JOURNAL DE THÉORIE DES NOMBRES DE BORDEAUX

DAVID FORD
PASCAL LETARD

Implementing the Round Four maximal order algorithm

Journal de Théorie des Nombres de Bordeaux, tome 6, nº 1 (1994), p. 39-80

<http://www.numdam.org/item?id=JTNB_1994__6_1_39_0>

Journal de Théorie des Nombres de Bordeaux **6** *(1994), 39–80*

Implementing the Round Four maximal order algorithm

by DAVID FORD and PASCAL LETARD

1. Introduction

Given p, a rational prime, and f, a monic separable polynomial in $\mathbb{Z}_p[x]$, let $\xi = x + f\mathbb{Q}_p[x]$. Define $\mathcal{A}_f = \mathbb{Q}_p[x]/f\mathbb{Q}_p[x] = \mathbb{Q}_p[\xi]$, and let $\mathcal{O}_f$ be the maximal order in $\mathcal{A}_f$.

For $\theta \in \mathcal{A}_f$, let χ_θ denote the characteristic polynomial of θ and Δ_θ the discriminant of χ_θ. An element $\theta \in \mathcal{O}_f$ is *primary* if χ_θ is congruent modulo p to a power of a polynomial ν_θ, with ν_θ monic and irreducible mod p.

The Round Four algorithm constructs an integral basis for $\mathcal{O}_f$.

The algorithm has three distinct branches:

0. For $\alpha \in \mathcal{O}_f$ with $\Delta_\alpha \neq 0$, the Dedekind test (see [Cohen 1993]) applied to χ_α efficiently determines a basis for $\mathcal{D}_\alpha$, the coefficient ring of the p-radical of $\mathbb{Z}_p[\alpha]$, which is an order satisfying $\mathbb{Z}_p[\alpha] \subseteq \mathcal{D}_\alpha \subseteq \mathcal{O}_f$, with $\mathbb{Z}_p[\alpha] = \mathcal{D}_\alpha$ if and only if $\mathbb{Z}_p[\alpha] = \mathcal{O}_f$.

1. If a non-primary element of $\mathcal{O}_f$ is found then this element can be used to construct orthogonal idempotents and decompose the algebra as the direct sum of subalgebras of lower degree. An integral basis for the algebra can then be constructed from integral bases of the subalgebras.

2. Failing cases 0 and 1, for a given primary $\alpha \in \mathcal{O}_f$ with $\Delta_\alpha \neq 0$ a sequence of elements in $\mathcal{O}_f - \mathbb{Z}_p[\alpha]$ with increasing p-adic value can be constructed. This sequence can not be extended indefinitely without leading to case 0 or case 1.

Manuscrit reçu le 20 juillet 1993, version définitive le 3 juin 1994.

In what follows we give the theoretical underpinnings of the algorithm. Descriptions in detail of major routines (`maxord`, `decomp`, `nilord`, ...) of our MAPLE implementation are in Appendix I. A complete MAPLE listing appears as Appendix II. Experimental results are given in Appendix III.

2. Denominators

Let

$$f = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n$$

with $a_0 = 1$, and $t_j = \mathrm{Tr}(\xi^j)$, so that $t_0 = n$. We define

$$A = \begin{pmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} \\ t_1 & t_2 & t_3 & \cdots & t_n \\ t_2 & t_3 & t_4 & \cdots & t_{n+1} \\ \vdots & \vdots & \vdots & & \vdots \\ t_{n-1} & t_n & t_{n+1} & \cdots & t_{2n-2} \end{pmatrix}.$$

Using Newton's relations

$$\sum_{j=0}^{m} t_{m-j}a_j = (n-m)a_m \qquad (m \le n)$$

$$\sum_{j=0}^{n} t_{m-j}a_j = 0 \qquad (m > n)$$

it can be shown inductively for $r \ge 0$ that

$$\mathtt{rem}(x^r f', f, x) = \sum_{k=1}^{n} b_{r,k}x^{n-k}$$

where

$$b_{r,k} = \sum_{j=0}^{k-1} t_{r+k-1-j}a_j.$$

We define

$$B = \begin{pmatrix} b_{0,1} & b_{0,2} & b_{0,3} & \cdots & b_{0,n} \\ b_{1,1} & b_{1,2} & b_{1,3} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & b_{2,3} & \cdots & b_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ b_{n-1,1} & b_{n-1,2} & b_{n-1,3} & \cdots & b_{n-1,n} \end{pmatrix}.$$

We call a matrix in $\mathbb{Z}_p^{n\times n}$ a *lower identity* if it has the form

$$\begin{pmatrix} 1 & * & * & \cdots & * \\ 0 & 1 & * & \cdots & * \\ 0 & 0 & 1 & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

For the lower identity

$$C = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & a_0 & a_1 & \cdots & a_{n-2} \\ 0 & 0 & a_0 & \cdots & a_{n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & a_0 \end{pmatrix} = \begin{pmatrix} 1 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & 1 & a_1 & \cdots & a_{n-2} \\ 0 & 0 & 1 & \cdots & a_{n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

we have

$$AC = B.$$

In essence, B is the "interesting part" of the Sylvester matrix of f and f'. Construction of the row-reduced Hermite normal form of B gives $L, T, D \in \mathbb{Z}_p^{n\times n}$ with L unimodular, T a lower identity, and diagonal

$$D = \mathtt{diag}(p^{\delta_1}, p^{\delta_2}, \ldots, p^{\delta_n})$$

with $\delta_1 \leq \delta_2 \leq \cdots \leq \delta_n$, such that

$$LB = DT.$$

Definition. For $\alpha \in O_f$ with $\Delta_\alpha \neq 0$, the *reduced discriminant* of χ_α is p^{d_α}, given by

$$p^{d_\alpha}\mathbb{Z}_p = (\chi_\alpha \mathbb{Z}_p[x] + \chi'_\alpha \mathbb{Z}_p[x]) \cap \mathbb{Z}_p.$$

It is clear that the reduced discriminant $p^{d_\xi} = p^{\delta_n}$ can be obtained from the (n, n) entry of the p-adic Hermite normal form of the Sylvester matrix B defined above.

Now let

$$X = (1, \xi, \ldots, \xi^{n-1})$$

and define

$$\mathcal{C}_f = \{z \in \mathbb{Q}_p[\xi] : \mathrm{Tr}(z\mathbb{Z}_p[\xi]) \subseteq \mathbb{Z}_p\}.$$

Clearly $\mathcal{C}_f$ is a $\mathbb{Z}_p$-module containing $\mathcal{O}_f$, and, since $A = \big(\mathrm{Tr}(\xi^{j-1}\xi^{i-1})\big)$, the entries of XA^{-1} form a $\mathbb{Z}_p$-basis for $\mathcal{C}_f$.

There exist a lower identity S and diagonal

$$E = \mathtt{diag}(p^{-\epsilon_1}, p^{-\epsilon_2}, \ldots, p^{-\epsilon_n})$$

with $0 = \epsilon_1 \le \epsilon_2 \le \cdots \le \epsilon_n$, such that the entries of XSE form a $\mathbb{Z}_p$-basis for $\mathcal{O}_f$. Because $p^{\epsilon_1 + \cdots + \epsilon_n}$ is the $\mathbb{Z}_p$-module index of $\mathbb{Z}_p[\xi]$ in $\mathcal{O}_f$, we have

$$2(\epsilon_1 + \cdots + \epsilon_n) \le \delta_1 + \cdots + \delta_n.$$

Since $\mathcal{O}_f \subseteq \mathcal{C}_f$ there exists a non-singular $M \in \mathbb{Z}_p^{n\times n}$ for which

$$\begin{aligned} XSE &= XA^{-1}M \\ SE &= A^{-1}M \\ &= CT^{-1}D^{-1}LM \\ DTC^{-1}SE &= LM. \end{aligned}$$

Then $H = TC^{-1}S$ is a lower identity and $DHE = LM \in \mathbb{Z}_p^{n\times n}$. Since $(DHE)_{j,j} = p^{\delta_j - \epsilon_j}$, $1 \le j \le n$, it follows that

$$\delta_j - \epsilon_j \ge 0 \qquad (1 \le j \le n).$$

THEOREM. *The diagonal entries of the p-adic Smith (or Hermite) normal form of the Sylvester matrix of f and f' give bounds on the successive denominators of an integral basis of $\mathcal{A}_f$ given in Hermite normal form.*

COROLLARY. *If $\alpha \in \mathcal{O}_f$ and $\Delta_\alpha \ne 0$ then $p^{d_\alpha}\mathcal{O}_f \subseteq \mathbb{Z}_p[\alpha]$.*

3. Structural stability

Assume $d \ge 0$, $h \in \mathbb{Z}_p[x]$, h monic, satisfying

$$p^d\mathcal{O}_f \subseteq \mathbb{Z}_p[\xi], \quad p^d\mathcal{O}_h \subseteq \mathbb{Z}_p[\theta], \quad h \equiv f \pmod{p^{2d}\mathbb{Z}_p[x]}$$

where $\theta = x + h\mathbb{Q}_p[x]$. Let $\sigma : \mathcal{A}_f \to \mathcal{A}_h$ be the $\mathbb{Q}_p$-module isomorphism determined by $\sigma(\xi^{j-1}) = \theta^{j-1}$ $(1 \le j \le n)$. For all $\alpha, \beta \in \mathbb{Z}_p[\xi]$ we have

$$\sigma(\alpha\beta) \equiv \sigma(\alpha)\sigma(\beta) \pmod{p^{2d}\mathbb{Z}_p[\theta]}.$$

Now define $\widetilde{\mathcal{O}} = \sigma(\mathcal{O}_f)$. For any $\alpha, \beta \in \mathcal{O}_f$ we have $p^d\alpha, p^d\beta \in \mathbb{Z}_p[\xi]$, so that

$$\sigma(\alpha)\sigma(\beta) = \sigma(\alpha\beta) + p^{-2d}(\sigma(p^d\alpha)\sigma(p^d\beta) - \sigma((p^d\alpha)(p^d\beta))) \in \widetilde{\mathcal{O}} + \mathbb{Z}_p[\theta] = \widetilde{\mathcal{O}}.$$

Thus the $\mathbb{Z}_p$-module $\widetilde{\mathcal{O}}$ is closed under multiplication, and therefore is an order in A_h. By similar reasoning, $\sigma^{-1}(\mathcal{O}_h)$ is an order in A_f, so

$$\mathcal{O}_f = \sigma^{-1}(\widetilde{\mathcal{O}}) \subseteq \sigma^{-1}(\mathcal{O}_h) \subseteq \mathcal{O}_f.$$

Consequently, $\widetilde{\mathcal{O}} = \mathcal{O}_h$.

