JOURNAL de Théorie des Nombres de BORDEAUX

anciennement Séminaire de Théorie des Nombres de Bordeaux

Karim BELABAS, Mark VAN HOEIJ, Jürgen KLÜNERS et Allan STEEL

Factoring polynomials over global fields

Tome 21, n° 1 (2009), p. 15-39.

<http://jtnb.cedram.org/item?id=JTNB_2009__21_1_15_0>

Journal de Théorie des Nombres
de Bordeaux **21** (2009), 15-39

Factoring polynomials over global fields

par Karim BELABAS, Mark VAN HOEIJ, Jürgen KLÜNERS
et Allan STEEL

Résumé. Nous démontrons une complexité polynomiale en temps pour l'algorithme de van Hoeij de factorisation de polynômes univariés à coefficients rationnels, ainsi que pour des variantes naturelles. En particulier, notre approche fournit aussi une complexité polynomiale pour les polynômes bivariés sur un corps fini.

Abstract. We prove that van Hoeij's original algorithm to factor univariate polynomials over the rationals runs in polynomial time, as well as natural variants. In particular, our approach also yields polynomial time complexity results for bivariate polynomials over a finite field.

1. Introduction

Let K be a global field. The goal of this paper is to present a practical algorithm which factors a polynomial $f \in K[X]$ in polynomial time. In the first three sections we treat the part that applies to all global fields; but we only complete the work for two global fields $K = \mathbb{F}_q(t)$ and $K = \mathbb{Q}$, see Sections 4 and 5 (the general case is more technical; to treat algebraic number fields we have to combine the first three sections of this paper with techniques from [Bel03]).

Let v be a prime ideal of a maximal order $\mathcal{O}$ of K. We denote by K_v the completion of K at v with maximal order $\mathcal{O}_v$ and denote by k the residue class field $\mathcal{O}/v$.

- If $K = \mathbb{Q}$ then $\mathcal{O} = \mathbb{Z}$ and v is a prime ideal generated by a prime number, which we denote by v, too. K_v is the field of v-adic numbers, $\mathcal{O}_v$ its ring of v-adic integers, and $k = \mathbb{Z}/v\mathbb{Z}$.
- If $K = \mathbb{F}_q(t)$ then we choose $\mathcal{O} = \mathbb{F}_q[t]$ and a prime ideal v generated by an irreducible polynomial in $\mathbb{F}_q[t]$, which we denote by v, too. If α is a root of this polynomial in an algebraic closure of $\mathbb{F}_q$, then $k \cong \mathbb{F}_q(\alpha)$, $\mathcal{O}_v \cong k[[t-\alpha]]$ and $K_v \cong k((t-\alpha))$.

The method of Zassenhaus [Zas69] to factor in $K[X]$ reduces to the case that f is in $\mathcal{O}[X]$, the reduction $\bar{f}$ of f is separable in $k[X]$, and the leading coefficient $\mathrm{lc}(f)$ of f does not vanish mod v. Since k is finite we can factor $\bar{f}$

in $k[X]$ using well-known algorithms. By Hensel's lemma, the factorization of $\bar{f}$ lifts to a factorization

$$f = \mathrm{lc}(f) f_1 \cdots f_r$$

in $K_v[X]$, where $\mathrm{lc}(f) \in K \subset K_v$ and $f_1, \ldots, f_r$ are monic and irreducible in $K_v[X]$. Since $\mathrm{lc}(f)$ does not vanish mod v, we have $f_1, \ldots, f_r \in \mathcal{O}_v[X]$. In practice, elements of $\mathcal{O}_v$ are computed modulo v^ℓ for some $\ell > 0$ and lifted to $\mathcal{O}$. For $a \in \mathcal{O}_v$ we write "$a \bmod v^\ell$" for such a lift of a to $\mathcal{O}$. This notation is extended to $\mathcal{O}_v[X]$ coefficient-wise. By Hensel lifting the irreducible factors of $\bar{f}$ we can compute $f_1, \ldots, f_r \mod v^\ell$ for any fixed $\ell > 0$.

Let $g \in K[X]$ be a monic irreducible factor of f. Then its coefficients can be bounded in terms of f. Write

$$g = f_1^{e_1} \cdots f_r^{e_r},$$

where $e_i \in \{0, 1\}$ for all $1 \leq i \leq r$. If ℓ is large enough compared to the bound on the coefficients of g, we may test for given $e_1, \ldots, e_r \in \{0, 1\}$ whether $f_1^{e_1} \cdots f_r^{e_r} \in K[X]$ by computing $\mathrm{lc}(f) f_1^{e_1} \cdots f_r^{e_r} \bmod v^\ell$ and checking whether this divides f in $K[X]$. This time, the lift "$\ldots \bmod v^\ell$" to $\mathcal{O}[X]$ is not arbitrary. Choosing the right lift is straightforward if K is $\mathbb{Q}$ or $\mathbb{F}_q(t)$ since there are canonical minimal lifts to $\mathcal{O}$, but care is needed for general global fields (see [Bel03] for the number field case, and [PO06] for the general case).

This reduces the problem to a finite computation. The Zassenhaus algorithm finds the e_i by an exhaustive enumeration, which works very well if r is small or the K-rational irreducible factors are plentiful. Otherwise, we face combinatorial explosion and exponential behavior. That is, about 2^r combinations are considered, where r can be of the order of $\deg f$.

The landmark paper by Lenstra *et al.* [LLL82] avoids this combinatorial problem by constructing K-rational factors via lattice basis reduction (LLL-reduction). The original paper assumes $K = \mathbb{Q}$, but was generalized by Arjen Lenstra [Len82] (K a number field), then Pohst and Méndez [PO06, PM06] (K any global field). Unfortunately, although this algorithm runs in polynomial time, it is rather slow in practice since it requires Hensel lifting to huge accuracy, followed by the LLL-reduction of correspondingly huge lattices. Van Hoeij [Hoe02] gave a different solution to the combinatorial problem for $K = \mathbb{Q}$ by reducing it to a knapsack problem using power sums. Belabas [Bel03] generalized van Hoeij's algorithm to number fields, but these two papers stated no complexity bound. This approach is reminiscent of Sasaki *et al.* [SSH93] who used power sums to factor a bivariate $f \in k[X, Y]$ over a field k using k-linear algebra for the recombination step.

We shall describe a similar approach over a general global field K, and show it runs in polynomial time, although details will only be provided for

the cases $K = \mathbb{Q}$ and $K = \mathbb{F}_q(t)$. Our key idea is to replace power sums by logarithmic derivatives multiplied by the target polynomial f. This not only simplifies complexity proofs, but also has practical algorithmic advantages: see Example 4.3. This is related to Miller's [Mil92] idempotents approach, but has the advantage that one obtains a rigorous complexity proof. We give a high-level unified description of the new algorithm in §3, then treat in detail the cases $K = \mathbb{F}_q(t)$ and $K = \mathbb{Q}$ in §4 and §5, respectively.

2. Notation

Throughout the paper we use the following notation. Let K be a global field of characteristic $p \geq 0$ with maximal order $\mathcal{O}$. In particular, the notation $\mathbb{Z}/p\mathbb{Z}$ denotes either the finite prime field $\mathbb{F}_p$ ($p > 0$) or the infinite ring $\mathbb{Z}$ ($p = 0$). We wish to factor over $K[X]$ a separable polynomial $f \in \mathcal{O}[X]$ of degree $n > 1$. Let v be a prime ideal of $\mathcal{O}$ such that the leading coefficient $\mathrm{lc}(f) \in \mathcal{O}$ of f does not vanish mod v, and f mod v is still separable. We denote by K_v the completion of K at v, with maximal order $\mathcal{O}_v$, maximal ideal v and finite residue field k. Let $\bar{f}$ be the image of f in $k[X]$, which is a separable polynomial of degree n. In the number field case, we can work within a subring of $\mathcal{O}$ if the computation of $\mathcal{O}$ is too costly (see [Bel03]), but we ignore this aspect here.

We have the factorizations into monic irreducible factors

$$f = \mathrm{lc}(f) f_1 \cdots f_r \in \mathcal{O}_v[X], \quad \bar{f} = \mathrm{lc}(\bar{f}) \bar{f}_1 \cdots \bar{f}_r \in k[X], \quad \text{and} \quad f = \mathrm{lc}(f) g_1 \cdots g_s \in K[X].$$

Obviously, $\mathrm{lc}(f) g_i \in \mathcal{O}[X]$ and $1 \leq s \leq r \leq n$. We call the f_i the *local factors* and the g_j the *K-factors*. We fix an integer $\ell \geq 1$. We cannot compute $f_i \in \mathcal{O}_v[X]$ with infinite accuracy, but we can compute f_i mod v^ℓ, which is in $\mathcal{O}[X]$.

In order to hide logarithmic factors in complexity estimates, we use the customary notation $\widetilde{O}(f)$ for $f(\log f)^{O(1)}$. Finally, let $3 \geq \omega \geq 2$ be a feasible matrix multiplication exponent, so that two $n \times n$ matrices can be multiplied within $O(n^\omega)$ field operations. Let Id_r be the r by r identity matrix and let w^{tr} denote the transpose of the vector w.

