation of these additional operations.

6.5. SPECIAL TECHNIQUES FOR THE NON-MONIC CASE

Relationship of Computed Factors to True Factors

The first step towards solving the leading coefficient problem is the realization that the factors computed by the Hensel construction are 'almost' the correct factors over the integers, in the following sense. Let l be large enough so that $\frac{1}{2}p^l > B$ where B bounds the magnitudes of all integer coefficients appearing in $a(x)$ and in its factors. Then Theorem 6.4 below proves that the factors $u^{(l)}(x)$ and $w^{(l)}(x)$ computed by the Hensel construction such that

$$u^{(l)}(x)w^{(l)}(x) \equiv a(x) \pmod{p^l}$$

differ from the true factors over the integers only by a unit in the ring $\mathbf{Z}_{p^l}[x]$ (if an appropriate factorization over the integers exists). In Example 6.7 of the preceding section we see by inspection of $a(x)$ and its known factors that $B = 36$ and therefore $l = 3$ is sufficient, so the factors

$$u^{(3)}(x) = 12x + 30; \;\; w^{(3)}(x) = x^2 + 40x + 22$$

computed in iteration step $k = 2$ must be the correct factors apart from units in the ring $\mathbf{Z}_{125}[x]$. Note that

$$u^{(3)}(x)w^{(3)}(x) = 12x^3 + 510x^2 + 1464x + 660 \in \mathbf{Z}[x]$$

so that $u^{(3)}(x)w^{(3)}(x) \neq a(x)$ but

$$u^{(3)}(x)w^{(3)}(x) \equiv a(x) \pmod{5^3}.$$

Now in this example it is known that the correct leading coefficient of $w(x)$ is 6, so we multiply $w^{(3)}(x)$ by 6 in the domain $\mathbf{Z}_{125}[x]$ and correspondingly multiply $u^{(3)}(x)$ by $6^{-1} \in \mathbf{Z}_{125}[x]$ so as to maintain the relationship

$$[6^{-1}u^{(3)}(x)][6w^{(3)}(x)] \equiv a(x) \pmod{5^3}.$$

Since $6^{-1} = 21 \in \mathbf{Z}_{125}[x]$ we obtain the factors

$$u(x) = 21u^{(3)}(x) = 2x + 5 \in \mathbf{Z}_{125}[x];$$

$$w(x) = 6w^{(3)}(x) = 6x^2 - 10x + 7 \in \mathbf{Z}_{125}[x].$$

Then $u(x)w(x) = a(x)$ in the domain $\mathbf{Z}[x]$ and the desired factors have been obtained.

The above example makes use of the knowledge that 6 is the correct leading coefficient of $w(x)$ and it would seem that such knowledge would not be available in the general case. However we will shortly describe a general method which, by slightly altering the original problem, leads to a situation in which the correct leading coefficients of both factors will always be known. For the moment we must prove the result that $u^{(l)}(x)$ and $w^{(l)}(x)$ are associates in the ring $\mathbf{Z}_{p^l}[x]$ of the true factors over the integers. To this end, recall that the units in a polynomial ring are precisely the units in its coefficient ring and therefore we must understand which elements in a ring of the form $\mathbf{Z}_{p^k}$ are units. Unlike the field $\mathbf{Z}_p$ in which every nonzero element is a unit, the ring $\mathbf{Z}_{p^k}$ (for $k > 1$) has some nonzero elements which fail to have multiplicative inverses (e.g. the element $p \in \mathbf{Z}_{p^k}$ is not a unit). The following theorem proves that most of the elements in the ring $\mathbf{Z}_{p^k}$ are units and identifies those elements which are not units.

Theorem 6.3.

Let p be a prime integer and let k be any positive integer. An element $a \in \mathbf{Z}_{p^k}$ is a unit in $\mathbf{Z}_{p^k}$ if and only if p does not divide a (in the integral domain $\mathbf{Z}$).

Proof: We first claim that the integer p is not a unit in $\mathbf{Z}_{p^k}$. For $k = 1$, p is the zero element in $\mathbf{Z}_p$ so the claim is obvious. For $k > 1$, if p is a unit in $\mathbf{Z}_{p^k}$ then there exist integers p^{-1} and c such that in the domain $\mathbf{Z}$

$$pp^{-1} = cp^k + 1$$

whence

$$p(p^{-1} - cp^{k-1}) = 1$$

which implies $p \mid 1$. The latter is impossible so the claim is proved.

'only if': Suppose $p \mid a$ so that $a = px$ for some integer x. If a is a unit in $\mathbf{Z}_{p^k}$ then there exists an integer a^{-1} such that

$$aa^{-1} \equiv 1 \pmod{p^k}.$$

But since $a = px$ it follows that

$$pxa^{-1} \equiv 1 \pmod{p^k}$$

which implies that p has an inverse modulo p^k. This contradicts the claim proved above.

'if': Suppose p does not divide a. Then $\mathrm{GCD}(a,p^k) = 1$ since the only nontrivial divisors of p^k are p^i $(1 \le i \le k)$. Therefore the extended Euclidean algorithm can be applied to compute $a^{-1} \pmod{p^k}$. □

Theorem 6.4.

Let $a(x) \in \mathbf{Z}[x]$ be a given polynomial over the integers, let p be a prime integer which does not divide $\mathrm{Lc}[a(x)]$, and let $u^{(1)}(x), w^{(1)}(x) \in \mathbf{Z}_p[x]$ be two relatively prime polynomials over the field $\mathbf{Z}_p$ such that

$$(65) \quad a(x) \equiv u^{(1)}(x)w^{(1)}(x) \ (\mathrm{mod}\ p).$$

Let l be an integer such that $\frac{1}{2}p^l > B$ where B bounds the magnitudes of all integer coefficients appearing in $a(x)$ and in any of its possible factors with degrees not exceeding $\max\{\deg[u^{(1)}(x)], \deg[w^{(1)}(x)]\}$. Let $u^{(k)}(x)$ and $w^{(k)}(x)$ be factors computed by the Hensel construction such that

$$(66) \quad a(x) \equiv u^{(k)}(x)w^{(k)}(x) \ (\mathrm{mod}\ p^k)$$

and

$$(67) \quad \begin{cases} u^{(k)}(x) \equiv u^{(1)}(x) \ (\mathrm{mod}\ p); \\ w^{(k)}(x) \equiv w^{(1)}(x) \ (\mathrm{mod}\ p) \end{cases}$$

for $k = 1, 2, \ldots, l$. If there exist polynomials $u(x), w(x) \in \mathbf{Z}[x]$ such that

$$(68) \quad a(x) = u(x)w(x) \in \mathbf{Z}[x]$$

and

$$(69) \quad \begin{cases} \mathfrak{n}[u(x)] \equiv \mathfrak{n}[u^{(1)}(x)] \ (\mathrm{mod}\ p); \\ \mathfrak{n}[w(x)] \equiv \mathfrak{n}[w^{(1)}(x)] \ (\mathrm{mod}\ p) \end{cases}$$

where $\mathfrak{n}$ denotes the normalization 'make the polynomial monic as an element of the domain $\mathbf{Z}_p[x]$' then the polynomials $u(x)$ and $u^{(l)}(x)$, as well as $w(x)$ and $w^{(l)}(x)$, are associates in the ring $\mathbf{Z}_{p^l}[x]$. More generally, for each $k \geq 1$ the polynomials $\phi_{p^k}(u(x))$ and $u^{(k)}(x)$, as well as $\phi_{p^k}(w(x))$ and $w^{(k)}(x)$, are associates in the ring $\mathbf{Z}_{p^k}[x]$.

Proof: Let $k \geq 1$ be any fixed positive integer. The assumption that p does not divide $\mathrm{Lc}[a(x)]$ implies, by Theorem 6.3, that $\mathrm{Lc}[a(x)]$ is a unit in $\mathbf{Z}_{p^k}[x]$. We may therefore define the monic polynomial

$$\tilde{a}(x) = \{\mathrm{Lc}[a(x)]\}^{-1}a(x) \in \mathbf{Z}_{p^k}[x].$$

Now (66) implies that

$$\mathrm{Lc}[a(x)] \equiv \mathrm{Lc}[u^{(k)}(x)]\mathrm{Lc}[w^{(k)}(x)] \ (\mathrm{mod}\ p^k)$$

so clearly p does not divide $\mathrm{Lc}[u^{(k)}(x)]$ and p does not divide $\mathrm{Lc}[w^{(k)}(x)]$ (for otherwise $p \mid \mathrm{Lc}[a(x)]$), so we may also define the monic polynomials

$$\tilde{u}^{(k)}(x) = \mathrm{Lc}[u^{(k)}(x)]^{-1}u^{(k)}(x) \in \mathbf{Z}_{p^k}[x];$$

$$\tilde{w}^{(k)}(x) = \mathrm{Lc}[w^{(k)}(x)]^{-1}w^{(k)}(x) \in \mathbf{Z}_{p^k}[x].$$

Obviously we may normalize the polynomials $u^{(1)}(x), w^{(1)}(x) \in \mathbf{Z}_p[x]$ yielding the monic polynomials

$$\tilde{u}^{(1)}(x) = \mathfrak{n}[u^{(1)}(x)]; \ \ \tilde{w}^{(1)}(x) = \mathfrak{n}[w^{(1)}(x)].$$

It is easy to verify that conditions (65), (66), and (67) remain valid when $a(x), u^{(1)}(x), w^{(1)}(x), u^{(k)}(x), w^{(k)}(x)$ are replaced by $\tilde{a}(x), \tilde{u}^{(1)}(x), \tilde{w}^{(1)}(x), \tilde{u}^{(k)}(x), \tilde{w}^{(k)}(x)$, respectively. Then by the Corollary to Theorem 6.2, conditions (66)-(67) in the monic case uniquely determine the monic polynomial factors $\tilde{u}^{(k)}(x), \tilde{w}^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$. Now suppose there exist polynomial factors $u(x), w(x) \in \mathbf{Z}[x]$ satisfying (68)-(69) and consider the polynomials $\phi_{p^k}(u(x)), \phi_{p^k}(w(x)) \in \mathbf{Z}_{p^k}[x]$. By reasoning as above, we may normalize these two polynomials in the ring $\mathbf{Z}_{p^k}[x]$ yielding monic polynomials $\tilde{u}(x), \tilde{w}(x) \in \mathbf{Z}_{p^k}[x]$ and these monic polynomials provide another factorization in $\mathbf{Z}_{p^k}[x]$ satisfying the monic versions of (66)-(67). Hence by uniqueness,

$$\tilde{\mathrm{u}}^{(k)}(x) = \tilde{\mathrm{u}}(x);\ \ \tilde{\mathrm{w}}^{(k)}(x) = \tilde{\mathrm{w}}(x).$$

It follows that $\mathrm{u}^{(k)}(x)$ and $\phi_{p^k}(\mathrm{u}(x))$ are associates in the ring $\mathbf{Z}_{p^k}[x]$ and similarly $\mathrm{w}^{(k)}(x)$ and $\phi_{p^k}(\mathrm{w}(x))$ are associates in the ring $\mathbf{Z}_{p^k}[x]$.

The above proof holds for any fixed $k \geq 1$. In particular when $k = l$, note that since $\frac{1}{2}p^l > \mathrm{B}$ we have $\phi_{p^l}(\mathrm{u}(x)) = \mathrm{u}(x)$ and $\phi_{p^l}(\mathrm{w}(x)) = \mathrm{w}(x)$. □

The Modified Hensel Construction

The result of Theorem 6.4 can be used to 'fix' the Hensel construction so that it will correctly generate the factors over the integers in the non-monic case. To this end we wish to create a situation in which the correct leading coefficients of the factors are known a priori and this can be achieved as follows. Let us assume that the polynomial $\mathrm{a}(x) \in \mathbf{Z}[x]$ to be factored is a primitive polynomial. (Note that this assumption simply means that to factor an arbitrary polynomial over the integers we will first remove the unit part and the content so that the problem reduces to factoring the primitive part). Let $\mathrm{a}(x)$ have the modulo p factorization (65) and suppose that there exist factors $\mathrm{u}(x), \mathrm{w}(x) \in \mathbf{Z}[x]$ satisfying (68)-(69). The leading coefficients

$$\alpha = \mathrm{Lc}[\mathrm{a}(x)];\ \ \mu = \mathrm{Lc}[\mathrm{u}(x)];\ \ \nu = \mathrm{Lc}[\mathrm{w}(x)]$$

clearly must satisfy

$$\alpha = \mu\nu$$

but at this point we do not know the correct splitting of α into μ and ν. However if we define the new polynomial

$$\tilde{\mathrm{a}}(x) = \alpha a(x)$$

and seek a factorization of $\tilde{\mathrm{a}}(x)$ then we have the relationship

$$\tilde{\mathrm{a}}(x) = \mu\nu\mathrm{u}(x)\mathrm{w}(x) = [\nu\mathrm{u}(x)][\mu\mathrm{w}(x)].$$

In other words, by defining $\tilde{\mathrm{u}}(x) = \nu u(x)$ and $\tilde{\mathrm{w}}(x) = \mu w(x)$ we see that there exists a factorization

$$\tilde{\mathrm{a}}(x) = \tilde{\mathrm{u}}(x)\tilde{\mathrm{w}}(x) \in \mathbf{Z}[x]$$

in which the leading coefficient of each factor is known to be α.