Therefore, if a $\mathbb{Z}_p$-basis of $\mathcal{O}_h$ is known in terms of $[1, \theta, \ldots, \theta^{n-1}]$ then the $\mathbb{Q}_p$-module isomorphism σ^{-1} immediately gives a $\mathbb{Z}_p$-basis of $\mathcal{O}_f$ in terms of $[1, \xi, \ldots, \xi^{n-1}]$.

To apply this result, recall that $p^d\mathcal{O}_f \subseteq \mathbb{Z}_p[\xi]$ whenever either $d \geq d_\xi$ or $p^{2d+2} \nmid \Delta_\xi$.

4. Case 1: algebraic decomposition

Assume $h, a_1, a_2 \in \mathbb{Z}_p[x]$, with

$$\begin{aligned} a_1h + a_2h' &\equiv p^d \pmod{p^{d+1}\mathbb{Z}_p[x]} \\ h(\xi) &\equiv 0 \pmod{p^m\mathbb{Z}_p[\xi]} \\ m &\geq 2d+1. \end{aligned}$$

Taking $\psi = -a_2(\xi)h(\xi)/p^m$ we have

$$\begin{aligned} \psi &\in \mathbb{Z}_p[\xi] \\ h(\xi + p^{m-d}\psi) &\equiv h(\xi) + p^{m-d}\psi h'(\xi) \equiv 0 \pmod{p^{m+1}\mathbb{Z}_p[\xi]}. \end{aligned}$$

It follows that there exists $\varphi \in \mathbb{Z}_p[\xi]$ such that

$$\begin{aligned} h(\varphi) &= 0 \\ \mathbb{Z}_p[\varphi] &= \mathbb{Z}_p[\xi] \\ \varphi &\equiv \xi \pmod{p^{m-d}\mathbb{Z}_p[\xi]}. \end{aligned}$$

Now, using the reduced discriminant p^{d_ξ} of f, set

$$m_\xi = 2d_\xi + 1.$$

If $h \equiv f \pmod{p^{m_\xi}\mathbb{Z}_p[x]}$, then $\mathcal{A}_f$ contains a root of h, so $\mathcal{A}_h \cong \mathcal{A}_f$ (as $\mathbb{Q}_p$-algebras).

In particular, if $f \equiv f_1 f_2 \pmod{p^{m_\xi}\mathbb{Z}_p[x]}$, f_1, f_2 monic, then $\mathcal{A}_f \cong \mathcal{A}_{f_1 f_2} \cong \mathcal{A}_{f_1} \oplus \mathcal{A}_{f_2}$. In this case an integral basis for $\mathcal{A}_f$ can be constructed directly by mapping integral bases of $\mathcal{A}_{f_1}$ and $\mathcal{A}_{f_2}$ into $\mathcal{A}_f$.

(Structural stability can be applied to reduce the modulus p^{m_ξ} by at least one p-adic digit. But as the example $f = x^2 - 5$, $f_1 = x - 1$, $f_2 = x + 1$, $p = 2$, $d_\xi = 1$ shows, $\mathcal{A}_f$ and $\mathcal{A}_{f_1 f_2}$ might then fail to be isomorphic $\mathbb{Q}_p$-algebras.)

Given a non-primary element of O_f, the Round Four algorithm finds orthogonal idempotents e_1, e_2 such that $e_j\mathcal{A}_f \cong \mathcal{A}_{f_j}$, $j = 1, 2$, giving explicit isomorphisms

$$\psi_j : \mathcal{A}_{f_j} \to e_j\mathcal{A}_f$$
$$\psi_j(\xi_j) = e_j\xi$$

where $\xi_j = x + f_j\mathbb{Q}_p[x]$, so that $\mathcal{A}_{f_j} = \mathbb{Q}_p[x]/f_j\mathbb{Q}_p[x] = \mathbb{Q}_p[\xi_j]$.

5. Computing the p-adic GCD

Suppose $f, b \in \mathbb{Z}[x]$, with f monic. Then for any m the MAPLE procedure

$$\texttt{ihermite(sylvester(}f,b\texttt{)) mod } p^m$$

gives, in its last non-zero row, $p^r f_1$, with $f_1 \in \mathbb{Z}[x]$, f_1 monic, such that

$$\begin{aligned} p^r f_1 &\in f\mathbb{Z}[x] + b\mathbb{Z}[x] + p^m\mathbb{Z}[x] \\ \texttt{rem(}f, f_1, x\texttt{)} &\equiv 0 \pmod{p^{m-r}\mathbb{Z}[x]}. \end{aligned}$$

So with

$$f_2 = \texttt{quo(}f, f_1, x\texttt{) mod } p^m$$

we have

$$f \equiv f_1 f_2 \pmod{p^{m-r}\mathbb{Z}[x]}.$$

Suppose further that $B \in \mathbb{Z}_p[x]$, $G = \gcd(f, B)$, $b \equiv B \pmod{p^m\mathbb{Z}_p[x]}$, and let

$$p^s\mathbb{Z}_p = \left(\frac{f}{G}\mathbb{Z}_p[x] + \frac{B}{G}\mathbb{Z}_p[x]\right) \cap \mathbb{Z}_p.$$

If $m > s$, then

$$r = s, \quad f_1 \equiv G \pmod{p^{m-s}\mathbb{Z}_p[x]};$$

otherwise

$$r < s, \quad \deg(f_1) > \deg(G).$$

6. Idempotents: a special case

Assume $e \in \mathbb{Q}_p[x]$ with $0 < \deg(e) < \deg(f)$ and $f \mid e(1-e)$.

Choose d_r so that $p^{d_r}e \in \mathbb{Z}_p[x]$, and let $B_1 = p^{d_r}(1-e)$, $B_2 = p^{d_r}e$.

For $j = 1, 2$ let

$$\begin{aligned} G_j &= \gcd(f, B_j) \\ H_j &= B_j/G_j \\ p^{s_j}\mathbb{Z}_p &= \left(\frac{f}{G_j}\mathbb{Z}_p[x] + \frac{B}{G_j}\mathbb{Z}_p[x]\right) \cap \mathbb{Z}_p \end{aligned}$$

so that

$$\begin{aligned} f &= G_1G_2 \\ p^{s_1}\mathbb{Z}_p &= (G_2\mathbb{Z}_p[x] + H_1\mathbb{Z}_p[x]) \cap \mathbb{Z}_p \\ p^{s_2}\mathbb{Z}_p &= (G_1\mathbb{Z}_p[x] + H_2\mathbb{Z}_p[x]) \cap \mathbb{Z}_p. \end{aligned}$$

Because

$$p^{d_r} = B_1 + B_2 = G_1H_1 + G_2H_2$$

we have $s_1 \leq d_r$, $s_2 \leq d_r$.

We wish to compute p-adic approximations to $\gcd(f, e)$ and $\gcd(f, 1-e)$.

We define $B = p^{d_r}(1-e)$, so that $B \in \mathbb{Z}_p[x]$ and $\gcd(f, B) = \gcd(f, 1-e)$.

For $m = m_r + d_r$ and $b \equiv B \pmod{p^m\mathbb{Z}_p[x]}$ we compute f_1, f_2 as before, giving

$$\begin{aligned} f_1 &\equiv \gcd(f, 1-e) \pmod{p^{m_r}\mathbb{Z}_p[x]} \\ f_2 &\equiv \gcd(f, e) \pmod{p^{m_r}\mathbb{Z}_p[x]} \\ f &\equiv f_1f_2 \pmod{p^{m_r}\mathbb{Z}_p[x]}. \end{aligned}$$

7. Case 2: search for a power basis

For $\theta \in \mathcal{A}_f$ with $\chi_\theta = x^n + c_1 x^{n-1} + \cdots + c_n$ we define

$$v^*(\theta) = \min_{1 \le k \le n} \frac{v_p(c_k)}{k}.$$

The p-radical of $\mathcal{O}_f$ is

$$\mathcal{J}_f = \{\theta \in \mathcal{O}_f : v^*(\theta) > 0\}.$$

For primary $\theta \in \mathcal{O}_f$ we define

$$\begin{aligned} D_\theta &= \deg(\nu_\theta) \\ L_\theta / M_\theta &= v^*(\nu_\theta(\theta)), \text{ with } L_\theta, M_\theta \ge 0, \gcd(L_\theta, M_\theta) = 1 \\ r_\theta L_\theta - s_\theta M_\theta &= 1, \text{ with } r_\theta > 0,\ s_\theta \ge 0 \\ \eta_\theta(\theta) &= \nu_\theta(\theta)^{r_\theta} / p^{s_\theta}, \text{ so that } v^*(\eta_\theta(\theta)) = 1/M_\theta. \end{aligned}$$

The algorithm attempts to find $\alpha \in \mathcal{O}_f$ with $\mathbb{Z}_p[\alpha] = \mathcal{O}_f$.

For any $\alpha \in \mathcal{O}_f$ with $\Delta_\alpha \neq 0$ we have $p^{d_\alpha}\mathcal{O}_f \subseteq \mathbb{Z}_p[\alpha]$. It follows that the integral elements of $\mathcal{A}_f$ lying outside $\mathbb{Z}_p[\alpha]$ have bounded p-adic value. Assuming $\mathbb{Z}_p[\alpha] \neq \mathcal{O}_f$, the algorithm constructs a sequence of elements $\beta \in \mathcal{O}_f - \mathbb{Z}_p[\alpha]$ with $v^*(\beta)$ strictly increasing. Since $v^*(\beta) < d_\alpha$ for each β, it suffices to compute β modulo $p^{d_\alpha}\mathbb{Z}_p[\alpha]$.

Tests **A**, **B**, **C** are applied to various elements $\theta \in \mathcal{O}_f$ that are produced by the algorithm.

A. If θ is non-primary, we break off the search for α and revert to case 1.

B. If $D_\theta \nmid D_\alpha$ then we find $\varphi \in \theta + \mathbb{Z}_p[\alpha]$ with either φ not primary or $D_\varphi = \operatorname{lcm}(D_\alpha, D_\theta) > D_\alpha$. We replace $\alpha \leftarrow \varphi$ and go to step **2**.