3. General description

3.1. Sketch of the algorithm. Our method relies on two main ideas:

3.1.1. *Linearize.* The logarithmic derivative is a group homomorphism from the multiplicative group $K_v(X)^*$ to the additive group $K_v(X)$, with kernel $K_v(X^p)^*$. The first idea is to multiply this by f and consider the

group homomorphism:

$$\begin{aligned} \Phi : K_v(X)^*/K_v(X^p)^* &\to K_v(X) \\ g &\mapsto fg'/g. \end{aligned}$$

Lemma 3.1. *If g is in the subgroup of $K_v(X)^*/K_v(X^p)^*$ generated by the local factors f_i, then $\Phi(g) \in \mathcal{O}_v[X]$. If g is in the subgroup generated by the K-factors g_j, then $\Phi(g) \in \mathcal{O}[X]$.*

Proof. It is enough to prove the first statement for $g = f_i$ a local factor of f. It then becomes obvious since both f/g and g' belong to $\mathcal{O}_v[X]$ (recall that the f_i are monic and in $\mathcal{O}_v[X]$). It is enough to prove the second statement for $g = g_j$ an irreducible K-factor. Take any non-zero prime ideal of $\mathcal{O}$ and let w be the corresponding valuation on K, which is extended to a valuation on $K[X]$ by taking $w(\sum c_i X^i) = \min_i\ w(c_i)$. Again, $\Phi(g)$ is the product of g' and f/g and hence in $K[X]$. Since $w(g') \geq w(g)$ we get

$$w(\Phi(g)) = w(g'\, f/g) \geq w(g\, f/g) = w(f) \geq 0$$

since $f \in \mathcal{O}[X]$. So the valuation of $\Phi(g) \in K[X]$ is non-negative at any prime ideal of $\mathcal{O}$ and hence $\Phi(g) \in \mathcal{O}[X]$. □

Compared to [Hoe02], we have replaced power sums by f times the logarithmic derivative. If $g \in K[X]$ is a monic separable polynomial, the i-th power sum (i-th "trace") of g is:

$$\mathrm{Tr}_i(g) := \sum_{\alpha} \alpha^i,$$

where the sum is taken over the roots α of g in an algebraic closure of K. Despite the relation

$$g'/g = \sum_{i \geq 0} \mathrm{Tr}_i(g) X^{-i-1},$$

our present approach turns out to be more convenient for complexity proofs than power sums, and also has practical advantages when f is sparse or not monic.

3.1.2. ***Approximately solve knapsack.*** Let $G_v \subset K_v(X)^*/K_v(X^p)^*$ be the subgroup generated by the local factors. Our goal is to find the subgroup $G \subset G_v$ generated by the irreducible K-factors of f. To do this we construct a "knapsack lattice" L in a similar way as in [Hoe02], except that instead of traces (power sums) of f_j we use the coefficients of $\Phi(f_j)$, lifted mod v^ℓ to $\mathcal{O}$ (see Sections 4 and 5 for details).

Compute a basis of *small vectors* for L, using lattice reduction if the characteristic is $p = 0$ and $\mathbb{F}_p$-linear Gaussian elimination if $p > 0$. The image of K-rational factors are small and are thus supported on the small basis vectors. Discarding large basis vectors of L yields a sublattice $L' \subset L$,

associated to a subgroup G' of G_v, such that we still have $G \subset G'$. Note that L' and G' depend implicitly on ℓ.

3.1.3. ***Conclusion.*** We claim that $G = G'$ provided ℓ is large enough.

3.2. Sketch of proof. Again, we are deliberately vague here, leaving the details to Sections 4 and 5. The proof is by contradiction. If G' is strictly larger than G, then by Lemma 3.2 below, it contains an element, represented by a rational function $g \in K_v(X)^*$, such that

(1) At least one f_i divides $\Phi(g)$,
(2) None of the $\overline{g}_j$ divide $\overline{\Phi(g)}$ where the bar indicates reduction to $k[X]$.
(3) $\Phi(g) \bmod v^\ell =: H \in \mathcal{O}[X]$ is *small*. In particular, the size of H is polynomial in the size of f.

Let $R := \mathrm{Res}(f, H) \in \mathcal{O}$ be the resultant of f and H. Then

- $\mathrm{Res}(f, \Phi(g)) = 0$, hence $v^\ell \mid R$. In fact, $v^{\ell d} \mid R$, where d is the sum of the degrees of the f_i that divide $\Phi(g)$. Item (1) implies $d > 0$.
- $R \neq 0$, because if R were zero then H would be divisible by some g_j, so $\overline{H}$ would be divisible by $\overline{g}_j$, contradicting item (2).
- The size of R is bounded by a polynomial in the size of f and H, and hence (by item (3)) in the size of f.

These three points contradict each other if v^ℓ is larger than the upper bound for R. For any strictly larger ℓ, we have $G = G'$.

Lemma 3.2. *Suppose $G \subsetneq G'$ and let $(b_1, \dots, b_u)$ be any $(\mathbb{Z}/p\mathbb{Z})$-basis of $\Phi(G') \bmod v^\ell$. Then there exists an element $g \in G' \setminus G$ such that*

(1) *$f_i \mid \Phi(g) \in \mathcal{O}_v[X]$ for some $1 \leq i \leq r$.*
(2) *$\overline{g}_j \nmid \overline{\Phi(g)}$ for all $1 \leq j \leq s$.*
(3) *$H := \Phi(g) \bmod v^\ell$ is of the form $b_i + e\Phi(g_j) + \sum_{k \in T} \Phi(g_k)$ for some indices i, j, some set $T \subset \{1, \dots, s\}$, and some integer $e \in \mathbb{Z}$.*

Proof. Elements $g \in G_v$ can be written in the form $g = f_1^{e_1} \cdots f_r^{e_r} \cdot K_v(X^p)^*$, where the integers e_i are defined mod p. We view e_i as an element of $\mathbb{Z}/p\mathbb{Z}$, and then define the support of g as $\mathrm{Supp}\, g = \{i \colon e_i \neq 0\}$. Since the f_i are pairwise coprime and irreducible in $\mathcal{O}_v[X]$, we have

$$f_i \mid \Phi(f_j) \iff i \neq j.$$

So $f_i \mid \Phi(g)$ iff e_i is zero in $\mathbb{Z}/p\mathbb{Z}$, so that $g_j \mid \Phi(g)$ iff $\mathrm{Supp}\, g \bigcap \mathrm{Supp}\, g_j = \emptyset$.

The supports of $g_1, \dots, g_s$ form a partition of $\{1, \dots, r\}$. Choose any element $g \in G' \setminus G$. For all $1 \leq j \leq s$ with $\mathrm{Supp}\, g_j \cap \mathrm{Supp}\, g = \emptyset$, replace g by $g_j g$. Then condition (2) is satisfied (recall that $\overline{f}$ is separable), and g is still in $G' \setminus G$. Write this g as $f_1^{e_1} \cdots f_r^{e_r} \cdot K_v(X^p)^*$ as above. Since g is not in the group G generated by $g_1, \dots, g_s$, there must be some g_j for which $S_j := \{e_i \colon i \in \mathrm{Supp}\, g_j\}$ contains more than one element. Take an

element $e \in S_j$ and replace g by g/g_j^e. Now g satisfies conditions (1), (2) and (3). □

We have sketched a general proof and omitted the details. Filling in these details is easy for the case $K = \mathbb{F}_q(t)$ (discussed in Section 4). The details for $K = \mathbb{Q}$ require more work, so are deferred to the last section.

3.3. Further general remarks. Concretely, the $(\mathbb{Z}/p\mathbb{Z})$-module G is represented as follows: For $1 \le j \le s$ write the monic irreducible K-factors as $g_j = f_1^{w_{j,1}} \cdots f_r^{w_{j,r}}$ with $w_{j,1}, \ldots, w_{j,r} \in \{0,1\}$ and write $w_j := (w_{j,1}, \ldots, w_{j,r})^{\mathrm{tr}} \in \mathbb{Z}^r$. Now let $W = (\mathbb{Z}/p\mathbb{Z})w_1 + \cdots + (\mathbb{Z}/p\mathbb{Z})w_s$. Knowing W enables us to compute the factors g_j of f. Indeed, given any $(\mathbb{Z}/p\mathbb{Z})$-basis $u_1, \ldots, u_s$ of W, we can find $\{w_1, \ldots, w_s\}$ by computing the reduced echelon form of $u_1, \ldots, u_s$, or by using the following shortcut:

Lemma 3.3. *For $1 \le a, b \le r$, write $a \sim b$ if the a-th and b-th entry of u are the same for every u in $u_1, \ldots, u_s$. This is an equivalence relation and each equivalence class $cl(a)$ corresponds to a unique $w_j = (w_{j,b})_{1 \le b \le r}$ (with $w_{j,b} = 1$ if $a \sim b$ and 0 otherwise).*

In order to determine W, let L be some subspace of $(\mathbb{Z}/p\mathbb{Z})^r$ that contains W, for instance $L = (\mathbb{Z}/p\mathbb{Z})^r$. The following lemma enables us to check whether $L = W$ (compare [Hoe02, Lemma 2.8]).

Lemma 3.4. *Let $L \supseteq W$ be generated by a $(\mathbb{Z}/p\mathbb{Z})$-basis $u_1, \ldots, u_{\tilde{s}}$ of $0-1$ vectors $u_i = (u_{i,1}, \ldots, u_{i,r})$ $(1 \le i \le \tilde{s})$. We assume that*

$$\tilde{g}_i(X) := \prod_{j=1}^{r} f_j^{u_{i,j}} \in K[X] \quad \text{for } 1 \le i \le \tilde{s}.$$

Then $s = \tilde{s}$ and $W = L$. Furthermore, if

$(*)$ *exactly r elements $u_{i,j}$ are equal to 1 (for $1 \le i \le \tilde{s}$, $1 \le j \le r$),*

then $\{\tilde{g}_i \colon 1 \le i \le s\} = \{g_i \colon 1 \le i \le s\}$.

Proof. Since the g_j are irreducible in $K[X]$, each $\tilde{g}_i$ is a product of some g_j's, hence $L \subset W$. Since $W \subset L$ by assumption, it follows that $W = L$ and $s = \tilde{s}$. Condition $(*)$ forces the $\tilde{g}_i$ to be coprime (otherwise, one f_j is unused and we cannot have $L = W$). Then $\prod_{i \le s} \tilde{g}_i = f/\operatorname{lc}(f) = \prod_{i \le s} g_i$ and we are done since none of the $\tilde{g}_i$ is equal to 1 (otherwise $u_i = 0$ could not be part of a basis). □

The criterion is used as in Zassenhaus's algorithm: given $(u_{i,j})$, compute the $\tilde{g}_i \bmod v^\ell$ from the $f_j \bmod v^\ell$ and check whether each $\tilde{g}_i$ divides f. We may first use Lemma 3.3 to ensure condition $(*)$ is satisfied before attempting to reconstruct the $\tilde{g}_i$.