The Hensel construction can now be modified for the polynomial $\tilde{\mathrm{a}}(x)$ so that for any $k \geq 1$, it computes factors $\tilde{\mathrm{u}}^{(k)}(x), \tilde{\mathrm{w}}^{(k)}(x) \in \mathbf{Z}_{p^k}[x]$ which satisfy not only the conditions of Hensel's lemma but, in addition, satisfy the relationships

$$\text{(70)}\quad \tilde{\mathrm{u}}^{(k)}(x) = \phi_{p^k}(\tilde{\mathrm{u}}(x));\ \ \tilde{\mathrm{w}}^{(k)}(x) = \phi_{p^k}(\tilde{\mathrm{w}}(x))$$

where $\tilde{\mathrm{u}}(x), \tilde{\mathrm{w}}(x) \in \mathbf{Z}[x]$ are the (unknown) factors of $\tilde{\mathrm{a}}(x)$ over the integers. (Note that the relationships (70) do not hold in Example 6.7). The modification which can be made to the Hensel construction is a simple adjustment of units in each iteration step. For if $\mathrm{u}^{(k)}(x)$ and $\mathrm{w}^{(k)}(x)$ denote modulo p^k factors of $\tilde{\mathrm{a}}(x)$ satisfying the conditions of Hensel's lemma then the modulo p^k factors of $\tilde{\mathrm{u}}^{(k)}(x)$ and $\tilde{\mathrm{w}}^{(k)}(x)$ which maintain the conditions of Hensel's lemma and, in addition, satisfy (70) can be defined by

$$\text{(71)}\quad \begin{cases} \tilde{\mathrm{u}}^{(k)}(x) = \phi_{p^k}(\alpha \times \mathrm{Lc}[\mathrm{u}^{(k)}(x)]^{-1} \times \mathrm{u}^{(k)}(x)); \\ \tilde{\mathrm{w}}^{(k)}(x) = \phi_{p^k}(\alpha \times \mathrm{Lc}[\mathrm{w}^{(k)}(x)]^{-1} \times \mathrm{w}^{(k)}(x)) \end{cases}$$

(where the modulo p^k inverses appearing here are guaranteed to exist by assuming the condition that p does not divide $\mathrm{Lc}[\mathrm{a}(x)]$). The associativity relationships stated in Theorem 6.4 have thus been strengthened to the equality relationships (70) by employing the knowledge that the correct leading coefficient of each factor is α. Finally when $k = l$, where l is large enough so that $\frac{1}{2}p^l$ bounds the magnitudes of all integer coefficients appearing in $\tilde{\mathrm{a}}(x)$ and its factors, the relationships (70) become

$$\tilde{\mathrm{u}}^{(l)}(x) = \tilde{\mathrm{u}}(x);\ \ \tilde{\mathrm{w}}^{(l)}(x) = \tilde{\mathrm{w}}(x)$$

so the factors of $\tilde{\mathrm{a}}(x)$ (which were assumed to exist) have been obtained. Note that if the bound B is defined for the original polynomial $\mathrm{a}(x)$ as in Theorem 6.4 then since the modified Hensel construction is being applied to the larger polynomial $\tilde{\mathrm{a}}(x)$, we must now require l to be large enough so that

$$\tfrac{1}{2}p^l > \mathrm{B}\,\mathrm{Lc}[\mathrm{a}(x)].$$

The final step of this modified Hensel construction is to deduce the factorization of $\mathrm{a}(x)$ from the computed factorization

$$\tilde{\mathrm{a}}(x) = \tilde{\mathrm{u}}(x)\tilde{\mathrm{w}}(x) \in \mathbf{Z}[x].$$

Since $\mathrm{a}(x)$ was assumed to be primitive, we have the relationship $\mathrm{a}(x) = \mathrm{pp}[\tilde{\mathrm{a}}(x)]$ from which it follows that the desired factors of $\mathrm{a}(x)$ are defined by

$$\mathrm{u}(x) = \mathrm{pp}[\tilde{\mathrm{u}}(x)];\ \ \mathrm{w}(x) = \mathrm{pp}[\tilde{\mathrm{w}}(x)].$$

The modification of the Hensel construction which is actually used in Algorithm 6.1 is a more efficient adaptation of the above ideas. Before discussing the improved version we consider an example.

Example 6.8.

Let us return to the problem of Example 6.7 where the Hensel construction failed to produce the factors over the integers. We have

$$\mathrm{a}(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbf{Z}[x]$$

and

$$\mathrm{a}(x) \equiv \mathrm{u}^{(1)}(x)\mathrm{w}^{(1)}(x) \pmod 5$$

where $\mathrm{u}^{(1)}(x) = 2x$; $\mathrm{w}^{(1)}(x) = x^2 + 2$. Note that $\mathrm{a}(x)$ is a primitive polynomial, and that the prime 5 does not divide the leading coefficient 12. In the new scheme, we define the new polynomial

$$\tilde{\mathrm{a}}(x) = 12\mathrm{a}(x) = 144x^3 + 120x^2 - 432x + 420.$$

We know that if there exists a factorization of $\mathrm{a}(x)$ satisfying conditions (68)-(69) then there also exists a corresponding factorization such that 12 is the leading coefficient of each factor. The initial factorization

$$\tilde{\mathrm{a}}(x) \equiv \tilde{\mathrm{u}}^{(1)}(x)\tilde{\mathrm{w}}^{(1)}(x) \pmod 5$$

such that the case $k = 1$ of (70) is satisfied can be obtained by applying the adjustment (71) to the given polynomials $\mathrm{u}^{(1)}(x)$ and $\mathrm{w}^{(1)}(x)$; we get

$$\tilde{\mathrm{u}}^{(1)}(x) = \phi_5(12 \times 2^{-1} \times (2x)) = 2x;$$

$$\tilde{\mathrm{w}}^{(1)}(x) = \phi_5(12 \times 1^{-1} \times (x^2 + 2)) = 2x^2 - 1.$$

Applying iteration step $k = 1$ of the usual Hensel construction to the polynomial $\tilde{\mathrm{a}}(x)$, we get

$$\mathrm{u}^{(2)}(x) = \tilde{\mathrm{u}}^{(1)}(x) + (-x + 1)5 = -3x + 5;$$

$$w^{(2)}(x) = \tilde{w}^{(1)}(x) + (x - 1)5 = 2x^2 + 5x - 6.$$

Applying the adjustment (71) yields

$$\tilde{u}^{(2)}(x) = \phi_{25}(12 \times (-3)^{-1} \times (-3x + 5)) = 12x + 5;$$

$$\tilde{w}^{(2)}(x) = \phi_{25}(12 \times 2^{-1} \times (2x^2 + 5x - 6)) = 12x^2 + 5x - 11.$$

In iteration step $k = 2$ we get

$$u^{(3)}(x) = \tilde{u}^{(2)}(x) + (1)5^2 = 12x + 30;$$

$$w^{(3)}(x) = \tilde{w}^{(2)}(x) + (-x + 1)5^2 = 12x^2 - 20x + 14$$

and the adjustment (71) leaves the factors unchanged – i.e. $\tilde{u}^{(3)}(x) = 12x + 30$; $\tilde{w}^{(3)}(x) = 12x^2 - 20x + 14$. At this point,

$$\tilde{a}(x) - \tilde{u}^{(3)}(x)\tilde{w}^{(3)}(x) = 0$$

so the iteration halts and the factorization of $\tilde{a}(x)$ has been obtained. Thus the desired factors of the original polynomial $a(x)$ are:

$$u(x) = pp[\tilde{u}^{(3)}(x)] = 2x + 5;$$

$$w(x) = pp[\tilde{w}^{(3)}(x)] = 6x^2 - 10x + 7. \quad \square$$

Applying a Smaller Multiplier

In the scheme described above and applied in Example 6.8, note that the polynomial $\tilde{a}(x)$ which is actually factored may contain integer coefficients which are significantly larger than the integer coefficients in the original polynomial $a(x)$. This may lead to a decrease in efficiency compared to a scheme which works directly with the original polynomial $a(x)$. Exercise 6-xx considers a scheme in which the Hensel construction is applied to the original polynomial $a(x)$ (exactly as in Example 6.7) until l is large enough so that

$$(72) \quad \tfrac{1}{2}p^l > B\, Lc[a(x)],$$

and then at the end a 'restore leading coefficient' operation is performed. One disadvantage of such a scheme is that the iteration then loses its 'automatic' stopping criterion – i.e. it is not generally possible in such a scheme to recognize that enough iterations have been performed prior to satisfying the bound (72). This disadvantage is aggravated by two additional facts: (i) in practice the bound B almost always will be a very pessimistic bound; and (ii) each successive iteration step is usually more costly than the previous step (e.g. in Example 6.7 note the growth in the size of the coefficients of $e(x)$ and the factors with successive iteration steps). Therefore the potential saving of costly iterations offered by an iterative scheme which can recognize termination independently of the bound (72) can be very significant. An even more serious disadvantage of a scheme using a final 'restore leading coefficient' operation arises in the multivariate case (see Exercise 6-yy).

The problem of coefficient size in the scheme which factors $\tilde{a}(x)$ rather than directly factoring $a(x)$ can be partially alleviated in certain circumstances as follows. Suppose we choose a multiplier γ which is smaller than $Lc[a(x)]$ in defining the new polynomial

$$(73) \quad \tilde{a}(x) = \gamma a(x).$$

Suppose it is known that γ is a multiple of the leading coefficient of one of the factors to be computed, let us say $u(x)$ – i.e. suppose it is known that

$$(74) \quad Lc[u(x)] \mid \gamma.$$

Then the polynomial $\tilde{a}(x)$ defined by (73) has a factorization in which the leading coefficients of the factors are known, where as usual we are assuming the existence of an appropriate factorization of the original polynomial $a(x)$. (Note that the choice $\gamma = Lc[a(x)]$ used previously is a particular case of a multiplier which satisfies (74)). In order to see this fact, let the assumed factorization of $\mathrm{a}(x)$ be

$$\mathrm{a}(x) = \mathrm{u}(x)\mathrm{w}(x) \in \mathbf{Z}[x]$$

and as before let us define the following leading coefficients:

$$\alpha = \mathrm{Lc}[\mathrm{a}(x)];\ \ \mu = \mathrm{Lc}[\mathrm{u}(x)];\ \ \nu = \mathrm{Lc}[\mathrm{w}(x)].$$

In addition, by (74) we may define the integer

$$\beta = \gamma/\mu.$$

Then the polynomial $\tilde{\mathrm{a}}(x)$ defined by (73) satisfies the following relationship:

$$\tilde{\mathrm{a}}(x) = \beta\mu\mathrm{u}(x)\mathrm{w}(x) = [\beta\mathrm{u}(x)][\mu\mathrm{w}(x)].$$

Hence by defining $\tilde{\mathrm{u}}(x) = \beta\mathrm{u}(x)$ and $\tilde{\mathrm{w}}(x) = \mu\mathrm{w}(x)$ we see that there exists a factorization

$$\tilde{\mathrm{a}}(x) = \tilde{\mathrm{u}}(x)\tilde{\mathrm{w}}(x) \in \mathbf{Z}[x]$$

in which

$$\mathrm{Lc}[\tilde{\mathrm{u}}(x)] = \beta\mu = \gamma;\ \ \mathrm{Lc}[\tilde{\mathrm{w}}(x)] = \mu\nu = \alpha$$

where α is the known integer $\mathrm{Lc}[\mathrm{a}(x)]$ and where γ has been specified. It is this generalization of the previously discussed scheme which is implemented in Algorithm 6.1, where γ is an optional input. If γ is unspecified on input then step 1 of the algorithm sets $\gamma = \mathrm{Lc}[\mathrm{a}(x)]$ by default. It might seem that the specification of a γ smaller than $\mathrm{Lc}[\mathrm{a}(x)]$ satisfying (74) would be impossible for most practical problems. However it turns out that in the application of the Hensel lifting algorithm to the important problem of polynomial GCD computation, the specification of γ is always possible (see chapter 7). Finally, note that by (73) the termination condition (72) (for the case when the factorization of $\mathrm{a}(x)$ does not exist) can be changed to the condition

$$\tfrac{1}{2}p^l > B\gamma.$$

The replaceLc Operation

The design of Algorithm 6.1 has now been fully explained except for one very significant modification. The scheme we have described (and applied in Example 6.8) requires that formulas (71) be applied to adjust units in each iteration step. However it can be seen that step 4 of Algorithm 6.1 contains no such adjustment of units in each iteration of the **while**-loop. Indeed step 4 of Algorithm 6.1 is simply an implementation of the pure unmodified Hensel construction. The reason that Algorithm 6.1 is able to avoid the extra cost of adjusting units in each iteration stems from the yet-to-be-explained 'replaceLc' operation appearing in step 3. This new operation is an ingenious modification described by Yun [Yun73] and attributed to a suggestion by Moses. Consider the polynomial $\tilde{\mathrm{a}}(x)$ defined by (73) and consider its modulo p factors $\tilde{\mathrm{u}}^{(1)}(x), \tilde{\mathrm{w}}^{(1)}(x) \in \mathbf{Z}_p[x]$ adjusted (as in step 1 of Algorithm 6.1) so that

$$(75)\quad \tilde{\mathrm{u}}^{(1)}(x) = \phi_p(\tilde{\mathrm{u}}(x));\ \ \tilde{\mathrm{w}}^{(1)}(x) = \phi_p(\tilde{\mathrm{w}}(x))$$

where $\tilde{\mathrm{u}}(x)$ and $\tilde{\mathrm{w}}(x)$ are the factors of $\tilde{\mathrm{a}}(x)$ over the integers as discussed above such that

$$(76)\quad \mathrm{Lc}[\tilde{\mathrm{u}}(x)] = \gamma;\ \ \mathrm{Lc}[\tilde{\mathrm{w}}(x)] = \mathrm{Lc}[\mathrm{a}(x)].$$