C. If $M_\theta \nmid M_\alpha$, we find $a, b, c \ge 0$ such that $aM_\alpha + bM_\theta - cM_\alpha M_\theta = \gcd(M_\alpha, M_\theta)$. Then $\varphi = \alpha + \eta_\alpha(\alpha)^b \eta_\theta(\theta)^a / p^c$ satisfies $D_\varphi = D_\alpha$ and $M_\varphi = \operatorname{lcm}(M_\alpha, M_\theta) > M_\alpha$. So we replace $\alpha \leftarrow \varphi$ and go to step **2**.

The sequence of elements β is constructed as follows.

1. Let $\xi = x + f(x)\mathbb{Z}_p[x]$.
   Set $\alpha = \xi$.

2. Apply test **A** to α.
   If $\Delta_\alpha = 0$, replace α by $\alpha + kp\xi$ for some $k > 0$ such that $\Delta_{\alpha+kp\xi} \neq 0$.
   Apply the Dedekind test to χ_α. If $\mathbb{Z}_p[\alpha] = \mathcal{O}_f$ the search terminates.
   If $v^*(\alpha) > 0$, replace α by $\alpha + 1$, so that $v^*(\alpha) = 0$.
   If $L_\alpha > 1$, replace α by $\alpha + \eta_\alpha(\alpha)$, so that $L_\alpha = 1$ (so $\eta_\alpha = \nu_\alpha$, with ν_α unchanged).

   We have $p^{d_\alpha}\mathcal{O}_f \subseteq \mathbb{Z}_p[\alpha]$. We determine r such that $\theta^{p^{rD_\alpha}} \in \mathbb{Z}_p[\alpha]$ whenever $\theta \in \mathbb{Z}_p[\alpha] + \mathcal{J}_f$. Let $\theta = \varphi + \psi$, with $\varphi \in \mathbb{Z}_p[\alpha]$, $\psi \in \mathcal{J}_f$. We have $v^*(\psi) \geq 1/n$, and $\psi^k \in \mathbb{Z}_p[\alpha]$ whenever $k \geq nd_\alpha$. Observe that $v^*\binom{p^s}{k} \geq s - v^*(k!) \geq s - (k-1)/(p-1)$. It therefore suffices to have $rD_\alpha \geq d_\alpha + (k-1)/(p-1)$ for all $k < nd_\alpha$.

3. Set $q = p^{rD_\alpha}$, with $r = \lceil (d_\alpha(n+p-1) - 2)/(D_\alpha(p-1)) \rceil$.
   Set $\pi = \nu_\alpha(\alpha)$, so that $v^*(\pi) = 1/M_\alpha$.
   Set $\beta = \pi^{M_\alpha}/p$.

4. Apply tests **A**, **B**, **C** to β.

5. Set $k = M_\alpha v^*(\beta)$, so that $v^*(\pi^k) = v^*(\beta)$. ($k \in \mathbb{Z}$ because $M_\beta \mid M_\alpha$.)
   Set $\gamma = \beta/\pi^k$, and apply tests **A**, **B**, **C** to γ.

6. Set $\delta = \gamma^q$, and apply tests **A**, **B**, **C** to δ.

7. If $\delta \in \mathbb{Z}_p[\alpha]$:
   We have $\pi^k\delta \in \mathbb{Z}_p[\alpha]$, $\pi^k\gamma = \beta \notin \mathbb{Z}_p[\alpha]$, and, because $D_\gamma \mid D_\alpha$, $\gamma - \delta \in \mathcal{J}_f$.
   So $\pi^k(\gamma - \delta) \notin \mathbb{Z}_p[\alpha]$, but $v^*(\pi^k(\gamma - \delta)) \geq v^*(\pi^k) + v^*(\gamma - \delta) > v^*(\pi^k) = v^*(\beta)$.
   Replace β by $\pi^k(\gamma - \delta) = \beta - \pi^k\delta$, and go to step 4.

8. If $\delta \notin \mathbb{Z}_p[\alpha]$:
   Then $\mathbb{Z}_p[\alpha, \gamma]/\mathcal{J}_f$ is not a field. We apply test **A** to elements of $\gamma + \mathbb{Z}_p[\alpha]$ until discovering a non-primary element of $\mathcal{O}_f$.

The preceding cycle of steps, increasing $v^*(\beta)$, cannot be repeated indefinitely. It must be interrupted by one of the following:

i) discovery of $\alpha \in \mathcal{O}_f$ with $\mathbb{Z}_p[\alpha] = \mathcal{O}_f$;
ii) discovery of a non-primary element;
iii) discovery of $\theta \in \mathcal{O}_f$ with $D_\theta \nmid D_\alpha$;
iv) discovery of $\theta \in \mathcal{O}_f$ with $M_\theta \nmid M_\alpha$.

Case i) terminates the algorithm. Case ii) lowers the degree of f, and so can occur at most $n-1$ times. Cases iii) and iv) occur finitely often, because D_α and M_α are bounded by n.

8. Polynomial factorization

The Round Four algorithm can readily be adapted for factorization of polynomials in $\mathbb{Q}_p[x]$.

First, the Dedekind test must be replaced by a test for irreducibility in $\mathbb{Q}_p[x]$. It suffices to test for two conditions:

1) χ_α is irreducible in $\mathbb{Q}_p[x]$ if χ_α is irreducible modulo p (i.e., the image of χ_α in $\mathbb{F}_p[x]$ is irreducible in $\mathbb{F}_p[x]$);
2) χ_α is irreducible in $\mathbb{Q}_p[x]$ if α is an *Eisenstein element,* i.e.,
   i) α is primary, and
   ii) $\Delta_\alpha \neq 0$, and
   iii) $v^*(\nu_\alpha(\alpha)) = \deg(\nu_\alpha)/\deg(\chi_\alpha)$.

It is a well-known theorem that

$$\begin{aligned} f \text{ is irreducible in } \mathbb{Q}_p[x] &\iff A_f \text{ is a field} \\ &\iff A_f \text{ contains an Eisenstein element.} \end{aligned}$$

Second, routines that returned integral bases must be modified to return factorizations instead.

As noted above, factorizations must be computed modulo $p^{2d_\epsilon+1}$ (at least) to ensure that the results can be lifted to $\mathbb{Q}_p[x]$; any such factorization agrees with the correct p-adic factorization modulo $p^{d_\epsilon+1}$.

Acknowledgements.

This work was done at the invitation of, and with generous support from, the Centre de Recherche en Mathématique de Bordeaux at Université Bordeaux I. Additional support was provided by the Natural Sciences and Engineering Research Council (Canada) and Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (Québec).

References

The essential mathematical background is given in [Zariski & Samuel 1958, Ch V]. The Round Two algorithm is described in [Zimmer 1972, Ch 6(b)] and [Cohen 1993, Ch 6.1]. Elements of the Round Four algorithm are evident in [Zassenhaus 1975]. The first implementation of Round Four was [Ford 1978] (also reported in [Ford 1987]), for which the ALGEB language was created. [Böffgen 1987] gives a straightforward account (in German) but with some proofs omitted. [Böffgen & Reichert 1987] discusses application of the algorithm to factorization. [Bradford 1988] provides a well-written survey. [Pohst & Zassenhaus 1989, Ch 4] describes the algorithm in a general setting.

1. A. Ash, R. Pinch, R. Taylor, *An $\widehat{A}_4$ extension of* $\mathbf{Q}$ *attached to a non-selfdual automorphic form on GL(3)*, Mathematische Annalen **291** (1991), 753–766.
2. R. Böffgen, *Der Algorithmus von Ford/Zassenhaus zur Berechnung von Genzheitsbasen in Polynomalgebren*, Annales Universitatis Saraviensis, Series Mathematicae 1 (1987), no. 3, 60–129.
3. R. Böffgen, M. A. Reichert, *Computing the Decomposition of Primes p and p-adic Absolute Values in Semi-simple Algebras over* $\mathbf{Q}$, Journal of Symbolic Computation 4 (1987), 3–10.
4. R. J. Bradford, *On the Computation of Integral Bases and Defects of Integrity*, PhD Dissertation, University of Bath, 1988.
5. B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. Leong, M. B. Monagan, S. M. Watt, *Maple V Language Reference Manual*, Springer-Verlag, New York, 1991.
6. H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, New York, 1993.
7. D. Ford, *On the Computation of the Maximal Order in a Dedekind Domain*, PhD Dissertation, Ohio State University, 1978.
8. D. Ford, *The Construction of Maximal Orders over a Dedekind Domain*, Journal of Symbolic Computation 4 (1987), 69–75.
9. M. Pohst, H. Zassenhaus, *Algorithmic Algebraic Number Theory*, Cambridge University Press, Cambridge, 1989.
10. O. Zariski, P. Samuel, *Commutative Algebra, Vol I*, Van Nostrand, Princeton, New Jersey, 1958.
11. H. Zassenhaus, *On Hensel Factorization II*, Symposia Mathematica XV (1975), Academic Press, London, 499–513.
12. H. Zimmer, *Computational Problems, Methods and Results in Algebraic Number Theory*, Lecture Notes in Mathematics **262**, Springer-Verlag, 1972.

Appendix I — The Major Procedures

maxord

Given p : a rational prime,
f : a monic, separable polynomial in $\mathbb{Z}_p[x]$,
m_f : the p-adic value of the discriminant of f.

Let $\xi = x + f\mathbb{Q}_p[x]$, so that $\mathbb{Q}_p[\xi] = \mathbb{Q}_p[x]/f\mathbb{Q}_p[x] = \mathcal{A}_f$.

Factorize f mod p, and let g be the square-free part of f.

If f satisfies the Dedekind criterion with respect to p, then the ring of coefficients in $\mathcal{A}_f$ of the p-radical of $\mathbb{Z}_p[\xi]$ is a p-maximal order.
Return the basis computed by `dbasis`(p, f, m_f, x, f, g).

If f fails to satisfy the Dedekind criterion, then $\mathbb{Z}_p[\xi]$ is not p-maximal.

Let h be an irreducible factor of f mod p.

If h is the only irredicible factor of f mod p, then ξ is primary.
Return the basis computed by `nilord`(p, f, m_f, h).