4. The case $K = \mathbb{F}_q(t)$

4.1. Representing $\mathcal{O}_v$. In this section $p > 0$, q is a power of p, $\mathcal{O} = \mathbb{F}_q[t]$, and the prime ideal v is generated by an irreducible polynomial in $\mathbb{F}_q[t]$, which we shall also denote as v. Let $f \in \mathcal{O}[X]$ be a separable polynomial. We wish to factor f, viewed as element of $\mathbb{F}_q(t)[X]$. Let α be a root of $v \in \mathbb{F}_q[t]$ in an algebraic closure of $\mathbb{F}_q$, so that the residue field $k = \mathbb{F}_q[t]/(v)$ is isomorphic to $\mathbb{F}_q(\alpha)$. Representing $t - \alpha$ with a new variable $\tilde{t}$, the map $t \mapsto \tilde{t} + \alpha$ is an isomorphism from $\mathbb{F}_q[t]/(v^\ell)$ to $\mathbb{F}_q(\alpha)[\tilde{t}]/(\tilde{t}^\ell)$. Taking limits, we obtain an isomorphism from

$$\mathcal{O}_v = \varprojlim_\ell \mathbb{F}_q[t]/(v^\ell)$$

to

$$\mathbb{F}_q(\alpha)[[\tilde{t}]] = \varprojlim_\ell \mathbb{F}_q(\alpha)[\tilde{t}]/(\tilde{t}^\ell).$$

If $g \in \mathcal{O}_v[X]$ we write "$g \bmod v^\ell$" for the unique lift of g to $\mathbb{F}_q[t, X]$ such that

$$\deg_t(g) < \deg_t(v^\ell) = \ell \deg v,$$

where $\deg_t$ denotes the degree with respect to t. We cannot compute $f_i \in \mathcal{O}_v$ with infinite accuracy; however, for any integer $\ell > 0$ we can compute $f_i \bmod v^\ell$, which is an element of $\mathbb{F}_q[t, X]$.

Note that the above technicalities with $\mathcal{O}_v$ become easier if we take $v = t$ so that $\tilde{t} = t$. However, we can not always do this: we can only take $v = t$ if $f(0, X)$ is separable and has degree n (and it may be impossible to transfer to such a situation via $\mathbb{F}_q$-linear transformations).

4.2. Bounds. An important ingredient of our algorithm is a bound for the coefficients of $\Phi(g)$, where g is a factor of f.

Lemma 4.1. *Let* $B_i := \min\big(\deg_t(f),\ \tilde{n} - i - 1\big)$ *where* $\tilde{n}$ *is the* total degree *of* f *as a bivariate polynomial. Let* $g \in \mathbb{F}_q[t][X]$ *be a polynomial which divides* f*. Then*

$$\Phi(g) = \sum_{i=0}^{n-1} a_i(t) X^i \in \mathbb{F}_q[t][X] \quad \textit{with } \deg_t(a_i) \le B_i.$$

Proof. From $\Phi(g) = fg'/g$ we get $\deg_t(\Phi(g)) + \deg_t(g) = \deg_t(g') + \deg_t(f)$. Since $\deg_t(g') \le \deg_t(g)$ we get $\deg a_i \le \deg_t(f)$. The second bound is proven in the same way. □

Example. Let $\mathbb{F}_q = \mathbb{F}_{41}$, $f = X^{10} + t^2X^8 + tX^5 + 1$. We obtain $B_0 = \cdots = B_7 = 2$, $B_8 = 1$, $B_9 = 0$.

With a little effort we can get a better bound. Denote by $N(f) \subset \mathbb{R}^2$ the Newton polygon of f, which is defined as the convex hull of all points (i, j) for which the coefficient of t^jX^i in f is non-zero.

Lemma 4.2. *Let* $B_i := \sup\{j \in \mathbb{N}\colon (i+1,j) \in N(f)\}$ *and suppose* $g \in \mathbb{F}_q[t][X]$ *is a polynomial which divides* f*. Then*

$$\Phi(g) = \sum_{i=0}^{n-1} a_i(t)X^i \in \mathbb{F}_q[t][X] \text{ with } \deg(a_i) \le B_i.$$

Proof. If $S_1, S_2 \subset \mathbb{R}^2$ then define $S_1 + S_2 := \{s_1 + s_2\colon s_1 \in S_1, s_2 \in S_2\}$. It is well known that $N(gh) = N(g) + N(h)$ for all $g, h \in \mathbb{F}_q[t, X]$. It is also clear that $N(g') \subset N(g) + \{(-1,0)\}$. Then

$$\begin{aligned} N(f/g \cdot g') &= N(f/g) + N(g') \\ &\subset N(f/g) + N(g) + \{(-1,0)\} = N(f) + \{(-1,0)\}. \end{aligned}$$

If $(i,j) \in N(\Phi(g)) \subset N(f) + \{(-1,0)\}$, then $(i+1,j) \in N(f)$. □

4.3. The main idea. We use the notation from §3.3. We start with $L = (\mathbb{Z}/p\mathbb{Z})^r$ containing W. Our goal is to compute a subspace $L' \subseteq L$ which still contains W. This procedure is repeated until we reach $L' = W$, which can be checked using Lemmata 3.3 and 3.4, given a basis of L'.

We now explain how to produce L'. Choose a precision ℓ and let $g \in G_v$. If the t-degree of one of the coefficients of $\Phi(g)$ mod v^ℓ exceeds the degree bound B_i, then g is not a K-factor of f. We use this to replace the Zassenhaus combinatorial search by linear algebra. More precisely, let $m_i = B_i + 1$ and $\sigma = \deg_t(v^\ell)$. Write

$$\Phi(f_j) \bmod v^\ell = \sum_{i=0}^{n-1} a_{i,j}X^i \quad (1 \le j \le r)$$

so the $a_{i,j}$ are in $\mathbb{F}_q[t]$ and have degree $< \sigma$. Define

$$\phi_{m_i}(c_0t^0 + \cdots + c_{\sigma-1}t^{\sigma-1}) := (c_{m_i}, \ldots, c_{\sigma-1})^{\mathrm{tr}}$$

and

$$A_i := (\phi_{m_i}(a_{i,1}) \cdots \phi_{m_i}(a_{i,r})) \tag{4.1}$$

which is a $(\sigma - m_i) \times r$ matrix with entries in $\mathbb{F}_q$, satisfying $A_i e = 0$ for all $e \in W$. Now let L' be the intersection of the kernels of $A_1, \ldots, A_{n-1}$ (viewed as $\mathbb{F}_p$-linear maps from L to $\mathbb{F}_q^{\sigma-m_i}$). Then L' contains W. Note that L and W are subspaces of $\mathbb{F}_p^r$ and A_i is defined over $\mathbb{F}_q$.

Let B be a degree bound for the resultant R in §3.1.3. Looking at the Sylvester matrix, Lemma 4.1 gives $B = \min\left((2n-1)\deg_t(f), \tilde{n}(\tilde{n}-1)\right)$, where $\tilde{n}$ is the total degree of f as bivariate polynomial. As in §3.1.3, one finds that L' must be W when $\deg_t(v^\ell) > B$. This proves

Theorem 4.1. *Let* B *be as above. If* $\deg_t(v^\ell) > B$ *and* L' *is the intersection of the kernels of* A_i*,* $i = 0, \ldots, n-1$ *then* $L' = W$*.*

This leads to an algorithm that factors a separable polynomial $f \in \mathbb{F}_q[t, X]$ in deterministic polynomial time, on input f, v and a (square-free) modular factorization of f modulo v. Such a modular factorization can be obtained from f only in *probabilistic* polynomial time, where the non-deterministic step lies in finding the roots of a degree n polynomial over the prime field $\mathbb{F}_p$. In particular, if $p = n^{O(1)}$ is small, a suitable v and the associated modular factorization can be obtained from f in deterministic polynomial time. We do not give a detailed running-time analysis of this algorithm, which the interested reader can find in [BLSSW04] (but see Corollary 4.1).

Now in that paper [BLSSW04], the authors followed our f times g'/g approach (presented in a previous version of the present paper) and, as a corollary of our quadratic bound $\tilde{n}(\tilde{n}-1)$, they obtained a *linear* bound when $p > \tilde{n}(\tilde{n}-1)$ is large enough. A striking consequence is

Theorem 4.2 (Bostan, Lecerf, Salvy, Schost, Wiebelt)**.** *If $p > \tilde{n}(\tilde{n}-1)$, then one can factor a polynomial $f \in \mathbb{F}_q[t][X]$ of total degree $\tilde{n}$ in $O(\tilde{n}^{\omega+1})$ operations in $\mathbb{F}_q$, given an oracle for univariate factorization in $\mathbb{F}_q[t]$. Allowing Las Vegas probabilistic algorithms, this drops down to $O(\tilde{n}^{\omega})$ expected operations.*

If f is monic and if we use the bound $B_i \leq \tilde{n} - i - 1$ from Lemma 4.1, then our present approach and the power sums in [Hoe02] are equivalent (see [BLSSW04]). However, if we use the sharper bound in Lemma 4.2, or if f is not monic, then the two approaches are no longer equivalent. In this case the f times g'/g approach is at least as good but can be better as the following examples show.

