Practical Aurifeuillian Factorization

Bill Allombert, Karim Belabas

JOURNAL
de Théorie des Nombres
de BORDEAUX
anciennement Séminaire de Théorie des Nombres de Bordeaux

Bill ALLOMBERT et Karim BELABAS

Practical Aurifeuillian factorization

Tome 20, n° 3 (2008), p. 543-553.

<http://jtnb.cedram.org/item?id=JTNB_2008__20_3_543_0>

Journal de Théorie des Nombres
de Bordeaux **20** (2008), 543-553

Practical Aurifeuillian factorization

par Bill ALLOMBERT et Karim BELABAS

Dedicated to Henri Cohen on his 60th birthday

Résumé. Nous décrivons un algorithme simple pour déterminer les facteurs d'Aurifeuille des entiers $\Phi_d(a)$, où Φ_d est le d-ème polynôme cyclotomique, et a un entier. Sous une hypothèse de Riemann convenable, l'algorithme termine en temps polynomial déterministe $\tilde{O}(d^2L)$, utilisant un espace $O(dL)$, où l'on a noté $L := \log(|a|+1)$.

Abstract. We describe a simple procedure to find Aurifeuillian factors of values of cyclotomic polynomials $\Phi_d(a)$ for integers a and $d > 0$. Assuming a suitable Riemann Hypothesis, the algorithm runs in deterministic time $\tilde{O}(d^2L)$, using $O(dL)$ space, where $L := \log(|a|+1)$.

Contents

Let Φ_d denote the d-th cyclotomic polynomial

$$\Phi_d(X) = \prod_{k \in (\mathbb{Z}/d\mathbb{Z})^*} (X - \zeta_d^k),$$

where ζ_d is a d-th primitive root of unity. To factor integers of the form $a^n - 1$, it is advantageous to start from the algebraic factors

$$a^n - 1 = \prod_{d|n} \Phi_d(a).$$

Manuscrit reçu le 22 mars 2008.

This trick generalizes to

$$a^n + 1 = \frac{a^{2n} - 1}{a^n - 1} = \prod_{d|2n,\, d\nmid n} \Phi_d(a),$$

and in fact to $a^n \pm b^n$ for integers a and b since rational factors of $\Phi_d(a/b)$ lead to integer factors of the requested integer.

Less widely known but still classical, it is often possible to refine further these algebraic factorization. An *Aurifeuillian factorization* exists if $a \in \mathbb{Q}$ is such that $a\zeta_d =: \alpha^2$ is a square in $\mathbb{Q}(\zeta_d)$. In that case, let Nx denote the absolute norm from $\mathbb{Q}(\zeta_d)$ to $\mathbb{Q}$; then

$$\Phi_d(a) = N(a - \zeta_d) = \pm N(\zeta_d a - \zeta_d^2) = \pm N(\alpha - \zeta_d) N(\alpha + \zeta_d), \tag{1}$$

where we have used $N\zeta_d \in \mathbb{Z}^* = \{-1, 1\}$. (In fact, $N\zeta_d = 1$ for $d \neq 2$.) We thus get two rational factors, the so-called *Aurifeuillian factors* of $\Phi_d(a)$. For all complex embeddings $\sigma : \mathbb{Q}(\zeta_d) \to \mathbb{C}$, we have $|\sigma(\alpha \pm \zeta_d)| \geqslant \sqrt{a} - 1$ by the triangle inequality. If $a \in \mathbb{Z}$ satisfies $|a| > 4$, then $\sqrt{a} - 1 > 1$ and we obtain a non-trivial factorization of $\Phi_d(a)$: both Aurifeuillian factors are integers larger than 1. In fact, essentially the same argument proves that both factors have roughly the same size.