If h is not the only irreducible factor of f mod p, then ξ is non-primary.
Return the basis computed by `decomp`(p, f, m_f, x, f, h).

decomp

Given	p :	a rational prime,
	f :	a monic, separable polynomial in $\mathbb{Z}_p[x]$,
	m_f :	the p-adic value of the discriminant of f,
	θ :	a non-primary element of $\mathcal{O}_f$,
	χ_θ :	the characteristic polynomial of θ,
	ν_θ :	an irreducible factor of χ_θ mod p.

Let $p^{d_r} = \texttt{respm}(f, f', p^{m_f})$, $m_r = 2d_r + 1$.

Then $p^{d_r}\mathcal{O}_f \subseteq \mathbb{Z}_p[\xi]$, where $\xi = x + f\mathbb{Q}_p[x]$, and $p^{d_r}\mathbb{Z}_p = (f\mathbb{Z}_p[x] + f'\mathbb{Z}_p[x]) \cap \mathbb{Z}_p$. Furthermore, if $f^* \equiv f \pmod{p^{m_r}\mathbb{Z}_p[x]}$, then $\mathcal{A}_{f^*} \cong \mathcal{A}_f$.

Modulo $p\mathbb{Z}_p[x]$, set $b_2 \equiv \nu_\theta^k$, with k such that $\chi_\theta \equiv b_1b_2$ and $\gcd(b_1, b_2) \equiv 1$. Compute a_1, a_2 so that $a_1b_2 + a_2b_1 \equiv 1 \pmod{p\mathbb{Z}_p[x]}$, and set

$$\begin{aligned} e(\xi) &= a_1(\theta(\xi))b_2(\theta(\xi)) \\ k &= 1. \end{aligned}$$

In the loop that follows we will always have $e(\xi)(1 - e(\xi)) \in p^k\mathcal{O}_f$.

Repeat the sequence

$$\begin{aligned} e(\xi) &\leftarrow 3e(\xi)^2 - 2e(\xi)^3 \pmod{p^{2k}\mathbb{Z}_p[\xi]} \\ k &\leftarrow 2k \end{aligned}$$

until $k \geq m_r + d_r$, then set

$$\begin{aligned} f_1 &\equiv \texttt{gcdpm}(f, p^{d_r}(1-e), p^{m_r+d_r}), \\ f_2 &\equiv \texttt{quo}(f, f_1, x) \end{aligned}$$

modulo $p^{m_r}\mathbb{Z}_p[x]$.

Define $e_1 = e$, $e_2 = 1 - e_1$, and for $j = 1, 2$ let $\xi_j = x + f_j\mathbb{Q}_p[x]$. Compute the integral bases $[\omega_{j,1}(\xi_j), \ldots \omega_{j,n_j}(\xi_j)] = \texttt{maxord}(p, f_j, m_{f_j})$.

Return the basis

$$[\,e_1(\xi)\omega_{1,1}(\xi),\ \ldots,\ e_1(\xi)\omega_{1,n_1}(\xi),\ e_2(\xi)w_{2,1}(\xi),\ \ldots,\ e_2(\xi)\omega_{2,n_2}(\xi)\,].$$

nilord

Given p : a rational prime,
f : a monic, separable polynomial in $\mathbb{Z}_p[x]$,
m_f : the p-adic value of the discriminant of f,
g : the unique irreducible factor of f mod p.

Initially set $\alpha = \xi = x + f\mathbb{Q}_p[x]$.

Perform the following steps, starting over whenever α is replaced.

Compute χ_α, the characteristic polynomial of α.
Factorize χ_α mod p, and let ν_α be an irreducible factor of χ_α mod p.
If χ_α has more than one irreducible factor mod p,
then **return** the basis given by $\texttt{decomp}(p, f, m_f, \alpha, \chi_\alpha, \nu_\alpha)$.

Compute Δ_α, the discriminant of χ_α.
If $\Delta_\alpha = 0$, replace $\alpha \leftarrow \alpha + p\xi$.
If χ_α satisfies the Dedekind criterion with respect to p,
then **return** the basis given by $\texttt{dbasis}(p, f, m_f, \alpha, \chi_\alpha, \nu_\alpha)$.

If $v^*(\alpha) > 0$, replace $\alpha \leftarrow \alpha + 1$.
Let $v^*(\nu_\alpha(\alpha)) = L_\alpha / M_\alpha$, with $\gcd(L_\alpha, M_\alpha) = 1$, $M_\alpha > 0$.
Compute η_α such that $v^*(\eta_\alpha(\alpha)) = 1/M_\alpha$.
If $L_\alpha > 1$, replace $\alpha \leftarrow \alpha + \eta_\alpha(\alpha)$.

Let φ be returned by $\texttt{bsrch}(p, \chi_\alpha, k_\alpha, \eta_\alpha, M_\alpha)$, where $p^{k_\alpha}\mathbb{Z}_p = \Delta_\alpha \mathbb{Z}_p$.
If φ is non-primary, **return** the basis given by $\texttt{decomp}(p, f, m_f, \varphi, \chi_\varphi, \nu_\varphi)$.
Otherwise, replace $\alpha \leftarrow \varphi$.

`bsrch`

Given
p : a rational prime,
f_α : the characteristic polynomial of α,
k_α : the p-adic value of the discriminant of f_α,
ν_α : the unique irreducible factor of f_α mod p,
M_α : rational integer such that $v^*(\nu_\alpha(\alpha)) = 1/M_\alpha$.

Let

$$\begin{aligned} n &= \deg(f_\alpha) \\ r &= \lceil (d_\alpha(n+p-1)-2)/(D_\alpha(p-1)) \rceil \\ \pi &= \nu_\alpha(\alpha). \end{aligned}$$

Initially set $\beta \equiv \pi^{M_\alpha}/p \pmod{p^{d_\alpha}\mathbb{Z}_p[\alpha]}$.

Perform the following sequence of tests for each choice of β.

If β is not primary, **return** β.
If $D_\beta \nmid D_\alpha$, find $\varphi \in \beta + \mathbb{Z}_p[\alpha]$ with φ not primary or $D_\varphi = \operatorname{lcm}(D_\alpha, D_\beta)$, and **return** φ.
If $M_\beta \nmid M_\alpha$, compute φ such that $D_\varphi = D_\alpha$ and $M_\varphi = \operatorname{lcm}(M_\alpha, M_\beta)$, and **return** φ.

Set $k = M_\alpha v^*(\beta)$, so $v^*(\pi^k) = v^*(\beta)$, and set $\gamma \equiv \beta/\pi^k \pmod{p^{d_\alpha}\mathbb{Z}_p[\alpha]}$.

If γ is not primary, **return** γ.
If $D_\gamma \nmid D_\alpha$, find $\varphi \in \gamma + \mathbb{Z}_p[\alpha]$ with φ not primary or $D_\varphi = \operatorname{lcm}(D_\alpha, D_\gamma)$, and **return** φ.
If $M_\gamma \nmid M_\alpha$, compute φ such that $D_\varphi = D_\alpha$ and $M_\varphi = \operatorname{lcm}(M_\alpha, M_\gamma)$, and **return** φ.

Set $q = p^{rD_\alpha}$, so that $\theta^q \in \mathbb{Z}_p[\alpha]$ whenever $\theta \in \mathbb{Z}_p[\alpha] + \mathcal{J}_f$, and set $\delta \equiv \gamma^q \pmod{p^{d_\alpha}\mathbb{Z}_p[\alpha]}$.

If δ is not primary, **return** δ.
If $D_\delta \nmid D_\alpha$, find $\varphi \in \delta + \mathbb{Z}_p[\alpha]$ with φ not primary or $D_\varphi = \operatorname{lcm}(D_\alpha, D_\delta)$, and **return** φ.

If $M_\delta \nmid M_\alpha$, compute φ such that $D_\varphi = D_\alpha$ and $M_\varphi = \mathrm{lcm}(M_\alpha, M_\delta)$, and **return** φ.

If $\delta \notin \mathbb{Z}_p[\alpha]$, examine elements $\varphi = \gamma + h(\alpha)$, with $h \in \mathbb{Z}_p[x]$ reduced modulo $p\mathbb{Z}_p[x]$, until finding non-primary φ; then **return** φ.

If $\delta \in \mathbb{Z}_p[\alpha]$, set $\beta \equiv \pi^k(\gamma - \delta) \pmod{p^{d_\alpha}\mathbb{Z}_p[\alpha]}$ and repeat the sequence of tests.

dbasis

Given p : a rational prime,
f : a monic, separable polynomial in $\mathbb{Z}_p[x]$,
m_f : the p-adic value of the discriminant of f,
α : an element of $\mathcal{O}_f$,
f_α : the (separable) characteristic polynomial of α,
g_α : the square-free part of f_α mod p.

Let

$$\begin{aligned} h_\alpha &= \mathtt{Quo}(f_\alpha, g_\alpha, x) \ \mathtt{mod}\ p \\ k_\alpha &= \tfrac{1}{p}(f_\alpha - g_\alpha h_\alpha) \\ g_\beta &= \mathtt{Gcd}(g_\alpha, h_\alpha, k_\alpha) \ \mathtt{mod}\ p \\ h_\beta &= \mathtt{Quo}(f_\alpha, g_\beta, x) \ \mathtt{mod}\ p \end{aligned}$$

and define

$$\beta = \frac{h_\beta(\alpha)}{p}.$$

The coefficient ring in $\mathcal{A}_f$ of the p-radical of $\mathbb{Z}_p[\alpha]$ is

$$\mathcal{D}_\alpha = \mathbb{Z}_p[\alpha] + \beta\mathbb{Z}_p[\alpha],$$

which is an order of $\mathcal{A}_f$ satisfying $\mathbb{Z}_p[\alpha] \subseteq \mathcal{D}_\alpha \subseteq \mathcal{O}_f$, with $\mathbb{Z}_p[\alpha] = \mathcal{D}_\alpha$ if and only if $\mathbb{Z}_p[\alpha] = \mathcal{O}_f$.

Moreover, if $\deg(g_\beta) = m$ then $[\mathcal{D}_\alpha : \mathbb{Z}_p[\alpha]] = p^m$.

Return the basis $[1, \alpha, \ldots, \alpha^{n-m-1}, \beta, \alpha\beta, \ldots, \alpha^{m-1}\beta]$.

gcdpm

Given f_1 : a monic polynomial in $\mathbb{Z}_p[x]$,
f_2 : a polynomial in $\mathbb{Z}_p[x]$,
p^m : a power of the rational prime p.

