Redundant τ-adic Expansions I: Non-Adjacent Digit Sets and their Applications to Scalar Multiplication

Roberto Maria Avanzi, Clemens Heuberger and Helmut Prodinger

Abstract. This paper investigates some properties of τ-adic expansions of scalars. Such expansions are widely used in the design of scalar multiplication algorithms on Koblitz Curves, but at the same time they are much less understood than their binary counterparts.

Solinas introduced the width-w τ-adic non-adjacent form for use with Koblitz curves. This is an expansion of integers $z = \sum_{i=0}^{\ell} z_i \tau^i$, where τ is a quadratic integer depending on the curve, such that $z_i \neq 0$ implies $z_{w+i-1} = \ldots = z_{i+1} = 0$, like the sliding window binary recodings of integers. It uses a redundant digit set, i.e., an expansion of an integer using this digit set need not be uniquely determined if the syntactical constraints are not enforced.

We show that the digit sets described by Solinas, formed by elements of minimal norm in their residue classes, are uniquely determined.

Apart from this digit set of minimal norm representatives, other digit sets can be chosen such that all integers can be represented by a width-w non-adjacent form using those digits. We describe an algorithm recognizing admissible digit sets. Results by Solinas and by Blake, Murty, and Xu are generalized.

In particular, we introduce two new useful families of digit sets.

The first set is syntactically defined. As a consequence of its adoption we can also present improved and streamlined algorithms to perform the precomputations in τ-adic scalar multiplication methods. The latter use an improvement of the computation of sums and differences of points on elliptic curves with mixed affine and López-Dahab coordinates.

The second set is suitable for low-memory applications, generalizing an approach started by Avanzi, Ciet, and Sica. It permits to devise a scalar multiplication algorithm that dispenses with the initial precomputation stage and its associated memory space. A suitable choice of the parameters of the method leads to a scalar multiplication algorithm on Koblitz Curves that achieves sublinear complexity in the number of expensive curve operations.

Keywords. Koblitz curves; Frobenius endomorphism; Scalar Multiplication; *tau*-adic expansions; Non-Adjacent-Forms; Digit Sets; Point halving; Efficient Implementation.

1. Introduction

Elliptic curves (EC) [23, 28] are now a well established and standardised [19, 31] cryptographic primitive. The performance of an EC cryptosystem depends on the efficiency of the fundamental

This paper was in part written while R. Avanzi and C. Heuberger were visiting the Department of Mathematical Sciences, Stellenbosch University, and during a visit of R. Avanzi at TU Graz supported by the Austrian Science Foundation FWF, project S9606. R. Avanzi's research described in this paper has been partly supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

C. Heuberger is supported by the Austrian Science Foundation FWF, project S9606, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory."

H. Prodinger is supported by the NRF grant 2053748 of the South African National Research Foundation and by the Center of Experimental Mathematics of the University of Stellenbosch.

operation, the *scalar multiplication*, i.e., the computation of the multiple sP of a point P by an integer s. Among all EC, *Koblitz curves* [24], defined by the equation

$$E_a\colon y^2+xy=x^3+ax^2+1 \qquad \text{with} \qquad a\in\{0,1\} \tag{1}$$

over the finite field $\mathbb{F}_{2^n}$, permit particularly efficient implementation of scalar multiplication. Key to their good performance is the Frobenius endomorphism τ, i.e., the map induced on $E_a(\mathbb{F}_{2^n})$ by the Frobenius automorphism of the field extension $\mathbb{F}_{2^n}/\mathbb{F}_2$, that maps field elements to their squares.

Set $\mu=(-1)^{1-a}$. It is known [37, Section 4.1] that τ permutes the points $P\in E_a(\mathbb{F}_{2^n})$, and $(\tau^2+2)P=\mu\tau(P)$. Hence, if the curve contains a subgroup G of large prime order with smaller cofactor the map acts on the points of G like multiplication by a complex constant s. We identify τ with a root of

$$\tau^2-\mu\tau+2=0\ . \tag{2}$$

If we write an integer z as $\sum_{i=0}^{\ell} z_i\tau^i$, where the digits z_i belong to a suitably defined digit set $\mathcal{D}$, then we can compute zP as $\sum_{i=0}^{\ell} z_i\tau^i(P)$ via a Horner scheme. The resulting method [24, 36, 37] is called a "τ-and-add" method since it replaces the doubling with a Frobenius operation in the classic double-and-add scalar multiplication algorithm. Since a Frobenius operation is much faster than a group doubling, scalar multiplication on Koblitz curves is a very fast operation.

The elements dP for all $d\in\mathcal{D}$ must be computed before the main loop of the Horner scheme begins. Larger digit sets usually correspond to representations $\sum_{i=0}^{\ell} z_i\tau^i$ with fewer non-zero coefficients, which in turn translates to fewer group additions. The recipe for optimal performance is a balance between digit set size and number of non-zero coefficients. This is studied in Section 3.2.2.

Solinas [36, 37] considers the residue classes in $\mathbb{Z}[\tau]$ modulo τ^w which are coprime to τ, and forms a digit set comprising the zero and an element of minimal norm from each residue class coprime to τ. We prove (Theorem 2) that such elements are unique, hence this digit set is uniquely determined. Since the resulting digit set is quite large, an additional condition has to be imposed to guarantee uniqueness of the representation. Solinas' condition is the *width-w non-adjacency property*

$$z_i\neq 0 \qquad \text{implies} \qquad z_{i+w-1}=\ldots=z_{i+1}=0\ . \tag{3}$$

This also forces the number of non-zero digits in a representation to be quite low.

We call a digit set that allows us to write each integer as a recoding satisfying property (3) a *(width-w) non-adjacent digit set*, or w-NADS for short. Theorem 1 is a criterion for establishing whether a given digit set is a w-NADS, which is very different in substance from the criterion of Blake, Murty, and Xu [13]. The characterisation of digit sets which allow recoding with a non-adjacency condition is a line of research started in the binary case by Muir and Stinson in [29, 30], see also Heuberger and Prodinger [18] and Avoine, Monnerat, and Peyrin [12].

We then turn our attention to families of digit sets (such as Solinas' "representatives of minimal norm" digit set). In Section 2.3 we introduce digit sets whose digits are syntactically defined in terms of their width-2 non-adjacent form. The result is almost always a w-NADS, the exceptions are explicitly listed in Theorem 3. Precise estimates of the length of the recoding are given in Theorem 4.

In Section 2.4 we study another family of digit sets corresponding, in a suitable sense, to "repeated point halvings" (cf. Theorem 5). It is used to design a width-w scalar multiplication algorithm without precomputations. These digit sets are not always a w-NADS (Theorem 6), but this difficulty can be overcome by stepping down the window size for the most significant digits (§ 3.2.1).

In Section 3 we discuss the relevance of our results for cryptographic applications and performance. § 3.1 is devoted to the syntactically defined digit set. We present streamlined techniques for the precomputations for a τ-adic scalar multiplication on Koblitz Curves – in fact, these are the first such techniques that work for all values of the parameter w. § 3.2 deals with the point halving digit sets. This results in a precomputationless scalar multiplication algorithm with a sublinear number of expensive curve operations (§ 3.2.2). The performance of these scalar multiplication methods is compared to the state of the art in § 3.3.

The questions of optimality of Solinas' digit set as well as of ours are discussed in [17].

Disclaimer: *The information in this document reflects only the authors' views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.*

This paper is an extended version of [8]*, with proofs and additional results.*

2. Digit Sets

Let $\mu \in \{\pm 1\}$, τ be a root of equation (2) and $\bar{\tau}$ the complex conjugate of τ. Note that $2/\tau = \bar{\tau} = \mu - \tau = -\mu(1+\tau^2)$. We consider expansions to the base of τ of integers in $\mathbb{Z}[\tau]$. It is well known that $\mathbb{Z}[\tau]$, which is the ring of algebraic integers of $\mathbb{Q}(\sqrt{-7})$, is a Euclidean domain and therefore a factorial ring. We write the norm of an element $z = a + b\tau \in \mathbb{Z}[\tau]$ with $a, b \in \mathbb{Z}$ as

$$N(z) = z\bar{z} = (a+b\tau)(a+b\bar{\tau}) = a^2 + ab(\tau+\bar{\tau}) + b^2\tau\bar{\tau} = a^2 + \mu ab + 2\,b^2.$$

It is the square of the absolute value of z.

Definition 2.1. Let $\mathcal{D}$ be a (finite) subset of $\mathbb{Z}[\tau]$ containing 0 and $w \geq 1$ be an integer. A *$\mathcal{D}$-expansion* of $z \in \mathbb{Z}[\tau]$ is a sequence $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j\geq 0} \in \mathcal{D}^{\mathbb{N}_0}$ such that

1. Only a finite number of the digits ε_j is nonzero.
2. $\mathsf{value}(\boldsymbol{\varepsilon}) := \sum_{j\geq 0} \varepsilon_j \tau^j = z$, i.e., $\boldsymbol{\varepsilon}$ is indeed an expansion of z.

The *Hamming weight* of $\boldsymbol{\varepsilon}$ is the number of nonzero digits ε_j. The *length* of $\boldsymbol{\varepsilon}$ is defined as

$$\mathsf{length}(\boldsymbol{\varepsilon}) := 1 + \max\{j : \varepsilon_j \neq 0\}\ .$$

A $\mathcal{D}$-expansion of z is called a *$\mathcal{D}$-w-Non-Adjacent-Form ($\mathcal{D}$-w-NAF) of z*, if it satisfies the following *width-w non-adjacency property*:

3. Each block $(\varepsilon_{j+w-1}, \ldots, \varepsilon_j)$ of w consecutive digits contains at most one nonzero digit ε_k, $j \leq k \leq j+w-1$.

A $\{0, \pm 1\}$-2-NAF is also called a *τ-NAF*.

The set $\mathcal{D}$ is called a *w-Non-Adjacent-Digit-Set (w-NADS)*, if each $z \in \mathbb{Z}[\tau]$ has a $\mathcal{D}$-w-NAF.

The set $\mathcal{D}$ is called *symmetric,* if $d \in \mathcal{D}$ implies $-d \in \mathcal{D}$. In this case we can partition $\mathcal{D}$ as $\mathcal{D} = \{0\} \cup \mathcal{D}^+ \cup (-\mathcal{D}^+)$ with $\mathcal{D}^+ \cap (-\mathcal{D}^+) = \emptyset$. The set $\mathcal{D}^+$ is called set of *positive* digits and the digits in $-\mathcal{D}^+$ are the *negative* digits.

Typically, we choose $\mathcal{D}$ to be a set of cardinality $1 + 2^{w-1}$, but we do not require this in the definition. One of our aims is to investigate which $\mathcal{D}$ are w-NADS, and we shall usually restrict ourselves to digit sets formed by adjoining 0 to a reduced residue system modulo τ^w, which is defined as usual:

Definition 2.2. Let $w \geq 1$ be an integer. A *reduced residue system $\mathcal{D}'$* for the number ring $\mathbb{Z}[\tau]$ modulo τ^w is a set of representatives of the prime congruence classes of $\mathbb{Z}[\tau]$ modulo τ^w, i.e., the congruence classes that are coprime to τ.

For a digit set $\mathcal{D}$ formed by 0 together with a reduced residue system modulo τ^w, Algorithm 1 either recodes an integer $z \in \mathbb{Z}[\tau]$ to the base of τ, or enters in a infinite loop for some inputs when $\mathcal{D}$ is not a NADS (the fact that the algorithm actually enters a loop if it does not terminate is shown in Proposition 2.6).

Example 2.3. Just using a digit set which consists of 0 and a reduced residue system does not imply that Algorithm 1 terminates. This has been observed in the binary case for NAF-like expansions of rational integers to the base of 2 by Muir and Stinson [29]. If we take $w = 1$ and the digit set $\{0, 1-\tau\}$ (here the corresponding reduced residue set modulo $\tau = \tau^1$ comprises the single element $1-\tau$) we see that the element 1 has an expansion

$$(1-\tau) + (1-\tau)\tau + (1-\tau)\tau^2 + (1-\tau)\tau^3 + \cdots\ .$$

Algorithm 1 does not terminate in this case.

For later use, we note the following:

Remark 2.4. A number $a + \tau b \in \mathbb{Z}[\tau]$ with $a, b \in \mathbb{Z}$ is relatively prime to τ if and only if a is odd. This follows from the fact that τ is a prime element in $\mathbb{Z}[\tau]$ and that τ divides a rational integer if and only if this rational integer is even.