Usually, Aurifeuille's trick is presented as polynomial identities of the form

$$\Phi_d(X) = U_{c,d}^2(X) - cXV_{c,d}^2(X),$$

for various constants c and polynomials U, V depending on c, d (Schinzel [9]). Stevenhagen [10] and Brent [2] give algorithms to compute U and V, using a Euclidean algorithm and Newton sums identities respectively. Both algorithms use $O(d^2)$ integer operations, and $\tilde{O}(d)$ using asymptotically fast arithmetic. We do not know a reference for their bit complexity but, as remarked by Brent, a straightforward implementation of Stevenhagen's Euclidean algorithm suffers from intermediate expression swell.

In this short note, we propose an algorithm to find Aurifeuillian factors, which is easier to describe and implement than the polynomial approaches sketched above. It is also more explicit in the sense that extracting from the literature a polynomial formula yielding a factor for a given factorization problem is not obvious, whereas we obtain directly an Aurifeuillian factor of $\Phi_d(a)$, whenever one exists.

1. When does an Aurifeuillian factorization exist ?

Proposition 1.1 (Granville-Pleasants [5])**.** *Let $a \in \mathbb{Q}^*$ and let ζ_d be a primitive d-th root of unity. Let a^* be the squarefree integer, which is the*

canonical representative of a in $\mathbb{Q}^*/(\mathbb{Q}^*)^2$. Then $a\zeta_d$ is a square in $\mathbb{Q}(\zeta_d)$ if and only if $a^* \mid d$ and one of the following is true:

- $a^* \equiv 1 \pmod 4$ *and d is odd.*
- $a^* \equiv 3 \pmod 4$ *and $v_2(d) = 1$.*
- a^* *is even and $v_2(d) = 2$.*

Note that the second case $v_2(d) = 1$ reduces to the first, because $\Phi_d(X) = \Phi_{d/2}(-X)$ in that case. Less obvious, but even more interesting: if

$$D = 2^{v_2(d)} \prod_{p \mid d, p \neq 2} p \quad \text{and} \quad A = a^{d/D},$$

we have $\Phi_d(a) = \Phi_D(A)$ and (d, a) satisfies the above conditions if and only if (D, A) does; in fact $A^* = a^*$ and $v_2(D) = v_2(d)$. On the other hand, if $D' = \prod_{p \mid d} p$ and $A' = a^{d/D'}$, we still have $\Phi_d(a) = \Phi_{D'}(A')$, but the pair (D', A') never satisfies the above conditions when $4 \mid d$: indeed D' is even but $(A')^* = 1$.

The proof of the Proposition is a straightforward case by case analysis, and provides an explicit square root in each case in terms of Gaussian sums. Namely, for p an odd prime and $\left(\frac{\cdot}{q}\right)$ the Legendre-Jacobi symbol modulo the integer $q > 1$, we have

$$g(p)^2 = \left(\frac{-1}{p}\right) p, \quad \text{where} \quad g(p) := \sum_{x \in \mathbb{F}_p} \left(\frac{x}{p}\right) \zeta_p^x \quad \text{and} \quad \zeta_p := \zeta_d^{d/p},$$

$$g(2)^2 = -2i, \quad \text{where} \quad g(2) := i - 1 \quad \text{and} \quad i := \zeta_d^{d/4},$$

assuming $p \mid d$ and $4 \mid d$ respectively. Let

$$G(a^*) = \prod_{p \mid a^*} g(p).$$

If $|a^*| = \prod_p p$ is odd, this yields for instance

$$G^2 = \left(\frac{-1}{|a^*|}\right) |a^*| = (-1)^{\frac{|a^*|-1}{2}} |a^*| = (-1)^{\frac{a^*-1}{2}} a^*$$

$$= \begin{cases} a^* & \text{if } a^* \equiv 1 \pmod 4, \\ -a^* & \text{if } a^* \equiv 3 \pmod 4. \end{cases}$$

Note that Proposition 1.1 implies that, if $a\zeta_d$ is a square, then $a^* \mid d$, hence all the primes $p \mid a^*$ also divide d; if further a^* is even, then $4 \mid d$.