Example. We continue with Example 4.2 and take $v = t$. If we use the power sums approach, or equivalently, if we use the bound $B_i \leq \tilde{n} - i - 1$ from Lemma 4.1 then we need at least $\ell = 6$ to prove that f is irreducible. However, the Newton polygon bounds are $B_0 = \cdots = B_3 = 0$, $B_4 = \cdots = B_6 = 1$, $B_7 = 2$, $B_8 = 1$, $B_9 = 0$ in this example. Taking $\ell = 2$ we already get some relations, and $\ell = 3$ now suffices to prove irreducibility.

Our new approach has the additional advantage that a reduction to the monic case, which may increase the size of f, is not needed.

Example. If $\mathbb{F}_q = \mathbb{F}_{13}$, $f = (t^{10}-1)X^{10} + tX^2 + 1$ and $v = t$, our algorithm would factor f mod t, Hensel lift to $\ell = 4$ and then stop because at this point $\dim(L')$ becomes 1, which implies that f is irreducible.

Experimentally, this last example is typical: for irreducible f, one often reaches $\dim(L') = 1$ for a small value of ℓ. Among polynomials of given degree, we expect to prove irreducibility faster than we would find non-trivial factors.

4.4. Algorithms. A practical implementation should not use the precision bound $\deg_t(v^\ell) > B$ because the equations defined by the matrices A_i, $0 \le i < n$ could already be sufficient for smaller values of ℓ. Since we have to use quadratic Hensel lifting anyway, we generate the new equations after each Hensel lifting step, and produce a new lattice L' containing W. Then we check with Lemma 3.4 whether $L' = W$. If not we repeat this procedure. We now write formally such an algorithm. In order to simplify the presentation we will assume that we can choose $v = t$ (see §4.1 if this is not possible).

Denote $L' \subseteq \mathbb{F}_q^r$ as the solution set of the equations defined by the matrices A_i. In the following algorithm the matrix $N \in \mathbb{F}_q^{r\times d}$ describes a basis of L'. Note that the algorithm works more efficiently if instead of L' we work with $\mathbb{F}_p^r \bigcap L'$. see section 4.5 for details how to do this.

Algorithm 4.3. *(Bivariate Factorization, with simplifying assumptions)*

Input: $f \in \mathbb{F}_q[t][X]$, such that the leading coefficient $\mathrm{lc}(f)$ does not vanish at $t = 0$, and $f(0, x) \in \mathbb{F}_q[X]$ is squarefree.

Output: Factorization $f = \mathrm{lc}(f)g_1 \cdots g_s \in \mathbb{F}_q(t)[X]$.

Step 1: Compute $f \equiv \mathrm{lc}(f)f_1 \cdots f_r \mod t$ in $\mathbb{F}_q[X]$.

Step 2: Compute bounds B_i using Lemma 4.2.

Step 3: Set $\ell \leftarrow 1$ and $N \leftarrow \mathrm{Id}_r \in \mathbb{F}_q^{r\times r}$ (identity matrix).

Step 4: While not finished do

(1) Set $\ell \leftarrow 2\ell$.

(2) Compute $f \equiv \mathrm{lc}(f)f_1 \cdots f_r \mod t^\ell$ using Hensel lifting.

(3) Compute $\Phi(f_j) \bmod t^\ell = \sum_{i=0}^{n-1} a_{i,j}(t)X^i \quad (1 \le j \le r)$.

(4) For $i \in \left\{\tilde{i} \in \{0, \ldots n-1\} : \ell/2 \ge B_{\tilde{i}} + 1\right\}$ do

(a) Write (for fixed i) $a_{i,j} = \sum_{k=0}^{\ell-1} c_{k,j}t^k$ $(1 \le j \le r)$.

(b) Set $k_0 \leftarrow \min\{B_i + 1, \ell/2\}$.

(c) Set $C_i \leftarrow \begin{pmatrix} c_{k_0,1} & \cdots & c_{k_0,r} \\ \vdots & \ddots & \vdots \\ c_{\ell-1,1} & \cdots & c_{\ell-1,r} \end{pmatrix} \in \mathbb{F}_q^{(\ell-k_0)\times r}$.

(d) Compute $\tilde{N} \in \mathbb{F}_q^{d\times\tilde{d}}$ of largest rank such that $(C_iN)\tilde{N} = 0$.

(e) $N \leftarrow N\tilde{N} \in \mathbb{F}_q^{r\times d}$, where d is the dimension of the current lattice L' corresponding to N.

(5) If $d = 1$ then f is irreducible and return f.

(6) Check using Lemma 3.4 if the lattice L' corresponding to N equals W. If so, return the factors g_i.

The assumption that f is squarefree is reasonable since squarefree factorization in $\mathbb{F}_q[t, X]$ is not expensive. But the assumption that $t = 0$ is a good evaluation point cannot be ignored. This assumption might not hold for any evaluation point in $\mathbb{F}_q$, in which case we need to use a prime ideal v of degree > 1. To handle this case, we implement the isomorphism between $\mathbb{F}_q[t]/(v^\ell)$ and $\mathbb{F}_q(\alpha)[\tilde{t}]/(\tilde{t}^\ell)$ mentioned in §4.1, using linear algebra to determine the inverse isomorphism.

It might happen that the check in Step 4 (6) fails even if $L' = W$ because ℓ is too small to compute the factors. In this case, additional Hensel lifting is necessary, but all subsequent matrices $C_i N$ are equal to 0.

4.5. Practical improvements. In the case when q is not a prime number but a prime power the following improvement helps in practical examples and improves the complexity when $n > p$ (see Theorem 4.4).

For $q = p^w$ write $\mathbb{F}_q = \mathbb{F}_p\gamma_1 + \cdots + \mathbb{F}_p\gamma_w$ and define

$$\psi : \mathbb{F}_q \to \mathbb{F}_p^w, \quad \sum_{l=1}^{w} c_l\gamma_l \mapsto (c_1, \ldots, c_w)^{\mathrm{tr}}.$$

Let A_i be as in equation (4.1). Then we define $\tilde{A}_i$ as follows: replace every entry c of A_i by $\psi(c)$. Since $\psi(c)$ is a column vector with w entries, $\tilde{A}_i$ is a $w(\sigma - m_i) \times r$ matrix with entries in $\mathbb{F}_p$ and we still have $\tilde{A}_i e = 0$ for all $e \in W$. Now let L' be the intersection of the kernels of $\tilde{A}_0, \ldots, \tilde{A}_{n-1}$. Then L' contains W. It is obvious that this intersection is contained in the intersection obtained when we compute the kernels of the A_i. The following example shows that it can be strictly smaller.

Example. Take a positive integer a, a prime number p, and let $n := p^a + 1$. Take $\mathbb{F}_q$ as the splitting field of $X^n - 1$ over $\mathbb{F}_p$, set $f := X^n + t^n - 1$ and $v := t$. We remark that f is irreducible. In this example, the intersection of the kernels of the A_i, which is what we would calculate without the trick, equals $W \otimes \mathbb{F}_q$ if and only if $\ell > n(n-1)$. With the trick we obtain W if and only if $\ell > n$.

Remark. In the algorithm we assumed for simplicity that we can choose $v = t$. Even if this is possible it can be a good idea to choose v of higher degree. If one chooses a prime ideal v of higher degree than necessary, then a smaller ℓ will suffice to reach the same $\deg_t(v^\ell)$. Using the isomorphism between $\mathcal{O}_v$ and $\mathbb{F}_q(\alpha)[[\tilde{t}]]$ and fast $\mathbb{F}_q(\alpha)$ arithmetic, this can save time in Hensel lifting when polynomial multiplication of degree-kd polynomials over $\mathbb{F}_p$ (in the classical range) is slower than multiplication of degree-k polynomials over $\mathbb{F}_{p^d}$ (because the Zech logarithm representation for small

extension fields is very efficient). For small p and reducible f this often reduces the running time in our implementation in the Magma [BCP97] system. The advantage disappears for very high degrees where one switches to asymptotically fast polynomial arithmetic and a drawback is that factoring $\overline{f}$ becomes more costly.

4.6. Improving the bounds. In this section we improve the quadratic bound B in Theorem 4.1 for the case of small characteristic p and provide examples where the improved bound is sharp, nicely supplementing the result of [BLSSW04]. The idea is that we can make sure that $\sigma > n/p$ in §3.1.3. For simplicity of notation we assume that $\tilde{n} = n$ and f is monic.

Theorem 4.4. *Let $f \in \mathbb{F}_q[t][X]$ be a monic polynomial of total degree n and $q = p^r$. Using Algorithm 4.3, we get $L' = W$ when $\ell > \min(q, n)(n-1)$. When we use the improvement described in §4.5 this bound is improved to $\ell > \min(p, n)(n-1)$.*

Proof. We imitate the proof of §3.1.3. Assume for a moment that f is irreducible and there exists an element in $g \in G' \setminus G$. As in the proof of Lemma 3.2 we get the corresponding exponent vector $(e_1, \ldots, e_r)$, where $e_i \in \{0, \ldots, p-1\}$. Choose a value $a \in \{0, \ldots, p-1\}$ such that

$$\sigma_a := \sum_{\{i:\ e_i = a\}} \deg(f_i)$$

becomes maximal. Then replace g by g/f^a and define $\sigma := \sigma_a \geq n/p$. Therefore, $v^{\sigma\ell}$ divides the resultant. Altogether we want $\sigma\ell > n(n-1)$. Therefore it suffices to choose $\ell > p(n-1)$.