Algorithm 1. General windowed integer recoding

INPUT: An element z from $\mathbb{Z}[\tau]$, a natural number $w \geq 1$ and a reduced residue system $\mathcal{D}'$ for the number ring R modulo τ^w.
OUTPUT: A representation $z = \sum_{j=0}^{\ell-1} \varepsilon_j \tau^j$ of length ℓ of the integer z satisfying the width-w non-adjacency property (3). If no such representation exists, the algorithm does not terminate.

```
$j \leftarrow 0$, $u \leftarrow z$
while $u \neq 0$ do
    if $\tau \mid u$ then
        $\varepsilon_j \leftarrow 0$                                  [Output 0]
    else
        Let $\varepsilon_j \in \mathcal{D}'$ s.t. $\varepsilon_j \equiv z \pmod{\tau^w}$     [Output $\varepsilon_j$]
    $u \leftarrow u - \varepsilon_j$
    $u \leftarrow u/\tau$
    $j \leftarrow j + 1$
$\ell \leftarrow j$
return $(\{\varepsilon_j\}_{j=0}^{\ell-1}, \ell)$
```

2.1. Algorithmic Characterization

As already mentioned, one aim of this paper is to investigate which digit sets $\mathcal{D}$ are in fact w-NADS. For concrete $\mathcal{D}$ and w, this question can be decided algorithmically using ideas of Matula [26]:

Theorem 1. *Let $\mathcal{D}$ be a finite subset of $\mathbb{Z}[\tau]$ containing 0 and $w \geq 1$ be an integer.*

Let

$$M := \left\lfloor \frac{\max\{N(d) : d \in \mathcal{D}\}}{\left(2^{w/2} - 1\right)^2} \right\rfloor,$$

where $N(z)$ denotes the norm of z.

Consider the directed graph $G = (V, A)$ defined by its set of vertices

$$V := \{0\} \cup \{z \in \mathbb{Z}[\tau] : N(z) \leq M \,, \tau \nmid z\}$$

and set of arcs

$$A := \{(y, z) \in V^2 \;:\; \text{There exist } d \in \mathcal{D} \setminus \{0\} \text{ and an integer } v \geq w \text{ s.t. } z = \tau^v y + d\}\ .$$

Then $\mathcal{D}$ is a w-NADS if and only if the following conditions are both satisfied.

1. *The set $\mathcal{D}$ contains a reduced residue system modulo τ^w.*
2. *In $G = (V, A)$, each vertex $z \in V$ is reachable from 0.*

If $\mathcal{D}$ is a w-NADS and $\mathcal{D} \setminus \{0\}$ is a reduced residue system modulo τ^w, then each $z \in \mathbb{Z}[\tau]$ has a unique $\mathcal{D}$-w-NAF.

The following lemma will be used in the proof of the theorem.

Lemma 2.5. *Let $z \in V$ and $d \in \mathcal{D}$ with $d \equiv z \pmod{\tau^w}$. Then there exists an integer $v \geq w$ such that $(z - d)/\tau^v$ is either 0 or not divisible by τ. Furthermore, $(z - d)/\tau^v \in V$.*

Proof. By construction, $(z - d)/\tau^w$ is an element of $\mathbb{Z}[\tau]$, and a power τ^v of τ possibly higher than τ^w divides $z - d$. Our v is the maximal exponent of such a power. We have

$$\sqrt{N\left(\frac{z-d}{\tau^v}\right)} = \frac{|z-d|}{2^{v/2}} \leq \frac{|z| + |d|}{2^{v/2}}$$

$$\leq \frac{|z| + |d|}{2^{w/2}} \leq \frac{\frac{\max\{|d| : d \in \mathcal{D}\}}{(2^{w/2}-1)} + |d|}{2^{w/2}} \leq \frac{\max\{|d| : d \in \mathcal{D}\}}{(2^{w/2} - 1)}\ .$$

Since $N((z - d)/\tau^v)$ is an integer, we conclude that $N((z - d)/\tau^v) \leq M$. □

Proof of Theorem 1. We first assume that $\mathcal{D}$ is a w-NADS. Let $z = a + b\tau$ be relatively prime to τ and let $\boldsymbol{\varepsilon}$ be its $\mathcal{D}$-w-NADS. Obviously, we have $1 \equiv a \equiv z \equiv \varepsilon_0 \pmod{\tau}$. Thus, $\varepsilon_0 \neq 0$ and therefore $\varepsilon_1 = \cdots = \varepsilon_{w-1} = 0$, whence $z \equiv \varepsilon_0 \pmod{\tau^w}$. Thus $\mathcal{D}$ contains a reduced residue system modulo τ^w.

Assume that $0 \neq z \in V$. Let $\boldsymbol{\varepsilon}$ be the $\mathcal{D}$-w-NADS of z. We have $\varepsilon_0 \neq 0$, we set $y = (z-\varepsilon_0)/\tau^v$, where v is the highest exponent for which τ^v divides $z - \varepsilon_0$. By Lemma 2.5 we have $v \geq w$, $y \in V$ and $(y, z) \in G$. A $\mathcal{D}$-w-NAF of y can be obtained by omitting the last digits of $\boldsymbol{\varepsilon}$. Repeating this finitely often, we arrive at 0. Using the arcs in reverse order we see that z is reachable from 0.

Conversely, we assume that the two conditions are fulfilled. We first show that every $z \in V$ has a $\mathcal{D}$-w-NAF by induction on the distance from 0 to z in G. Let $0 \neq z \in V$. Then there is a $y \in V$ with has a smaller distance from 0 than z and is a predecessor of z in G. By induction, y has a $\mathcal{D}$-w-NAF. Since $z = \tau^v y + d$ for some nonzero $d \in \mathcal{D}$ and an integer $v \geq w$, we get a $\mathcal{D}$-w-NAF of z by appending $(0, 0, \ldots, 0, d)$ (with $v - 1$ zeros) to the $\mathcal{D}$-w-NAF of y.

Next, we prove that all $z \in \mathbb{Z}[\tau]$ have a $\mathcal{D}$-w-NAF by induction on $N(z)$. Let $z \in \mathbb{Z}[\tau] \setminus \{0\}$. If $N(z) \leq M$ and $\tau \mid z$ then let τ^v be the highest power of τ dividing z. Put $y = z/\tau^v$. Now $N(y) = N(z)/2^v < N(z) \leq M$ and y is not divisible by τ, hence y is in V and we know it has a $\mathcal{D}$-w-NAF. The $\mathcal{D}$-w-NAF of z will be obtained by appending v zeros to the $\mathcal{D}$-w-NAF of z/τ^v.

We may now assume that $N(z) > M$ and therefore

$$|z|(2^{w/2} - 1) > \max\{|d| : d \in \mathcal{D}\} \ .$$

If τ divides z, we set $y = z/\tau$ with $N(y) = N(z)/2$. If τ does not divide z, we have $\gcd(z, \tau) = 1$. Thus there are $d \in \mathcal{D}$ and an integer $v \geq w$ and $y \in \mathbb{Z}[\tau]$ with $z = \tau^v y + d$ such that $y = 0$ or y is not divisible by τ. We have

$$\sqrt{N(y)} = \frac{|z-d|}{2^{v/2}} \leq \frac{|z-d|}{2^{w/2}} < \frac{|z| + |z|(2^{w/2}-1)}{2^{w/2}} = \sqrt{N(z)} \ .$$

Thus we may take a $\mathcal{D}$-w-NAF of y and append 0 or $(0, 0, \ldots, 0, d)$ (with $v - 1$ zeros) respectively to obtain a $\mathcal{D}$-w-NAF of z.

Finally we assume that $\mathcal{D} \setminus \{0\}$ is a reduced residue system modulo τ^w and $\mathcal{D}$ is a w-NADS. Assume that some z has two $\mathcal{D}$-w-NAFs $\boldsymbol{\varepsilon}$ and $\boldsymbol{\eta}$. If $z \equiv 0 \pmod{\tau}$, we must have $\varepsilon_0 = \eta_0 = 0$ and we continue with z/τ. If $z \not\equiv 0 \pmod{\tau}$, then we must have $\varepsilon_0 \neq 0$ and $\eta_0 \neq 0$, and the w-NAF property implies that $\varepsilon_j = \eta_j = 0$ for $1 \leq j < w$. Therefore, we have $\varepsilon_0 \equiv \eta_0 \pmod{\tau^w}$, whence $\varepsilon_0 = \eta_0$. Thus we continue with $(z - \varepsilon_0)/\tau^w$. By induction, we see that $\boldsymbol{\varepsilon} = \boldsymbol{\eta}$. □

We now investigate the case of sets which are not w-NADS:

Proposition 2.6. *Let $w \geq 1$ be an integer, $\mathcal{D}$ be a finite subset of $\mathbb{Z}[\tau]$ consisting of 0 and a reduced residue system modulo τ^w. Then the following conditions are equivalent:*

1. *The set $\mathcal{D}$ is not a w-NADS.*
2. *There is a $z \in \mathbb{Z}[\tau]$ such that Algorithm 1 enters an infinite loop.*
3. *There is a $z \in \mathbb{Z}[\tau]$, a positive integer ℓ and a $\mathcal{D}$-w-NAF $\boldsymbol{\varepsilon}$, such that*

$$z(1 - \tau^\ell) = \mathsf{value}(\boldsymbol{\varepsilon}) \ \textit{and}\ \ell \geq \mathsf{length}(\boldsymbol{\varepsilon}) + w - 1. \tag{4}$$

Proof. If $\mathcal{D}$ is not a w-NADS, then there is a vertex $z' \in V$ which is not reachable from 0 by Theorem 1. By Lemma 2.5, every vertex except 0 has indegree 1. Since V is finite, this implies that there is a cycle in G. For any vertex z contained in such a cycle, Algorithm 1 visits the vertices of this cycle in reverse order, thus it enters an infinite loop.

On the other hand, if Algorithm 1 enters an infinite loop, then $\mathcal{D}$ is not a w-NADS by definition.

Algorithm 1 enters an infinite loop containing some $z \in \mathbb{Z}[\tau]$ if and only if it produces an expansion of the form

$$z = \sum_{j=0}^{\ell-1} \varepsilon_j \tau^j + z\tau^\ell$$

for some integer ℓ and some sequence $(\varepsilon_j)_{0 \leq j < \ell}$ satisfying the width-w non-adjacency property. This is equivalent to (4). □

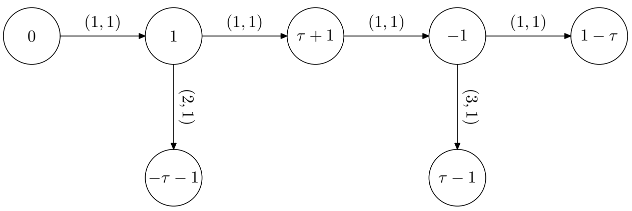

FIGURE 1. Directed Graph G for $\mu=-1$, $w=1$, $\mathcal{D}=\{0,1\}$. The arcs are labeled with (v,d) as in the definition of the graph, i.e., $y \xrightarrow{(v,d)} z$ means that $z=\tau^v y+d$.

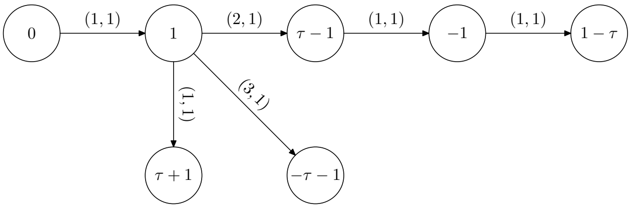

FIGURE 2. Directed Graph G for $\mu=1$, $w=1$, $\mathcal{D}=\{0,1\}$. The arcs are labeled as in Figure 1.

We now discuss two well-known examples.

Example 2.7. Let $w=1$ and $\mathcal{D}=\{0,1\}$. By Remark 2.4, there is only one residue class prime to τ. In this case $M=5$, so $V=\{0,\pm1,\pm1\pm\tau\}$. The corresponding directed graphs are shown in Figures 1 and 2 for $\mu=-1$ and $\mu=1$, respectively. We see that in both cases, all 7 states are reachable from 0. Thus, $\{0,1\}$ is a 1-NADS. This is equivalent to saying that τ is the base of a canonical number system in $\mathbb{Z}[\tau]$ in the sense of Kátai and Szabó [21]. This result is contained in the general characterisation of canonical number systems in imaginary quadratic number fields by Kátai and Kovács [20] and Gilbert [16].