Remark 1.2. The interesting special case $a = p$ prime was our original motivation for this work. Namely, to compute the order of elements in $\mathbb{F}_{p^n}^*$,

in particular to test prospective primitive roots, we need to complete the factorization of

$$p^n - 1 = \prod_{d|n} \Phi_d(p).$$

If p divides n, Aurifeuille's trick provides extra useful factors of the $\Phi_d(p)$ such that p divides d.

2. A product formula for an Aurifeuillian factor

There are beautiful and unexpected formulas for Aurifeuillian polynomials, see [2, Theorem 1]. Our formula for Aurifeuillian factors is neither beautiful nor unexpected, but algorithmically useful nevertheless. The Galois action on the Gaussian sum G is explicit and we write down a variation on (1) optimized for computational purposes:

Proposition 2.1. *Let $d > 2$, and (d, a) satisfy the conditions of Proposition 1.1. Write $a = a^* f^2$, $f \in \mathbb{Q}^*$ and let $G(a) = f \prod_{p|a^*} g(p) \in \mathbb{Q}(\zeta_d)$. Then*

$$\prod_{j \in (\mathbb{Z}/d\mathbb{Z})^*} (\chi(j)G - \zeta_d^j)$$

is an Aurifeuillian divisor of $\Phi_d(a)$, where

$$\chi(j) = \begin{cases} \left(\frac{j}{|a^*|}\right) & \text{if } a^* \text{ odd,} \\ \left(\frac{j}{|a^*/2|}\right) & \text{if } a^* \text{ even and } j \equiv 1 \pmod 4, \\ \left(\frac{j}{|a^*/2|}\right) i & \text{if } a^* \text{ even and } j \equiv 3 \pmod 4. \end{cases}$$

Proof. The $\sigma_j : \zeta_d \mapsto \zeta_d^j$, $j \in (\mathbb{Z}/d\mathbb{Z})^*$, run over the Galois group of $\mathrm{Gal}(\mathbb{Q}(\zeta_d)/\mathbb{Q})$, and $Nx = \prod_j \sigma_j x$ for all $x \in \mathbb{Q}(\zeta_d)$.

1) We first treat the case d odd, $a^* \equiv 1 \pmod 4$: then $a = G(a)^2$ and ζ_d^2 is a primitive d-th root of 1. In particular

$$\Phi_d(a) = N(G^2 - \zeta_d^2) = N(G - \zeta_d)N(G + \zeta_d).$$

Since $\sigma_j g(p) = \left(\frac{j}{p}\right) g(p)$ for all *odd* primes p, we obtain

$$\sigma_j G = \left(\frac{j}{|a^*|}\right) G, \quad \text{hence} \quad \sigma_j(G - \zeta_d) = \chi(j)G - \zeta_d^j.$$

2) For completeness, we include the case $v_2(d) = 1$ and $a^* \equiv 3 \pmod 4$: then $a = -G(a)^2$ and $-\zeta_d^2$ is a primitive d-th root of 1. In particular

$$\Phi_d(a) = N(-G^2 + \zeta_d^2) = N(G - \zeta_d)N(G + \zeta_d).$$

The computation of $\sigma_j(G - \zeta_d)$ is still valid.

3) Assume finally that $v_2(d) = 2$ and a^* even: then $\pm i\zeta_d^2 = \zeta_d^{2\pm d/4}$ are primitive d-th roots of 1, since $\gcd(2 \pm d/4, d) = 1$: indeed $2 \pm d/4$ is odd and any odd prime divisor of d and $2 \pm d/4$ would divide 2. Reusing the previous computations,

$$G(a)^2 = f^2(-2i)(-1)^{\frac{(a^*/2)-1}{2}}(a^*/2) = (-1)^{\frac{(a^*/2)+1}{2}}ai,$$