If f is not irreducible we have to check the second condition of Lemma 3.2 that $\bar{g}_j \nmid \bar{\Phi}(g)$ for $1 \leq j \leq s$, where g_j are the true factors of f. Suppose there are g_j with this bad property after changing the element as in the first part of the proof. Then this means that we (theoretically) can replace f by $\tilde{f} := f/g_j$. Furthermore we can replace the corresponding lattices G and G' accordingly. If we repeat this we arrive at a situation where we do not have the problem with the second condition. Now we can proceed as in the first part of the proof, where we replace f by $\tilde{f}$. We obtain the condition $\sigma \geq \deg(\tilde{f})/p$, but now it suffices to choose $\sigma\ell > \deg(\tilde{f})(\deg(\tilde{f} - 1))$. Altogether we obtain

$$\ell > p(\deg(\tilde{f}) - 1)$$

and the result follows since $\deg(f) \geq \deg(\tilde{f})$. □

Corollary 4.1. *Let f be separable in $\mathbb{F}_q[t][X]$ and let p be the characteristic of $\mathbb{F}_q$. For simplicity, assume that f has total degree n and that $\mathrm{Res}_t(f', f)(0) \neq 0$. Disregarding the time needed for one univariate factorization in $\mathbb{F}_q[X]$ in degree n, the irreducible factors of f can be computed in*

deterministic time $\widetilde{O}(q \min(n,p) n^{\omega+1})$. *One may omit the* $\min(n,p)$ *factor if* $p > n(n-1)$ *or* $p = O(1)$.

Proof. Apply [BLSSW04, Proposition 1] with accuracy $\sigma = \min(n,p)(n-1)$ for $p \leq n(n-1)$. That proposition counts basic operations in $\mathbb{F}_q$, each of which is done in time $\widetilde{O}(q)$. The result for large p is the main result of [BLSSW04]. □

The condition on the resultant of f and f' is not restrictive: if q is sufficiently large then a linear transformation achieves $\mathrm{Res}_t(f', f)(0) \neq 0$. A constant field extension of degree $O(\log n)$ makes q sufficiently large, and this logarithmic increase in complexity disappears in the $\widetilde{O}$ estimates. However, in practice it is both inefficient and unnecessary to make a field extension; instead we can choose some v of degree $O(\log n)$, see the remark right after Algorithm 4.3.

Quite surprisingly, the following example shows that our accuracy bound is sharp: for small and large p a linear accuracy is sufficient, but if $n \leq p \leq n(n-1)$, a quadratic accuracy $n(n-1)$ may be needed:

Example. Let $h(X) := X^p - X + t^{p-1}$ which is irreducible since one of the slopes of the Newton polygon is $(p-1)/p$, whose denominator equals the degree. Let us define $f(X) := h(X+t) = X^p - X + t^p - t + t^{p-1}$. The latter polynomial f has the property that Lemma 4.2 does not provide better bounds than Lemma 4.1. Taking $v = t$ we find that $\bar{f}$ factors into n linear factors. If we choose $\ell = n+1$ we arrive at a lattice L' generated by $(1, \ldots, 1)$ and $(1, 2, \ldots, p-1, 0)$. This last vector disappears exactly when $\ell > \min\{n, p\}\,(n-1)$.

The proof of the following result is in the same spirit as Theorem 4.4 and related to Proposition 4 in [BLSSW04]:

Theorem 4.5. *Let* $f \in \mathbb{F}_q[t][X]$ *be a polynomial. Compute* L' *using precision* $\ell > 2(n-1)$. *Then all* $0-1$ *vectors* e *in* L' *belong to* W.

Proof. Assume $e \in L' \setminus W$ is a $0-1$ vector. By subtracting the $0-1$ vectors of true factors g_j we can make sure that

$$\sum_{\{i:\, e_i = 0\}} \deg(f_i) \geq n/2.$$

In the case that the second condition of Lemma 3.2 is not true for g_j we replace f by $\tilde{f} := f/g_j$ as in the proof of Theorem 4.4. Altogether we obtain the desired bound. □

This shows that there is no wrong $0-1$ vector in our lattice L', provided $\ell > 2n$. Unfortunately it may be the case that there are other vectors not belonging to W. In fact, the same example 4.6 exhibits "wrong" elements

in the lattice L' which are not $0-1$ vectors. Of course, in this bad example it would be a big practical improvement to stop at $\ell = n+1$ and perform an "exponential" search for $0-1$ vectors. This is much cheaper than the additional Hensel lifting needed to reach the theoretical bound.

5. The case $K = \mathbb{Q}$

5.1. Setup. For $f \in \mathbb{C}[X]$ with leading coefficient $\mathrm{lc}(f)$, let

$$M(f) := |\mathrm{lc}(f)| \prod_{|\alpha|>1} |\alpha|^{m_\alpha}$$

be the Mahler measure of f, where the product is taken over all roots $\alpha \in \mathbb{C}$ of f with absolute value > 1, and m_α is the multiplicity of the root α.

Lemma 5.1. *If $f, g \in \mathbb{C}[X]$, where g is a non-constant divisor of f, then*

$$\Phi(g) = \sum_{i=0}^{n-1} a_i X^i \in \mathbb{C}[X].$$

Further:

(1) $M(\Phi(g)) \le nM(f)$,
(2) $|a_i| \le B_i := \binom{n-1}{i} nM(f)$ *for all* $i < n$.

Proof. It is clear that $\Phi(g) \in \mathbb{C}[X]$, with degree $n-1$. The Mahler measure of $\Phi(g)$ is bounded by $\deg(g)M(f) \le nM(f)$ since $M(A') \le \deg(A)M(A)$ (see [Mah61]), and $M(AB) = M(A)M(B)$ for any $A, B \in \mathbb{C}[X]$ [MPS99, p. 79], proving (1). The upper bound (2) now follows by [MPS99, Lemma 2.1.9]. □

We now restrict to the case $K = \mathbb{Q}$; for the additional details needed in the number field case, see [Bel03]. Then $\mathcal{O} = \mathbb{Z}$ and v is a prime number. We use $||.||_2$ for the L^2 norm on $\mathbb{Z}^n$ and on $\mathbb{Z}[X]_{<n}$, which denotes the set of polynomials in $\mathbb{Z}[X]$ of degree $< n$. We assume $f \in \mathbb{Z}[X]$ is separable. By equation (3.9) in [LLL82] there exists a small prime $v = O(n \log n + n \log ||f||_2)$ such that f remains separable mod v.

Corollary 5.1. *With $f \in \mathbb{Z}[X]$ and g any factor of f in $\mathbb{Q}[X]$, we have $\Phi(g) \in \mathbb{Z}[X]_{<n}$ and*

$$||\Phi(g)||_2 \le B(f) := 2^{n-1} n ||f||_2.$$

Proof. That $\Phi(g)$ is in $\mathbb{Z}[X]$ was proved in Section 3.1.1. Lemmata 2.1.8 and 2.1.9 in [MPS99] prove that

$$||\Phi(g)||_2 \le 2^{n-1} M(\Phi(g)) \le n 2^{n-1} M(f) \le n 2^{n-1} ||f||_2,$$

using Lemma 5.1 (1), then Landau's inequality (see e.g. [MPS99, Corollary 2.1.5]). □

Fix some integer $\ell \geq 1$. Recall that "$x \bmod v^\ell$" denotes the canonical minimal lift to $\mathbb{Z}$ of $x \in \mathbb{Z}_v$, extended to $\mathbb{Z}_v[X]$ coefficientwise. Let the $a_{i,j} \in \mathbb{Z}$ be the coefficients of $\Phi(f_j) \bmod v^\ell$, so that

$$\Phi(f_j) \bmod v^\ell = \sum_{i=0}^{n-1} a_{i,j} x^i \quad (1 \leq j \leq r).$$

Then let the *all-coefficients* lattice L be the column space of the following matrix:

$$A := \begin{pmatrix} \mathrm{Id}_r & 0 \\ \mathcal{A} & v^\ell \mathrm{Id}_n \end{pmatrix}, \quad \text{where } \mathcal{A} := \begin{pmatrix} a_{0,1} & \cdots & a_{0,r} \\ \vdots & \ddots & \vdots \\ a_{n-1,1} & \cdots & a_{n-1,r} \end{pmatrix}.$$

Note that L depends on ℓ.

Corollary 5.2. *Each K-factor $g_j = \prod f_k^{w_{j,k}}$, associated to $w_j = (w_{j,k}) \in \{0,1\}^r$, corresponds to a vector*

$$\tilde{w}_j = A \begin{pmatrix} w_j \\ * \end{pmatrix} = \begin{pmatrix} w_j \\ \Phi(g_j) \end{pmatrix} \in L.$$

Then

$$||\tilde{w}_j||_2 \leq \sqrt{||w_j||_2^2 + B^2} \leq B' := \sqrt{r + B^2},$$

where $B = B(f)$ is as in Corollary 5.1.

5.2. A truncated all-coefficient lattice. Before we state the algorithm we introduce an improvement which corresponds to [Hoe02, Definition 2]. As in the bivariate case it would be nice to ignore the lower digits in the p-adic expansion of the coefficients. For $i < n$, let ℓ_i be the smallest value such that $v^{\ell_i} > 2B_i$, where B_i is the bound from Lemma 5.1. Then we define the following rounding procedure:

$$\bar{a}_{i,j} := a_{i,j} \bmod v^{\ell_i} \in \left]-\frac{v^{\ell_i}}{2}, \frac{v^{\ell_i}}{2}\right] \quad \text{and} \quad \Psi_{\ell_i}^\ell(a_{i,j}) := (a_{i,j} - \bar{a}_{i,j})/v^{\ell_i}.$$

In other words, $\Psi_{\ell_i}^\ell(a_{i,j})$ is the quotient of the Euclidean division of $a_{i,j}$ by v^{ℓ_i} with centered remainders. Note that $\Psi_{\ell_i}^\ell(a_{i,j}) = 0$ if $\ell \leq \ell_i$. Applying the truncation operators $\Psi_{\ell_i}^\ell$ to the lines of $\mathcal{A}$, we consider instead of A the truncated matrix