It is also easy to see that the expected density of a τ-adic espansion of an element of $\mathbb{Z}[\tau]$ using this digit set is 1/2. Also the expansion of integers given in [27] has density 1/2, but this is not the same representation. In fact, in [27] the digit set $\{0,\pm1\}$ is used, but the algorithm generating the expansion is not optimal.

Remark 2.8. Example 2.7 immediately shows that there are exactly 2^w residue classes modulo τ^w; a complete residue system is given by $\sum_{j=0}^{w-1}\varepsilon_j\tau^j$ with $\varepsilon_j\in\{0,1\}$ for $0\le j<w$. There are 2^{w-1} residue classes coprime to τ^w, a reduced residue system is given by $1+\sum_{j=1}^{w-1}\varepsilon_j\tau^j$ with $\varepsilon_j\in\{0,1\}$ for $1\le j<w$.

Example 2.9. Let $w=2$ and $\mathcal{D}=\{0,\pm1\}$. Using Remark 2.8, it is easily seen that $\{\pm1\}$ is a reduced residue system modulo τ^2. In this case, $M=1$, the graph G consists of the three states $V=\{0,\pm1\}$ only, and those are obviously reachable from 0. Thus $\{0,\pm1\}$ is a 2-NADS. This has been proved by Solinas [36, 37].

Example 2.10. One might consider the digit set $\mathcal{D}=\{0\}\cup\{\pm1,\pm3,\ldots,\pm(2^{w-1}-1)\}$. The odd digits form a reduced residue system modulo τ^w, since τ^w divides a rational integer if and only if 2^w divides it (note that τ and $\bar\tau$ are coprime primes in $\mathbb{Z}[\tau]$).

However, this digit set is not a w-NADS for all w. For instance, for $w=6$, the number $1-\mu\tau$ has no $\mathcal{D}$-6-NAF, since

$$(1-\mu\tau)(1-\tau^{24})=-9\tau^{18}-27\tau^{12}+9\tau^6+27$$

and therefore by Proposition 2.6 Algorithm 1 enters an infinite loop.

Using Theorem 1, we can verify that for $w\in\{2,3,4,5,7,8,9,10\}$, this set $\mathcal{D}$ is a w-NADS.

2.2. Representatives of Minimal Norm

In this section, we revisit the digit set proposed by Solinas [36, 37]. He proposed to take an element of minimal norm from each prime residue class modulo τ^w. At first sight, it is not clear whether this choice is unique. But it can be proved to be the case:

Theorem 2. *Let τ, $w \geq 2$ be as above, and $\mathsf{MNR}(w)$ a digit set consisting of 0 together with one element of minimal norm from each prime residue class modulo τ^w.*

The digit set $\mathsf{MNR}(w)$ is uniquely determined. In other words, in each prime residue class modulo τ^w there exists a unique element of minimal norm.

Proof. Let α, β be distinct elements of minimal norm in the same prime residue class modulo τ^w. Then, $\beta = \alpha + \gamma\tau^w$ with $\gamma \in \mathbb{Z}[\tau] \setminus \{0\}$.

By [27, Lemma 2] (see also [37, Corollary 59 and Equation 64]) we have $N(\alpha), N(\beta) \leq \frac{4}{7}N(\tau^w)$, hence

$$\sqrt{N(\gamma)N(\tau^w)} = \sqrt{N(\alpha-\beta)} \leq \sqrt{N(\alpha)} + \sqrt{N(\beta)} \leq \frac{4}{\sqrt{7}}\sqrt{N(\tau^w)} \ .$$

This implies $N(\gamma) \leq \frac{16}{7}$ and therefore $N(\gamma) \leq 2$. Being $\gamma \neq 0$, it can only be $N(\gamma) = 1$ or 2.

Now we make use of the fact that τ is prime and does not divide α nor $\bar{\tau}$. Writing down the relation $N(\alpha + \gamma\tau^w) = N(\alpha)$ explicitly we obtain $\alpha\bar{\gamma}\bar{\tau}^w + \bar{\alpha}\gamma\tau^w + \gamma\bar{\gamma}\tau^w\bar{\tau}^w = 0$. This implies that τ^w divides $\alpha\bar{\gamma}\bar{\tau}^w$ and thus $\bar{\gamma}$. Therefore $2^w = N(\tau^w)$ divides $N(\bar{\gamma}) = N(\gamma)$ which we know to be either 1 or 2, implying in turn $w \leq 1$. This is a contradiction. □

Remark 2.11. The result does not really depend on Meier and Staffelbach's result [27, Lemma 2]: Since $\mathbb{Z}[\tau]$ is known to be an Euclidean domain with respect to the norm function, Euclidean division by τ^w shows that every residue class modulo τ^w has a representative of norm less than 2^w. This yields $N(\gamma) < 4$ in the above proof, which is still sufficient.

2.3. Syntactic Sufficient Conditions

The aim of this section is to prove sufficient conditions for families of sets $\mathcal{D}$ to be a w-NADS at the level of digits of the τ-NAF. In contrast to Theorem 1, where a decision can be made for any concrete set $\mathcal{D}$, we will now focus on families of such sets. Blake, Murty, and Xu [13] gave such sufficient conditions based on the norm of the numbers involved.

Proposition 2.12. *Let $w \geq 1$ and $\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}'$ two τ-NAFs. Then $\mathsf{value}(\boldsymbol{\varepsilon}) \equiv \mathsf{value}(\boldsymbol{\varepsilon}') \pmod{\tau^w}$ if and only if*

$$\varepsilon_j = \varepsilon_j' \text{ for } 0 \leq j \leq w-2 \text{ and } |\varepsilon_{w-1}| = |\varepsilon_{w-1}'| \ . \tag{5}$$

Proof. Assume first that (5) holds. If $\varepsilon_{w-1} = \varepsilon_{w-1}'$, then it is clear that $\mathsf{value}(\boldsymbol{\varepsilon}) \equiv \mathsf{value}(\boldsymbol{\varepsilon}') \pmod{\tau^w}$. W.l.o.g., we may now assume that $\varepsilon_{w-1} = 1$ and $\varepsilon_{w-1}' = -1$. In this case we have $\mathsf{value}(\boldsymbol{\varepsilon}) - \mathsf{value}(\boldsymbol{\varepsilon}') \equiv 2\tau^{w-1} \equiv 0 \pmod{\tau^w}$ by (2).

To prove the converse direction, we proceed by induction on w. For $w = 1$, we note that $\varepsilon_0 \equiv \mathsf{value}(\boldsymbol{\varepsilon}) \equiv \mathsf{value}(\boldsymbol{\varepsilon}') \equiv \varepsilon_0' \pmod{\tau}$ implies $|\varepsilon_0| = |\varepsilon_0'|$ since both least significant digits are elements of $\{0, \pm 1\}$. We now consider the case of general w. Assume that $\mathsf{value}(\boldsymbol{\varepsilon}) \equiv \mathsf{value}(\boldsymbol{\varepsilon}') \pmod{\tau^w}$. By induction hypothesis, we have $\varepsilon_j = \varepsilon_j'$ for $0 \leq j \leq w-3$ and $|\varepsilon_{w-2}| = |\varepsilon_{w-2}'|$.

We first consider the case that $\varepsilon_{w-2} = \varepsilon_{w-2}'$. In that case we conclude that $\mathsf{value}(\boldsymbol{\varepsilon}) - \mathsf{value}(\boldsymbol{\varepsilon}') \equiv (\varepsilon_{w-1} - \varepsilon_{w-1}')\tau^{w-1} \pmod{\tau^w}$, which implies that $\varepsilon_{w-1} \equiv \varepsilon_{w-1}' \pmod{\tau}$ and therefore $|\varepsilon_{w-1}| = |\varepsilon_{w-1}'|$. Thus (5) is proved in this case.

Finally, we consider the case that $\varepsilon_{w-2} \neq \varepsilon_{w-2}'$. W.l.o.g., we may assume that $\varepsilon_{w-2} = 1$ and $\varepsilon_{w-2}' = -1$. Since $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}'$ are both τ-NAFs, the subsequent digits ε_{w-1} and ε_{w-1}' must both vanish. But this implies that $\mathsf{value}(\boldsymbol{\varepsilon}) - \mathsf{value}(\boldsymbol{\varepsilon}') \equiv 2\tau^{w-2} \equiv \mu\tau^{w-1} \pmod{\tau^w}$, a contradiction. Thus this case cannot occur. □

w	μ	$\mathcal{D}$	Remark
3	-1	$\{1, -1, -\tau^2+1, -\tau^2-1\}$	$(-\tau-1)\left(1-\tau^3\right) = -\tau^2+1$
3	-1	$\{1, -1, -\tau^2+1, \tau^2-1\}$	$(-\tau-1)\left(1-\tau^3\right) = -\tau^2+1$
3	-1	$\{1, -1, \tau^2+1, \tau^2-1\}$	$(\tau+1)\left(1-\tau^3\right) = \tau^2-1$
3	1	$\{1, -1, -\tau^2+1, \tau^2-1\}$	$(-\tau+1)\left(1-\tau^6\right) = \left(-\tau^2+1\right)\tau^3+\tau^2-1$

Table 1. List of sets of short τ-NAF representatives which are not a w-NADS. The "Remark" column contains an example of an element which cannot be represented.

Definition 2.13. Let w be a positive integer and $\mathcal{D}$ be a subset of

$$\{\,0\,\} \cup \{\, \mathsf{value}(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon} \text{ is a } \tau\text{-NAF of length at most } w \text{ with } \varepsilon_0 \neq 0 \,\}$$

consisting of 0 and a reduced residue system modulo τ^w. Then $\mathcal{D}$ is called a *set of short τ-NAF representatives* for τ^w.

By Proposition 2.12, an example for a set of short τ-NAF representatives is

$$\begin{aligned}\mathsf{SNR}(w) = \{\,0\,\} \cup \Big\{\, \mathsf{value}(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon} \text{ is a } \tau\text{-NAF of length at most } w \\ \text{with } \varepsilon_0 \neq 0 \text{ and } \varepsilon_{w-1} \in \{0, \varepsilon_0\} \,\Big\}\ .\end{aligned} \tag{6}$$

All other sets of short τ-NAF representatives are obtained by changing the signs of ε_{w-1} without changing ε_0 in some of the $\boldsymbol{\varepsilon}$. Note that for $w \geq 3$, the digits of $\mathsf{SNR}(w)$ given in (6) can be seen as the values of the words of length w of the language given by the regular expression

$$0^* + (0 + 01 + 0\bar{1})^*(01 + 0\bar{1}) + 1(0 + 01 + 0\bar{1})^*01 + \bar{1}(0 + 01 + 0\bar{1})^*0\bar{1}\ .$$

From this regular expression, it is easy to check that the cardinality of $\mathsf{SNR}(w)$ is indeed $1 + 2^{w-1}$. However, this is also a consequence of Example 2.7 and Proposition 2.12.

The main result of this section is the following theorem, which states that in almost all cases, a set of short τ-NAF representatives is a w-NADS:

Theorem 3. *Let w be a positive integer and $\mathcal{D}$ a set of short τ-NAF representatives. Then $\mathcal{D}$ is a w-NADS if and only if it is not listed in Table* 1.

In particular, if $w \geq 4$, then $\mathcal{D}$ is always a w-NADS. Moreover, $\mathsf{SNR}(w)$ *as defined in* (6) *is a w-NADS for all $w \geq 2$.*

Proof. For $w \in \{1, 2\}$, all choices of $\mathcal{D}$ are those studied in Examples 2.7 and 2.9. These turned out to be w-NADS. For $w = 3$, there are only the possibilities $\mathcal{D} = \{0, 1, -1, \pm\tau^2 + 1, \pm\tau^2 - 1\}$ for independent signs in front of τ^2. Using Theorem 1, these have been checked and Table 1 has been established based on the results.