Writing the modulo p factors in the form

$$\tilde{\mathrm{u}}^{(1)}(x) = \mu_m x^m + \mu_{m-1}x^{m-1} + \cdots + \mu_0;$$

$$\tilde{\mathrm{w}}^{(1)}(x) = \nu_n x^n + \nu_{n-1}x^{n-1} + \cdots + \nu_0$$

where $\mu_m \neq 0$ and $\nu_n \neq 0$, it follows from (75)-(76) that $\mu_m = \phi_p(\gamma)$ and $\nu_n = \phi_p(\mathrm{Lc}[\mathrm{a}(x)])$. Now suppose that the factors $\tilde{\mathrm{u}}^{(1)}(x)$ and $\tilde{\mathrm{w}}^{(1)}(x)$ are changed by simply replacing the leading coefficients μ_m and ν_n by γ and $\alpha = \mathrm{Lc}[a(x)]$, respectively. To this end we define the algorithmic operation ***replaceLc*** as follows:

> Given a polynomial $a(x) \in R[x]$ over a coefficient ring R and given an element $r \in R$, the result of the operation replaceLc$(a(x), r)$ is the polynomial obtained from $a(x)$ by replacing the leading coefficient of $a(x)$ by r.

In this algorithmic notation, the polynomials $\tilde{u}^{(1)}(x)$ and $\tilde{w}^{(1)}(x)$ are replaced by the polynomials replaceLc$(\tilde{u}^{(1)}(x), \gamma)$ and replaceLc$(\tilde{w}^{(1)}(x), \alpha)$. Let $\tilde{u}^{(1)}(x)$ and $\tilde{w}^{(1)}(x)$ now denote the modified factors – i.e.

$$(77) \quad \begin{cases} \tilde{u}^{(1)}(x) = \gamma x^m + \mu_{m-1}x^{m-1} + \cdots + \mu_0; \\ \tilde{w}^{(1)}(x) = \alpha x^n + \nu_{n-1}x^{n-1} + \cdots + \nu_0. \end{cases}$$

Then the leading coefficients of $\tilde{u}^{(1)}(x)$ and $\tilde{w}^{(1)}(x)$ are no longer represented as elements of the field $\mathbf{Z}_p$ in the usual representation, but nonetheless we still have the property

$$\tilde{a}(x) \equiv \tilde{u}^{(1)}(x)\tilde{w}^{(1)}(x) \pmod{p}.$$

The Hensel construction can therefore be applied using (77) as the initial factors.

Let us consider the form of the successive factors which will be computed by the Hensel construction based on (77). Using the notation of step 4 of Algorithm 6.1, we first compute

$$c(x) = \frac{e(x)}{p} = \frac{\tilde{a}(x) - \tilde{u}^{(1)}(x)\tilde{w}^{(1)}(x)}{p}$$

(where the domain of this computation is $\mathbf{Z}[x]$). Now since $\mathrm{Lc}[\tilde{a}(x)] = \gamma\alpha$, it is clear from (77) that we will have

$$\deg[c(x)] < \deg[\tilde{a}(x)] = \deg[\tilde{u}^{(1)}(x)] + \deg[\tilde{w}^{(1)}(x)].$$

This strict inequality implies that the Hensel construction will then perform exactly as in the monic case in the following sense. The solution $\sigma(x), \tau(x) \in \mathbf{Z}_p[x]$ of the polynomial diophantine equation solved in step 4.1 of the algorithm will satisfy (as usual) the condition

$$\deg[\sigma(x)] < \deg[\tilde{w}^{(1)}(x)]$$

and, in addition, we will have the following condition

$$\deg[\tau(x)] < \deg[\tilde{u}^{(1)}(x)]$$

(see Theorem 2.6). Therefore when the factors are updated in step 4.2 of the algorithm the leading coefficients of both factors will remain unchanged. This is a desirable property since the leading coefficients are already known to be the correct integer coefficients. By the same reasoning, each successive iteration of the Hensel construction will also leave the leading coefficients unchanged. Finally since the successive factors computed by this scheme must satisfy Hensel's lemma, Theorem 6.4 guarantees that after a sufficient number of iterations the computed factors will be associates of the true factors of $\tilde{a}(x)$ over the integers (if such factors exist); but since the computed factors have the same leading coefficients as the true factors over the integers, they can be associates only if they are identically equal. Therefore the desired factors of $\tilde{a}(x)$ will be computed (and will be recognized) by the iteration in step 4 of Algorithm 6.1 and no further adjustment of units is required. (For a discussion of further efficiency improvements to Algorithm 6.1 see Exercise 6-zz).

Example 6.9.

Consider once again the problem of factoring the non-monic polynomial of Example 6.7:

$$a(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbf{Z}[x].$$

This time we will apply Algorithm 6.1 in its full generality. The input to the algorithm is the primitive polynomial $a(x)$, the prime $p = 5$ (note that p does not divide $\mathrm{Lc}[a(x)]$), and the two relatively prime modulo 5 factors of $a(x)$ given by $u^{(1)}(x) = 2x$ and $w^{(1)}(x) = x^2 + 2$. The value of the bound B required by the algorithm is not needed in this example because the iterations will terminate by finding a factorization. The integer γ is undefined on input.

In step 1 of Algorithm 6.1 the following values are assigned:

$$\alpha = 12;$$

$$\gamma = 12;$$

$$a(x) = 144x^3 + 120x^2 - 432x + 420;$$

$$u^{(1)}(x) = 2x;\;\; w^{(1)}(x) = 2x^2 - 1.$$

In step 2 the extended Euclidean algorithm is applied to $u^{(1)}(x)$ and $w^{(1)}(x)$ in the domain $\mathbf{Z}_5[x]$ yielding

$$s(x) = x;\;\; t(x) = -1.$$

In step 3 the leading coefficients of $u^{(1)}(x)$ and $w^{(1)}(x)$ are replaced by the correct integer coefficients; we get:

$$u(x) = 12x;\;\; w(x) = 12x^2 - 1;$$

$$e(x) = 120x^2 - 420x + 420;$$

$$\mathit{modulus} = 5.$$

In step 4 the sequence of values computed for $\sigma(x), \tau(x), u(x), w(x)$, and $e(x)$ is as follows.

End of iteration no.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
0	–	–	$12x$	$12x^2 - 1$	$120x^2 - 420x + 420$
1	$x - 2$	1	$12x + 5$	$12x^2 + 5x - 11$	$-325x + 475$
2	$-x + 1$	1	$12x + 30$	$12x^2 - 20x + 14$	0

Finally in step 5 the following values are obtained:

$$\delta = \mathrm{cont}[12x + 30] = 6;$$

$$u(x) = \frac{12x + 30}{6} = 2x + 5;$$

$$w(x) = \frac{12x^2 - 20x + 14}{\frac{12}{6}} = 6x^2 - 10x + 7.$$

Note that this computation was essentially equivalent to the computation in Example 6.8 except that we have avoided the cost of adjusting units in each iteration step. □

6.6. MULTIVARIATE GENERALIZATION OF HENSEL'S LEMMA

We return now to the general multivariate lifting problem which was discussed at the beginning of section 6.3. Specifically, we wish to find multivariate polynomials $u(x_1, \ldots, x_v), w(x_1, \ldots, x_v) \in \mathbf{Z}[x_1, \ldots, x_v]$ which satisfy equations (39)-(40) where $u_0(x_1)$ and $w_0(x_1)$, the images mod $<I, p>$, are given. Here $I = <x_2-\alpha_2, \ldots, x_v-\alpha_2>$ is the kernel of a multivariate evaluation homomorphism and p is a prime integer.

A Homomorphism Diagram

We consider the lifting process in two separate stages. Firstly, the solution in $\mathbf{Z}_p[x_1]$ is lifted to the solution in $\mathbf{Z}_{p^l}[x_1]$ for some sufficiently large l such that the ring $\mathbf{Z}_{p^l}$ can be identified with $\mathbf{Z}$ for the particular problem being solved. This first stage of lifting is accomplished by the univariate Hensel lifting algorithm (Algorithm 6.1). Secondly, the solution in $\mathbf{Z}_{p^l}[x_1]$ is lifted to the desired solution in $\mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ (which is identified with the original domain $\mathbf{Z}[x_1, \ldots, x_v]$) by the multivariate Hensel lifting algorithm to be described. The latter algorithm is given the solution mod $<\mathrm{I}, p^l>$ and, using an iteration analogous to the univariate case, it lifts to the solution mod $<\mathrm{I}^{k+1}, p^l>$ for $k = 1, 2, \ldots, d$ where d is the maximum total degree in the indeterminate $x_2, \ldots, x_v$ of any term in the solution polynomials.

Figure 6.2 shows a homomorphism diagram for solving a problem using the univariate and multivariate Hensel lifting algorithms. It should be noted that the order of the univariate and multivariate operations has been reversed compared with the homomorphism diagram of Figure 6.1 presented at the end of section 6.2.

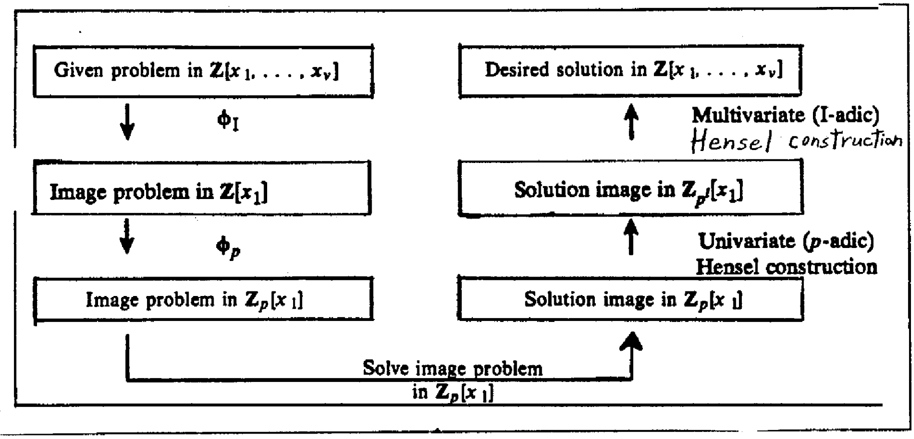
Given problem in $\mathbf{Z}[x_1, \ldots, x_v]$
ϕ_{I}
Image problem in $\mathbf{Z}[x_1]$
ϕ_p
Image problem in $\mathbf{Z}_p[x_1]$
Desired solution in $\mathbf{Z}[x_1, \ldots, x_v]$
Multivariate (I-adic)
Hensel construction
Solution image in $\mathbf{Z}_{p^l}[x_1]$
Univariate (p-adic)
Hensel construction
Solution image in $\mathbf{Z}_p[x_1]$
Solve image problem
in $\mathbf{Z}_p[x_1]$

Figure 6.2: Homomorphism diagram for univariate and multivariate Hensel constructions.

In the setting of section 6.2 (solving a polynomial equation f(u) = 0 via Newton's iteration), the computation could be organized in either order. In the current setting (solving a bivariate polynomial equation F(u, w) = 0 via Hensel constructions), there is a fundamental reason for organizing the computation in the order specified by Figure 6.2. Before pursuing this point some further remarks about the diagram in Figure 6.2 should be noted. In the box at the end of the arrow labelled "Univariate (p-adic) Hensel construction", the domain is specified as $\mathbf{Z}_{p^l}[x_1]$. As we have already noted, l will be chosen to be sufficiently large for the particular problem so that the ring $\mathbf{Z}_{p^l}$ can be identified with $\mathbf{Z}$ (an identification we have made in the box following the multivariate Hensel construction). The specification of the domain in the form $\mathbf{Z}_{p^l}[x_1]$ is deliberate, in order to emphasize the fact (to be seen shortly) that the domain of the operations required by the multivariate Hensel construction is $\mathbf{Z}_{p^l}[x_1]$ and not $\mathbf{Z}[x_1]$. Another point to be noted about the organization of this diagram is that the multivariate problem has been conceptually separated from the univariate problem, such that a diagram for the multivariate problem could read: "Apply the homomorphism ϕ_{I}, solve the univariate problem in $\mathbf{Z}[x_1]$ (by any method of your choosing), and finally apply the multivariate Hensel construction." However the operations for the multivariate Hensel construction require that the univariate domain $\mathbf{Z}[x_1]$ must be replaced by a domain $\mathbf{Z}_{p^l}[x_1]$ for some prime p and integer l, even if the univariate problem was solved by a non-Hensel method.