(5.1)

$$\tilde{A} := \left(\begin{array}{ccc|ccc} & \mathrm{Id}_r & & & 0 & \\ \hline \Psi_{\ell_0}^\ell(a_{0,1}) & \cdots & \Psi_{\ell_0}^\ell(a_{0,r}) & v^{\ell-\ell_0} & & 0 \\ \vdots & \ddots & \vdots & & \ddots & \\ \Psi_{\ell_{n-1}}^\ell(a_{n-1,1}) & \cdots & \Psi_{\ell_{n-1}}^\ell(a_{n-1,r}) & 0 & & v^{\ell-\ell_{n-1}} \end{array}\right).$$

The columns of $\tilde{A}$ generate another *all-coefficients* lattice which we still denote L, since from now on we shall only work with this truncated version. Again, a K–factor g_j corresponds to a vector in L whose entries comes from w_j and $\Phi(g_j)$, the second component being truncated. Unfortunately the rounding operator $\Psi_{\ell_i}^{\ell}$ is not linear; i.e., in general

$$\Psi_{\ell_i}^{\ell}(a_{i,j_1}) + \Psi_{\ell_i}^{\ell}(a_{i,j_2}) \neq \Psi_{\ell_i}^{\ell}(a_{i,j_1} + a_{i,j_2}).$$

This phenomenon does not occur in the bivariate case, thus allowing us to use linear system of equations to solve our problem. Fortunately the rounding error is small, bounded by $r/2$ as in [Hoe02, Lemma 2.6]:

Lemma 5.2. *If $a_1, \ldots, a_s$ are integers and $a = a_1 + \cdots + a_s$, then*

$$\Psi_k^{\ell}(a) = \varepsilon + \sum_{i=1}^{s} \Psi_k^{\ell}(a_i), \quad \textit{where} \quad \varepsilon \in \mathbb{Z},\ |\varepsilon| \leq \frac{s}{2}.$$

Each vector associated to a rational factor in $\operatorname{Im} \tilde{A}$ is the sum of at most r input vectors, hence has L^2-norm less than $B' := \sqrt{r + n(r/2)^2}$ by the lemma. We obtain the following algorithm.

Algorithm 5.1. *(Rational factorization)*

Input: $f \in \mathbb{Z}[X]$, v a prime. Assume f is primitive (content 1), square-free modulo v, and that the leading coefficient $\operatorname{lc}(f)$ of f is not divisible by v.

Output: Factorization $f = \operatorname{lc}(f) g_1 \cdots g_s \in \mathbb{Q}[X]$.

Step 1: Compute $f \equiv \operatorname{lc}(f) \bar{f}_1 \cdots \bar{f}_r \bmod v$.

Step 2: Compute minimal $\ell_0, \ldots, \ell_{n-1}$ such that $v^{\ell_i} > 2B_i$, where B_i is given by Lemma 5.1.

Step 3: Let $\ell := \max\{\ell_0, \ldots, \ell_{n-1}\} + 1$ and $B' := \sqrt{r + n(r/2)^2}$.

Step 4: Compute $f \equiv \operatorname{lc}(f) f_1 \cdots f_r \bmod v^{\ell}$ (Hensel lift).

Step 5: Compute $\Phi(f_j) \bmod v^{\ell} = \sum_{i=0}^{n-1} a_{i,j} x^i \quad (1 \leq j \leq r)$.

Step 6: Compute a reduced basis $b_1, \ldots, b_m$ of the lattice $L = \operatorname{Im} \tilde{A}$ defined by the columns of (5.1).

Step 7: Set $t \leftarrow \min\left\{ i \colon ||b_j^*||_2 > B', \text{ for all } j > i \right\}$.

Step 8: Let L' be the projection of $\mathbb{Z}b_1 + \ldots + \mathbb{Z}b_t$ on its first r coordinates.

Step 9: Try to construct a canonical $0-1$ basis of L' and check using Lemma 3.4 if $L' = W$.

Step 10: If that is successful, then reconstruct the factors $g_1, \ldots, g_t$ from W ($s{=}t$ here). Otherwise replace ℓ by 2ℓ and go to Step 4.

Since the final result is checked to be correct, we only need to prove that this algorithm eventually terminates. In order to get good (polynomial) running times we must prove that the corresponding ℓ is not too big.

Before we do this, we must explain what a *reduced* basis is in Step 6. Let us recall the main property of LLL–reduced bases, where we choose the Lovász constant $1/4 < \gamma < 1$, see [LLL82, Proposition 1.6 & following remark]:

Proposition 5.1. *Let $b_1, \ldots, b_m$ be a LLL-reduced basis of a lattice L with corresponding Gram–Schmidt basis $b_1^*, \ldots, b_m^*$. Then the following inequalities hold:*

$$||b_j||_2 \leq \left(\frac{4}{4\gamma - 1}\right)^{(i-1)/2} ||b_i^*||_2, \quad \textit{for } 1 \leq j \leq i \leq m.$$

Any basis satisfying analogous bounds can be used in our algorithm, for instance Schönhage's semi-reduced bases ([Sch84, §2]), which satisfy the weaker bound $||b_j||_2 \leq 2^{m-1+i/2}||b_i^*||_2$. To cater for all sensible basis reduction algorithms, we assume henceforth that, whatever reduction algorithm is chosen, a reduced basis $(b_1, \ldots, b_m)$ satisfies

$$||b_j||_2 \leq C^m ||b_i^*||_2, \quad \text{for } 1 \leq j \leq i \leq m, \tag{5.2}$$

for some constant C. Using LLL-reduction with γ sufficiently close to 1, we may choose any $C > \sqrt{4/3}$.

The following lemma justifies Step 7: all short vectors are in the span of the first t basis vectors. This cutting is independent of the reduction property.

Lemma 5.3. *Let $B' \in \mathbb{R}^{>0}$ and L be a lattice with basis $b_1, \ldots, b_m$ and corresponding orthogonal Gram–Schmidt basis $b_1^*, \ldots, b_m^*$. Define $t := \min\left\{i \colon ||b_j^*||_2 > B', \textit{ for all } j > i\right\}$. Then all vectors b with $||b||_2 \leq B'$ are contained in $\mathbb{Z}b_1 + \ldots + \mathbb{Z}b_t$.*

Proof. We imitate the proof of (1.11) in [LLL82]. Let

$$b = \sum_{i=1}^{m} r_i b_i = \sum_{i=1}^{m} r_i' b_i^* \quad \text{with } r_i \in \mathbb{Z},\ r_i' \in \mathbb{R} \text{ for all } 1 \leq i \leq m.$$

Let j be the largest index with $r_j \neq 0$. Then $r_j = r_j'$ and therefore

$$||b||_2 \geq \left|r_j'\right| \cdot ||b_j^*||_2 \geq ||b_j^*||_2,$$

since r_j is a non-zero integer. The result follows. □

Theorem 5.2. *Let $f \in \mathbb{Z}[X]$ of degree n and v as in Algorithm 5.1. Then the algorithm terminates when*

$$v^\ell > c^n \cdot (2C)^{n^2} ||f||_2^{2n-1} (\log ||f||_2)^n, \tag{5.3}$$

where c is a constant which can be explicitly computed.

Proof. We use the notation of Section 3. For $e = (e_1, \dots, e_{r+n})^{\mathrm{tr}} \in L$, let $\mathrm{POL}(e) := \Phi(f_1^{e_1} \cdots f_r^{e_r}) \in \Phi(G_v)$; this polynomial only depends on the first r coordinates of e. Lemma 5.3 shows that every $w \in L$ with $||w||_2 \le B'$ is in $\mathbb{Z}b_1 + \cdots + \mathbb{Z}b_t$, in particular the $\tilde{w}_j$, hence L' contains W in Step 8. Assume $W \subsetneq L'$, then $\mathrm{POL}(b_u) \notin \Phi(G)$ for some $1 \le u \le t$. On the other hand, (5.2) implies that $||b_u||_2 \le C^n B'$.

Using Lemma 3.2 as in §3.1.3, there exists a vector $g \in L'$ such that $f_i \mid \mathrm{POL}(g)$ for some $1 \le i \le r$, and $v^\ell \mid \mathrm{Res}(f, H) \ne 0$, where $H := \mathrm{POL}(g) \bmod v^\ell$. From the proof of Lemma 3.2, this vector g may be obtained in the following way:

- first adding a subset of $\{\tilde{w}_1, \dots, \tilde{w}_s\}$ to b_u, yielding a vector b such that
$$||b||_2 \le (C^n + s)B',$$
- then by adding to b a vector of the form $e\tilde{w}_i$ for some integer e with $|e| \le ||b||_\infty \le ||b||_2$. This gives a new vector $\tilde{b} = b + e\tilde{w}_i$ with
$$||\tilde{b}||_2 \le ||b||_2(1 + B') \le (C^n + s)B'(1 + B'),$$
using $||\tilde{w}_i||_2 \le B'$ and the preceding bound.
- g is the projection of $\tilde{b}$ to L'.

We now need an upper bound for $||H||_2$. Let $\tilde{\ell} := \max\{\ell_0, \dots, \ell_{n-1}\}$, satisfying

$$v^{\tilde{\ell}} \le v \cdot 2^{n-1} n ||f||_2$$

by Corollary 5.1. If $H(X) = \sum_{i<n} h_i X^i \in \mathbb{Z}[X]$, the projection of $\tilde{b}$ on its n last coordinates is $\left(\Psi^\ell_{\ell_0}(h_0), \dots, \Psi^\ell_{\ell_{n-1}}(h_{n-1})\right)$, up to truncation errors, which we bound using Lemma 5.2. Using $v = O(n \log n + n \log ||f||_2)$, we obtain

$$\begin{aligned} ||H||_2 \le v^{\tilde{\ell}}(||\tilde{b}||_2 + ns/2) &\le v^{\tilde{\ell}}\big((C^n + s)B'(B' + 1) + ns/2\big) \\ &\le c \cdot (2C)^n ||f||_2 \log ||f||_2, \end{aligned}$$

where the constant c can be explicitly computed, using $s \le n$, $B' = O(n^{3/2})$. From the preceding discussion and Hadamard's bound on determinants,

$$v^\ell \le |\mathrm{Res}(f, H)| \le ||f||_2^{n-1} ||H||_2^n \le c^n (2C)^{n^2} ||f||_2^{2n-1} (\log ||f||_2)^n,$$

and we obtain a contradiction with (5.3). □

Let $h := \log ||f||_2$. From this theorem, we obtain $\ell \log v = O(n(n+h))$. Since Hensel lifting and LLL-reduction are polynomial time algorithms, we see that W can be computed in polynomial time.