hence $a = \pm iG(a)^2$. It follows

$$\Phi_d(a) = N(a - (\pm i)\zeta_d^2) = N((\pm i)G^2 - (\pm i)\zeta_d^2) = N(G - \zeta_d)N(G + \zeta_d),$$

where we use $N(\pm i) = 1$ in the last equality. As for the Galois action, we have

$$\frac{\sigma_j\, g(2)}{g(2)} = \varepsilon(j) := \begin{cases} 1 & \text{if } j \equiv 1 \pmod 4 \\ i & \text{if } j \equiv 3 \pmod 4), \end{cases}$$

and it follows that

$$\sigma_j G = \varepsilon(j)\left(\frac{j}{|a^*/2|}\right)G.$$

□

Many analogous products can be written, involving terms of the form $\pm\zeta^i G \pm \zeta^j$; our product is written so as to

- always multiply G by a trivial factor in a given term: $\chi(j)G$ takes values in $\{\pm G, \pm iG\}$ which is easily precomputed.
- require the powers of ζ in increasing order, so there is no need to precompute and store them: they can be obtained by successive multiplications.

Remark 2.2. In fact, storing a few powers of ζ is still useful: if $j_1 < j_2$ are two consecutive integer representatives for elements in $(\mathbb{Z}/d\mathbb{Z})^*$, ζ^{j_2} is computed as $\zeta^{j_1} \times \zeta^{j_2-j_1}$ and the latter lives in a small set which should be precomputed. As an obvious example, when $2 \mid d$ is even, $j_2 - j_1$ is even and only powers of ζ^2 occur, but we can be more thorough and store all $\zeta^{j_2-j_1}$.

Estimating the maximal gap $j_2 - j_1$ in terms of d is a famous question of Jacobstahl, and Iwaniec [7] proved that $j_2 - j_1 \ll (r\log(r+1))^2$ if d has r distinct prime divisors. In particular, $j_2 - j_1 = \tilde{O}(\log d)^2$ remains small. Consequently, the ζ^j for $j \in (\mathbb{Z}/d\mathbb{Z})^*$ are obtained using $\varphi(d) + \tilde{O}(\log d)^2$ modular multiplications, storing no more than $\tilde{O}(\log d)^2$ values at a time.

Compared to the obvious algorithm using $d-1$ modular multiplications, we save a factor $(d-1)/\varphi(d)$ which can be of the order of $\log\log d$ (when d is a product of small primes). Of course this technique becomes less useful if d has few prime factors, in particular it is useless if d is prime!

Remark 2.3. If d is an odd prime such that 2 is a primitive root mod d, a different optimization applies, even though we no longer save on the number of multiplications: we write $j \in (\mathbb{Z}/d\mathbb{Z})^*$ as 2^k, and compute

$$\prod_{k\in\mathbb{Z}/(d-1)\mathbb{Z}} \left(\left(\frac{2}{|a^*|}\right)^k G - \zeta_d^{2^k}\right),$$

where the $\zeta_d^{2^k}$ are computed by successive squarings, which are slightly faster than general multiplications. Of course, $(\frac{2}{|a^*|}) = \pm 1$ is constant and computed only once. The condition on d implies $d \equiv 3, 5 \pmod 8$ (otherwise 2 is a square), and is not very restrictive otherwise: out of the 332365 primes congruent to $3, 5 \pmod 8$ and less than 10^7, 248491 satisfy it, about 74%.

Remark 2.4. In the case $v_2(d) = 2$, replacing our ad hoc $g(2)$ by the customary $g(2) := \zeta_{2d} + \zeta_{2d}^{-1}$ yields nicer formulas since we then have $\sigma_j\, g(p) = (\frac{j}{p}) g(p)$ for all primes p and $g(2)^2 = 2$. Unfortunately, we would then factor

$$\Phi_d(a)^2 = N_{\mathbb{Q}(\zeta_{2d})/\mathbb{Q}}(G^2 - \zeta_{2d}^2) = N_{\mathbb{Q}(\zeta_{2d})/\mathbb{Q}}(G - \zeta_{2d}) N_{\mathbb{Q}(\zeta_{2d})/\mathbb{Q}}(G + \zeta_{2d}),$$

producing essentially the *squares* of the requested Aurifeuillian factors, which would force us to work at double accuracy in the next section.