Recall from section 6.3 that the basic computation to be performed in applying a step of a Hensel iteration (i.e., Newton's iteration applied to the equation

$$\mathrm{F}(\mathrm{u}, \mathrm{w}) = \mathrm{a}(x_1, \ldots, x_v) - \mathrm{uw} = 0\,)$$

is to solve a polynomial diophantine equation of the form

$$(78) \quad A^{(k)}\Delta\mathrm{u}^{(k)} + B^{(k)}\Delta\mathrm{w}^{(k)} = C^{(k)}$$

for the correction terms $\Delta\mathrm{u}^{(k)}, \Delta\mathrm{w}^{(k)}$ (where $A^{(k)}, B^{(k)}, C^{(k)}$ are given polynomials). Now if the order of the univariate and multivariate lifting steps is to be that of Figure 6.1 (i.e. I-adic lifting preceding p-adic lifting) then during I-adic lifting equation (78) will have to be solved in the domain $\mathbf{Z}_p[x_1]$, and during p-adic lifting equation (78) will have to be solved in the domain $\mathbf{Z}_p[x_1, \ldots, x_v]$. As we have already seen in the development of Algorithm 6.1, Theorem 2.6 shows how to solve equation (78) in the Euclidean domain $\mathbf{Z}_p[x_1]$. However the necessity to solve equation (78) in the multivariate polynomial domain $\mathbf{Z}_p[x_1, \ldots, x_v]$ poses serious difficulties. Theorem 2.6 does not apply because this multivariate domain is certainly not a Euclidean domain. It is possible to develop methods to solve equation (78) in multivariate domains but the computational expense of these methods makes the Hensel construction impractical when organized in this way. On the other hand, if the computation is organized as specified in the diagram of Figure 6.2 then during p-adic lifting equation (78) will be solved in the Euclidean domain $\mathbf{Z}_p[x_1]$, and during I-adic lifting equation (78) will be solved in the ring $\mathbf{Z}_{p^l}[x_1]$. Again we have the apparent difficulty that the latter ring is not a Euclidean domain. However this univariate polynomial ring is "nearly a Euclidean domain" in the sense that the ring $\mathbf{Z}_{p^l}$ is "nearly a field" (see Theorem 6.3 in the preceding section). The constructive proof of Theorem 2.6 (which is based on applying the extended Euclidean algorithm) will often remain valid for solving equation (78) in the univariate polynomial ring $\mathbf{Z}_{p^l}[x_1]$, with a little luck in the choice of the prime p for the particular problem being solved. In general though, we cannot guarantee the existence of the inverses required by the extended Euclidean algorithm when it is applied in the ring $\mathbf{Z}_{p^l}[x_1]$.

Polynomial Diophantine Equations in $\mathbf{Z}_{p^l}[x_1]$

A general solution to the problem of solving polynomial diophantine equations in the ring $\mathbf{Z}_{p^l}[x_1]$ is obtained by applying Newton's iteration to lift the solution in $\mathbf{Z}_p[x_1]$ up to a solution in $\mathbf{Z}_{p^l}[x_1]$. The "extended Euclidean" problem to be solved is to find polynomials $\mathrm{s}^{(l)}(x_1), \mathrm{t}^{(l)}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ which satisfy the equation

$$(79) \quad \mathrm{s}^{(l)}(x_1)\mathrm{u}(x_1) + \mathrm{t}^{(l)}(x_1)\mathrm{w}(x_1) \equiv 1 \; (\mathrm{mod}\; p^l)$$

where $\mathrm{u}(x_1), \mathrm{w}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ are given polynomials such that $\phi_p(\mathrm{u}(x_1)), \phi_p(\mathrm{w}(x_1))$ are relatively prime polynomials in the Euclidean domain $\mathbf{Z}_p[x_1]$. The equation to which we will apply Newton's iteration is

$$\mathrm{G}(\mathrm{s}, \mathrm{t}) = \mathrm{su}(x_1) + \mathrm{tw}(x_1) - 1 = 0.$$

Proceeding as in previous sections, if we have the order-k p-adic approximations $\mathrm{s}^{(k)}, \mathrm{t}^{(k)}$ to the solution pair $\bar{\mathrm{s}}, \bar{\mathrm{t}}$ and if we obtain correction terms $\Delta\mathrm{s}^{(k)}, \Delta\mathrm{t}^{(k)}$ which satisfy the equation

$$(80) \quad \mathrm{G}_{\mathrm{s}}(\mathrm{s}^{(k)}, \mathrm{t}^{(k)})\Delta\mathrm{s}^{(k)} + \mathrm{G}_{\mathrm{t}}(\mathrm{s}^{(k)}, \mathrm{t}^{(k)})\Delta\mathrm{t}^{(k)} \equiv -G(\mathrm{s}^{(k)}, \mathrm{t}^{(k)}) \;\; (\mathrm{mod}\; p^{k+1})$$

then the order-$(k+1)$ p-adic approximations are given by

$$\mathrm{s}^{(k+1)} = \mathrm{s}^{(k)} + \Delta\mathrm{s}^{(k)}, \quad \mathrm{t}^{(k+1)} = \mathrm{t}^{(k)} + \Delta\mathrm{t}^{(k)}.$$

Writing the correction terms in the form

$$\Delta s^{(k)} = s_k(x_1)p^k, \quad \Delta t^{(k)} = t_k(x_1)p^k$$

where $s_k(x_1), t_k(x_1) \in \mathbf{Z}_p[x_1]$, substituting for the partial derivatives, and dividing through by p^k, equation (80) becomes

$$\text{(81)} \quad u(x_1)s_k(x_1) + w(x_1)t_k(x_1) \equiv \frac{1-s^{(k)}u(x_1) - t^{(k)}w(x_1)}{p^k} \pmod{p}.$$

The order-1 p-adic approximations $s^{(1)}, t^{(1)} \in \mathbf{Z}_p[x_1]$ for the solution of equation (79) are obtained by the extended Euclidean algorithm (or, in the context of Figure 6.2, they have already been computed in Algorithm 6.1 for the univariate Hensel construction). For $k = 1, 2, \cdots, l-1$, equation (81) can be solved for the correction terms $s_k(x_1), t_k(x_1) \in \mathbf{Z}_p[x_1]$ by Theorem 2.6, thus generating the desired solution of equation (79).

The following theorem shows that we can solve, in the ring $\mathbf{Z}_{p^l}[x_1]$, the polynomial diophantine equations which arise in the multivariate Hensel construction.

Theorem 6.5.

For a prime integer p and a positive integer l, let $u(x_1), w(x_1) \in \mathbf{Z}_{p^l}[x_1]$ be univariate polynomials satisfying the following conditions:

(*i*) $p \nmid \mathrm{Lc}[u(x_1)]$ and $p \nmid \mathrm{Lc}[w(x_1)]$;

(*ii*) $\phi_p(u(x_1))$ and $\phi_p(w(x_1))$ are relatively prime polynomials in $\mathbf{Z}_p[x_1]$.

Then for any polynomial $c(x_1) \in \mathbf{Z}_{p^l}[x_1]$ there exist unique polynomials $\sigma(x_1), \tau(x_1) \in \mathbf{Z}_{p^l}[x_1]$ such that

$$\text{(82)} \quad \sigma(x_1)u(x_1) + \tau(x_1)w(x_1) \equiv c(x_1) \pmod{p^l}$$

and

$$\text{(83)} \quad \deg[\sigma(x_1)] < \deg[w(x_1)].$$

Moreover, if $\deg[c(x_1)] < \deg[u(x_1)] + \deg[w(x_1)]$ then $\tau(x_1)$ satisfies

$$\text{(84)} \quad \deg[\tau(x_1)] < \deg[u(x_1)].$$

Proof:

Existence:

The extended Euclidean algorithm can be applied to compute polynomials $s^{(1)}(x_1), t^{(1)}(x_1) \in \mathbf{Z}_p[x_1]$ satisfying the equations

$$\text{(85)} \quad s^{(1)}(x_1)u(x_1) + t^{(1)}(x_1)w(x_1) \equiv 1 \pmod{p}.$$

By Theorem 2.6, equation (81) can be solved for polynomials $s_k(x_1), t_k(x_1) \in \mathbf{Z}_p[x_1]$ for successive integers $k \geq 1$, where we define

$$s^{(k)}(x_1) = s^{(1)}(x_1) + s_1(x_1)p + \cdots + s_{k-1}(x_1)p^{k-1};$$

$$t^{(k)}(x_1) = t^{(1)}(x_1) + t_1(x_1)p + \cdots + t_{k-1}(x_1)p^{k-1};$$

and we must prove that

$$\text{(86)} \quad s^{(k)}(x_1)u(x_1) + t^{(k)}(x_1)w(x_1) \equiv 1 \pmod{p^k}.$$

We will prove (86) by induction. The case $k = 1$ is given by equation (85). Suppose (86) holds for some $k \geq 1$. Then noting that

$$\mathrm{s}^{(k+1)}(x_1) = \mathrm{s}^{(k)}(x_1) + \mathrm{s}_k(x_1)p^k \text{ and } \mathrm{t}^{(k+1)}(x_1) = \mathrm{t}^{(k)}(x_1) + \mathrm{t}_k(x_1)p^k$$

where $\mathrm{s}_k(x_1)$ and $\mathrm{t}_k(x_1)$ are the solutions of equation (81), we have

$$\mathrm{s}^{(k+1)}(x_1)\mathrm{u}(x_1) + \mathrm{t}^{(k+1)}(x_1)\mathrm{w}(x_1) =$$
$$\mathrm{s}^{(k)}(x_1)\mathrm{u}(x_1) + \mathrm{t}^{(k)}(x_1)\mathrm{w}(x_1) + p^k[\mathrm{s}_k(x_1)\mathrm{u}(x_1) + \mathrm{t}_k(x_1)\mathrm{w}(x_1)] \equiv 1 \ (\mathrm{mod}\ p^{k+1})$$

where we have applied equation (81) after multiplying it through by p^k. Thus (86) is proved for all $k \geq 1$, and in particular for $k = l$.

Now the desired polynomials $\sigma(x_1)$, $\tau(x_1) \in \mathbf{Z}_{p^l}[x_1]$ satisfying equation (82) can be calculated exactly as in the proof of Theorem 2.6. Specifically, the polynomials

$$\tilde{\sigma}(x_1) = \mathrm{s}^{(l)}(x_1)\mathrm{c}(x_1) \text{ and } \tilde{\tau}(x_1) = \mathrm{t}^{(l)}(x_1)\mathrm{c}(x_1)$$

satisfy equation (82) and then to reduce the degree we apply Euclidean division of $\tilde{\sigma}(x_1)$ by $\mathrm{w}(x_1)$ yielding $\mathrm{q}(x_1)$, $\mathrm{r}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ such that

$$\tilde{\sigma}(x_1) \equiv \mathrm{w}(x_1)\mathrm{q}(x_1) + \mathrm{r}(x_1) \ (\mathrm{mod}\ p^l)$$

where $\deg[\mathrm{r}(x_1)] < \deg[\mathrm{w}(x_1)]$. This division step will be valid in the ring $\mathbf{Z}_{p^l}[x_1]$ because condition (i) guarantees that $\mathrm{Lc}[\mathrm{w}(x_1)]$ is a unit in the ring $\mathbf{Z}_{p^l}$ (see Theorem 6.3). Finally, defining

$$\sigma(x_1) = \mathrm{r}(x_1) \text{ and } \tau(x_1) = \tilde{\tau}(x_1) + \mathrm{q}(x_1)\mathrm{u}(x_1) \in \mathbf{Z}_{p^l}[x_1]$$

equation (82) and the degree constraint (83) are readily verified.