Corollary 5.3. *Let $h := \log ||f||_2$. Algorithm 5.1 factors f using $O(n^9 + n^7 h^2)$ bit operations, assuming classical (quadratic) arithmetic throughout. The time becomes $\widetilde{O}(n^8 + n^6 h^2)$ if fast arithmetic is used.*

Proof. The running time is dominated by the basis reduction: see e.g. [GvzG00, §14 and §15] to estimate the factorization over $\mathbb{Z}/v\mathbb{Z}$ and Hensel lifting respectively. (Since $v = \widetilde{O}(nh)$ is small, we can factor over $\mathbb{Z}/v\mathbb{Z}$ using a deterministic algorithm.) The naive bound uses Nguyen and Stehlé's L^2 algorithm [NS05]. From an input basis $(b_1, \ldots, b_n)$, where the b_i belong to $\mathbb{Z}^n$ and satisfy $\log ||b_i||_2 = O(\log B)$, the algorithm produces an LLL-reduced basis in $O(n^5 \log B(n + \log B))$ bit operations. The bound using fast arithmetic follows from Schönhage's algorithm [Sch84, Theorem 2.1] which produces a semi-reduced basis in time $\widetilde{O}(n^4 \log^2 B)$. In both cases, we can take $\log B = O(\log v^\ell) = O(n(n+h))$. Note that, switching to fast multiplication, the complexity becomes slightly worse in the h term (by logarithmic factors). □

Although there is no practical reason for doing so, since power sums do not offer advantages over coefficients of Φ, one could now use the relation between power sums and Φ to show the algorithm in [Hoe02] is polynomial time, provided that one uses what we call the *all-traces* version of the algorithm. This version uses all of the traces numbered $1, \ldots, n-1$ at the same time, so the basis reduction takes place in $\mathbb{Z}^{r+n-1}$.

5.3. Bit accuracy of the modular computations. Our estimate for the accuracy to which modular factors are approximated is $\ell \log v = O(n(n+h))$. It is the same as in the proof of [LLL82, Proposition 3.4]. The best complexity so far for our problem of factoring univariate polynomials over $\mathbb{Q}$ was obtained by Schönhage [Sch84], namely $\widetilde{O}(n^6 + n^4 h^2)$ bit operations. Schönhage uses complex diophantine approximation, together with his landmark divide and conquer procedure to approximate the complex roots of f within guaranteed time and relative error bounds, and a kind of lattice semi-reduction which turns out to be weaker but asymptotically faster than LLL. Interestingly, he uses complex floating point computation instead of p-adics, but still to the same accuracy of $O(n(n+h))$ bits, see [Sch84, Lemma 6.2]. Note that both Corollary 5.1 and the bounds needed for Zassenhaus reconstruction, e.g. Landau-Mignotte bound, are $O(n+h)$. So, there is an annoying extra factor n in the accuracy estimates, coming from a resultant upper bound in all three cases.

Building on Schönhage's root-finding approach, Miller [Mil92] recast the factorization problem as a search for primitive idempotents in the algebra

$A = \mathbb{Q}[X]/(f)$, found as integer combinations of complex floating point approximations of the Lagrange idempotents in $A \otimes \mathbb{C}$, which are easy to refine from an initial rough approximation. Using an integer relation-finding algorithm instead of more general basis reductions, he works in *linear* precision $O(n \log^2 n + h)$. This would be essentially optimal (of the order of Laudau-Mignotte bound), and would constitute a breakthrough analogous to the linear lifting bound of [BLSSW04]. Unfortunately, apart from a number of harmless typos[1], there is a subtle gap in Miller's proof, which we do not know how to fix. The crucial relation-finding algorithm from [HJLS89] uses an unorthodox floating point model: computations in $\mathbb{R}$ are done to infinite accuracy in unit time. We know the height of the final relations, but this does not imply that we can truncate all real numbers to that many significant digits and obtain the same results, with respect to correctness or running times. In fact even termination may be a problem as reported in [NS05] for the closely related LLL basis reduction algorithm. An apparently difficult perturbation result is needed at this point. It looks plausible that a rigorous result will require increasing the accuracy, presumably losing that factor n saved over all other known polynomial time algorithms for this problem.

This means that in all three proven algorithms (ours, [LLL82] and [Sch84]), the matrix input to the basis reduction algorithm has the same bit-size. In our case, a single lattice basis reduction finds all factors at once, compared to $O(\log n)$ basis reductions per rational factor in [LLL82] and [Sch84]. Thus we do improve on [LLL82] but still lose a factor n^2 over [Sch84] because Schönhage's semi-reduction of a $\mathbb{Z}$-basis $b_1, \ldots, b_n$ actually takes times $\tilde{O}(n^2 \log^2 \tilde{B})$ where $\tilde{B}$ is a bound such that

$$||b_i||_2 \leq \tilde{B}, \quad \text{and} \quad \mathrm{Gram}(b_1, \ldots, b_i) \leq \tilde{B} \quad (i \leq n).$$

Due to this bound on the Gramians, we are not able to do better than the trivial bound

$$\log \tilde{B} \ll n \log B \ll n^2(n + h).$$

Schönhage treats a one or two-dimensional knapsack, where the much better bound

$$\log \tilde{B} \ll \log B \ll n(n + h)$$

is available [Sch84, Lemma 6.1]; we, on the other hand, handle simultaneously n coefficients.

In conclusion, the proven complexity of our basic algorithm is the best one so far in the h (height) aspect, though only by logarithmic factors if

[1] Conflicting forms of the main complexity result are given. Only sketches of proof are provided, but we believe the author's claim is that his algorithm factors the polynomial f using $O(n^3(n \log^2 n + h))$ operations performed on integers of $O(n \log^2 n + h)$ bits. Which indeed would make it a factor n faster than Schönhage's.

fast arithmetic is used. But it is worse in the n (degree) aspect, which is due to two factors: the accuracy $O(n(n+h))$ is too large, and we treat too many coefficients simultaneously.

The reason our algorithm is still very much of interest is that a practical implementation will start with $O(n+h)$ bits of accuracy and solve *many* one-dimensional knapsacks ("one coefficient at a time"), with increasing accuracy. As new coefficients are taken into account, rational factors are found and others are proven irreducible. As long as modular factors are left over, the accuracy is increased and Theorem 5.2 guarantees termination (all factors found and proven irreducible) when $O(n(n+h))$ precision is reached and all coefficients have been incorporated. We discuss this idea further in the next section, although we do not give a sharp complexity result which would capture that practical algorithm's true efficiency.

5.4. One coefficient at a time. From a practical point of view, the all-traces and all-coefficients versions are slow and thus not interesting. The main question is whether practical versions of the algorithm can be proven to run in polynomial time. Using one trace at a time works very well in practice (see [Bel03]). We will show that the "one coefficient at a time" version factors in $\mathbb{Q}[X]$ in polynomial time. The same must then also be true for one trace at a time, confirming a conjecture in [Bel03, Remark 2.5].

Let B_i be the bound for $|a_i|$ given in Lemma 5.1. For $0 \le i < n$ and $g \in K_v(X)^*$ write $T_i'(g) \in \mathbb{Z}$ the coefficient of X^i in $\Phi(g) \bmod v^\ell$. Let $T_i(g) := T_i'(g)/B_i \in \mathbb{Q}$. Note that T_i depends on ℓ, which is a fixed integer ≥ 1 that may be chosen arbitrarily. Lemma 5.1 says that if g is a K-factor of f, then $|T_i(g)| \le 1$.

Proposition 5.2. *Let C be the basis reduction constant, as in* (5.2)*. One can compute a sequence of lattices $L_{n-1}, L_{n-2}, \dots, L_0$ with the following properties:*

(1) $\mathbb{Z}^r = L_{n-1} \supseteq L_{n-2} \cdots \supseteq L_0 \supseteq W$
(2) $L_i = \mathbb{Z}b_{i,1} + \cdots + \mathbb{Z}b_{i,r_i}$ *for some integer r_i and some vectors $b_{i,j} \in \mathbb{Z}^r$ with the following properties:*
  (a) $||b_{i,j}||_2 \le (r+2)C^r$.
  (b) *If* $b_{i,j} = (e_1, \dots, e_r)^{\mathrm{tr}}$ *then* $T_i(f_1^{e_1} \cdots f_r^{e_r}) \le (r+2)C^r$

Proof. If $i = n-1$ we may take $b_{i,1}, \dots, b_{i,r_i}$ as the standard basis of $\mathbb{Z}^r$. If $i < n-1$ then we may assume that $L_{i+1} = \mathbb{Z}b_{i+1,1} + \cdots + \mathbb{Z}b_{i+1,r_{i+1}}$ has been computed and define b_j' as follows: First write $b_{i+1,j} = (e_1, \dots, e_r)^{\mathrm{tr}}$, then compute $a := e_1T_i(f_1) + \cdots + e_rT_i(f_r)$ and set $b_j' := (e_1, \dots, e_r, a)^{\mathrm{tr}} \in \mathbb{Z}^r \times \mathbb{Q}$. Now let $L' := \mathbb{Z}b_1' + \cdots + \mathbb{Z}b_{r_{i+1}}' + \mathbb{Z}P$ where $P = (0, \dots, 0, v^\ell/B_i)^{\mathrm{tr}}$. Let $b_1, b_2, \dots$ be an LLL-reduced basis of L', let $b_1^*, b_2^*, \dots$ the associated orthogonalized basis, and let r_i be the smallest index such that $||b_j^*||_2 > r+2$

for all $j > r_i$. Now define $b_{i,j}$ as the projection of b_j on the first r entries and let $L_i := \mathbb{Z}b_{i,1} + \cdots + \mathbb{Z}b_{i,r_i}$.