3. An ℓ-adic algorithm and its complexity ($a \in \mathbb{Z}$)

There are two main ideas to implement easily and efficiently the previous formula. The first one is to compute the product as an ℓ-adic number for a suitable ℓ, not as a complex number: as usual, this avoids tedious estimates of roundoff errors. The second one is to compute the product of local Gaussian sums G directly, as a single ℓ-adic square root of a known number. We now restrict to $a \in \mathbb{Z}$, and defer the general case $a \in \mathbb{Q}$ to the next section.

Algorithm 3.1 (Aurifeuillian factorization)
Input: Integers $d \in \mathbb{Z}_{>0}$ and $a \in \mathbb{Z}$, $a \neq 0$.
Output: An Aurifeuillian factor of $\Phi_d(a)$, if one exists.

(1) [*Handle trivial cases* $d \leqslant 2$]. If $d > 2$, goto (2).
If $d = 2$, set $a \leftarrow -a$.
Return $A + 1$ if $a =: A^2$ is a square in $\mathbb{Z}$ and fail otherwise.

(2) Use Sub-Algorithm 3.2: fail if (d, a) does not satisfy the Granville-Pleasants criterion. Replace (d, a) by the simpler pair returned by the algorithm; at this point we also know a^* and $\varphi(d)$.

(3) Find ℓ, the smallest prime $\equiv 1 \mod d$, and $\zeta \in \mathbb{F}_\ell^*$ of exact order d.

(4) Let $B = (\sqrt{|a|}+1)^{\varphi(d)}$ and e the smallest integer such that $\ell^e > B$.
(5) ζ lifts to a primitive d-th root of 1 in $\mathbb{Z}_\ell$, still denoted ζ. Using Hensel lifting, compute $z \in (\mathbb{Z}/\ell^e\mathbb{Z})$ such that $z \equiv \zeta \pmod{\ell^e}$.
[*End of ℓ-adic initializations.*]
(6) Define $\gamma \in (\mathbb{Z}/\ell^e\mathbb{Z})$ in the following way: if d is odd, let $\gamma \leftarrow a$; else let $i \leftarrow z^{d/4}$ and $\gamma \leftarrow (-1)^{\frac{(a^*/2)+1}{2}} ai$.
(7) Compute an approximate ℓ-adic square root G of γ: an integer $0 \leqslant G < \ell^e$, such that $G^2 \equiv \gamma \pmod{\ell^e}$ (Hensel lift).
(8) Let χ as in Proposition 2.1 and compute the integer $0 \leqslant F < \ell^e$ such that

$$F \equiv \prod_{j \in (\mathbb{Z}/d\mathbb{Z})^*} \left(\chi(j)G - z^j\right) \pmod{\ell^e}.$$

[*Compute z^j by successive multiplications; if d is even, precompute iG.*]
(9) Return F.

Sub-Algorithm 3.2 Input: Integers $d \in \mathbb{Z}_{>0}$ and $a \in \mathbb{Z}$, $a \neq 0$.
Output: Fail if the Granville-Pleasants criterion is not satisfied. Otherwise returns a pair (D, A) with $\Phi_D(A) = \Phi_d(a)$, admitting an Aurifeuillian factor; $D = \delta$ or 4δ, where δ is odd and squarefree. Byproducts: computes $a^* = A^*$, and $\varphi(D)$.