Uniqueness:

Let $\sigma_1(x_1)$, $\tau_1(x_1) \in \mathbf{Z}_{p^l}[x_1]$ and $\sigma_2(x_1)$, $\tau_2(x_1) \in \mathbf{Z}_{p^l}[x_1]$ be two pairs of polynomials satisfying (82)-(83). Subtracting the two different equations of the form (82) yields

$$\text{(87)} \quad (\sigma_1(x_1) - \sigma_2(x_1))\mathrm{u}(x_1) \equiv -(\tau_1(x_1) - \tau_2(x_1))\mathrm{w}(x_1) \ (\mathrm{mod}\ p^l).$$

Also, the degree constraint (83) satisfied by $\sigma_1(x_1)$ and $\sigma_2(x_1)$ yields

$$\text{(88)} \quad \deg[\sigma_1(x_1) - \sigma_2(x_1)] < \deg[\mathrm{w}(x_1)].$$

Now taking the congruence (87) modulo p we have a relationship in the domain $\mathbf{Z}_p[x_1]$ which, together with condition (ii), implies that $\phi_p(\mathrm{w}(x_1))$ divides $\phi_p(\sigma_1(x_1) - \sigma_2(x_1))$ in the domain $\mathbf{Z}_p[x_1]$. Noting from condition (i) that $\phi_p(\mathrm{w}(x_1))$ has the same degree as $\mathrm{w}(x_1)$, (88) implies that

$$\sigma_1(x_1) - \sigma_2(x_1) \equiv 0 \ (\mathrm{mod}\ p)$$

and then it follows from (87) that

$$\tau_1(x_1) - \tau_2(x_1) \equiv 0 \ (\mathrm{mod}\ p)$$

We now claim that the polynomials $\sigma_1(x_1) - \sigma_2(x_1)$ and $\tau_1(x_1) - \tau_2(x_1)$ satisfying (87) are divisible by p^k for all positive integers $k \leq l$. The proof is by induction. The case $k = 1$ has just been proved. Suppose that they are divisible by p^k for some $k < l$. Then we may define the polynomials

$$\alpha(x_1) = (\sigma_1(x_1) - \sigma_2(x_1)) / p^k \text{ and } \beta(x_1) = (\tau_1(x_1) - \tau_2(x_1)) / p^k$$

and, dividing through by p^k in congruence (87) we have

$$\alpha(x_1)\mathrm{u}(x_1) \equiv -\beta(x_1)\mathrm{w}(x_1) \ (\mathrm{mod}\ p^{l-k}).$$

By repeating the argument used above, we conclude that

$$\alpha(x_1) \equiv 0 \pmod p \quad \text{and} \quad \beta(x_1) \equiv 0 \pmod p;$$

i.e., $\sigma_1(x_1) - \sigma_2(x_1)$ and $\tau_1(x_1) - \tau_2(x_1)$ are divisible by p^{k+1}, which proves the claim.

Finally, we have proved that

$$\sigma_1(x_1) \equiv \sigma_2(x_1) \pmod{p^l} \quad \text{and} \quad \tau_1(x_1) \equiv \tau_2(x_1) \pmod{p^l}$$

which proves uniqueness in the ring $\mathbf{Z}_{p^l}[x_1]$.

Final Degree Constraint:

It remains to prove (84). From (82) we can write

$$\tau(x_1) \equiv (\mathrm{c}(x_1) - \sigma(x_1)\mathrm{u}(x_1)) \,/\, \mathrm{w}(x_1) \pmod{p^l}$$

and the division here is valid in the ring $\mathbf{Z}_{p^l}[x_1]$ because $\mathrm{Lc}[\mathrm{w}(x_1)]$ is a unit in $\mathbf{Z}_{p^l}$, by condition (i). By this same condition, we have

$$\text{(89)} \quad \deg[\tau(x_1)] = \deg[\mathrm{c}(x_1) - \sigma(x_1)\mathrm{u}(x_1)] - \deg[\mathrm{w}(x_1)].$$

Now if $\deg[\mathrm{c}(x_1)] \geq \deg[\sigma(x_1)\mathrm{u}(x_1)]$ then from (89)

$$\deg[\tau(x_1)] \leq \deg[\mathrm{c}(x_1)] - \deg[\mathrm{w}(x_1)] < \deg[\mathrm{u}(x_1)]$$

as long as $\deg[\mathrm{c}(x_1)] < \deg[\mathrm{u}(x_1)] + \deg[\mathrm{w}(x_1)]$ as stated. Otherwise if $\deg[\mathrm{c}(x_1)] < \deg[\sigma(x_1)\mathrm{u}(x_1)]$ (in which case the stated degree bound for $\mathrm{c}(x_1)$ also holds because of (83)) then from (89)

$$\deg[\tau(x_1)] = \deg[\sigma(x_1)\mathrm{u}(x_1)] - \deg[\mathrm{w}(x_1)] < \deg[\mathrm{u}(x_1)]$$

where the last inequality follows from (83). Thus (84) is proved. □

Multivariate Hensel Construction

We are now ready to develop the multivariate generalization of Hensel's Lemma. We pose the problem of finding multivariate polynomials $\mathrm{u}(x_1, \ldots, x_v)$, $\mathrm{w}(x_1, \ldots, x_v) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ which satisfy the congruence

$$\text{(90)} \quad \mathrm{a}(x_1, \ldots, x_v) - \mathrm{uw} \equiv 0 \pmod{p^l}$$

such that

$$\text{(91)} \quad \begin{cases} \mathrm{u}(x_1, \ldots, x_v) \equiv \mathrm{u}^{(1)}(x_1) \pmod{<\mathrm{I}, p^l>}; \\ \mathrm{w}(x_1, \ldots, x_v) \equiv \mathrm{w}^{(1)}(x_1) \pmod{<\mathrm{I}, p^l>}; \end{cases}$$

where $\mathrm{u}^{(1)}(x_1)$, $\mathrm{w}^{(1)}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ are given univariate polynomials which satisfy (90) modulo I. Here, p is a prime integer, l is a positive integer, $\mathrm{a}(x_1, \ldots, x_v) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ is a given multivariate polynomial, and $\mathrm{I} = <x_2 - \alpha_2, \ldots, x_v - \alpha_v>$ is the kernel of a multivariate evaluation homomorphism. Denoting the desired solution polynomials by $\bar{\mathrm{u}}$ and $\bar{\mathrm{w}}$, we will develop these solutions in their I-adic forms:

$$\text{(92)} \quad \begin{cases} \bar{\mathrm{u}} = \mathrm{u}^{(1)} + \Delta\mathrm{u}^{(1)} + \Delta\mathrm{u}^{(2)} + \cdots + \Delta\mathrm{u}^{(d)}; \\ \bar{\mathrm{w}} = \mathrm{w}^{(1)} + \Delta\mathrm{w}^{(1)} + \Delta\mathrm{w}^{(2)} + \cdots + \Delta\mathrm{w}^{(d)} \end{cases}$$

where d is the maximum total degree of any term in $\bar{\mathrm{u}}$ or $\bar{\mathrm{w}}$, $\mathrm{u}^{(1)} = \phi_{\mathrm{I}}(\bar{\mathrm{u}})$, $\mathrm{w}^{(1)} = \phi_{\mathrm{I}}(\bar{\mathrm{w}})$, and $\Delta\mathrm{u}^{(k)}, \Delta\mathrm{w}^{(k)} \in \mathrm{I}^k$, for $k = 1, 2, \cdots, d$. From section 6.1 we know that the I-adic representation of the polynomial $\bar{\mathrm{u}}$ is precisely the multivariate Taylor series representation (14). The k-th correction term $\Delta\mathrm{u}^{(k)} \in I^k$ is the term in (14) of total degree k with respect to I and it is represented by k nested summations in the form:

$$\text{(93)}\quad \Delta u^{(k)} = \sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} u_{\mathbf{i}}(x_1)(x_{i_1} - \alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k})$$

where $\mathbf{i} = (i_1, \cdots, i_k)$ is a vector subscript and $u_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$. Similarly, in the I-adic representation of $\overline{w}$ the k-th correction term takes the form

$$\text{(94)}\quad \Delta w^{(k)} = \sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} w_{\mathbf{i}}(x_1)(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k})$$

where $w_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$.

Our problem now is to compute, for each $k = 1, 2, \ldots, d$, the k-th correction terms $\Delta u^{(k)}$, $\Delta w^{(k)}$ in (92). Let $u^{(k)}, w^{(k)}$ denote the order-k I-adic approximations to $\overline{u}, \overline{w}$ given by the first k terms in (92). Letting $F(u, w)$ denote the left-hand-side of (90), Newton's iteration for solving $F(u, w) = 0$ in $\mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ takes the form of the congruence equation

$$\text{(95)}\quad w^{(k)}\Delta u^{(k)} + u^{(k)}\Delta w^{(k)} \equiv a(x_1, \ldots, x_v) - u^{(k)}w^{(k)} \pmod{<I^{k+1}, p^l>}$$

which must be solved for the correction terms $\Delta u^{(k)}, \Delta w^{(k)} \in I^k$ and then

$$u^{(k+1)} = u^{(k)} + \Delta u^{(k)},\ \ w^{(k+1)} = w^{(k)} + \Delta w^{(k)}$$

will be order-$(k+1)$ I-adic approximations to $\overline{u}, \overline{w}$. Now since $u^{(k)}, w^{(k)}$ are order-k I-adic approximations we have

$$a(x_1, \ldots, x_v) - u^{(k)}w^{(k)} \in I^k$$

and therefore the right-hand-side of (95) may be expressed in the form

$$\sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} c_{\mathbf{i}}(x_1)(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k})$$

for some coefficients $c_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$. Substituting into (95) this nested-summation representation for the right-hand-side and also the nested-summation representations (93)-(94), the congruence (95) may be solved by separately solving the following congruence for each term in the I-adic representation:

$$w^{(k)}u_{\mathbf{i}}(x_1) + u^{(k)}w_{\mathbf{i}}(x_1) \equiv c_{\mathbf{i}}(x_1) \pmod{<I, p^l>}$$

yielding the desired I-adic coefficients $u_{\mathbf{i}}(x_1), w_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ which define the correction terms (93)-(94). Now note that since this is a congruence modulo I we may apply the evaluation homomorphism ϕ_I to the left-hand-side, yielding the following polynomial diophantine equation to solve in the ring $\mathbf{Z}_{p^l}[x_1]$ for each term in the I-adic representation:

$$\text{(96)}\quad w^{(1)}(x_1)u_{\mathbf{i}}(x_1) + u^{(1)}(x_1)w_{\mathbf{i}}(x_1) \equiv c_{\mathbf{i}}(x_1) \pmod{p^l}$$

where $u^{(1)}(x_1)$, $w^{(1)}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ are the given polynomials in the problem (90)-(91) being solved. Theorem 6.5 states the conditions under which the congruence (96) has a unique solution $u_{\mathbf{i}}(x_1)$, $w_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$.

The following theorem formally proves the validity of the above method which is known as the *multivariate Hensel construction*.

Theorem 6.6. *Multivariate Hensel construction.*

Let p be a prime integer, let l be a positive integer, and let $a(x_1, \ldots, x_v) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ be a given multivariate polynomial. Let $I = <x_2-\alpha_2, \ldots, x_v-\alpha_v>$ be the kernel of a multivariate evaluation homomorphism such that $p \,/\, \mathrm{Lc}[\phi_I(a(x_1, \ldots, x_v))]$. Let $u^{(1)}(x_1)$, $w^{(1)}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ be two univariate polynomials which satisfy the following conditions:

$$(i)\quad a(x_1, \ldots, x_v) \equiv u^{(1)}(x_1)w^{(1)}(x_1) \pmod{<I, p^l>};$$

(*ii*) $\phi_p(\mathrm{u}^{(1)}(x_1))$ and $\phi_p(\mathrm{w}^{(1)}(x_1))$ are relatively prime polynomials in $\mathbf{Z}_p[x_1]$.