Consider the vector w_j corresponding to the K-factor g_j and let w'_j be the corresponding vector in L'. The first r entries of w'_j are in $\{0, 1\}$, and the last entry equals $T_i(g_j) \in \mathbb{Q}$ which has absolute value ≤ 1 by Lemma 5.1. Hence, $||w'_j||_2 \leq \sqrt{r+1} < r+2$. Then it follows from Lemma 5.3 that $w_j \in L_i$ and hence $W \subseteq L_i$. By (5.2), we have $||b_j||_2 \leq (r+2)C^r$ when $j \leq r_i$ which implies (2a), resp. (2b), since projecting on the first r entries, resp. last entry, does not make a vector longer.

The lattice L' to be reduced was in $\mathbb{Z}^r \times \mathbb{Q}$. Lattice reduction in $\mathbb{Z}^{r+1}$ is more efficient, so we round each of the numbers $T_i(f_1), \ldots, T_i(f_r), v^\ell/B_i$ to the nearest integer. Then we obtain a lattice $L' \subseteq \mathbb{Z}^{r+1}$ but now we have introduced rounding errors. Consider again the vectors $w_j \in W$ and $w'_j \in L'$. If w_j has σ entries equal to 1, then the last entry of w'_j is the sum of σ elements of $\{T_i(f_1), \ldots, T_i(f_r)\}$ plus an integer in the interval $(-\sigma/2, \sigma/2)$ times v^ℓ/B_i. We introduced an error ≤ 0.5 in each of the numbers $T_i(f_1), \ldots, T_i(f_r), v^\ell/B_i$. Then the total rounding error in the last entry of w'_j is less than $0.5(\sigma + \sigma/2)$ which is less than r, so this entry will have absolute value $< r+1$. Then $||w'_j||_2 < \sqrt{\sigma + (r+1)^2} < r+2$. The proposition is stated with $r+2$ instead of $\sqrt{r+1}$ so that the bounds can still be used for practical implementations that round $T_i(f_1), \ldots, T_i(f_r), v^\ell/B_i$ to $\mathbb{Z}$. □

Lemma 5.4. *With the notation of Proposition 5.2, the following holds for every $n-1 \geq i \geq i' \geq 0$. If $e = (e_1, \ldots, e_r)^{\mathrm{tr}}$ is an element of $\left\{b_{i',1}, \ldots, b_{i',r_{i'}}\right\}$ then*

$$T_i(f_1^{e_1} \cdots f_r^{e_r}) \leq C^{O(r^2)}.$$

Proof. The entries of the $b_{i,j}$ and e are bounded by $(r+2)C^r = C^{O(r)}$. Since $e \in L_{i'} \subseteq L_i$ we can write $e = \sum_{j=1}^{r_i} c_j b_{i,j}$ for some $c_j \in \mathbb{Z}$ that can be found by solving linear equations. With Cramer's rule one finds $|c_j| \leq C^{O(r^2)}$. Multiplying this by r_i and by the bound in Proposition 5.2 part (2b) leads to the bound $C^{O(r^2)}$. □

Theorem 5.3. *$L_0 = W$ for some ℓ polynomially bounded in terms of n and $\log ||f||_2$.*

Proof. If $L_0 \neq W$ then let e be one of the vectors $b_{0,j}$ from Proposition 5.2 that is not in W. Write $e = (e_1, \ldots, e_r)^{\mathrm{tr}}$ and $g = f_1^{e_1} \cdots f_r^{e_r}$. Write $\Phi(g) = \sum c_i X^i$. Then the corresponding vector in the all-coefficients lattice (see Theorem 5.2) is $\tilde{e} := (e_1, \ldots, e_r, c_0, \ldots, c_{n-1})^{\mathrm{tr}}$ where $c_0, \ldots, c_{n-1}$ are bounded in absolute value by $C^{O(r^2)}$ by Lemma 5.4. Applying the process in the proof of Lemma 3.2 we obtain a new vector e' whose length differs at

most by $(s+\max\{e_1,\ldots,e_r\})B'$ from e. The last n entries of this vector are the coefficients of a polynomial $H \in \mathbb{Z}[X]_{<n}$ and we have $v^\ell \mid \mathrm{Res}(f,H) \neq 0$ in the same way as in Theorem 5.2. This implies that $\log v^\ell$ is polynomially bounded. □

We propose to implement the "one coefficient at a time" approach in the following way: start with a value for ℓ that is *at most* as large as what one would use in the Zassenhaus approach. Then, compute $L_{n-1}, L_{n-2}, \ldots$ until we find W. If we reach L_0 and we still have not found W then we must increase ℓ. The computation of each L_i should be done using the incremental strategy of [Bel03, §2.4], reducing one large-determinant LLL-reduction to a sequence of smaller LLL-reductions that at the end produce the same result. Then one has a polynomial time algorithm that runs very well in practice, with running times that should be the same as those reported in [Bel03] for $\mathbb{Q}[X]$.

This incremental strategy is very efficient because after each LLL-reduction one can check if vectors can be removed. This way, there are no long vectors in the input of the next LLL-reduction. In fact, one can keep the bit-length of these vectors below a bound that depends solely on r. This means that the cost of each individual call to the LLL-reduction algorithm can be bounded by a polynomial that *depends only on* r and is independent of both n and $||f||_2$, which explains why the algorithm performs so well in practice.

Ideally, the practical performance of the incremental strategy would lead to a theoretical complexity that is better than that of Algorithm 5.1. Unfortunately, Lemma 5.4 leads to a pessimistic theoretical bound for the number of calls to the LLL-reduction algorithm. Thus, at the moment the proofs of Theorem 5.3 and Lemma 5.4 do not lead to a good complexity result for the incremental strategy. In fact, working out the details leads to a bound that is worse than the bound for Algorithm 5.1, despite the fact that the incremental strategy is much faster. The second author and his graduate student are currently working to find a better complexity result that accurately reflects the actual performance of the incremental strategy.

References

[Bel03] K. Belabas, *A relative van Hoeij algorithm over number fields*. Journal of Symbolic Computation **37** (2004), 641–668.

[BCP97] W. Bosma, J. Cannon, C. Playoust, *The Magma Algebra System I: The User Language*. Journal of Symbolic Computation **24** (3) (1997), 235–265.

[BLSSW04] A. Bostan, G. Lecerf, B. Salvy, É. Schost and B. Wiebelt, *Complexity issues in bivariate polynomial factorization*. Proceedings of ISSAC (2004).

[GvzG00] J. von zur Gathen and J. Gerhard, *Modern computer algebra*. Cambridge University Press, New York, 1999.

[HJLS89] J. Håstad, B. Just, J. C. Lagarias and C.-P. Schnorr, *Polynomial time algorithms for finding integer relations among real numbers*. SIAM J. Comput. **18** (1989), 859–881.

[Hoe02] M. van Hoeij, *Factoring polynomials and the knapsack problem*. J. Number Theory **95** (2002), 167–189.

[Len82] A. K. Lenstra, *Lattices and factorization of polynomials over algebraic number fields*. Springer Lecture Notes in Computer Science **144** (1982), 32–39.

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász, *Factoring polynomials with rational coefficients*. Math. Ann. **261** (1982), no. 4, 515–534.

[MPS99] M. Mignotte, L. Pasteur and D. Stefanescu, *Polynomials: An Algorithmic Approach*. Springer, 1999.

[Mah61] K. Mahler, *On the zeros of the derivative of a polynomial*. Proc. Roy. Soc. Ser. A **264** (1961), 145–154.

[Mil92] V. Miller, *Factoring Polynomials via Relation-Finding*. ISTCS '92, Springer Lecture Notes in Computer Science **601** (1992), 115–121.

[NS05] P. Nguyen and D. Stehlé, *Floating point LLL revisited*. Eurocrypt'05, Springer Lecture Notes in Computer Science **3494** (2005), 215–233.

[PZ89] M. E. Pohst and H. Zassenhaus, *Algorithmic algebraic number theory*. Cambridge University Press, 1989.

[PO06] M. E. Pohst, *Factoring polynomials over global fields. I*. Journal of Symbolic Computation **39** (2005), 617–630.

[PM06] M. E. Pohst and J. Méndez Omaña, *Factoring polynomials over global fields. II*. Journal of Symbolic Computation **40** (2005), 1325–1339.

[SSH93] T. Sasaki, M. Sasaki, *A unified method for multivariate polynomial factorization*. Japan J. Industrial and Applied Math **10** (1993), no. 1, 21–39.

[Sch84] A. Schönhage, *Factorization of univariate integer polynomials by Diophantine approximation and an improved basis reduction algorithm*. In *Automata, languages and programming (Antwerp, 1984)*, Springer Lecture Notes in Computer Science **172** (1984), 436–447.

[Zas69] H. Zassenhaus, *On Hensel factorization I*. Journal of Number Theory **1** (1969), 291–311.

Karim Belabas
Université Bordeaux 1
351 cours de la Libération
F-33405 Talence, France
E-mail: Karim.Belabas@math.u-bordeaux.fr

Mark van Hoeij
Florida State University
Dept. of Mathematics
Tallahassee, FL 32306, USA
Supported by NSF grants 0098034, 0511544 and 0728853
E-mail: hoeij@math.fsu.edu

Jürgen Klüners
Universität Paderborn
Institut für Mathematik
33095 Paderborn, Germany
E-mail: klueners@math.uni-paderborn.de

Allan Steel
School of Mathematics and Statistics F07
University of Sydney
NSW 2006, Australia
E-mail: allan@maths.usyd.edu.au