So the only remaining case is that of $w \geq 4$. Let $z \in \mathbb{Z}[\tau]$ be relatively prime to τ, choose $d = \mathsf{value}(\boldsymbol{\varepsilon}) \in \mathcal{D}$ such that $d \equiv z \pmod{\tau^w}$ and set $y = (z - d)/\tau^w$. Denote the τ-NAF of z by $\boldsymbol{\eta}$. We set $y' := \sum_{j\geq 0} \eta_{j+w}\tau^j$, i.e., the number created by truncating the least significant w digits of the τ-NAF of z.

We claim that either $y = y'$ or y' is a multiple of τ and $y = y' \pm \bar{\tau}$. If $\eta_{w-1} = 0$ then the number formed by the w least significant digits of $\boldsymbol{\eta}$, which is $(0\,\eta_{w-2} \ldots \eta_1\,\eta_0)_\tau$, is in $\mathcal{D}$, hence it is d and $y = y'$. Otherwise $\eta_w = 0$ and y' is divisible by τ, and from Proposition 2.12 together with the fact that $\bar{\tau} = 2 \cdot \tau^{-1}$ we see that $y \in \{y', y' \pm \bar{\tau}\}$.

Next, we want to show that the length of the τ-NAF of $y' \pm \bar{\tau}$ is at most the length of the τ-NAF of y' increased by 3. If we can prove this, since the length of the τ-NAF of y' equals the length of the τ-NAF of z decreased by w, we conclude that the length of the τ-NAF of y is smaller than the length of the τ-NAF of z. From this it follows that repeatedly choosing $d \equiv z \pmod{\tau^w}$ in $\mathcal{D}$ and replacing z with $(z - d)/\tau^w$ will eventually terminate with 0 and yield a $\mathcal{D}$-w-NAF of z.

To prove our claim about the length of the τ-NAF of $y' \pm \bar{\tau}$ we study the behaviour of even τ-adic NAFs upon addition or subtraction of $\bar{\tau}$. We therefore consider transducer automata which compute the τ-NAF of $y' \pm \bar{\tau}$ from the τ-NAF of y'.

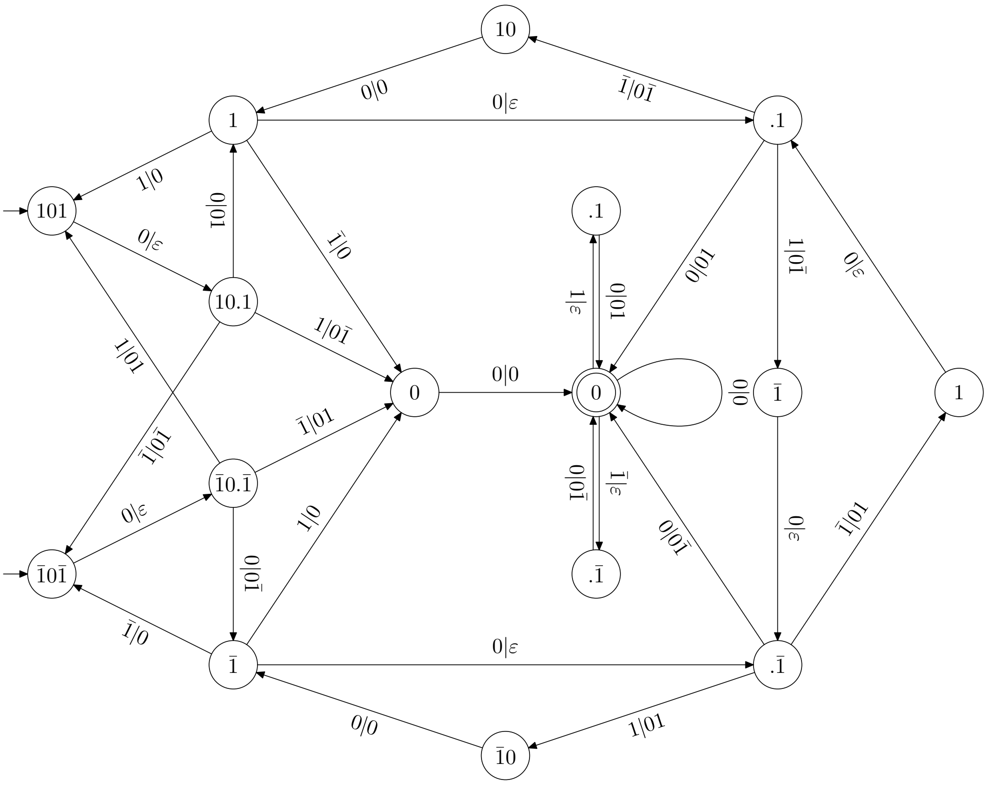

FIGURE 3. Transducer for the addition of $\pm\bar{\tau}$ for $\mu = -1$. Addition of $\bar{\tau}$ and $-\bar{\tau}$ corresponds to starting at states 101 and $\bar{1}0\bar{1}$, respectively.

These are shown in Figures 3 and 4 for $\mu = -1$ and $\mu = 1$, respectively. Similar transducers have already been shown in [7, Figures 14 and 15] for the addition of ± 1 to a τ-NAF. Apart from the new initial states the last simplification step has been omitted in Figures 3 and 4 in order to make the labels of the states more consistent: they have now been chosen to represent the carry, and the "τ-point" corresponds to the look-ahead.

From these transducers, it is easily seen that the length of the τ-NAF of $y' \pm \bar{\tau}$ is at most the length of the τ-NAF of y' increased by 3. This concludes the proof of the theorem. □

The following result is concerned with the lengths of recodings that make use of the set of short τ-NAF representatives.

Theorem 4. *Let $w \geq 2$ be a positive integer, $\mathcal{D}$ a set of short τ-NAF representatives, and $\boldsymbol{\varepsilon}$ a $\mathcal{D}$-w-NAF of some $z \in \mathbb{Z}[\tau]$.*

Then the length of $\boldsymbol{\varepsilon}$ can be bounded by

$$2\log_2|z| - w - 0.18829 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 7.08685 \ , \qquad \textit{if } w \geq 4 \ , \tag{7}$$

$$2\log_2|z| - 2.61267 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 5.01498 \ , \qquad \textit{if } w = 3 \ , \tag{8}$$

$$2\log_2|z| - 0.54627 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 3.51559 \ , \qquad \textit{if } w = 2 \ . \tag{9}$$

Note that (9) is Solinas' [37] Equation (53).

Proof. We first consider the case $w \geq 4$. By definition of $\mathcal{D}$, every nonzero digit of $\mathcal{D}$ has a τ-NAF of length at most w. Replacing each block $(0, \dots, 0, d)$ of $\boldsymbol{\varepsilon}$ by the τ-NAF of $d \in \mathcal{D}$ yields a $\{0, \pm 1\}$-expansion $\boldsymbol{\eta}$ of z. By construction, this expansion $\boldsymbol{\eta}$ has the following property: if $|\eta_j| = |\eta_{j+1}|$ holds for some j, then the block $(\eta_{j+w}, \dots, \eta_{j+1})$ satisfies the 2-NAF condition, i.e., $\eta_{k+1} \cdot \eta_k = 0$ for $j + 1 \leq k \leq j + w - 1$. Furthermore, we have

$$\mathsf{length}(\boldsymbol{\varepsilon}) \leq \mathsf{length}(\boldsymbol{\eta}) \leq \mathsf{length}(\boldsymbol{\varepsilon}) + w - 1 \ . \tag{10}$$

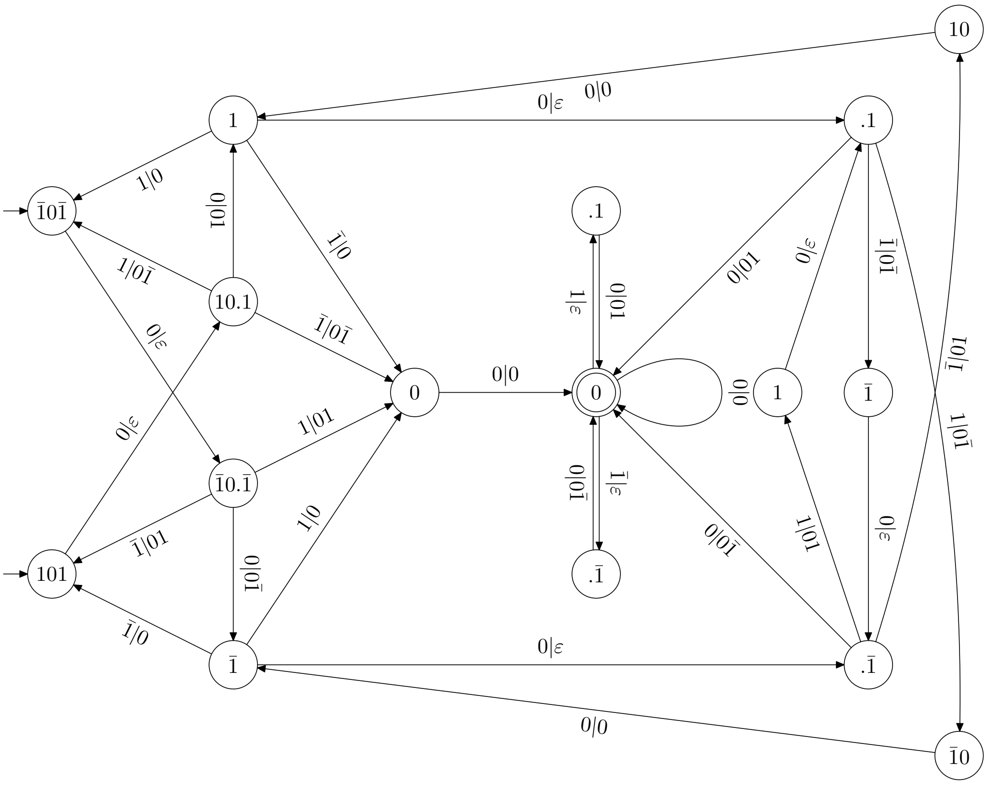

FIGURE 4. Transducer for the addition of $\pm\bar{\tau}$ for $\mu = +1$. Addition of $\bar{\tau}$ and $-\bar{\tau}$ corresponds to starting at states $\bar{1}0\bar{1}$ and 101, respectively.

We now derive a bound for $\mathsf{length}(\boldsymbol{\eta})$ which is independent of w. To that aim, we relax the above syntactical condition. More precisely, we only use that

$$\begin{aligned}\boldsymbol{\eta} \in \mathcal{L} := \Big\{ \boldsymbol{\theta} \in \{0, \pm 1\}^{\mathbb{N}_0} \ :\ & \text{There is no } j \in \mathbb{N}_0 \text{ such that} \\ & |\theta_j| = |\theta_{j+1}| = |\theta_{j+2}| = 1 \text{ or such that} \\ & |\theta_j| = |\theta_{j+1}| = |\theta_{j+3}| = |\theta_{j+4}| = 1 \Big\} \ .\end{aligned}$$

We denote the maximum and the minimum of the norm of words of $\mathcal{L}$ of length d by

$$\begin{aligned} N^{\mathcal{L}}_{\max}(d) &:= \max\{N(\mathsf{value}(\boldsymbol{\theta})) : \boldsymbol{\theta} \in \mathcal{L} \text{ and } \mathsf{length}(\boldsymbol{\theta}) = d\} \ , \\ N^{\mathcal{L}}_{\min}(d) &:= \min\{N(\mathsf{value}(\boldsymbol{\theta})) : \boldsymbol{\theta} \in \mathcal{L} \text{ and } \mathsf{length}(\boldsymbol{\theta}) = d\} \ . \end{aligned}$$

Solinas' [37] estimates (from Lemma 35 to Corollary 51) in the case of the τ-NAF remain valid for our quantities $N^{\mathcal{L}}_{\max}(d)$ and $N^{\mathcal{L}}_{\min}(d)$. Thus in our case, Solinas' Theorem 2 reads

$$\left(\sqrt{N^{\mathcal{L}}_{\min}(d)} - \frac{\sqrt{N^{\mathcal{L}}_{\max}(d)}}{2^{d/2}-1}\right)^2 \cdot 2^{\mathsf{length}(\boldsymbol{\eta})-d} < N(z) < \frac{N^{\mathcal{L}}_{\max}(d)}{(2^{d/2}-1)^2} \cdot 2^{\mathsf{length}(\boldsymbol{\eta})}$$

for $\mathsf{length}(\boldsymbol{\eta}) > 2d$. We calculate that

$$\begin{aligned} N^{\mathcal{L}}_{\min}(13) &= 86 \ , & N^{\mathcal{L}}_{\max}(13) &= 18288 \ , \\ N^{\mathcal{L}}_{\min}(15) &= 289 \ , & N^{\mathcal{L}}_{\max}(15) &= 73850 \ . \end{aligned}$$

This yields

$$2\log_2|z| - 1.18830 < \mathsf{length}(\boldsymbol{\eta}) < 2\log_2|z| + 7.08685 \tag{11}$$

for $\mathsf{length}(\boldsymbol{\eta}) > 30$. This bound is present also in Solinas [37, Eq. (53) and Section 9], but it is an unnecessary restriction: If $\mathsf{length}(\boldsymbol{\eta}) \leq 30$, we consider $\tau^k z$ for a sufficiently large integer k and the expansion $\boldsymbol{\eta}' \in \mathcal{L}$ defined by

$$\eta'_j = \begin{cases} 0, & \text{if } j < k, \\ \eta_{j-k}, & \text{if } j \geq k, \end{cases}$$

i.e., $\boldsymbol{\eta}'$ is $\boldsymbol{\eta}$ shifted left by k digits. Since (11) holds for $\tau^k z$ and $\boldsymbol{\eta}'$, it also holds for z and $\boldsymbol{\eta}$.