(1) If $d \equiv 2 \pmod 4$ set $d \leftarrow d/2$, $a \leftarrow -a$. [*Now d is odd or divisible by 4.*]
(2) [*Early abort.*] Fail if $8 \mid d$, or if $d \equiv v_2(a) \pmod 2$, or if (d odd and $a/2^{v_2(a)} \not\equiv 1 \pmod 4$).
(3) Factor $d = \prod p^{d_p}$.
(4) Compute the $a_p := v_p(a)$ for the above $p \mid d$, to obtain a partial factorization $a = \mathrm{sign}(a) \prod p^{a_p} b$, where $b > 0$, $(b, d) = 1$. Fail if b is not a square in $\mathbb{Z}$.
(5) Let $a^* = \mathrm{sign}(a) \prod_{a_p \text{ odd}} p$. Fail if $a^* \equiv 3 \pmod 4$, or $a^* \not\equiv d \pmod 2$.
(6) Compute $D = 2^{d_2} \prod_{p \mid d, p \neq 2} p$, and let $A = a^{d/D}$.
(7) Compute $\varphi(D)$; note that the factorization of D is known.
(8) Return $(D, A, a^*, \varphi(D))$.

Proof. Sub-Algorithm 3.2 is a straightforward implementation of Proposition 1.1. Now on to the main Algorithm.

Since $d > 2$ from step (2) on, the d-th cyclotomic field has no real embeddings and the norm has non-negative values. In particular, the Aurifeuillian factors $N(\alpha \pm \zeta_d)$ are non-negative. Since they are obviously less than $B < \ell^e$, knowing them mod ℓ^e is enough to reconstruct them.

For $d > 2$, a primitive d-th root of 1 exists in $\mathbb{Z}_\ell$ if and only if $\ell \equiv 1 \pmod d$; Hensel lifting a solution of $X^d = 1$ of exact order d in $\mathbb{F}_\ell^*$, we can approximate it to any desired ℓ-adic accuracy (note that $(d, \ell) = 1$).

Since the computed G has the correct square, it is equal to the one defined in Proposition 2.1 up to sign, but changing G into $-G$ corresponds to swapping the Aurifeuilian factors, i.e. computing $N(G + \zeta_d)$ instead of $N(G - \zeta_d)$. □

Remark 3.3. Recall that $a = p$ a small prime is an important special case, useful in basic computations involving $\mathbb{F}_{p^n}^*$. In the case $a = 2$, Step (7) of the main Algorithm simplifies since $a = a^* = 2$ and an ℓ-adic square root G of $-ai$ is $i - 1$. An analogous simplification applies if we only assume $a^* = 2$, since a square root G of $-ai$ is $f(i - 1)$, where $a = a^* f^2$.

Theorem 3.4. *Let $L := \log(|a| + 1)$, and $\mathsf{M}(n)$ an upper bound for the bit complexity of multiplication of two n-bits integers. Assume that for all $d > 1$, there exists a prime $\ell \equiv 1 \pmod d$ satisfying $\ell \leqslant Dd^C$ for some constants $C < 8$ and D. Given such an ℓ, Algorithm 3.1 runs in deterministic time $O(d\mathsf{M}(dL) + d^{C/4+\varepsilon}) = \tilde{O}(d^2 L)$, using $O(dL)$ space.*

Proof. The Sub-Algorithm handles numbers $\leqslant d$ for a negligible time $O(d^\varepsilon)$ (including the factorization of d), then computes $O(\log d)$ valuations of a at small primes $p \leqslant d$ in time $O((\log d)^2 L)$, then computes an approximate square root of $b \leqslant |a|$ in time $O(\mathsf{M}(L))$.

Finding an element of order o in $\mathbb{F}_\ell^*$ is done quickly using randomization. To do it in deterministic time, we may look for a primitive root and raise it to the $(\ell - 1)/o$-th power. Unconditionally, the least primitive root mod ℓ is $\ll \ell^{1/4+\varepsilon}$ by Burgess's famous result [3].