Let $n = \deg(f_1)$.

For $1 \le j \le n$, set $\sum_{k=1}^{n} B_{j,k}x^{n-k} \equiv \texttt{rem}(x^{n-j}f_2, f_1, x) \pmod{p^m\mathbb{Z}_p[x]}$.

Let I be the $n \times n$ identity matrix, and set $A = \begin{pmatrix} B \\ p^m I \end{pmatrix} \in \mathbb{Z}^{2n\times n}$.

Compute the row-reduced Hermite normal form of A.

Determine the maximum k, $1 \le k \le n$, satisfying $A_{kk} \not\equiv 0 \pmod{p^m}$.

Return $\sum_{j=k}^{n} \frac{A_{kj}}{A_{kk}} x^{n-j}$.

respm

Given f_1 : a monic polynomial in $\mathbb{Z}_p[x]$,
f_2 : a polynomial in $\mathbb{Z}_p[x]$,
p^m : a power of the rational prime p,
with $p^{m+1} \nmid \texttt{resultant}(f_1, f_2, x)$.

Compute $a_1, a_2 \in \mathbb{Q}_p[x]$ such that

$$a_1 f_1 + a_2 f_2 = 1, \quad \deg(a_1) < \deg(f_2), \quad \deg(a_2) < \deg(f_1).$$

Set $p^c = \gcd(p^m, \texttt{content}(p^m a_1, x), \texttt{content}(p^m a_2, x))$. We have

$$p^{m-c}\mathbb{Z}_p = (f_1\mathbb{Z}_p[x] + f_2\mathbb{Z}_p[x]) \cap \mathbb{Z}_p.$$

Return p^{m-c}.