Together with (10), we obtain (7) for $w \geq 4$.

To obtain the bound for $w = 3$, we consider all 3-NADS (by Theorem 3, there are 4 of them). In these concrete cases, the above calculations can be performed with the concrete language instead of a relaxation $\mathcal{L}$. This yields the given bound. The case $w = 2$ is contained in Solinas [37, Equation (53)]. □

2.4. Point Halving

For any given point P, point halving [22, 35, 34] consists in computing a point Q such that $2Q = P$. This inverse operation to point doubling applies to all elliptic curves over binary fields. Its evaluation is (at least two times) faster than that of a doubling and a halve-and-add scalar multiplication algorithm based on halving instead of doubling can be devised. This method is not useful for Koblitz curves because halving is slower than a Frobenius operation.

In [3] it is proposed to insert a halving in the "τ-and-add" method to speed up Koblitz curve scalar multiplication. This approach brings a non-negligible speedup and was further refined in [6, 7], where the insertion of a halving was implicitly interpreted as a digit set extension. This interpretation is the following: Inserting a halving in the scalar multiplication is equivalent to adding $\pm\bar{\tau}$ to the digit set $\{0, \pm 1\}$. Note that, by Theorem 3, $\mathcal{D} = \{0, \pm 1, \pm\bar{\tau}\} = \mathsf{SNR}(3)$ is the only symmetric 3-NADS of short τ-NAF representatives. In particular $\mathcal{D}' = \{\pm 1, \pm\bar{\tau}\}$ is a reduced residue system modulo τ^3.

As the following theorems state, adjoining more powers of $\bar{\tau}$ to $\mathcal{D}'$ one still gets reduced residue systems, and in some cases these give rise to w-NADS.

Theorem 5. *Let $w \geq 2$. Then $\mathcal{D}' := \{\pm\bar{\tau}^k : 0 \leq k < 2^{w-2}\}$ is a reduced residue system modulo τ^w.*

Proof. The assertion has already been proved for $w = 2$ in Example 2.9, so we assume that $w \geq 3$ in the sequel. We first claim that for $w \geq 3$, we have

$$v_\tau(\bar{\tau}^{2^{w-2}} - 1) = w \ , \tag{12a}$$

$$v_\tau(\bar{\tau}^{2^{w-2}} + 1) = 1 \ , \tag{12b}$$

where for $z \in \mathbb{Z}[\tau]$, $v_\tau(z)$ denotes the maximal integer k such that τ^k divides z.

Now, (12b) is an immediate consequence of (12a) and the fact that $v_\tau(2) = 1$. For $w = 3$, we have $\bar{\tau}^2 = \mu\tau^3 + 1$, which proves (12a) in this case. For $w \geq 4$, we note that $v_\tau(\bar{\tau}^{2^{w-2}} - 1) = v_\tau(\bar{\tau}^{2^{w-3}} - 1) + v_\tau(\bar{\tau}^{2^{w-3}} + 1) = (w-1) + 1 = w$, thus (12a) is proved by induction.

Since the unit group of $\mathbb{Z}[\tau]/\tau^w\mathbb{Z}[\tau]$ has order 2^{w-1} by Remark 2.8, the order of $\bar{\tau}$ modulo τ^w is a power of 2. By (12a), we have $\bar{\tau}^{2^k} \equiv 1 \pmod{\tau^w}$ if and only if $k \geq w - 2$, thus

$$\bar{\tau} \text{ has order } 2^{w-2} \text{ modulo } \tau^w \ . \tag{13}$$

Assume that $\bar{\tau}^\ell \equiv -\bar{\tau}^k \pmod{\tau^w}$ for some $0 \leq k < \ell < 2^{w-2}$. We get $\bar{\tau}^{\ell-k} \equiv -1 \pmod{\tau^w}$. By (13), squaring this congruence shows that 2^{w-2} divides $2(\ell-k) < 2^{w-1}$, thus $(\ell-k) \in \{0, 2^{w-3}\}$. Taking into account (12b), we see that both cases lead to a contradiction.

Since all elements of $\mathcal{D}'$ are relatively prime to τ^w, they are pairwise incongruent modulo τ^w and the cardinality of $\mathcal{D}'$ equals 2^{w-1}, the proof is completed. □

We consider the digit set

$$\mathsf{P}\bar{\tau}(w) := \{0\} \cup \{\pm\bar{\tau}^k : 0 \leq k < 2^{w-2}\} \tag{14}$$

and call it the digit set of "Powers of $\bar{\tau}$." For those w which are relevant in practice, we determine whether this set is indeed a w-NADS.

Theorem 6. *Let $\mathsf{P}\bar{\tau}(w)$ be defined as in* (14). *If $w \in \{2, 3, 4, 5, 6\}$ then $\mathsf{P}\bar{\tau}(w)$ is a w-NADS. If $w \in \{7, 8, 9, 10, 11, 12\}$ then $\mathsf{P}\bar{\tau}(w)$ is not a w-NADS.*

w	$\mathsf{MNR}(w) = \mathsf{P}\bar\tau(w)$	$\mathsf{MNR}(w) = \mathsf{SNR}(w)$	Max length of τ-NAF of a digit	
			$\mathsf{MNR}(w)$	$\mathsf{P}\bar\tau(w)$
2	True	True	1	1
3	True	True	3	3
4	True	True iff $\mu = 1$	4	4
5	False	False	6	8
6	False	False	8	17

Table 2. Comparison between some digit sets.

Proof. For every pair (w, μ) with $w \le 6$ the conditions of Theorem 1 have been verified by heavy symbolic computations.

For $7 \le w \le 12$ and both values of μ the graph G contains loops that are not reachable from 0. In other words, there are elements in $\mathbb{Z}[\tau]$ that have periodic expansions, cf. Proposition 2.6. If $w = 7$ we have $(-9 + 34\mu\tau)(1 - \tau^{16}) = \mu(-\bar\tau^6\tau^7 + \bar\tau^{27})$ and for $8 \le w \le 12$ we have $(371 - 20\mu\tau)(1 - \tau^{24}) = \mu(-\bar\tau^5\tau^{12} + \bar\tau^{41})$. □

Interesting phenomena happen when computing expansions of the same integer for different values of w. Let us consider expansions of 3. For $w \le 7$ the expansions are well-behaved. For example, for $w = 3$ the integer 3 is represented as $-\mu\tau^3 + \mu\bar\tau$ and for $w = 7$ the expansion of 3 is $\mu\bar\tau\tau^{26} - \mu\bar\tau^{15}\tau^{14} - \mu\tau^7 + \mu\bar\tau^{27}$

But, for $w = 8$ we get $3 = -\tau^{538} - \bar\tau^8\tau^{528} - \mu\bar\tau^{14}\tau^{519} + \bar\tau^9\tau^{509} + \bar\tau^{28}\tau^{500} + \bar\tau^{46}\tau^{490} - \mu\bar\tau^8\tau^{481} - \mu\bar\tau^{48}\tau^{473} - \bar\tau\tau^{465} - \bar\tau^{35}\tau^{455} + \bar\tau^{48}\tau^{444} - \mu\bar\tau^{35}\tau^{436} + \bar\tau^{22}\tau^{426} - \mu\bar\tau^{25}\tau^{418} + \bar\tau^{58}\tau^{408} - \bar\tau^{48}\tau^{400} + \mu\bar\tau\tau^{392} - \bar\tau^{42}\tau^{382} + \mu\bar\tau^{21}\tau^{374} - \mu\bar\tau^{27}\tau^{366} - \bar\tau^{22}\tau^{358} + \bar\tau^{40}\tau^{350} - \mu\bar\tau^{47}\tau^{342} + \bar\tau^{39}\tau^{333} + \mu\bar\tau^{16}\tau^{325} + \bar\tau^{46}\tau^{314} + \bar\tau^{20}\tau^{306} + \mu\bar\tau^{39}\tau^{298} + \bar\tau^{10}\tau^{286} - \mu\bar\tau^{61}\tau^{278} + \bar\tau^{35}\tau^{269} - \bar\tau^{62}\tau^{260} - \mu\bar\tau^{51}\tau^{252} - \bar\tau^{53}\tau^{243} - \bar\tau^7\tau^{235} + \bar\tau^{43}\tau^{227} + \bar\tau^{62}\tau^{216} + \mu\bar\tau^{54}\tau^{207} - \mu\bar\tau^{58}\tau^{197} + \mu\bar\tau^{60}\tau^{185} - \bar\tau\tau^{177} - \mu\bar\tau^{54}\tau^{167} + \bar\tau^{37}\tau^{159} - \bar\tau^{16}\tau^{150} - \bar\tau^{22}\tau^{142} + \mu\bar\tau^5\tau^{134} + \bar\tau^{15}\tau^{125} - \mu\bar\tau^{41}\tau^{110} + \bar\tau^{57}\tau^{99} + \mu\bar\tau^{59}\tau^{90} + \mu\bar\tau^{33}\tau^{78} - \mu\bar\tau^{59}\tau^{70} - \mu\bar\tau^{58}\tau^{61} + \bar\tau^{28}\tau^{52} + \mu\bar\tau^{17}\tau^{44} - \mu\bar\tau^{39}\tau^{36} - \mu\bar\tau^{27}\tau^{28} - \mu\bar\tau\tau^{20} - \mu\bar\tau^{42}\tau^{11} + \mu\bar\tau^{59}$.

2.5. Comparing the Digit Sets

So far, three digit sets have been studied: the minimal norm representatives $\mathsf{MNR}(w)$, short NAF representatives $\mathsf{SNR}(w)$, and the powers of $\bar\tau$ digit set $\mathsf{P}\bar\tau(w)$. It is a natural question to ask what are the relations between these sets when they are w-NADS.

As Table 2 shows, $\mathsf{MNR}(w)$ and $\mathsf{P}\bar\tau(w)$ are equal for $w \le 4$. For the same range of w, all digits in these digit sets have a τ-NAF of length at most w, which implies that they are also digit sets of short NAF representatives.

If symmetry is required, there is only *one* w-NADS of short NAF representatives for $w \le 3$ by Theorem 3, namely $\mathsf{SNR}(3)$: it coincides with $\mathsf{MNR}(3)$ and $\mathsf{P}\bar\tau(3)$.

For $w = 4$, there are two symmetric w-NADS of short NAF representatives, namely $\mathsf{SNR}(4)$ and another one. For $\mu = 1$, $\mathsf{SNR}(4)$ coincides with $\mathsf{MNR}(4)$ and $\mathsf{P}\bar\tau(4)$, for $\mu = -1$ the other set coincides with $\mathsf{MNR}(4)$ and $\mathsf{P}\bar\tau(4)$.

For $w \ge 5$, the three concepts are different: the lengths of the τ-NAFs of the digits in $\mathsf{P}\bar\tau(w)$ grow exponentially in w, and the lengths of the digits in $\mathsf{MNR}(w)$ exceed w slightly at most by 2 for the values of w that we have considered (it is a consequence of (9) and of [27, Lemma 2] that $\mathsf{length}(\varepsilon) \le w + 3$ for all $\varepsilon \in \mathsf{MNR}(w)$ for all w).