Then for any integer $k \geq 1$ there exist multivariate polynomials $\mathrm{u}^{(k)}, \mathrm{w}^{(k)} \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v] / \mathrm{I}^k$ such that

$$\text{(97)} \quad \mathrm{a}(x_1, \ldots, x_v) \equiv \mathrm{u}^{(k)}\mathrm{w}^{(k)} \pmod{<\mathrm{I}^k, p^l>}$$

and

$$\text{(98)} \quad \begin{cases} \mathrm{u}^{(k)} \equiv \mathrm{u}^{(1)}(x_1) \pmod{<\mathrm{I}, p^l>}; \\ \mathrm{w}^{(k)} \equiv \mathrm{w}^{(1)}(x_1) \pmod{<\mathrm{I}, p^l>}. \end{cases}$$

Proof: The proof is by induction on k. The case $k = 1$ is given by condition (i). Assume for $k \geq 1$ that we have $\mathrm{u}^{(k)}, \mathrm{w}^{(k)} \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v] / \mathrm{I}^k$ satisfying (97) and (98). Define

$$\text{(99)} \quad \mathrm{e}^{(k)} = \mathrm{a}(x_1, \ldots, x_v) - \mathrm{u}^{(k)}\mathrm{w}^{(k)} \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v] / \mathrm{I}^{k+1}$$

and from (97) it follows that $\mathrm{e}^{(k)} \in \mathrm{I}^k$. Define the polynomial coefficients $\mathrm{c}_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ by expressing $\mathrm{e}^{(k)}$ in I-adic form:

$$\text{(100)} \quad \mathrm{e}^{(k)} = \sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} \mathrm{c}_{\mathbf{i}}(x_1)(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k}).$$

By Theorem 6.5 (noting that since $p \nmid \mathrm{Lc}[\phi_{\mathrm{I}}(\mathrm{a}(x_1, \ldots, x_v))]$, condition (i) implies the first condition of Theorem 6.5 and condition (ii) is the second required condition), we can find unique polynomials $\sigma_{\mathbf{i}}(x_1), \tau_{\mathbf{i}}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ such that

$$\text{(101)} \quad \sigma_{\mathbf{i}}(x_1)\mathrm{u}^{(1)}(x_1) + \tau_{\mathbf{i}}(x_1)\mathrm{w}^{(1)}(x_1) \equiv \mathrm{c}_{\mathbf{i}}(x_1) \pmod{p^l}$$

and

$$\text{(102)} \quad \deg[\sigma_{\mathbf{i}}(x_1)] < \deg[\mathrm{w}^{(1)}(x_1)],$$

for each index $\mathbf{i}$ which appears in the I-adic representation of $\mathrm{e}^{(k)}$. Then by defining

$$\text{(103)} \quad \begin{cases} \mathrm{u}^{(k+1)} = \mathrm{u}^{(k)} + \displaystyle\sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} \tau_{\mathbf{i}}(x_1)(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k}); \\ \mathrm{w}^{(k+1)} = \mathrm{w}^{(k)} + \displaystyle\sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} \sigma_{\mathbf{i}}(x_1)(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k}) \end{cases}$$

we have by performing multiplication modulo I^{k+1}:

$$\mathrm{u}^{(k+1)}\mathrm{w}^{(k+1)} \equiv \mathrm{u}^{(k)}\mathrm{w}^{(k)} +$$

$$\sum_{i_1=2}^{v} \sum_{i_2=i_1}^{v} \cdots \sum_{i_k=i_{k-1}}^{v} (\sigma_{\mathbf{i}}(x_1)(\mathrm{u}^{(1)}(x_1) + \tau_{\mathbf{i}}(x_1)\mathrm{w}^{(1)}(x_1))(x_{i_1}-\alpha_{i_1})(x_{i_2}-\alpha_{i_2}) \cdots (x_{i_k}-\alpha_{i_k}$$

$$\pmod{<\mathrm{I}^{k+1}, p^l>}$$

$$\equiv \mathrm{u}^{(k)}\mathrm{w}^{(k)} + \mathrm{e}^{(k)} \pmod{<\mathrm{I}^{k+1}, p^l>}, \text{ by (101) and (100)}$$

$$\equiv \mathrm{a}(x_1, \ldots, x_v) \pmod{<\mathrm{I}^{k+1}, p^l>}, \text{ by (99).}$$

Thus (97) holds for $k + 1$. Also, from (103) it is clear that

$$u^{(k+1)} \equiv u^{(k)} \pmod{<I,p^l>};$$
$$w^{(k+1)} \equiv w^{(k)} \pmod{<I,p^l>}$$

and therefore since (98) holds for k it also holds for $k + 1$. □

The multivariate Hensel construction of Theorem 6.6 generates unique factors $u^{(k)}, w^{(k)}$ in the case where $a(x_1, \ldots, x_v)$ is "monic with respect to x_1"; i.e., in the case where the coefficient in $a(x_1, \ldots, x_v)$ of $x_1^{d_1}$ is 1, where d_1 denotes the degree in x_1. For in such a case, we may choose $u^{(k)}$ and $w^{(k)}$ each to be "monic with respect to x_1" and uniqueness follows just as in the univariate case. This result is stated as the following corollary, whose proof is a straightforward generalization of the proof of the Corollary to Theorem 6.2 and is omitted.

Corollary to Theorem 6.6. *Uniqueness of the Multivariate Hensel construction.*

In Theorem 6.6, if the given polynomial $a(x_1, \ldots, x_v) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v]$ has leading coefficient 1 with respect to the indeterminant x_1 and correspondingly if the univariate factors $u^{(1)}(x_1), w^{(1)}(x_1) \in \mathbf{Z}_{p^l}[x_1]$ are chosen to be monic, then for any integer $k \geq 1$ conditions (97)-(98) uniquely determine factors $u^{(k)}, w^{(k)} \in \mathbf{Z}_{p^l}[x_1, \ldots, x_v] / I^k$ which each have leading coefficient 1 with respect to the indeterminate x_1.

6.7. THE MULTIVARIATE HENSEL LIFTING ALGORITHM

The algorithm which follows directly from Theorem 6.6 has some deficiencies which must be corrected before we can present an efficient algorithm for the multivariate Hensel construction. One such deficiency is the *leading coefficient problem*. For this problem, we will adopt a solution which is directly analogous to the solution developed in section 6.5 and implemented in Algorithm 6.1 for the univariate case. Less obvious are the efficiency problems associated with the construction presented in the proof of Theorem 6.6. This construction exhibits poor performance in cases where some of the evaluation points α_j are nonzero and this problem is sometimes called the *bad-zero problem*. We will examine this problem now.

The Bad-Zero Problem

The source of the performance problems is the requirement in the proof of Theorem 6.6 that the error $e^{(k)}$ must be expressed in the I-adic form (100). This step can lead to very large intermediate expression swell resulting in an exponential cost function for the algorithm. The following example will serve to illustrate.

Example 6.10.

Let $p = 5$, $l = 1$,

$$a(x,y,z) = x^2y^4z - xy^9z^2 + xyz^3 + 2x - y^6z^4 - 2y^5z,$$

and $I = <y-1, z-1>$. Noting that

$$a(x,y,z) \equiv x^2+2x+2 \pmod{<I,5>}$$

we have

$$a(x,y,z) \equiv (x-2)(x-1) \pmod{<I,5>}.$$

Choosing $u^{(1)}(x) = x-2$ and $w^{(1)}(x) = x-1$, the conditions of Theorem 6.6 are satisfied. Since $a(x,y,z)$ is not monic we might expect the Hensel construction to fail to produce true factors in $\mathbf{Z}[x,y,z]$, but in this example the factor $w(x,y,z)$ is monic so the Hensel construction will succeed even though we are ignoring the leading coefficient problem.

The effect of representing the error at each step of the iteration in I-adic form can be seen by considering the I-adic form of $a(x,y,z)$:

$$\begin{aligned}
a(x,y,z) \equiv\ & (x^2+2x+2) \\
& +(-x^2-1)(y-1)+(x^2+x-1)(z-1) \\
& +(x^2-x)(y-1)^2+(-x^2+1)(y-1)(z-1) + (2x-1)(z-1)^2 \\
& +(-x^2+x)(y-1)^3+(x^2-2x)(y-1)^2(z-1)+(-x-1)(y-1)(z-1)^2+(x+1)(z-1)^3 \\
& +(x^2-x)(y-1)^4+(-x^2+2x)(y-1)^3(z-1)+(-x)(y-1)^2(z-1)^2+(x+1)(y-1)(z-1)^3+(-1)(z- \\
& +(-x+2)(y-1)^5+(x^2-2x)(y-1)^4(z-1)+(x)(y-1)^3(z-1)^2+(-1)(y-1)(z-1)^4 \\
& +(x-1)(y-1)^6+(-2x-1)(y-1)^5(z-1)+(-x)(y-1)^4(z-1)^2 \\
& +(-x)(y-1)^7+(2x+1)(y-1)^6(z-1)+(-x-1)(y-1)^5(z-1)^2 \\
& +(x)(y-1)^8+(-2x)(y-1)^7(z-1)+(x-1)(y-1)^6(z-1)^2+(1)(y-1)^5(z-1)^3 \\
& +(-x)(y-1)^9+(2x)(y-1)^8(z-1)+(-x)(y-1)^7(z-1)^2+(1)(y-1)^6(z-1)^3+(-1)(y-1)^5(z-1) \\
& +(-2x)(y-1)^9(z-1)+(x)(y-1)^8(z-1)^2+(-1)(y-1)^6(z-1)^4 \\
& +(-x)(y-1)^9(z-1)^2 \pmod 5.
\end{aligned}$$

We see that the I-adic representation contains 38 terms compared with 6 terms in the original expanded representation (which is an I-adic representation with respect to the ideal $I = <y,z>$). The number of polynomial diophantine equations of the form (101) which must be solved is proportional to the number of terms in the I-adic form of $a(x,y,z)$.

Carrying out the Hensel construction for this example, the factors are developed in I-adic form as follows:

$$\begin{aligned}
u^{(7)} =\ & (x-2)+(-x+1)(y-1)+(x-2)(z-1)+(x)(y-1)^2+(-x-2)(y-1)(z-1)+(-2)(z-1)^2 \\
& +(-x)(y-1)^3+(x)(y-1)^2(z-1)+(-2)(y-1)(z-1)^2+(1)(z-1)^3 \\
& +(x)(y-1)^4+(-x)(y-1)^3(z-1)+(1)(y-1)(z-1)^3+(x)(y-1)^4(z-1); \\
w^{(7)} =\ & (x-1)+(-1)(z-1)+(-1)(y-1)^5+(-1)(y-1)^5(x-1).
\end{aligned}$$

Expressing these factors in expanded form (and noting that the coefficient arithmetic is being done modulo 5), we have

$$u^{(7)} \equiv xy^4z + yz^3 + 2 \pmod 5;$$

$$w^{(7)} \equiv x - y^5z \pmod 5.$$

At this point the iteration can be halted because

$$e^{(7)} = a(x,y,z) - u^{(7)}w^{(7)} = 0.$$

Again note that there are many more terms in the I-adic representation of the factors than in the expanded representation. □

It is clear from Example 6.10 that the use of nonzero evaluation points can cause a severe case of intermediate expression swell. However it is not always possible to choose the evaluation points to be zero because in the applications of the Hensel construction (see chapter 7), a necessary condition is that the leading coefficient must not vanish under the evaluation homomorphism. The original implementation of the multivariate Hensel construction (the EZ algorithm) degraded significantly on problems requiring non-zero evaluation points.

One method of dealing with the I-adic representation in an implementation of the multivariate Hensel construction is to initially perform the changes of variables:

$$x_j \leftarrow x_j + \alpha_j,\ 2 \le j \le \nu,$$

if the ideal is $I = <x_2 - \alpha_2, \ldots, x_\nu - \alpha_\nu>$. The required I-adic representation is then a straightforward expanded representation based on the new ideal $<x_2, \ldots, x_v>$. However it is important to note that this method suffers from the problem of intermediate expression swell exactly as exhibited in Example 6.10. For in the original polynomial $a(x,y,z)$ in Example 6.10, the result of performing the changes of variables:

$$y \leftarrow y + 1; \quad z \leftarrow z + 1$$

and then expanding, is precisely the 38-term form of $a(x,y,z)$ displayed in the example, with $y - 1$ replaced by y and $z - 1$ replaced by z.

An improvement to the algorithm can be obtained by avoiding the changes of variables (or any other explicit representation of the I-adic form) as follows. At iteration step k let $e^{(k)}$ be represented as a multivariate polynomial in expanded form. It is desired to compute the coefficients $c_{\mathbf{i}}(x_1)$ appearing in (100), the I-adic representation of $e^{(k)}$, for all order-k vector indices

$$\mathbf{i} = (i_1, i_2, \cdots, i_k).$$

Noting that some of the indices in the vector $\mathbf{i}$ may be repeated, let the term in $e^{(k)}$ corresponding to a particular vector index $\mathbf{i}$ be of the form

$$c_{\mathbf{i}}(x_1)(x_{j_1} - \alpha_{j_1})^{n_1}(x_{j_2} - \alpha_{j_2})^{n_2} \cdots (x_{j_m} - \alpha_{j_m})^{n_m}$$

where all factors appearing here are distinct. Then the coefficient $c_{\mathbf{i}}(x_1)$ can be computed directly from the expanded representation of $e^{(k)}$ by using the following differentiation formula:

$$\textbf{(104)} \quad c_{\mathbf{i}}(x_1) = \frac{1}{n_1! \cdots n_m!} \phi_I \left(\left(\frac{\partial}{\partial x_{j_1}} \right)^{n_1} \cdots \left(\frac{\partial}{\partial x_{j_m}} \right)^{n_m} e^{(k)} \right).$$

This leads to an organization of the main iteration loop of the Hensel construction which can be expressed as follows (where d is the maximum total degree with respect to the indeterminates $x_2, \ldots, x_v$ over all terms in the input polynomial $a(x_1, \ldots, x_v)$):

```
for $k$ from 1 to $d$ while $e^{(k)} \neq 0$ do
   for each order-$k$ vector index $\mathbf{i} = (i_1, \cdots, i_k)$ such that $2 \le i_1 \le i_2 \le \cdots \le i_k \le v$ do
      Calculate $c_{\mathbf{i}}(x_1)$ using (104);
      Solve equation (101) for $\sigma_{\mathbf{i}}(x_1)$ and $\tau_{\mathbf{i}}(x_1)$;
      Update $u^{(k)}$ and $w^{(k)}$ according to (103);
   od
   Update $e^{(k)}$;
od.
```

This organization of the iteration loop is in contrast to the organization which follows more directly from the proof of Theorem 6.6, using the "changes of variables" concept, as follows: (In both of the above program segments, it is understood that

$$e^{(k)} = a(x_1, \ldots, x_v) - u^{(k)}w^{(k)}$$

computed in $\mathbf{Z}[x_1, \ldots, x_v]$ in expanded form).