Hensel lifting a root of $X^d = 1$ to accuracy ℓ^e is done in time $\tilde{O}(d\mathsf{M}(\log \ell^e))$, and the square root computation yielding G in time $O(\mathsf{M}(\log \ell^e))$. Finally we have $O(\varphi(d)) = O(d)$ multiplications in $\mathbb{Z}/\ell^e\mathbb{Z}$, in time $O(d\mathsf{M}(\log \ell^e))$, and $O(d)$ Jacobi symbols mod a^*, each in time $\tilde{O}(d)$ (note that $a^* \leqslant d$ at this point).

From $\ell^e > B \geqslant \ell^{e-1}$, we obtain $\ell^e \leqslant \ell B$, hence

$$\log \ell^e \leqslant \log \ell + \log B \ll \log d + \varphi(d)L \ll dL,$$

using $\ell \leqslant Dd^C$, which implies $\log \ell \ll_{C,D} \log d$.

The space complexity follows from noting that the computation stores $O(1)$ integers less than ℓ^e, provided we compute the z^j successively. Note that using Remark 2.2 increases our space requirements by a factor $r^{2+\varepsilon}$ if d has r prime divisors. □

The existence of $\ell \leqslant Dd^C$ as in the Theorem is ensured by Linnik's theorem, and the best unconditional bound so far is $C = 5.5$ (Heath-Brown [6]), which is indeed less than 8. Obviously, such an ℓ can be found in deterministic polynomial time $\tilde{O}(d^{C-1})$ by applying primality tests to successive members of the arithmetic progression $1+d, 1+2d, \ldots$. Unfortunately, this becomes dominant, and in order to obtain a realistic estimate, we must make it conditional:

Corollary 3.5. *Assuming the Generalized Riemann Hypothesis, Algorithm 3.1 runs in time $\tilde{O}(d^2 L)$.*

Proof. Assuming the Riemann Hypothesis, $\ell \leqslant 2(d \log d)^2$, see [1, Theorem 5.3], and may be found using $\tilde{O}(d)$ compositeness tests. □

Remark 3.6. For a practical *randomized* way to find $\zeta \in \mathbb{F}_\ell^*$ of order d, factor $\ell - 1$ (notice that the factorization of $d \mid \ell - 1$ is known and the cofactor $(\ell-1)/d$ is expected to be small). Then pick $z \in [1, \ell-1]$ uniformly at random until the following test succeeds: compute the order o of z, using [4, Algorithm 1.4.3]; if $d \mid o$, set $\zeta = z^{o/d}$ and stop. The probability to find an element whose order is a multiple of d in a cyclic group of order $n = \ell - 1$ is

$$\frac{1}{n} \sum_{k \mid \frac{n}{d}} \varphi(kd) \geqslant \frac{1}{n} \sum_{k \mid \frac{n}{d}} \varphi(k)\varphi(d) = \frac{\varphi(d)}{d},$$

with equality when $\gcd(n/d, d) = 1$. This lower bound does not depend on ℓ.

Remark 3.7. The ℓ-adic initialization is almost independent from a. To obtain Aurifeuillian factors of $\Phi_d(a_i)$ for fixed d and varying a_i, we can set $B = (\sqrt{\max_i |a_i|} + 1)^{\varphi(d)}$; the corresponding z may be reused in all computations.

Remark 3.8. Our straightforward upper bound $B = (\sqrt{|a|}+1)^{\varphi(d)}$ is rather sharp since both factors are also $\geqslant (\sqrt{|a|} - 1)^{\varphi(d)}$. This also means that, even if only one factor is desired, the output size is of order Ld; thus our runtime $\tilde{O}(Ld^2)$ is essentially optimal in the L aspect, but d times slower than an optimal, as yet unknown, quasi-linear algorithm.