Table 2 summarizes the above considerations and provides further information. The last two columns show the maximum length of the τ-NAFs of the digits in $\mathsf{MNR}(w)$ and $\mathsf{P}\bar\tau(w)$.

In Figure 5, a comparison between shortest NAF representatives and representatives of minimal norm is shown: For a given triple $(\eta_{-1}, \eta_{-2}, \eta_{-3}) \in \{\pm(1,0,1), \pm(1,0,-1), (0,\pm1,0), (0,0,\pm1), (0,0,0)\}$) an approximation of the set

$$\Big\{\sum_{j\ge1}\varepsilon_{-j}\tau^{-j} : (\varepsilon_{-1}, \varepsilon_{-2}, \varepsilon_{-3}) = (\eta_{-1}, \eta_{-2}, \eta_{-3}), \varepsilon_{-j} \in \{0, \pm1\} \text{ and } \varepsilon_{-j}\varepsilon_{-j-1} = 0 \text{ for all } j \ge 1\Big\}$$

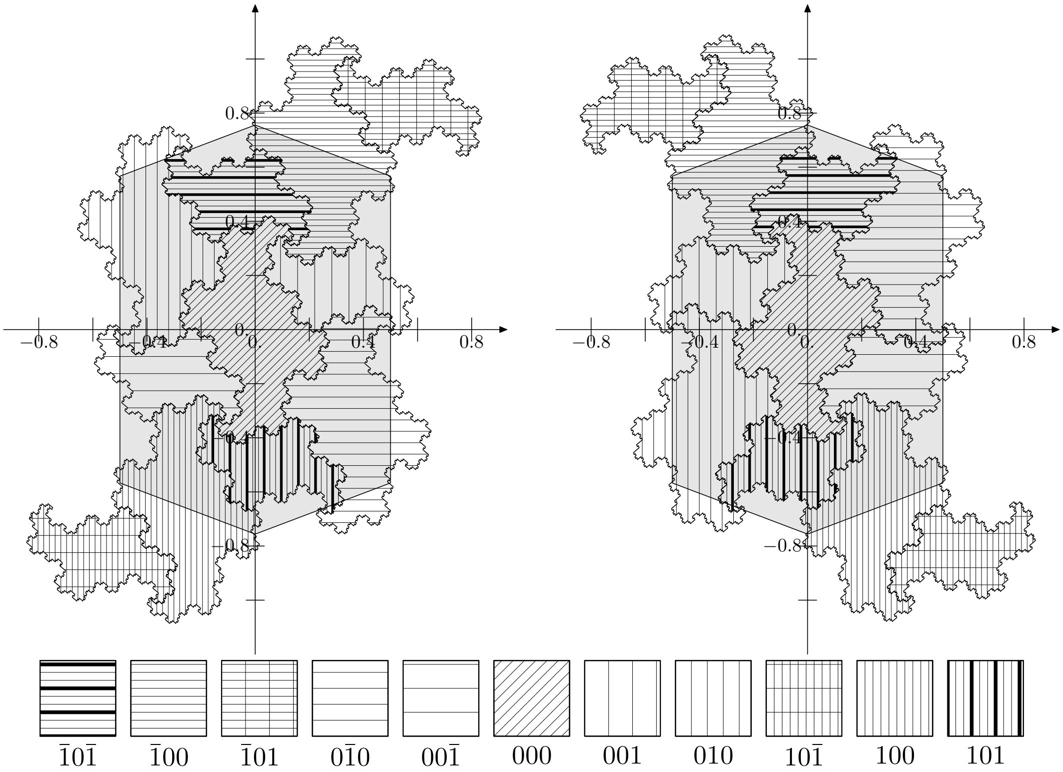

Figure 5. Comparison between shortest NAF representatives and representatives of minimal norm.

is shown and hatched according to the triple $(\eta_{-1}, \eta_{-2}, \eta_{-3})$. The approximation consists of truncating summands with $j > 15$. Thus the possible digits of shortest τ-NAF representatives are the lattice points in the set obtained by multiplying the "fractal" set in Figure 5 by τ^w. The corresponding set for the representatives of minimal norm is the shaded hexagon already shown in Solinas' papers [36, 37]. It is the Voronoĭ region of 0 with respect to the lattice $\mathbb{Z}[\tau]$.

3. Applications to Koblitz Curves

All digit sets seen so far can be used in a τ-and-add scalar multiplication, where we first precompute dP for all $d \in \mathcal{D} \setminus \{0\}$ and then we evaluate the scheme $\sum z_i \tau^i(P)$; in fact, only a half of the precomputations usually suffice since the digits sets $\mathsf{MNR}(w)$, $\mathsf{SNR}(w)$ and $\mathsf{P}\bar{\tau}(w)$ are all symmetric, i.e., the non-zero digits come in pairs of opposite sign.

The digit set $\mathsf{SNR}(w)$ from Section 2.3 simplifies the precomputation phase. The digit set $\mathsf{P}\bar{\tau}(w)$ from Section 2.4 allows us to perform precomputations very quickly or to get rid of them completely. In the next two subsections we shall consider these facts in detail. In Section 3.2.1 we explain how to use digit sets which are not w-NADS when they contain a subset that is a k-NADS for smaller k.

3.1. Using the Short-NAF Digit Set

Let us consider here the digit set $\mathsf{SNR}(w)$ defined in (6). With respect to Solinas' set it has the advantage of being syntactically defined. If a computer has to work with different curves, different scalar sizes and thus with different optimal choices for the window size, the representatives in Solinas' set must be recomputed – or they must be retrieved from a set of tables. In some cases, the time to compute representatives of minimal norm may have to be subsumed in the total scalar multiplication time. This is not the case with our set. This flexibility is also particularly important for computer algebra systems.

3.1.1. Computation of the expansion. When using Solinas' digit set $\mathsf{MNR}(w)$, one technical problem is the computation of the w-NAF expansion. Any element of $\mathbb{Z}[\tau]$ is either divisible by τ – in which case the least significant digit of its expansion is 0 – or congruent to exactly one element of the digit set – which is then chosen as the least significant element of the expansion. The full expansion is constructed recursively. This implies that we must be able to associate an element of $\mathbb{Z}[\tau]$ to its residue class modulo the chosen power of τ, and this requires some arithmetic operations involving multiplication of the (truncated) inputs by elements computed from Lucas sequences (cf. [37, § 7.3]). In particular, these elements are dependent on the chosen window width and must be precomputed and stored.

With the short-NAF digit set $\mathsf{SNR}(w)$, we can get away with these computations. The scalar is first recoded as a τ-NAF, and the elements of $\mathsf{SNR}(w)$ are associated to NAFs of length at most w with non-vanishing least significant digit, and thus to certain *odd integers* in the interval $[-a_w, a_w]$ where a_w is the largest odd integer that can be written as a binary NAF of length w. This integer is $(1010\ldots101)$ for odd w, and $(1010\ldots1001)$ for even w. Explicitly, $a_w = \frac{2^{w+1}-2(-1)^w}{3} - 1$. These integers can be used to index the elements in the precomputation table. We need only to precompute the multiples of the base point by "positive" short NAFs (that is with most significant digit equal to 1) – and the corresponding integers are the odd integers in the interval $[0, a_{w-1}]$ together with the integers $\equiv 1 \pmod 4$ in $[a_{w-1}+2, a_w]$. The indices in the table are then obtained by easy compression. The precomputed elements for the scalar multiplication loop can thus be retrieved upon direct reading the τ-NAF, of which we need only to compute the least w significant places. If the least and the w-th least significant digits of this segment of the τ-NAF are both non-zero and have different signs, a carry is generated: Thus, the computation of the τ-NAF should be interleaved with the parsing for short NAFs. This can be achieved by simple modifications to the algorithms for the τ-NAF in [36, 37].

3.1.2. Precomputations. In a scalar multiplication algorithm with variable base point P and designed around a scalar expansion with a large symmetric digit set $\mathcal{D}$, it is necessary to compute $d \cdot P$ for all the positive digits $d \in \mathcal{D}^+$.

For $\mathsf{MNR}(w)$ the positive part can be defined as the set of minimal norm representatives of the residue classes $a + \tau^w\mathbb{Z}[\tau]$ with a odd, positive, not larger than 2^{w-2}. For $\mathsf{SNR}(w)$ we can define $\mathsf{SNR}(w)^+$ to be the set of the short-NAF representatives whose most significant digit is 1.

One problem that arises with Solinas' digit set is that in principle it has an irregular structure and it is not easy to optimize the computation of the $d \cdot P$ for all the positive digits $d \in \mathsf{MNR}(w)^+$. This is in fact usually done ad hoc for each w.

For $w = 5$, $a = 1$ Solinas lists the following elements of $\mathsf{MNR}(5)^+$, where α_i belongs to the class $i + \tau^5\mathbb{Z}[\tau]$:

$$\begin{aligned}
\alpha_1 &= 1 & \alpha_3 &= \tau^2 - 1\\
\alpha_5 &= \tau^2 + 1 & \alpha_7 &= -\tau^3 - 1\\
\alpha_9 &= -\tau^5 - \tau^3 + 1 = -\tau^3\alpha_5 + 1 & \alpha_{11} &= -\tau^4 - \tau^2 - 1 = -\tau^2\alpha_5 - 1\\
\alpha_{13} &= -\tau^4 - \tau^2 + 1 = -\tau^2\alpha_5 + 1 & \alpha_{15} &= \tau^4 - 1
\end{aligned}$$

The precomputations then are performed as follows $P_3 = \tau^2P - P$, $P_5 = \tau^2P + P$, $P_7 = -\tau^3P - P$, $P_9 = -\tau^3P + P$, $P_{11} = -\tau P_5 + P$, $P_{13} = -\tau^2P_5 + P$, and $P_{15} = \tau^4P - P$. We observe that in order to compute them efficiently we need to represent each element in a compact way in terms of previously computed elements – this after having computed each digit by explicit modular reduction by a power of τ in the ring $\mathbb{Z}[\tau]$.

The syntactically defined digit set allows us to solve this problem very easily.

For fixed w, let $\mathsf{S}[i]$ be the subset of $\mathsf{SNR}(w)^+$ of elements whose τ-NAFs have weight i, and let $\mathsf{S}[0] = \{0\}$. The sets $\mathsf{S}[i]$ with $i \le \lfloor \frac{w}{2} \rfloor$ define a partition of $\mathsf{SNR}(w)^+$, and $\mathsf{S}[0]$ together with $\mathsf{S}[i]$ and $-\mathsf{S}[i]$ with $i \le \lfloor \frac{w}{2} \rfloor$ define a partition of $\mathsf{SNR}(w)$.

Furthermore, each element d of $\mathsf{S}[i+1]$ can be obtained as $\tau^a \cdot d' + u$ for some integer $a \ge 2$, $d' \in \mathsf{S}[i]$, and $u = \pm 1$.

Algorithm 2. Smart precomputations for a $\mathsf{SNR}(w)$-based scalar multiplication

```
INPUT: $w$ and a base point $P$
OUTPUT: The set $\{d \cdot P : d \in \mathsf{SNR}(w)^+\}$

precompute $P$ and $-P$ (nothing to do)
for $i = 2$ to $\lfloor w/2 \rfloor$ do
    Compute $d \cdot P$ for all $d \in \mathsf{S}[i]$ from the (already computed) set $\mathsf{S}[i-1] \cdot P$ as follows:
        let $d = \tau^a \cdot d' + u$ with $d' \in \mathsf{S}[i-1]$, $a$ integer with $a \geq 2$, and $u = \pm 1$
        compute $d \cdot P$ as $\tau^a \cdot (d' \cdot P) + u \cdot P$ by means of Frobenius operations and one addition
return $(\{d \cdot P : d \in \mathsf{SNR}(w)^+\})$
```

Algorithm 2 does all required precomputations according to the weight of the τ-NAFs of the digits, in order of ascending weight.

This algorithm has already the advantage of being streamlined: as the digits are *defined* to have a certain τ-NAF, their weight is known by construction. Given d, it is trivial to determine a, the *prefix* d' and u, as they are given by the representation of d.

3.1.2.1. Precomputations using affine coordinates. This algorithm also leads to further optimizations, in the case we are using affine coordinates. We denote by $\mathcal{A}$ the system of affine coordinates.