Appendix II — Maple Source Listings

```
###
###     Usage:
###
###
###     > f := x^7 - 7*x + 3;
###                                           7
###                                     f := x  - 7 x + 3
###
###     > ifactor(discrim(f,x));
###                                                8    8
###                                         (3)  (7)
###
###     > print(maxord(3,f,8));
###                              2   3   4   5
###                     [ 1, x, x , x , x , x ,
###
###                                          2        3        4        5        6
###                       - 1/3 x + 1/3 x  - 1/3 x  + 1/3 x  - 1/3 x  + 1/3 x  ]
###
###     > print(maxord(7,f,8));
###                                             2   3   4   5   6
###                                    [ 1, x, x , x , x , x , x  ]
###


        with(linalg,coldim,delrows,matrix,rowdim,stack,vectdim,vector):


###     p-maximal order of Af

        maxord := proc (p, f, mf)
        local w, r, g, h, j;

        w := Factors(f) mod p;

        r := rowdim(matrix(w[2]));

        g := product('w[2][j][1]','j'=1..r);   ### g = square-free part   ###
        h := w[2][1][1];                       ### h = irreducible factor ###

        if dedek(p,f,g,mf) then                ### p-maximal by Dedekind  ###
            RETURN(dbasis(p,f,mf,x,f,g))
        elif r = 1 then                        ### xi is primary          ###
            RETURN(nilord(p,f,mf,h))
        else                                   ### xi is non-primary      ###
            RETURN(decomp(p,f,mf,x,f,h))
        fi

        end:
```

```
###     Af = Af1 + Af2

        decomp := proc (p, f, mf, theta, chi, nu)
        local pdr, pmr, ph, pk, j, n1, n2, t, v1, v2,
              a1, a2, b1, b2, b3, f1, f2, e, ib1, ib2, ibas;

        pdr := respm(f,diff(f,x),p^mf);         ### reduced resultant ###
        pmr := pdr*pdr*p;

        b1  := chi mod p;  a2 := 0;
        b2  :=   1 mod p;  a1 := 1;
        b3  :=  nu mod p;
        while degree(b3,x) > 0 do
            b1 :=              Quo(b1,b3,x) mod p;
            b2 :=                     b2*b3 mod p;
            b3 := Gcdex(b2,b1,x,'a1','a2') mod p        ### monic ###
        od;

        e   := eleval(f,a1*b2,theta);
        e   := (pdr*e mod pdr*p) / pdr;

        pk  := p;
        ph  := pdr*pmr;

###     E(t) - e(t) belongs to p^k*Op, which is contained in p^(k-dr)*Zp[xi]

        while pk < ph do
            e  := rem (e*e*(3 - 2*e), f, x);
            pk := pk*pk;
            e  := (pdr*e mod pdr*pk) / pdr
        od;

        f1 := gcdpm(f,pdr*(1-e),pdr*pmr);  f1 := mods (f1, pmr);
        f2 :=                quo(f,f1,x);  f2 := mods (f2, pmr);

        n1  := degree(f1,x);            n2  := degree(f2,x);
        v1  := ordp(discrim(f1,x),p);   v2  := ordp(discrim(f2,x),p);
        ib1 := maxord(p,f1,v1);         ib2 := maxord(p,f2,v2);

        ibas := vector(n1+n2);

        for j from 1 to n1 do
            ibas[j] := rem(pdr*e*ib1[j],f,x) mod pdr
        od;
        for j from n1+1 to n1+n2 do
            ibas[j] := rem(pdr*(1-e)*ib2[j-n1],f,x) mod pdr
        od;

        RETURN(nbasis(ibas,pdr))

        end:
```

```
###     polynomial gcd mod p^m (assumes f1 monic)

        gcdpm := proc (f1, f2, pm)
        local n, a, b, h, j, k, n;

        n := degree(f1,x);
        a := matrix(n,n);

        for j from 1 to n do            ### Sylvester’s matrix, mod p^m ###
            if j = 1 then
                h := rem (f2, f1, x) mod pm
            else
                h := rem (x*h, f1, x) mod pm
            fi;
            for k from 1 to n do
                a[n-j+1,k] := coeff(h,x,n-k)
            od
        od;

        a := hnfpm(a,pm);

        k := 0;
        for j from 1 to n do
        if a[j,j] mod pm <> 0 then
            k := j
        fi;
        od;

        b := sum(’(a[k,j]/a[k,k])*x^(n-j)’,’j’=k..n);

        RETURN(b)

        end:


###     reduced resultant mod p^m

        respm := proc (f1, f2, pm)
        local g, a1, a2, pc;

        g  := gcdex (f1, f2, x, ’a1’, ’a2’);

        a1 := pm*a1 mod pm;
        a2 := pm*a2 mod pm;

        pc := igcd (pm, content(a1,x));
        pc := igcd (pc, content(a2,x));

        RETURN(pm/pc)

        end:
```

```
###     Hermite normal form (n x n) mod p^m

        hnfpm := proc (a, pm)
        local b, c, g, h, j, k, m, n, q, r,
              c00, c11, c12, c13, c21, c22, c23;

        m := rowdim(a);  n := coldim(a);
        b := stack(a,pm*array(1..n,1..n,identity));

        for k from 1 to m do
        for j from 1 to n do
            b[k,j] := mods (b[k,j], pm)         ### Q int Zp not subs Z ###
        od
        od;

        for c from 1 to n do
            h := c + m;
            g := b[c+m,c];
            for k from c to c+m-1 do
            if b[k,c] <> 0 then
                c00 := igcd(b[k,c],b[c+m,c]);
                if c00 < g then
                    h := k;  g := c00
                fi
            fi
            od;
            k := c + m;
            if h <> k then
                c00 := igcdex(b[h,c],b[k,c],'c11','c12');
                c21 := -b[k,c]/c00;  c22 := b[h,c]/c00;
                b[h,c] := c00;
                b[k,c] := 0;
                for j from c+1 to n do
                    c13 := c11*b[h,j] + c12*b[k,j];
                    c23 := c21*b[h,j] + c22*b[k,j];
                    b[h,j] := mods (c13, pm);
                    b[k,j] := mods (c23, pm)
                od
            fi;
            if h <> c then
            for j from c to n do
                k := b[c,j];  b[c,j] := b[h,j];  b[h,j] := k
            od
            fi;
            for k from 1 to c+m do
            if k <> c then
                r := mods (b[k,c], b[c,c]);
                q := iquo (b[k,c] - r, b[c,c]);
                if q <> 0 then
                for j from c to n do
                    b[k,j] := mods (b[k,j] - q*b[c,j], pm)
                od
                fi
            fi
            od
        od;

        RETURN(delrows(b,n+1..n+m))

        end:
```

```
###     xi is primary

        nilord := proc (p, fx, mf, gx)
        local alpha, chi, nu, eta, La, Ma, w, Dchi, phi, pm;

        alpha := x;  chi := fx;  nu := gx;  Dchi := p^mf;  pm := p^(mf+1);

        while true do

            if Dchi = 0 then
                alpha := alpha + p*x
            elif dedek(p,chi,nu,ordp(Dchi,p)) then
                RETURN(dbasis(p,fx,mf,alpha,chi,nu))     ### Dedekind ###
            else
                if vstar(chi,p) > 0 then
                    alpha := alpha + 1;
                    chi   := subs(x=x-1,chi);
                    nu    := subs(x=x-1,nu) mod p
                fi;
                w := setup(chi,p,x,nu);
                eta := w[2];  La := w[3];  Ma := w[4];
                if La > 1 then
                    alpha := alpha + eleval(fx,eta,alpha)
                else
                    w := bsrch (p, chi, ordp(Dchi,p), eta, Ma);
                    phi := eleval(fx,w[2],alpha);
                    if w[1] = 1 then
                        RETURN(decomp(p,fx,mf,phi,w[3],w[4]))
                    else
                        alpha := phi
                    fi
                fi
            fi;

            w := factcp(fx,p,alpha);  chi := w[1];  nu := w[2];
            if w[4] > 1 then
                RETURN(decomp(p,fx,mf,alpha,chi,nu))
            fi;
            Dchi := discrim(mods(chi,pm),x) mod pm;
            if Dchi = 0 then
                Dchi := discrim(chi,x)
            fi

        od

        end:
```

```
###     Returns
###     [1, phi, chi, nu]  if phi non-primary
###     [2, phi, chi, nu]  if D_phi > D_alpha or M_phi > M_alpha

        bsrch := proc (p, fa, ka, eta, Ma)
        local n, c, pc, pcc, beta, gamma, delta,
              j, pik, Da, Vb, w, r, field;

        Da  := degree(eta,x);
        n   := degree(fa,x);
        pc  := respm(fa,diff(fa,x),p^ka);
        c   := ordp(pc,p);
        pcc := pc*pc;

        r := 1 + trunc ( (c*(n + p - 1) - 2)/(Da*(p - 1)) );

        beta := eltpow (fa, eta, Ma) / p;

        while true do

            beta := (pc*beta mod pcc) / pc;

            w := testd (p, fa, c, Da, eta, Ma, beta);
            if w[1] < 3 then RETURN(w) fi;

            Vb := vstar(w[3],p);                ### w[3] = chi ###

            pik := eltpow (fa, eta, Ma*Vb);

            gamma := rem (beta*eltinv(fa,pik), fa, x);
            gamma := (pc*gamma mod pcc) / pc;

            w := testd (p, fa, c, Da, eta, Ma, gamma);
            if w[1] < 3 then RETURN(w) fi;

            delta := eltppm (fa, p^c, gamma, p^(r*Da));
            delta := (pc*delta mod pcc) / pc;

            w := testd (p, fa, c, Da, eta, Ma, delta);
            if w[1] < 3 then RETURN(w) fi;

            field := true;
            for j from 1 to n do
            if ordp(coeff(delta,x,n-j),p) < 0 then
                field := false
            fi
            od;

            if field then
                beta := beta - rem (pik*delta, fa, x)
            else
                RETURN(csrch(p,fa,gamma))
            fi

        od

        end:
```

```
###     Returns  [1, theta, chi, nu]  with theta non-primary

        csrch := proc (p, fa, gamma)
        local t, v, h, j, r, theta, w, b;

        b := vector(4);

        t := 0;
        while true do
            t := t + 1;
            v := t;
            h := 0;
            j := 0;
            while v <> 0 do
                r := irem(v,p);
                v := iquo(v,p);
                h := h + r*x^j;
                j := j + 1
            od;
            theta := gamma + rem (h, fa, x);
            w := factcp(fa,p,theta);
            if w[4] > 1 then
                b[1] := 1;              ### non-primary ###
                b[2] := theta;
                b[3] := w[1];
                b[4] := w[2];
                RETURN(b)
            fi
        od

        end:
```

```
###     Returns  [1, phi, chi, nu]  if phi non-primary
###              [2, phi, chi, nu]  if D_phi = lcm(D_alpha,D_theta)

        testb := proc (p, fa, Da, theta, Dt)
        local Dat, t, v, h, j, r, phi, w, b;

        Dat := ilcm(Da,Dt);
        b := vector(4);
        t := 0;
        while true do
            t := t + 1;
            v := t;
            h := 0;
            j := 0;
            while v <> 0 do
                r := irem(v,p);
                v := iquo(v,p);
                h := h + r*x^j;
                j := j + 1
            od;
            phi := theta + rem (h, fa, x);
            w   := factcp (fa, p, phi);
            if w[4] > 1 then            ### phi non-primary ###
                b[1] := 1;
                b[2] := phi;
                b[3] := w[1];
                b[4] := w[2];
                RETURN(b)
            fi;
            if w[3] = Dat then          ### D_phi = lcm(D_alpha,D_theta) ###
                b[1] := 2;
                b[2] := phi;
                b[3] := w[1];
                b[4] := w[2];
                RETURN(b)
            fi
        od

        end:
```

```
###     Returns  [1, phi, chi, nu]  if phi non-primary
###              [2, phi, chi, nu]  if M_phi > M_alpha

        testc := proc (p, fa, c, alph2, Ma, thet2, Mt)
        local g, r, s, t, c1, c2, c3, pc, ppc, psi, phi, w, b;

        b := vector(4);  pc := p^c;  ppc := p*pc;

        g := igcdex (Ma, Mt, 'r', 's');
        t := 0;                         ### r Ma + s Mt - t Ma Mt = g ###
        while r < 0 do
            r := r + Mt;
            t := t + 1
        od;
        while s < 0 do
            s := s + Ma;
            t := t + 1
        od;
        c1  := eltpow(fa,alph2,s);
        c2  := eltpow(fa,thet2,r);
        c3  := rem (c1*c2, fa, x) / p^t;
        psi := (pc*c3 mod ppc) / pc;    ### psi = c3 mod p ###
        phi := x + psi;

        w   := factcp (fa, p, phi);
        if w[4] > 1 then                ### phi non-primary ###
            b[1] := 1;
            b[2] := phi;
            b[3] := w[1];
            b[4] := w[2];
            RETURN(b)
        else                            ### M_phi = lcm(M_alpha,M_theta) ###
            b[1] := 2;
            b[2] := phi;
            b[3] := w[1];
            b[4] := w[2];
            RETURN(b)
        fi

        end:
```

```
###     Returns   [1, phi, chi, nu]  if phi non-primary
###               [2, phi, chi, nu]  if D_phi > D_alpha or M_phi > M_alpha
###               [3, phi, chi, nu]  otherwise

        testd := proc (p, fa, c, Da, alph2, Ma, theta)
        local chit, nut, Dt, thet2, Mt, w, b;

        b := vector(4);

        w := factcp (fa, p, theta);
        chit := w[1];  nut := w[2];  Dt := w[3];

        if w[4] > 1 then                 ### theta non-primary ###
            b[1] := 1;
            b[2] := theta;
            b[3] := chit;
            b[4] := nut;
            RETURN(b)
        fi;

        if Da < ilcm(Da,Dt) then         ### D_phi > D_alpha ###
            RETURN(testb(p,fa,Da,theta,Dt))
        fi;

        w := setup (fa, p, theta, nut);  thet2 := w[2];  Mt := w[4];

        if Ma < ilcm(Ma,Mt) then         ### M_phi > M_alpha ###
            RETURN(testc(p,fa,c,alph2,Ma,thet2,Mt))
        else
            b[1] := 3;
            b[2] := theta;
            b[3] := chit;
            b[4] := nut;
            RETURN(b)
        fi

        end:
```

```
###     Factorize char poly mod p

        factcp := proc (f, p, theta)
        local w, b;

        b    := vector(4);
        b[1] := chpol(f,theta);                ### chi_theta           ###
        w    := Factors(b[1]) mod p;
        b[2] := w[2][1][1];                    ### nu_theta            ###
        b[3] := degree(b[2],x);                ### D_theta             ###
        b[4] := rowdim(matrix(w[2]));          ### Nr of mod p factors ###

        RETURN(b)

        end:


###     Returns [theta_1, theta_2, L_theta, M_theta]
###             [1]       [2]       [3]      [4]

        setup := proc (f, p, theta, nut)
        local b, t1, t2, v1, Lt, Mt, c, r, s;

        b := vector(4);

        t1 := eleval(f,nut,theta);
        v1 := vstar(chpol(f,t1),p);
        Lt := numer(v1);
        Mt := denom(v1);
        c  := igcdex(Lt,-Mt,'r','s');          ### r Lt - s Mt = 1 ###
        while r <= 0 do
            r := r + Mt;
            s := s + Lt
        od;
        t2 := eltpow(f,t1,r) / p^s;

        b[1] := t1;
        b[2] := t2;
        b[3] := Lt;
        b[4] := Mt;

        RETURN(b)

        end:
```

```
###     evaluate g(a)

        eleval := proc (f, h, a)
        local n, g, y, k;

        g := collect(h,x);
        n := degree(g,x);
        y := 0;
        for k from n by -1 to 0 do
            y := rem (y*a + coeff(g,x,k), f, x)
        od;

        RETURN(y)

        end:


###     inverse of theta in Af

        eltinv := proc (f, theta)
        local g, a, b;

        g := gcdex (theta, f, x, 'a', 'b');

        RETURN(a)

        end:


###     Power of an element

        eltpow := proc (f, theta, k)
        local phi, psi, q, r;

        phi := 1;
        psi := theta;
        q   := k;

        while q <> 0 do
            r := irem(q,2);
            if r <> 0 then
                phi := rem (phi*psi, f, x)
            fi;
            q := iquo(q,2);
            if q <> o then
                psi := rem (psi*psi, f, x)
            fi
        od;

        RETURN(phi)

        end:
```

```
###     Modular power of an element

        eltppm := proc (f, pd, theta, k)
        local pdd, phi, psi, q, r;

        pdd := pd*pd;
        phi := pd;
        psi := pd*theta;
        q   := k;

        while q <> 0 do
            r := irem(q,2);
            if r <> 0 then
                phi := rem (phi*psi/pd, f, x);
                phi := phi mod pdd
            fi;
            q := iquo(q,2);
            if q <> o then
                psi := rem (psi*psi/pd, f, x);
                psi := psi mod pdd
            fi
        od;

        RETURN(phi/pd)

        end:


###     Dedekind test for p-maximality

        dedek := proc (p, fa, ga, mfa)
        local ha, gb;

        ha := Quo(fa,ga,x) mod p;
        gb := (fa - ga*ha)/p;
        gb := Gcd(gb,ga) mod p;
        gb := Gcd(gb,ha) mod p;
        if member (2*degree(gb,x), {0, mfa-1, mfa}) then
            RETURN(true)
        else
            RETURN(false)
        fi

        end:
```

```
###     Dedekind basis

        dbasis := proc (p, f, mf, alpha, fa, ga)
        local pd, n, m, i, j, k, h, a, b, ha, gb, hb;

        n  := degree(f,x);
        pd := p^trunc(mf/2);

        ha := Quo(fa,ga,x) mod p;
        gb := (fa - ga*ha)/p;
        gb := Gcd(gb,ga) mod p;
        gb := Gcd(gb,ha) mod p;

        a := matrix(n,n);
        m := degree(gb,x);

        for i from 1 to n do            ### Zp[a] + bZp[a] is maximal ###
            if i = 1 then
                ha := pd
            elif i = n-m+1 then
                hb := Quo(fa,gb,x) mod p;
                ha := pd*eleval(f,hb,alpha)/p
            else
                ha := rem(alpha*ha,f,x)
            fi;
            for j from 1 to n do
                a[i,j] := coeff(ha,x,n-j)
            od
        od;

        a := hnfpm(a,pd);

        b := vector(n);

        for j from 1 to n do
            b[j] := sum('a[n+1-j,n-k]*x^k','k'=0..n-1) / pd
        od;

        RETURN(b)

        end:
```

```
###     Normalized integral basis

        nbasis := proc (ibas, pd)
        local n, j, k, a, b, h;

        n := vectdim(ibas);

        a := matrix(n,n);

        for j from 1 to n do
        for k from 1 to n do
            a[j,k] := coeff(ibas[j],x,n-k)
        od
        od;

        a := hnfpm(a,pd);

        b := vector(n);

        for j from 1 to n do
            b[j] := sum('a[n+1-j,n-k]*x^k','k'=0..j-1) / pd
        od;

        RETURN(b)

        end:


###     characteristic polynomial of beta

        chpol := proc (f, beta)
        local g, y;

        g := resultant (y-beta, f, x);
        g := subs(y=x,g);
        g := collect(g,x);
        g := g/lcoeff(g,x);

        RETURN(g)

        end:
```

```
###     p-adic valuation

        ordp := proc (w, p)
        local nw, dw, vn, vd;

        nw := numer(w);
        dw := denom(w);
        vn := 0;
        vd := 0;

        if w = 0 then
            RETURN(0)
        else
            while nw mod p = 0 do
                nw := nw / p;
                vn := vn + 1
            od;
            while dw mod p = 0 do
                dw := dw / p;
                vd := vd + 1
            od;
            RETURN(vn-vd)
        fi

        end:


###     minimum extension valuation

        vstar := proc (h, p)
        local j, m, v, w, first;

        m := degree(h,x);
        first := true;

        for j from 1 to m do
        if coeff(h,x,m-j) <> 0 then
            w := ordp(coeff(h,x,m-j),p) / j;
            if first then
                v := w
            elif w < v then
                v := w
            fi;
            first := false
        fi
        od;

        if first then
            v := 0
        fi;

        RETURN(v)

        end:
```