A careful examination of these two organizations of the iteration loop shows that neither one is fully satisfactory for dealing with sparse multivariate polynomials. Recall our observation at the beginning of this chapter that, in practice, multivariate polynomials are generally sparse and the advantage of the Hensel construction over Chinese remainder (interpolation) algorithms is the ability to take advantage of sparseness. In the approach which applies the changes of variables, there is potentially a serious loss of sparsity because the

```
Substitute $x_j \leftarrow x_j + \alpha_j$ $(2 \le j \le v)$ in a$(x_1, \ldots, x_v)$;
for $k$ from 1 to $d$ while $e^{(k)} \ne 0$ do
    for each term of total degree $k$ which appears in the expanded form of $e^k$ do
        Pick off the coefficient $c_{\mathbf{i}}(x_1)$;
        Solve equation (101) for $\sigma_{\mathbf{i}}(x_1)$ and $\tau_{\mathbf{i}}(x_1)$;
        Update $u^{(k)}$ and $w^{(k)}$ according to (103);
    od
    Update $e^{(k)}$;
od.
Substitute $x_j \leftarrow x_j - \alpha_j$ $(2 \le j \le v)$ in $u^{(k)}$ and $w^{(k)}$.
```

representation of the polynomial a$(x_1, \ldots, x_v)$ after substituting the changes of variables can have many more terms than the original representation (see Example 6.10). Note, however, that after this substitution step, the iteration then goes on to perform calculations only for terms that *actually appear* in the expanded form of $e^{(k)}$. In contrast, in the approach which avoids the changes of variables but uses instead the differentiation formula (104), the inner **for**-loop iterates over *all possible* order-k vector indices $\mathbf{i} = (i_1, \cdots, i_k)$ and, in practice, a large proportion of the coefficients $c_{\mathbf{i}}(x_1)$ will be found to be zero. Since the differentiations and substitutions required by formula (104) can be performed relatively efficiently for polynomials (particularly if it is programmed to 'remember' computed derivatives since higher-order derivatives rely on lower-order derivatives) and since we would program the inner loop to check if $c_{\mathbf{i}}(x_1) = 0$ and avoid any additional work in that case, the method using formula (104) is generally preferable. However, the overhead of calculating $c_{\mathbf{i}}(x_1)$ for all possible choices of the vector index $\mathbf{i}$ is significant and the cost of this overhead grows exponentially in the number of variables, independently of the sparsity of the polynomials. In particular, note that in the (relatively common) case where all of the evaluation points are zero the method using (104) will be much more costly than the direct approach.

Polynomial Diophantine Equations in $\mathbf{Z}_{p^l}[x_1, \ldots, x_j]$

A significantly more efficient organization of the multivariate Hensel construction was developed by Paul Wang [Wang78] and he called it the EEZ (Enhanced EZ) algorithm. The main feature of the new algorithm is that it uses a variable-by-variable approach to avoid the "exponential overhead" discussed above. In the context of Figure 6.2, the multivariate Hensel construction lifting the solution from $\mathbf{Z}_{p^l}[x_1]$ to $\mathbf{Z}[x_1, \ldots, x_v]$ is replaced by a sequence of $v-1$ single-variable Hensel constructions to lift the solution

from $\mathbf{Z}_{p^l}[x_1]$ to $\mathbf{Z}_{p^l}[x_1, x_2]$;

from $\mathbf{Z}_{p^l}[x_1, x_2]$ to $\mathbf{Z}_{p^l}[x_1, x_2, x_3]$;

.

.

.

from $\mathbf{Z}_{p^l}[x_1, \ldots, x_{v-1}]$ to $\mathbf{Z}_{p^l}[x_1, \ldots, x_{v-1}, x_v]$.

(As usual, p^l is chosen large enough so that the final solution over the ring $\mathbf{Z}_{p^l}$ is equated with the desired solution over $\mathbf{Z}$).

Recall that the basic computation to be performed in applying a step of a Hensel iteration is to solve a polynomial diophantine equation in the "base domain". For the univariate Hensel construction in Figure 6.2, the "base domain" is $\mathbf{Z}_p[x_1]$ and Theorem 2.6 gives a method for solving the polynomial diophantine equations. For the "base domain" $\mathbf{Z}_{p^l}[x_1]$, we developed a method in Theorem 6.5 for solving the polynomial diophantine equations. In order to carry out the variable-by-variable Hensel construction, we need a method for solving polynomial diophantine equations in multivariate "base domains" $\mathbf{Z}_{p^l}[x_1, \ldots, x_j]$ and we turn now to the development of such a method. Just as in the proof of Theorem 6.5, we will apply Newton's iteration to the problem and indeed we will employ a variable-by-variable technique for solving this sub-problem.

The polynomial diophantine equation to be solved is to find multivariate polynomials $\sigma_j(x_1, \ldots, x_j)$, $\tau_j(x_1, \ldots, x_j) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_j]$ such that

$$\text{(105)} \quad \sigma_j(x_1, \ldots, x_j)\mathrm{u}(x_1, \ldots, x_j) + \tau_j(x_1, \ldots, x_j)\mathrm{w}(x_1, \ldots, x_j) \equiv \mathrm{c}(x_1, \ldots, x_j) \pmod{<\mathrm{I}_j^{d+1}, p^l>}$$

where $\mathrm{I}_j = <x_2 - \alpha_2, \ldots, x_j - \alpha_j>$, d is the maximum total degree of the solution polynomials with respect to the indeterminates $x_2, \ldots, x_j$, and $\mathrm{u}(x_1, \ldots, x_j)$, $\mathrm{w}(x_1, \ldots, x_j)$, $\mathrm{c}(x_1, \ldots, x_j) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_j]$ are given polynomials with $\phi_{<\mathrm{I},p>}(\mathrm{u}(x_1, \ldots, x_j))$ and $\phi_{<\mathrm{I},p>}(\mathrm{w}(x_1, \ldots, x_j))$ relatively prime polynomials in the Euclidean domain $\mathbf{Z}_p[x_1]$. The equation to which we will apply Newton's iteration is

$$\mathrm{G}(\sigma_j, \tau_j) = \sigma_j \mathrm{u}(x_1, \ldots, x_j) + \tau_j \mathrm{w}(x_1, \ldots, x_j) - \mathrm{c}(x_1, \ldots, x_j) = 0.$$

Choosing the particular variable x_j for lifting and proceeding as in previous sections, if we have the order-k approximations $\sigma_j^{(k)}, \tau_j^{(k)}$ satisfying

$$\mathrm{G}(\sigma_j^{(k)}, \tau_j^{(k)}) \equiv 0 \pmod{<(x_j - \alpha_j)^k, \mathrm{I}_{j-1}^{d+1}, p^l>}$$

and if we obtain correction terms $\Delta\sigma_j^{(k)}, \Delta\tau_j^{(k)}$ which satisfy the equation

$$\text{(106)} \quad \mathrm{G}_{\sigma_j}(\sigma_j^{(k)}, \tau_j^{(k)})\Delta_j^{(k)} + \mathrm{G}_{\tau_j}(\sigma_j^{(k)}, \tau_j^{(k)})\Delta\tau_j^{(k)} \equiv -\mathrm{G}(\sigma_j^{(k)}, \tau_j^{(k)}) \pmod{<(x_j - \alpha_j)^{k+1}, \mathrm{I}_{j-1}^{d+1}, p^l>}.$$

then

$$\sigma_j^{(k+1)} = \sigma_j^{(k)} + \Delta\sigma_j^{(k)}, \quad \tau_j^{(k+1)} = \tau j^{(k)} + \Delta\tau_j^{(k)}$$

will be order-$(k+1)$ approximations satisfying

$$\mathrm{G}(\sigma_j^{(k+1)}, \tau_j^{(k+1)}) \equiv 0 \pmod{<(x_j - \alpha_j)^{k+1}, \mathrm{I}_{j-1}^{d+1}, p^l>}.$$

Writing the correction terms in the form

$$\Delta\sigma_j^{(k)} = \mathrm{s}_{j,k}(x_1, \cdots, x_{j-1})(x_j - \alpha_j)^k,$$
$$\Delta\tau_j^{(k)} = \mathrm{t}_{j,k}(x_1, \cdots, x_{j-1})(x_j - \alpha_j)^k$$

where $\mathrm{s}_{j,k}(x_1, \ldots, x_{j-1})$, $\mathrm{t}_{j,k}(x_1, \ldots, x_{j-1}) \in \mathbf{Z}_{p^l}[x_1, \ldots, x_{j-1}]$, substituting for the partial derivatives, and dividing through by $(x_j - \alpha_j)^k$, equation (106) becomes

$$\text{(107)} \quad \mathrm{u}(x_1, \ldots, x_j)\mathrm{s}_{j,k}(x_1, \ldots, x_{j-1}) + \mathrm{w}(x_1, \ldots, x_j)\mathrm{t}_{j,k}(x_1, \cdots, x_{j-1})$$
$$\equiv \frac{\mathrm{c}(x_1, \ldots, x_j) - \sigma_j^{(k)}\mathrm{u}(x_1, \ldots, x_j) - \tau_j^{(k)}\mathrm{w}(x_1, \ldots, x_j)}{(x_j - \alpha_j)^k} \quad (\mathrm{mod} <(x_j - \alpha_j), \mathrm{I}_{j-1}^{d+1}$$

(Note that $\mathrm{I}_{j-1} = <x_2 - \alpha_2, \ldots, x_{j-1} - \alpha_{j-1}>$, the interpretation of I_1 is as the empty ideal, and note that the above development has assumed $j > 1$ since if $j = 1$ then the solution of the polynomial diophantine equation (105) is given by Theorem 6.5).

We thus have a recursive algorithm for solving the polynomial diophantine equation (105). The order-1 approximations $\sigma_j^{(1)}, \tau_j^{(1)}$ with respect to the ideal $<x_j - \alpha_j>$ are obtained by solving equation (105) modulo $<x_j - \alpha_j>$ - i.e., by solving the $(j - 1)$-variable problem

$$\sigma_{j-1}(x_1, \ldots, x_{j-1})\mathrm{u}(x_1, \ldots, x_{j-1}, \alpha_j) + \tau_{j-1}(x_1, \ldots, x_{j-1})\mathrm{w}(x_1, \ldots, x_{j-1}, \alpha_j)$$
$$\equiv \mathrm{c}(x_1, \ldots, x_{j-1}, \alpha_j) \pmod{<\mathrm{I}_{j-1}^{d+1}, p^l>}$$

and then setting

$$\sigma_j^{(1)} = \sigma_{j-1};\ \tau_j^{(1)} = \tau_{j-1}.$$

Then for $k = 1,2,\ldots,d$, we solve equation (107) which, noting that it is to be solved modulo $<x_j - \alpha_j>$, takes the form of the $(j-1)$-variable problem

$$\mathrm{u}(x_1, \ldots, x_{j-1}, \alpha_j)\mathrm{s}_{j,k}(x_1, \ldots, x_{j-1}) + \mathrm{w}(x_1, \ldots, x_{j-1}, \alpha_j)\mathrm{t}_{j-k}(x_1, \ldots, x_{j-1})$$
$$\equiv e_k(x_1, \ldots, x_{j-1}) \pmod{<\mathrm{I}_{j-1}^{d+1}, p^l>}$$

where $e_k(x_1, \ldots, x_{j-1})$ denotes the coefficient of $(x_j - \alpha_j)^k$ in the $<x_j - \alpha_j>$-adic representation of the polynomial

$$e(x_1, \ldots, x_j) = \mathrm{c}(x_1, \ldots, x_j) - \sigma_j^{(k)}\mathrm{u}(x_1, \ldots, x_j) - \tau_j^k\mathrm{w}(x_1, \ldots, x_j).$$

The base of the recursion is the univariate polynomial diophantine equation in $\mathbf{Z}_{p^l}[x_1]$ which can be solved by the method of Theorem 6.5.

This recursive algorithm for solving multivariate polynomial diophantine equations is presented as Algorithm 6.2. The conditions which must be satisfied by the input polynomials are the conditions required by Theorem 6.5 for the univariate case at the base of the recursion. Note that the solution of equation (105) computed by the algorithm satisfies the degree constraint

$$\partial_1[\sigma_j(x_1, \ldots, x_j)] < \partial_1[\mathrm{w}(x_1, \ldots, x_j)]$$

(where ∂_1 is the "degree in x_1" function) since the solution of the univariate case of (105) satisfies such a constraint (by Theorem 6.5), as does the solution of the univariate case of equation (107) which defines the correction terms, leading by induction to the general result.