All the computations of $d \cdot P$ as $\tau^a \cdot (d' \cdot P) + u \cdot P$ for fixed i are clearly independent from each other. Therefore we can first compute and store all the elements $\tau^a \cdot (d' \cdot P)$ and then simultaneously add all the $\pm P$. Note also that, for all d of length strictly smaller than w, to each $\tau^a \cdot (d' \cdot P)$ we have to both add and subtract P. Since P and $-P$ have the same (affine) X-coordinate, in the addition formulae we have to invert the same element to compute $A + P$ and $A - P$ for an arbitrary point A. We can perform all inversions (again, for fixed i) simultaneously, using a well known trick attributed to Peter L. Montgomery: n field inversions can be computed via one field inversion and $3(n-1)$ field multiplications – in other words each saved inversion is replaced by 3 multiplications. As soon as one inversion costs more than three multiplications, this approach is faster.

We thus have to perform $2^{w-2} - 1$ point additions in $\lfloor \frac{w}{2} \rfloor$ blocks (hence with $\lfloor \frac{w}{2} \rfloor$ inversions). Each point addition costs 1 field inversion, 1 field squaring, and 2 field multiplication – but, the number of inversions to merge must be decreased by the amount of digits different from ± 1 and of length strictly smaller than w in the digit set, divided by two, because these digits all come in pairs whose expansions differ only in the least significant digit. There are $\frac{2^{w-1} + (-1)^w}{3} - 1$ NAFs with positive leading digit, non vanishing least significant digit, and of length greater than 1 and at most $w - 1$. Hence, we save $\frac{2^{w-1} + (-1)^w}{6} - \frac{1}{2}$ inversions.

The $(2^{w-2} - 1) - \left(\frac{2^{w-1} + (-1)^w}{6} - \frac{1}{2}\right) = \frac{2^{w-1}}{3} - \frac{(-1)^w}{6} - \frac{1}{2}$ inversions come in $\lfloor \frac{w}{2} \rfloor$ groups, and are therefore in fact computed by $\lfloor \frac{w}{2} \rfloor$ inversions and $2^{w-1} - \frac{(-1)^w}{2} - \frac{3}{2} - 3 \lfloor \frac{w}{2} \rfloor$ multiplications.

Also the number of Frobenius operations required to compute the pairs of digits is halved, being the prefix the same. More precisely, we have:

Lemma 3.1. *The minimal number of Frobenius operations Φ_w required to do all the precomputations with window width w is given by*

$$\Phi_w = 2^{w-2} - \frac{1}{2} - \frac{(-1)^w}{2}$$

for $w \geq 2$.

Proof. We calculate the number of Frobenius operations needed by Algorithm 2. Since we are computing $d = \tau^a(d' \cdot P) + 1 \cdot P$ and $d = \tau^a(d' \cdot P) - 1 \cdot P$ at the same time, all NAFs with least significant digit -1 do not necessitate a Frobenius operation. If $a > 2$, then $\tau^{a-1}(d' \cdot P)$ has already been computed in the previous step, so only one Frobenius operation is needed. This corresponds to NAFs described by the regular expression

$$0^*1(0 + 0\bar{1} + 01)^*001. \tag{15}$$

On the other hand, if $a = 2$, we need two Frobenius operations. This corresponds to NAFs described the regular expression

$$0^*1(0 + 0\bar{1} + 01)^*(01 + 0\bar{1})01 \tag{16}$$

or

$$0^*101. \tag{17}$$

We denote the number of Frobenius operations for computing $\{d \cdot P : d \in \mathsf{SNR}(w)^+\}$ by Φ_w and consider the generating function $G(z) = \sum_{w \geq 0} \Phi_w z^w$. The three regular expressions (15), (16), and (17) correspond to the generating functions

$$\begin{aligned} G_1(z) &= \frac{1}{1-z} \cdot z \cdot \frac{1}{1-(z+2z^2)} \cdot z^3 \ , \\ G_2(z) &= \frac{1}{1-z} \cdot z \cdot \frac{1}{1-(z+2z^2)} \cdot 2z^2 \cdot z^2 \ , \\ G_3(z) &= \frac{1}{1-z} \cdot z^3 \ , \end{aligned}$$

respectively. Thus $G(z)$ is given by the appropriate linear combination of $G_1(z)$, $G_2(z)$, and $G_3(z)$ as

$$\begin{aligned} G(z) &= 1 \cdot G_1(z) + 2 \cdot G_2(z) + 2 \cdot G_3(z) \\ &= \frac{(2-z)z^3}{(1-z)(1+z)(1-2z)} = \frac{3}{4} - \frac{z}{2} - \frac{1}{2(1-z)} - \frac{1}{2(1+z)} + \frac{1}{4(1-2z)} \ . \end{aligned}$$

Extracting the coefficient of z^w yields

$$\Phi_w = 2^{w-2} - \frac{1+(-1)^w}{2} + \frac{3}{4}\,[w=0] - \frac{1}{2}\,[w=1] \ ,$$

where Iverson's notation [ ] for conditional expressions is used. □

Finally, we can now quantify the total cost of the precomputation:

Proposition 3.2. *Computing $\{d \cdot P : d \in \mathsf{SNR}(w)\}$ using affine coordinates takes*

$$\left(2^w - 3\left\lfloor\frac{w}{2}\right\rfloor - \frac{7}{2} - \frac{(-1)^w}{2}\right)\mathtt{M} + \left(\frac{3}{4}2^w - 2 - (-1)^w\right)\mathtt{S} + \left\lfloor\frac{w}{2}\right\rfloor\mathtt{I}$$

where M, S *and* I *denote field multiplication, squaring, and inversion, respectively.*

3.1.2.2. Precomputations using López-Dahab coordinates. We now consider the cost of the precomputation using López-Dahab coordinates [25], denoted by $\mathcal{LD}$. These are set of coordinates for elliptic curves in which the curve is given by the equation

$$Y^2 + XYZ = X^3Z + a_2X^2Z^2 + a_6Z^4 \ .$$

The triple $(X_1 : Y_1 : Z_1)$ represents the affine point $(X_1/Z_1, Y_1/Z_1^2)$ when $Z_1 \neq 0$ and the neutral point is $(1 : 0 : 0)$. The opposite of $(X_1 : Y_1 : Z_1)$ is $(X_1 : X_1Z_1 + Y_1 : Z_1)$. These coordinates provide the fastest mixed-coordinate addition formula on binary elliptic curves [1], taking a point in $\mathcal{LD}$ and a point in $\mathcal{A}$ and returning theit sum in $\mathcal{LD}$.

The idea it to keep the base point in $\mathcal{A}$, perform all the additions in mixed coordinates and the Frobenius operations on the points in $\mathcal{LD}$ (each Frobenius operation costs now 3 squarings), and finally convert all the precomputations to $\mathcal{A}$ at once using Montgomery's trick – in order to speed up the additions in the main loop by using again mixed coordinate systems. We need to invert all the Z-coordinates, and for each point $(X : Y : Z)$ there are two additional multiplications, one by Z^{-1} and one by $(Z^{-1})^2$.

Furthermore, also in this case we can save operations exploiting the fact that we are often adding and subtracting the same point.

According to [1] (see also [4]), to add a $\mathcal{LD}$-point $(X_1 : Y_1 : Z_1)$ to a $\mathcal{A}$-point (X_2, Y_2), we perform the following operations:

$$\begin{array}{lll} A = Y_1 + Y_2 Z_1^2 \ , & B = X_1 + X_2 Z_1 \ , & C = B Z_1 \ , \\ Z_3 = C^2 \ , & D = X_2 Z_3 \ , & \\ \multicolumn{3}{l}{X_3 = A^2 + \ C(A + B^2 + a_2 C) \ ,} \\ \multicolumn{3}{l}{Y_3 = (D + X_3)(AC + Z_3) + (Y_2 + X_2) Z_3^2 \ .} \end{array} \tag{18}$$

If $a_2 \in \{0, 1\}$ the multiplication by a_2, denoted by $\mathtt{M}_2$, can be saved. The cost of this mixed addition is $8\,\mathtt{M} + \mathtt{M}_2 + 5\,\mathtt{S}$.

Suppose now that we want to add and subtract the affine point $P_2 = (X_2, Y_2)$ to the point $P_1 = (X_1 : Y_1 : Z_1)$ in $\mathcal{LD}$. Note that $-(X_2, Y_2) = (X_2, X_2 + Y_2)$.

Let $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2, Y_2)$ and $(X_3' : Y_3' : Z_3') = (X_1 : Y_1 : Z_1) + (X_2, X_2 + Y_2)$. To compute both points we need perform the following sequence of operations

	Steps to compute $P_1 + P_2$	Steps to compute $P_1 - P_2$	Operations
1	$A = Y_1 + Y_2 Z_1^2$	$A' = Y_1 + (X_2 + Y_2) Z_1^2$	$2\,\mathtt{M} + 1\,\mathtt{S}$
2	$B = X_1 + X_2 Z_1$	$B' = X_1 + X_2 Z_1 = B$	$1\,\mathtt{M}$
3	$C = B Z_1$	$C' = B' Z_1 = B Z_1 = C$	$1\,\mathtt{M}$
4	$Z_3 = C^2$	$Z_3' = C'^2 = C^2 = Z_3$	$1\,\mathtt{S}$
5	$D = X_2 Z_3$	$D' = X_2 Z_3' = X_2 Z_3 = D$	$1\,\mathtt{M}$
6	$X_3 = A^2 + C(A + B^2 + a_2 C)$ $= A^2 + AC + C(B^2 + a_2 C)$	$X_3' = A'^2 + C'(A' + B'^2 + a_2 C')$ $= A'^2 + A'C + C(B^2 + a_2 C)$	$3\,\mathtt{M} + 1\,\mathtt{M}_2 + 3\,\mathtt{S}$
7	$Y_3 = (D + X_3)(AC + Z_3) + (Y_2 + X_2) Z_3^2$	$Y_3' = (D' + X_3')(A'C' + Z_3') + Y_2 Z_3'^2$ $= (D + X_3')(A'C + Z_3) + Y_2 Z_3^2$	$4\,\mathtt{M} + 1\,\mathtt{S}$
		Total operation count:	$12\,\mathtt{M} + 1\,\mathtt{M}_2 + 6\,\mathtt{S}$

We observe that the intermediate results B, C, D as well as the Z-coordinates of the results are the same, and therefore are computed once. It is possible to compute, by associativity of the multiplication, X_3 and X_3' with only 2 multiplications instead of 3, but then AC and $A'C$ must be computed explicitly again for Y_3 and Y_3', bringing the multiplication count in Step 7 to 6. As before, if $a_2 = 0$ or 1 the multiplication by a_2 need not be counted.

In general we perform $12\,\mathtt{M} + 1\,\mathtt{M}_2 + 6\,\mathtt{S}$ instead of $16\,\mathtt{M} + 2\,\mathtt{M}_2 + 10\,\mathtt{S}$, and in our case the cost is $12\,\mathtt{M} + 6\,\mathtt{S}$ instead of $16\,\mathtt{M} + 10\,\mathtt{S}$, a saving of $4\,\mathtt{M} + 4\,\mathtt{S}$.

The fact that $Z_3 = Z_3'$ further reduced the number of Z-coordinates to invert (moreover, these inverses are then squared) by the number of pairs of digits as above.

As with the precomputations in $\mathcal{A}$, we need to compute $2^{w-2} - 1$ points, all by Frobenius operations followed by addition or subtraction of P.

We have $\frac{2^{w-1} + (-1)^w}{6} - \frac{1}{2}$ pairs of digits that differ only in the least significant digit, other than the pair $\{1, -1\}$. For each of these pairs we save $4\,\mathtt{M} + 4\,\mathtt{S}$ because of the additions, then we need to invert their Z-coordinates only once, and we therefore save one further squaring per pair during the conversion from $\mathcal{LD}$ to $\mathcal{A}$.

We thus obtain the following result:

Proposition 3.3. *The precomputation of* $\{d \cdot P : d \in \mathsf{SNR}(w)\}$ *using López-Dahab coordinates and converting these points at the end in affine coordinates can be performed in*

$$\left(\frac{8}{3} 2^w - \frac{25}{2} - \frac{7}{6} (-1)^w\right) \mathtt{M} + \left(\frac{11}{6} 2^w - 5 - \frac{7}{3} (-1)^w\right) \mathtt{S} + \mathtt{I} \ .$$

Remark 3.4. Further optimizations are possible when implementing formula (18) using a technique introduced in [10] and already applied to the optimization of explicit arithmetic of low genus Jacobians [11]. This technique consists in locally reusing the precomputations associated to one of the two multiplicands in a binary polynomial product. For example, the two multiplications by C to obtain X_3 and Y_3 can be fused, and with a small change of the code also the precomputations associated to Z_1 to obtain B can be reused for C.