Appendix III — Examples

The different steps of the algorithm are illustrated in two first examples; the third is an extreme case for which the Round Two algorithm is faster. We also give tables of comparative performance statistics for several other examples.

All computations for these examples were done with version 1.38.40 of PARI on a dual-processor Sparc 10 system at the Centre de Recherche en Mathématiques de Bordeaux.

Example 1.

Let $f = x^4 + 3x^3 - x^2 + 8x + 8$, a monic separable polynomial in $\mathbb{Z}[x]$, and let $\mathcal{A}_f = \mathbb{Q}[x]/f\mathbb{Q}[x]$. Compute

$$\texttt{discrim}(f) = -3 \cdot 13 \cdot 19 \cdot 31 \cdot 2^5.$$

1.1 A 2-maximal order of $\mathcal{A}_f$.

Define $\mathcal{A}_{f,2} = \mathbb{Q}_2[x]/f\mathbb{Q}_2[x]$ and let $\mathcal{O}_{f,2}$ be the 2-maximal order of $\mathcal{A}_{f,2}$.

The element $\xi = x + f\mathbb{Q}_2[x]$ is a non-primary element of $\mathcal{O}_{f,2}$. So we construct orthogonal idempotents and decompose the algebra as the direct sum of subalgebras of lower degree.

We seek orthogonal idempotents e and $1 - e$ and polynomials f_1 and f_2 such that

$$\mathcal{A}_{f,2} = \mathcal{A}_{f_1,2} \oplus \mathcal{A}_{f_2,2} = \mathbb{Q}_2[x]/f_1\mathbb{Q}_2[x] \oplus \mathbb{Q}_2[x]/f_2\mathbb{Q}_2[x].$$

The `decomp` procedure gives

$$e = 47077x^3 + 12438x^2 + 58712x + 27857,$$
$$f_1 = x^2 + 32x + 184, \quad f_2 = x^2 + 483x + 231.$$

We recursively compute 2-maximal orders $\mathcal{O}_{f_1,2}$ of $\mathcal{A}_{f_1,2}$, $\mathcal{O}_{f_2,2}$ of $\mathcal{A}_{f_2,2}$.

1.1.1 Integral basis for $\mathcal{O}_{f_1,2}$.

Given $f_1 = x^2 + 32x + 184$ and $\xi_1 = x + f_1\mathbb{Q}_2[x]$, the `nilord` procedure constructs an Eisenstein element $\frac{3}{2}(\xi_1 + 2)$ with minimal polynomial $x^2 + 42x + 279$.

So a basis for the 2-maximal order of $\mathcal{A}_{f_1,2}$ is:

$$\left[1,\ \tfrac{1}{2}x\right]$$

Let $w_{1,f_1}(\xi_1)$, $w_{2,f_1}(\xi_1)$ be the two vectors for this basis.

1.1.2 Integral basis for $\mathcal{O}_{f_2,2}$.

For $f_2 = x^2 + 483x + 231$ the Dedekind test gives a basis for $\mathcal{O}_{f_2,2}$:

$$[1,\ x]\,.$$

Define $\xi_2 = x + f_2\mathbb{Q}_2[x]$ and let $w_{1,f_2}(\xi_2)$, $w_{2,f_2}(\xi_2)$ be the two vectors for this basis.

1.1.3 Integral basis for $\mathcal{O}_{f,2}$.

The 2-maximal order of $\mathcal{A}_{f,2}$ can be constructed directly by mapping integral bases of $\mathcal{A}_{f_1,2}$ and $\mathcal{A}_{f_2,2}$ into $\mathcal{A}_{f,2}$.

The Hermite Form of

$$\left[e(\xi)w_{1,f_1}(\xi),\ e(\xi)w_{2,f_1}(\xi),\ (1-e)(\xi)w_{1,f_2}(\xi),\ (1-e)(\xi)w_{2,f_2}(\xi)\right]$$

gives a basis for $\mathcal{O}_{f,2}$:

$$\left[1,\ x,\ x^2,\ \tfrac{1}{2}(x^3+x^2+x)\right].$$

1.2 Integral basis for $\mathcal{A}_f$.

Only $p = 2$ divides F, so the integral basis is the 2-maximal order of $\mathcal{A}_f$:

$$\left[1,\ x,\ x^2,\ \tfrac{1}{2}(x^3+x^2+x)\right]$$

and the discriminant of $\mathcal{A}_f$ is: $-3\cdot 13\cdot 19\cdot 31\cdot 2^3 = -183768$.

Example 2.

Let $f = x^5 + 4x^4 + 19x^3 + 3x^2 + 12x + 9$, a monic separable polynomial in $\mathbb{Z}[x]$, and let $\mathcal{A}_f = \mathbb{Q}[x]/f\mathbb{Q}[x]$. Compute

$$\texttt{discrim}(f) = 17 \cdot 42239 \cdot 2^8 \cdot 3^4.$$

2.1 A 2-maximal order of $\mathcal{A}_f$.

Define $\mathcal{A}_{f,2} = \mathbb{Q}_2[x]/f\mathbb{Q}_2[x]$ and let $\mathcal{O}_{f,2}$ be the 2-maximal order of $\mathcal{A}_{f,2}$.

The `decomp` procedure gives:

$$e = 52756x^4 + 32469x^3 + 21251x^2 + 56955x + 31530,$$
$$f_1 = x^3 + 191x^2 + 207x + 833, \quad f_2 = x^2 + 1861x + 713.$$

We recursively compute 2-maximal orders $\mathcal{O}_{f_1,2}$ of $\mathcal{A}_{f_1,2}$, $\mathcal{O}_{f_2,2}$ of $\mathcal{A}_{f_2,2}$.

2.1.1 A 2-maximal order of $\mathcal{O}_{f_1,2}$.

For $f_1 = x^3 + 191x^2 + 207x + 833$ the `decomp` procedure gives:

$$e = \tfrac{1}{4}(124823x^2 + 69006x + 47623),$$
$$f_{1,1} = x^2 + 162x + 1653, \quad f_{1,2} = x + 29.$$

2.1.1.1 Integral basis for $\mathcal{O}_{f_{1,1},2}$.

Given $f_{1,1} = x^2+162x+1653$ and $\xi_{1,1} = x+f_{1,1}\mathbb{Q}_2[x]$, the `nilord` procedure constructs an Eisenstein element $\frac{3}{2}(\xi_{1,1} + 1)$ with minimal polynomial $x^2 + 240x + 3357$.

So a basis for the 2-maximal order of $\mathcal{A}_{f_{1,1},2}$ is:

$$\left[1,\ \tfrac{1}{2}(x+1)\right].$$

2.1.1.2 Integral basis for $\mathcal{O}_{f_{1,2},2}$.

Given $f_{1,2} = x + 29$ the Dedekind test gives a basis for $\mathcal{O}_{f_{1,2},2}$:

$$[1,\ x].$$

2.1.1.3 Integral basis for $\mathcal{O}_{f_1,2}$.

The Hermite Form of

$$\big[\, e_{1,1}(\xi_1)w_{1,f_{1,1}}(\xi_1),\ e_{1,1}(\xi_1)w_{2,f_{1,1}}(\xi_1),\\ e_{1,2}(\xi_1)w_{1,f_{1,2}}(\xi_1),\ e_{1,2}(\xi_1)w_{2,f_{1,2}}(\xi_1)\,\big]$$

gives a basis for $\mathcal{O}_{f_1,2}$:

$$\left[1,\ \tfrac{1}{2}(x+1),\ \tfrac{1}{4}(x^2-1)\right].$$

Let $w_{1,f_1}(\xi_1)$, $w_{2,f_1}(\xi_1)$, $w_{3,f_1}(\xi_1)$ be these three vectors.

2.1.2 Integral basis for $\mathcal{O}_{f_2,2}$.

For $f_2 = x^2 + 1861x + 713$ the Dedekind test gives a basis for $\mathcal{O}_{f_{1,2},2}$:

$$[1,\ x].$$

Define $\xi_2 = x + f_2\mathbb{Q}_2[x]$ and let $w_{1,f_2}(\xi_2)$, $w_{2,f_2}(\xi_2)$ be the two vectors for this basis.

2.1.3 Integral basis for $\mathcal{O}_{f,2}$.

The 2-maximal order of $\mathcal{A}_{f,2}$ can be constructed directly by mapping integral bases of $\mathcal{A}_{f_1,2}$ and $\mathcal{A}_{f_2,2}$ into $\mathcal{A}_{f,2}$.