4. Rational inputs

To factor $\Phi(a)$ where $a \in \mathbb{Q}$, essentially the same algorithm applies with the following modifications:

(1) We now compute explicitly $f \in \mathbb{Q}^*$ such that $a = a^* f^2$, say $f = u/v$ for coprime integers u, v.
(2) Our prime $\ell \equiv 1 \pmod{d}$ must now also satisfy $\ell \nmid v$.

(3) The product

$$F = \prod_{j \in (\mathbb{Z}/d\mathbb{Z})^*} (\chi(j)G - \zeta_d^j)$$

is now a rational number, whose denominator divides $v^{\varphi(d)}$. So we use the bound $B = \left(v(\sqrt{a}+1)\right)^{\varphi(d)}$ and compute $Fv^{\varphi(d)} \bmod \ell^e$ as before. This is an integer which we can now recognize, and divide by $v^{\varphi(d)}$ to obtain a rational factor of $\Phi_d(a)$.

5. A gratuitous example

Recall that Henri Cohen's favourite small integer is 49; to celebrate his 60th birthday, we let our PARI/GP [8] implementation compute the Aurifeuillian factors of $\Phi_{6049}(6049)$:

```
? install(factor_Aurifeuille, GL);
? d = a = 6049;
? F = factor_Aurifeuille(a,d); \\ one factor.
                                  Output suppressed!
time = 13,760ms
? polcyclo(d, a) / F;          \\ the cofactor.
time = 0ms.
```

The computation was run on an Opteron 880 at 2.4Ghz using PARI/GP version `2.4.3` (with GMP-4.1.4 multiprecision kernel), producing two factors having 10899 and 10900 decimal digits in about 14 seconds.

Increasing d by a factor about 10, our implementation computes the Aurifeuillian factors of $\Phi_{60049}(60049)$ (126726 and 126727 decimal digits) in about 99 minutes on the same machine.

References

[1] E. Bach & J. Sorenson, *Explicit bounds for primes in residue classes*. Math. Comp. **65** (1996), no. 216, pp. 1717–1735.

[2] R. P. Brent, *Computing Aurifeuillian factors, in Computational algebra and number theory (Sydney, 1992)*. Math. Appl., vol. 325, Kluwer Acad. Publ., 1995, pp. 201–212.

[3] D. A. Burgess, *On character sums and primitive roots*. Proc. London Math. Soc. (3) **12** (1962), pp. 179–192.

[4] H. Cohen, *A course in computational algebraic number theory*. Graduate Texts in Mathematics, vol. 138, Springer-Verlag, Berlin, 1993.

[5] A. Granville & P. Pleasants, *Aurifeuillian factorization*. Math. Comp. **75** (2006), no. 253, pp. 497–508.

[6] D. R. Heath-Brown, *Zero-free regions for Dirichlet L-functions, and the least prime in an arithmetic progression*. Proc. London Math. Soc. (3) **64** (1992), no. 2, pp. 265–338.

[7] H. Iwaniec, *On the problem of Jacobsthal*. Demonstratio Math. **11** (1978), no. 1, pp. 225–231.

[8] PARI/GP, version `2.4.3`, Bordeaux, 2008, `http://pari.math.u-bordeaux.fr/`.

[9] A. Schinzel, *On primitive prime factors of* $a^n - b^n$. Proc. Cambridge Philos. Soc. **58** (1962), pp. 555–562.
[10] P. Stevenhagen, *On Aurifeuillian factorizations*. Nederl. Akad. Wetensch. Indag. Math. **49** (1987), no. 4, pp. 451–468.

Bill Allombert
Université Montpellier 2
CNRS I3M/LIRMM
Place Eugène Bataillon
F-34095 Montpellier cedex, France
E-mail: Bill.Allombert@univ-montp2.fr

Karim Belabas
Université Bordeaux I
Institut de mathématiques de Bordeaux (A2X)
351 cours de la Libération
F-33405 Talence cedex, France
E-mail: Karim.Belabas@math.u-bordeaux.fr