Maple Implementation of Algorithm 6.2

```
# diophant: Multivariate polynomial diophantine equations.
#       Algorithm 6.2 in Geddes textbook.
#
#       Solve in the domain Z/<p^k>[x1,...,xv] the (multivariate)
#       polynomial diophantine equation
#               sigma * a  +  tau * b  ===  c   ( mod <I^(d+1), p^k> )
#       for  sigma,tau  satisfying degree(sigma,x1) < degree(b,x1) .
#
#    Necessary conditions:
#       a mod <I,p>  and  b mod <I,p>  must be relatively prime in Zp[x1];
#       lcoeff(b mod I) must be a unit in the ring Z/<p^k> .
#
# INPUT:
#       a, b, c - polynomials in the domain Z/<p^k>[x1,...,xv];
#       I - list of equations [x2=alpha2, . . . , xv=alphav]
#               (possibly null, in which case it is a univariate problem)
#               representing an evaluation homomorphism --
#               mathematically, we view it as the ideal
#                       I = <x2-alpha2, . . . , xv-alphav> ;
#       d - a nonnegative integer specifying the maximum total degree
#               with respect to x2,...,xv of the desired result (the
#               value is irrelevant if I is null);
#       p - a prime integer;
#       k - (optional) a positive integer specifying that the coefficient
#               arithmetic is to be performed modulo p^k, and if this
#               parameter is not specified then k = 1 .
#
# OUTPUT:
#       The value returned is the list  [sigma, tau] .
```

```
diophant := proc (a, b, c, I, d, p, k)
        local modulus,n,eqn,monomial,anew,bnew,cnew,Inew,sigma,tau,e,
                m,cm,deltas,deltat,x1,terms,i;
        if nargs<6 or nargs>7 or not type(I,list) or
           not type(d,integer) or d<0 or not type(p,integer) or p<2 or
           nargs=7 and (not type(k,integer) or k<1) then
                ERROR(`wrong number (or type) of arguments in diophant`)
        fi;
        if nargs < 7 then modulus := p else modulus := p^k fi;

        n := nops(I);
        if n > 0 then
                eqn := op(n,I);  monomial := op(1,eqn) - op(2,eqn);
                anew := subs(eqn,a);  bnew := subs(eqn,b);
                cnew := subs(eqn,c);
                Inew := [op(1..n-1, I)];
                diophant(anew, bnew, cnew, Inew, args[5..nargs]);
                sigma := op(1,");  tau := op(2,"");
                e := mods(c - expand(sigma * a) - expand(tau * b), modulus);
                for m to d while e <> 0 do
                   cm := coeftayl(e, eqn, m);
                   if cm <> 0 then
                        diophant(anew, bnew, cm, Inew, args[5..nargs]);
                        deltas := expand(op(1,")*monomial^m);
                        deltat := expand(op(2,"")*monomial^m);
                        sigma := sigma + deltas;  tau := tau + deltat;
                        e := e - expand(deltas * a) - expand(deltat * b);
                        e := mods(e, modulus)
                   fi
                od
        else
                # Univariate case.
                indets(a) + indets(b) + indets(c);
                if nops(")=1 then x1 := op(") else
                        ERROR(`invalid indeterminates in procedure diophant`)
                fi;

                # Method:  For each power of x1, call procedure 'diophant1'
                # which will return remembered values after the first pass.
                if type(c,`+`) then terms := [op(c)] else terms := [c] fi;
                sigma := 0;  tau := 0;
                for i to nops(terms) do
                        op(i, terms);
                        m := degree(", x1);
                        cm := lcoeff("");
                        diophant1(a, b, x1, m, args[6..nargs]);
                        sigma := sigma + op(1,") * cm;
                        tau := tau + op(2,"") * cm
                od
        fi;
        [mods(sigma, modulus), mods(tau, modulus)]
end;
```

```
# diophant1: Solve in the domain Z/<p^k>[x] the polynomial diophantine
#        equation
#                sigma * a  +  tau * b  ==  x^m  (mod p^k)
#        for  sigma,tau  satisfying  degree(sigma,x) < degree(b,x) .
#        Necessary conditions:
#                a mod p, b mod p  must be relatively prime in Zp[x];
#                lcoeff(b) must be a unit in the ring Z/<p^k> .
#
#        The sixth parameter is optional, and if it is not present then k = 1.
#
#        The value returned is the list  [sigma, tau] .

diophant1 := proc (a, b, x, m, p, k)
        local modulus,s,t;
        option remember;

        if nargs<5 or nargs>6 or not type(x,name) or not type(m,integer)
           or m<0 or not type(p,integer) or p<2 or
           nargs=6 and (not type(k,integer) or k<1) then
                ERROR(`wrong number (or type) of parameters in diophant1`)
        fi;
        if nargs < 6 then modulus := p else modulus := p^k fi;

        # Apply mgcdex to a,b in Z/<p^k>[x] yielding s,t such that
        #        s * a  +  t * b  ==  1  (mod p^k)
        # (if GCD(a mod p, b mod p) = 1 in Zp[x] ,  else error)
        # where degree(s,x) < degree(b,x) .

        mgcdex(a, b, args[5..nargs]);
        s := op(1,"); t := op(2,"");
        if op(3,""") <> 1 then
                ERROR(`polynomials in diophant1 are not relatively prime`)
        fi;

        # 'quorem' computes q, r such that
        #        x^m * s  =  b*q  +  r  (mod modulus)
        # with degree(r,x) < degree(b,x).
        # Then  sigma = r  and  tau = x^m*t + q*a .

        quorem(x, m, s, b, modulus);
        [ op(2, "),  mods(expand(x^m*t + op(1,")*a), modulus) ]
end;
```

```
# quorem:  Compute q, r such that
#                  x^m * s  =  b*q  +  r  (mod modulus)
#         with degree(r,x) < degree(b,x).
#         Use recursion on m, and termination is guaranteed by the assumption:
#                  degree(s,x) < degree(b,x).
#         The value returned is the list  [q, r].

quorem := proc (x, m, s, b, modulus)
         local q,r,const;
         option remember;

         if m + degree(s,x) < degree(b,x) then
                  [ 0, expand(x^m * s) ]
         else
                  # If  x^(m-1)*s = b*q + r  with degree(r,x) < degree(b,x)
                  # then  x^m*s = b*(x*q) + (x*r).
                  # CASE 1:  degree(x*r,x) < degree(b,x)  and we are finished.
                  # CASE 2:  degree(x*r,x) = degree(b,x)  so use the fact that
                  #          x^m*s = b*(x*q + const)  +  (x*r - const*b)
                  # where const is chosen to decrease the degree of the
                  # remainder, namely:  const = lcoeff(r)/lcoeff(b).

                  quorem(x, m-1, s, b, modulus);
                  q := op(1,");  r := op(2,"");
                  if degree(r,x) < degree(b,x)-1 then
                           [ expand(x*q), expand(x*r) ]
                  else
                           const := mods(lcoeff(r)/lcoeff(b), modulus);
                           [ expand(x*q) + const,
                                    mods(expand(x*r - const*b), modulus) ]
                  fi
         fi
end;


mgcdex := 'readlib('mgcdex')';
coeftayl := 'readlib('coeftayl')';
```

Maple Implementation of Algorithm 6.3

```
# Multivariate Hensel lifting algorithm.
#       Algorithm 6.3 in Geddes textbook.
#
# INPUT:
#       a - multivariate polynomial in Z[x1,x2,...,xv] which is
#               primitive as a polynomial in the special variable x1;
#       I - list of equations [x2=alpha2, x3=alpha3, . . . , xv=alphav]
#               representing the evaluation homomorphism used;
#               mathematically, we view it as the ideal
#                       I = <x2-alpha2, x3-alpha3, . . . , xv-alphav>
#               and the following condition must hold:
#                       lcoeff(a, x1) <> 0  (mod I) .
#       p - prime integer which does not divide lcoeff(a mod I);
#       l - positive integer such that 1/2*p^l bounds the magnitudes of
#               all integers appearing in a and in either of its factors
#               to be computed;
#       u,w - univariate polynomials in Z/<p^l>[x1] such that u mod p  and
#               w mod p  are relatively prime polynomials in Zp[x1], and
#                       a == u * w  (mod <I,p^l>) ;
#       gamma - (optional) a polynomial in Z[x2,...,xv] which is known to
#               be a multiple of lcoeff(U,x1), where U (see OUTPUT below)
#               is one of the factors of a in Z[x1,x2,...,xv] to be computed
#               -- gamma should be not larger than lcoeff(a,x1), and if
#               gamma is not specified then the value lcoeff(a,x1) is used.
#
# OUTPUT:
#       If there exist polynomials U,W in Z[x1,x2,...,xv] such that
#               a = U * W
#       and
#               U/lcoeff(U,x1) == u/lcoeff(u,x1)  (mod <I,p^l>)
#               W/lcoeff(W,x1) == w/lcoeff(w,x1)  (mod <I,p^l>)
#       (where the divisions here are in the ring of integers mod p^l)
#       then the list [U, W] will be the value returned.
#       Otherwise, the value returned will be @FAIL.
#
# Functions required:  replcoeff, coeftayl, diophant, content.
#
#                                               KOG (Oct./83)
```

```
hensel := proc (a,I,p,l,u,w,gamma)

local x,modulus,v,lca,G,A,U,W,j,eqn,alpha,i,maxdeg,U1,W1,lcU,lcW,e,k,c,
        sigma,tau,deltaU,deltaW,contU,q;

if nargs<6 or nargs>7 or not type(I,list) or has(map(type,I,'='),false) or
        not type(p,integer) or p<2 or not type(l,integer) or l<1 then
            ERROR('wrong number (or type) of parameters in hensel')
fi;

convert(map(indets,I), '+');
indets(a)+indets(u)+indets(w);
" - "";
if nops(")=1 then x[1] := op(") else
        ERROR('inconsistent indeterminates in hensel')
fi;
if degree(a,x[1]) <> degree(u,x[1])+degree(w,x[1]) then
        ERROR('inconsistent degrees in hensel')
fi;

# Define new polynomial A[v] and its factors modulo <I,p^l>.
modulus := p^l;
v := nops(I) + 1;
lca := lcoeff(a, x[1]);
if nargs < 7 then G := lca else G := gamma fi;
if G = 1 then A[v]:= a else A[v]:= expand(G*a) fi;
lcoeff(u);
if " = 1 then 1 else modp("**(-1), modulus) fi;
if G = 1 and " = 1 then U := u else
        U := mods(subs(op(I),G) * " * u, modulus)
fi;
lcoeff(w);
if " = 1 then 1 else modp("**(-1), modulus) fi;
if lca = 1 and " = 1 then W := w else
        W := mods(subs(op(I),lca) * " * w, modulus)
fi;

# Initialization for the multivariate iteration.
for j from v by -1 to 2 do
        eqn := I[j-1];
        A[j-1] := mods(subs(eqn, A[j]), modulus);
        x[j] := op(1,eqn);  alpha[j] := op(2,eqn)
od;
map( proc(x,e) degree(e,x) end,  {seq(x[i], i = 2..v)}, A[v] );
maxdeg := max(op("));
```

```
# Variable-by-variable Hensel iteration.
for j from 2 to v do
        U1 := U;  W1 := W;
        if G <> 1 then
                if j = v then
                        lcU := G;  lcW := lca
                else
                        lcU := mods( subs(I[j..v-1], G), modulus );
                        lcW := mods( subs(I[j..v-1], lca), modulus )
                fi;
                U := replcoeff(U, x[1], lcU);  W := replcoeff(W, x[1], lcW)
        fi;
        e := A[j] - expand(U*W);
        for k to degree(A[j], x[j]) while e <> 0 do
           c := coeftayl(e, x[j]=alpha[j], k);
           if c <> 0 then
                diophant(U1, W1, c, [op(1..j-2,I)], maxdeg, p, 1);
                sigma := "[1];  tau := ""[2];
                deltaU := expand(tau * (x[j]-alpha[j])^k);
                deltaW := expand(sigma * (x[j]-alpha[j])^k);
                e := mods(e - expand(deltaU*W) - expand(deltaW*U) -
                                        expand(deltaU*deltaW), modulus);
                U := mods(U + deltaU, modulus);
                W := mods(W + deltaW, modulus);
                if e <> 0 then
                   if deltaU = 0 then divide(A[j],U,'W')
                   elif deltaW = 0 then divide(A[j],W,'U')
                   fi;
                   if " = true then e := 0 fi
                fi
           fi
        od
od;

# Check termination status.
if A[v]-expand(U*W) = 0 then  # Factorization obtained -- remove contents
        if G <> 1 then
                contU := content(U, x[1]);
                divide(U, contU, 'U');
                divide(G, contU, 'q');  divide(W, q, 'W');
                RETURN( [U,W] )
                # Note that the value A[v]/G is the original input a
        else
                RETURN( [U,W] )
        fi
else  # No such factorization exists
        RETURN(@FAIL)
fi
end;

replcoeff := 'readlib('replcoeff')';
coeftayl := 'readlib('coeftayl')';
diophant := 'readlib('diophant')';
content := 'readlib('content')';
```