The Hermite Form of

$$\big[\, e_1(\xi)w_{1,f_1}(\xi),\ e_1(\xi)w_{2,f_1}(\xi),\ e_1(\xi)w_{3,f_1}(\xi),\\ e_2(\xi)w_{1,f_2}(\xi),\ e_2(\xi)w_{2,f_2}(\xi)\,\big].$$

gives a basis for the 2-maximal order:

$$\left[1,\ x,\ x^2,\ \tfrac{1}{2}(x^3+1),\ \tfrac{1}{4}(x^4+x^3-x-1)\right].$$

2.2 A 3-maximal order of $\mathcal{A}_f$.

Define $\mathcal{A}_{f,3} = \mathbb{Q}_3[x]/f\mathbb{Q}_3[x]$ and let $\mathcal{O}_{f,3}$ be the 3-maximal order of $\mathcal{A}_{f,3}$.

The `decomp` procedure gives:

$$\begin{gathered} e = 4755x^4 + 1610x^3 + 5814x^2 + 5955x + 2071, \\ f_1 = x^3 + 12x + 9, \quad f_2 = x^2 + 4x + 7. \end{gathered}$$

We recursively compute 3-maximal orders $\mathcal{O}_{f_1,3}$ of $\mathcal{A}_{f_1,3}$, $\mathcal{O}_{f_2,3}$ of $\mathcal{A}_{f_2,3}$.

A basis for $\mathcal{O}_{f_1,3}$ is $[1, x, \frac{1}{3}x^2]$ and for $\mathcal{O}_{f_2,3}$ is $[1, x]$.

So, a basis for $\mathcal{O}_{f,3}$ is

$$\left[1,\ x,\ x^2,\ x^3,\ \tfrac{1}{3}(x^4 + x^3 + x^2)\right].$$

2.3 Integral basis for $\mathcal{A}_f$:

Computing

$$\begin{aligned} &\mathrm{HNF}\Big(12\mathcal{O}_{f,2}\ \Big|\ 12\mathcal{O}_{f,3}\Big) \\ &\qquad = \mathrm{HNF}\begin{pmatrix} 12 & 0 & 0 & 6 & -3 & 12 & 0 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & -3 & 0 & 12 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 & 0 & 0 & 12 & 0 & 4 \\ 0 & 0 & 0 & 6 & 3 & 0 & 0 & 0 & 12 & 4 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \end{pmatrix} \end{aligned}$$

gives an integral basis for $\mathcal{A}_f$:

$$\left[1,\ x,\ x^2,\ \tfrac{1}{2}(x^3 + 1),\ \tfrac{1}{12}(x^4 + x^3 + 4x^2 + 3x + 3)\right].$$

Example 3.

This example is from [Ash, Pinch & Taylor, 1991]. Let

$$\begin{aligned} f(x) = x^{12} & \\ & - 181170x^{11} \\ & + 13676070375x^{10} \\ & - 564635734535475x^{9} \\ & + 14120575648656756795x^{8} \\ & - 224213861531349946866060x^{7} \\ & + 2299324928127100837257833640x^{6} \\ & - 15120132032108410885407953780505x^{5} \\ & + 61607021939453175254804920116967515x^{4} \\ & - 144536083330213614666317706146365094565x^{3} \\ & + 170426077617455313511361437803852538934904x^{2} \\ & - 83139235455474245627641509862888062014092560x \\ & + 12253655221465755667504199645608996691723374656. \end{aligned}$$

Then f is irreducible in $\mathbb{Z}[x]$. Factoring the discriminant of f requires 46.15 CPU-seconds and gives

$$\begin{aligned} 2^{54} \cdot 3^{210} \cdot 61^{98} \cdot 233^{2} \cdot 419^{8} \cdot 1627^{6} \cdot 246319^{2} \cdot 1986499^{8} \cdot 156994183^{2} & \\ \cdot 102830099^{2} \cdot 369279563^{2} \cdot 712707529^{6} \cdot 63568512603919^{2}. & \end{aligned}$$

Computing the discriminant of $\mathbb{Q}[x]/f\mathbb{Q}[x]$ requires 208.41 CPU-seconds, giving

$$1397546311175017849 = 3^{6} \cdot 61^{8}.$$

So the index is

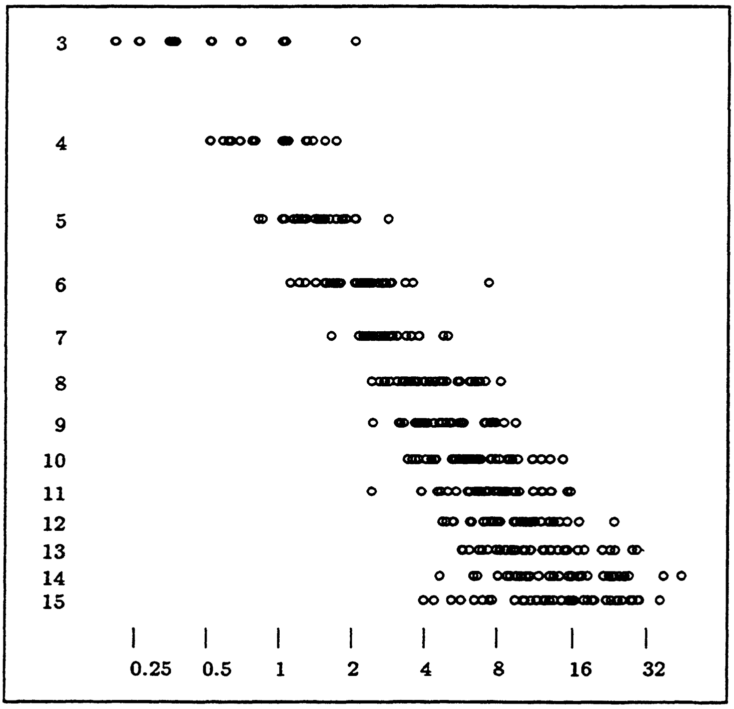
$$\begin{aligned} 2^{27} \cdot 3^{102} \cdot 61^{45} \cdot 233 \cdot 419^{4} \cdot 1627^{3} \cdot 246319 \cdot 1986499^{4} \cdot 156994183 & \\ \cdot 102830099 \cdot 369279563 \cdot 712707529^{3} \cdot 63568512603919. & \end{aligned}$$

Remark: For this example the Round Two algorithm is faster than Round Four (Round Two takes 120 CPU-seconds). However, the computation of each p-maximal order is faster with Round Four, except when $p = 61$. This case takes a considerable amount of time. Apparently working modulo such a large power of 61 is slower than handling matrices of size $12^2 \times 12$. If a means were known to identify such cases in advance, a better algorithm (combining Round Two and Round Four) could be found.

Statistical Comparison of Round Four to Round Two.

Using the 650 test polynomials of degrees 3 through 15 from [Ford, 1978] we identified 468 pairs (f, p) for which the Dedekind test for f does *not* immediately give a p-maximal order. A least-squares straight line fit to the points $(\ln n, \ln t_2/t_4)$ for these pairs gives the relation

$$\frac{t_2}{t_4} \approx 0.0426 n^{2.16}.$$



Execution Times: n vs t_2/t_4

Comparisons of Individual Examples.

We compare the CPU times used by the Round Four and Round Two algorithms for several examples. Times are given in milliseconds, with the time required to factorize the polynomial discriminant omitted.

Polynomial	Rd 4	Rd 2	Ratio
$x^9 - 2x^4 - 10x^3 + x - 2$	470	1550	3.3
$x^9 - 2x^5 + 17x^3 + 4$	270	920	3.4
$x^9 - 2x^3 - 10$	200	2940	14.7
$x^{10} + 7x^9 - 2x^8 - 2x^7 - 3x^5 + x^4 + 1$	70	810	11.6
$x^{10} - 4x^9 - 8x^5 + 5x^4 + 1$	70	1890	27.0
$x^{10} - 2x^9 - 15$	270	3900	14.4
$x^{11} + x^8 - 2x^2 + 4$	810	4000	4.9
$x^{11} - x^6 - 2x^3 - 12x^2 - 6$	630	7050	11.1
$x^{11} - x^{10} - x^4 - 4$	560	7150	12.7
$x^{12} - 3x^9 + 4x^8 - x^6 - x^2 + 10$	120	3360	28.0
$x^{12} + 4x^{11} + 5x^{10} + 6x^6 - 3x^4 + 12$	640	10920	17.0
$x^{12} + x^9 - 9x^7 - 2x^6 - 9x^5 - 6$	80	2080	26.0
$x^{13} + 6x^{10} - 10x^5 + 9x^2 - 2$	230	4290	18.7
$x^{13} + x^{10} + x^9 - 4x^8 - x^4 + x^2 - 1$	130	4730	36.4
$x^{13} + x^{11} - 8$	1100	17500	15.9
$x^{14} - x^{12} - x^7 + 10x^5 - 4$	710	14630	20.6
$x^{14} + 2x^8 + 6x - 1$	810	5280	6.5
$x^{14} - 8x^7 + 418$	550	19910	36.2
$x^{15} + 4x^{11} + 12x^{10} + x^3 - 4$	900	15360	17.1
$x^{15} + 9x^5 + 1$	1210	13030	10.8
$x^{15} - 13x^5 - 2$	420	17400	41.4
$x^{15} - 30x^{13} + 360x^{11} - 2200x^9 + 7200x^7$ $-12096x^5 + 8960x^3 - 120x - 249$	70	7670	109.5
$x^{15} - 30x^{13} + 360x^{11} - 2200x^9 + 7200x^7$ $-12096x^5 + 8960x^3 - 120x - 257$	1370	11670	8.5
$x^{16} + 132x^{14} + 6868x^{12} + 179570x^{10}$ $+2494972x^8 + 18111820x^6 + 65000173x^4$ $+102234000x^2 + 46240000$	5010	87230	17.4
$x^{21} - 42x^{19} + 756x^{17} - 7616x^{15} + 47040x^{13}$ $-183456x^{11} + 448448x^9 - 658944x^7$ $+532224x^5 - 197120x^3 + 21504x - 1691$	670	67720	101.0

David Ford
Centre Interuniversitaire en Calcul Mathématique Algébrique
Department of Computer Science
Concordia University
1455 de Maisonneuve Boulevard West
Montréal, Québec
Canada

Pascal Letard
Laboratoire d'Algorithmique Arithmétique et Expérimentale
Université Bordeaux I CNRS
351 cours de la Libération
33405 Talence Cedex,
France