In [10] it is reported that 2, 3, 4 multiplications with a common multiplicand can be performed at roughly the cost of 1.75, 2.35, 3 single multiplications respectively. The resulting savings for the mixed $\mathcal{LD}/\mathcal{A}$ addition formula are thus about a half the time of multiplication per group operation.

In the operations sequence to compute $P_1 \pm P_2$ more savings are possible. For instance, the products $Y_2 Z_1^2$ and $(X_2 + Y_2)Z_1^2$ can be fused, as well as the three multiplications by C in the expressions for X_3, Y_3, X_3' and Y_3' and the two multiplications by Z_3^2 in the last step. We can expect a saving of about 1.15 field multiplications per pair of group operations.

3.1.2.3. Comparing the two strategies. It is difficult to determine which approach (the one whose complexity is given in Proposition 3.2 or the one analyzed in Proposition 3.3) is faster in all situations, but we can try to estimate the difference by assuming the cost of a squaring to be negligible and $\mathtt{M}_2 = 0$, replacing $\lfloor \frac{w}{2} \rfloor$ with $w/2 - (1-(-1)^w)/4$. Since there is nothing to do if $w = 2$, we shall assume $w \geq 3$.

Subtracting the cost of the precomputation strategy from § 3.1.2.2 from the cost of the approach described in § 3.1.2.1, we obtain

$$\left(\frac{w}{2} - \frac{5-(-1)^w}{4}\right)\mathtt{I} - \left(\frac{5}{3}\,2^w + \frac{3}{2}\,w - \frac{39}{4} - \frac{(-1)^w}{12}\right)\mathtt{M}\ . \tag{19}$$

For $w = 3$, expression (19) is always negative, which means that the approach of § 3.1.2.1 is faster. Otherwise (19) vanishes for

$$\mathtt{I} = \frac{(20 \cdot 2^w + 18\,w - 117 + (-1)^w)}{3(2\,w - 5 + (-1)^w)}\mathtt{M}\ .$$

If an inversion costs more than this, then the second approach using only López-Dahab coordinates is faster – if the cost of an inversion is below this threshold, then the first approach is faster.

For $w = 4, 5, 6, 7, 8$ the threshold is 23, 51, 53, 107 and 143 $\mathtt{M}$ respectively. These will increase a bit if we do not ignore the cost of squaring. For example, if we assume $\mathtt{S} = \frac{1}{10}\,\mathtt{M}$, these thresholds become 24.3, 54.3, 56.25, 113.85 and 152.1 $\mathtt{M}$ respectively.

Remark 3.5. For $i \geq 2$, a digit $d \in \mathsf{S}[i]$ can also be written as $d = d' \cdot \tau^a + d''$ with $d' \in \mathsf{S}\big[\lfloor \frac{i}{2} \rfloor\big]$, $d'' \in \mathsf{S}\big[\lceil \frac{i}{2} \rceil\big]$, and an integer $a \geq 2$. We can exploit this fact to reduce the number of inversions in the first precomputation approach to $O(\log_2 w)$, and we proceed similarly to the above, the details being more complicated. This is bound to give a perceptible performance advantage only for large w, and being $w = O(\log_2 n)$ (for a more precise statement, see § 3.2.2), this means that the extension degree is large enough to guarantee that better performance will be obtained by using only López-Dahab coordinates.

3.2. τ-adic Scalar Multiplication with Repeated Halvings

Let $w \geq 2$ be an integer and $\mathsf{P}\bar{\tau}(w)$ the digit set defined in Section 2.4. Let P be a point on an elliptic curve and $Q_j := \tau^j(2^{-j}P)$ for $0 \leq j < 2^{w-2}$ and $R := Q_{2^{w-2}-1}$. To compute zP, we have to compute yR for $y := \bar{\tau}^{2^{w-2}-1}z$. Computing a $\mathcal{D}$-w-NAF of y, this can be done by using the points Q_j, $0 \leq j < 2^{w-2}$ as precomputations.

Now, a point halving on an elliptic curve is much faster than a point doubling, and a point addition is not faster than a doubling – with affine coordinates a doubling and an addition have similar timings, and with other coordinate systems an addition is much slower than a doubling. Now, with, say, Solinas' set or the short τ-NAF representatives the precomputations always involve one group addition per digit set element (even though something can be saved: see § 3.1.2)

With the digit set $\mathsf{P}\bar{\tau}(w)$ from Section 2.4 we require a halving per digit set element. Hence, our approach with the points Q_j and halvings is already faster than traditional ones.

But we can do even better, especially if normal bases are used to represent the field $\mathbb{F}_{2^n}$. Algorithm 3 computes zP using an expansion $y = \sum_{i=0}^{\ell} y_i \tau^i$ of the integer $y := \bar{\tau}^{2^{w-2}-1}z$ where the digits y_i belong to the digit set $\mathsf{P}\bar{\tau}(w)$.

To explain how it works we introduce some notation. Write $y_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$ and $0 \leq k < 2^{w-2}$. We also define

$$y^{(k)} = \sum_{i\,:\,0 \leq i \leq \ell,\ y_i = \pm\bar{\tau}^k} \varepsilon_i \tau^i\ .$$

Algorithm 3. τ-adic Scalar Multiplication on Koblitz curves with Repeated Halvings, basic variant

INPUT: A Koblitz curve E_a, a point P of odd order on it, and a scalar z.
OUTPUT: zP

```
1.  $y \leftarrow \bar{\tau}^{2^{w-2}-1} z$
    Write $y = \sum_{i=0}^{\ell} y_i \tau^i$  where  $y_i \in \mathsf{P}\bar{\tau}(w)$
    Write $y_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2.  $\ell_k \leftarrow \max\left(\{-1\} \cup \{i : y_i = \pm\bar{\tau}^k \text{ for some } k\}\right)$
3.  $Q \leftarrow 0$
4.  for  $k = 0$  to  $2^{w-2} - 1$  do
5.      if  $k > 0$  then  $Q \leftarrow \tau^{n-\ell_k} Q$, $Q \leftarrow \frac{1}{2} Q$
6.      for  $i = \ell_k$  to  $0$  do
7.          $Q \leftarrow \tau Q$
8.          if  $y_i = \pm\bar{\tau}^k$  then  $Q \leftarrow Q + \varepsilon_i P$
9.      $\left[Q = \sum_{j=0}^{k} \left(\frac{\tau}{2}\right)^{k-j} y^{(j)} P\right]$
10. return  $Q$
```

Now $y = \sum_{k=0}^{2^{w-2}-1} y^{(k)} \bar{\tau}^k$ and therefore

$$\begin{aligned} zP = \bar{\tau}^{-(2^{w-2}-1)} yP &= \left(\sum_{k=0}^{2^{w-2}-1} y^{(k)} \bar{\tau}^k \right) \bar{\tau}^{-(2^{w-2}-1)} P \\ &= \sum_{k=0}^{2^{w-2}-1} y^{(k)} \bar{\tau}^{k-(2^{w-2}-1)} P = \sum_{k=0}^{2^{w-2}-1} \left(\frac{\tau}{2}\right)^{2^{w-2}-1-k} (y^{(k)}) P \ . \end{aligned}$$

The last expression is evaluated by a Horner scheme in $\frac{\tau}{2}$, i.e., by repeated applications of τ and a point halving, interleaved with additions of $y^{(0)}P$, $y^{(1)}P$, etc. The elements $y^{(k)}P$ are computed by a τ-and-add loop as usual. To save a memory register, instead of computing $y^{(k)}P$ and then adding it to a partial evaluation of the Horner scheme, we apply τ to the negative of the length of $y^{(k)}$ (which is $1 + \ell_k$) to the intermediate result X and perform the τ-and-add loop to evaluate $y^{(k)}P$ starting with this X instead of a "clean" zero. In Step 5 there is an optimization already present in [3]: n is added to the exponent (since $n \approx \ell_k$ and τ^n acts like the identity on the curve; recall that we are working in $\mathbb{F}_{2^n}$) and the operation is also partially fused to the subsequent $\frac{\tau}{2}$.

Apart from the input, we only need to store the additional variable X and the recoding of the scalar. The multiplication of z by $\bar{\tau}^{2^{w-2}-1}$ is an easy operation, and the negative powers of τ can be easily eliminated by multiplying by a suitable power of τ^n, which operates trivially on the points of the curve. Reduction of this scalar by $(\tau^n - 1)/(\tau - 1)$ following [36, 37] is also necessary.

An issue with Algorithm 3 is that the number of Frobenius operations may increase exponentially with w, since the internal loop is repeated up to 2^{w-2} times. This is not a problem if a normal basis is used to represent the field, but may induce a performance penalty with a polynomial basis. A similar problem was faced by the authors of [32], and they solved it adapting an idea from [33]. The idea consists in keeping a copy R of the point P in normal basis representation. Instead of computing $y^{(k)}P$ by a Horner scheme in τ, the summands $\varepsilon_i \tau^i P$ are just added together. The power of the Frobenius is applied to R *before* converting the result back to a polynomial basis representation and accumulating it. According to [15] a basis conversion takes about the same time as one polynomial basis multiplication, and the two conversion routines require each a matrix that occupies $O(n^2)$ bits of memory. However, in our implementation we find the cost of basis conversion to be between 1.5 and 3 field multiplications, depending on the field size – this is due to the fact that the field multiplication routines which we have employed are particularly fast [10]. We shall then use these higher ratios when estimating the computational cost of our methods with respect to other techniques (§ 3.3).

Algorithm 4 is our realisation of this approach. It is suited in the context where a polynomial basis is used for a field and the cost of an inversion is not prohibitive. The routines normal_basis

Algorithm 4. Low-memory τ-adic Scalar Multiplication on Koblitz Curves with Repeated Halvings, for Fast Inversion

```
INPUT: $P \in E(\mathbb{F}_{2^n})$, scalar $z$
OUTPUT: $zP$
1.  $y \leftarrow \bar{\tau}^{2^{w-2}-1} z$
    Write $y = \sum_{i=0}^{\ell} y_i \tau^i$  where  $y_i \in \mathsf{P}\bar{\tau}(w)$
    Write $y_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2.  $R \leftarrow \mathrm{normal\_basis}(P)$
3.  $Q \leftarrow 0$
4.  for  $k = 0$  to  $2^{w-2} - 1$
5.      if  $k > 0$  then  $Q \leftarrow \tau Q$,  $Q \leftarrow \frac{1}{2} Q$
6.      for  $i = 0$  to  $\ell$
7.          if  $y_i = \pm\bar{\tau}^k$  then  $Q \leftarrow Q + \varepsilon_i \mathrm{polynomial\_basis}(\tau^i R)$
8.      $\left[Q = \sum_{j=0}^{k} \left(\frac{\tau}{2}\right)^{k-j} y^{(j)} \cdot P\right]$
9.  return  $Q$
```

Algorithm 5. Low-memory τ-adic Scalar Multiplication on Koblitz Curves with Repeated Doublings, for Slow Inversion

```
INPUT: $P \in E(\mathbb{F}_{2^n})$, scalar $z$
OUTPUT: $zP$
1.  Write $z = \sum_{i=0}^{\ell} z_i \tau^i$  where  $z_i \in \mathsf{P}\bar{\tau}(w)$
    Write $z_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2.  $R \leftarrow \mathrm{normal\_basis}(P)$                                   [Keep in affine coordinates]
3.  $Q \leftarrow 0$                                              [$Q$ is in López-Dahab coodinates]
4.  for  $k = 2^{w-2} - 1$  to  $0$
5.      if  $k > 0$  then  $Q \leftarrow \tau^{-1} Q$,  $Q \leftarrow 2Q$              [$\tau^{-1}$ is three square roots]
6.      for  $i = 0$  to  $\ell$
7.          if  $z_i = \pm\bar{\tau}^k$  then  $Q \leftarrow Q + \varepsilon_i \mathrm{polynomial\_basis}(\tau^i R)$     [Mixed coord.]
8.      $\left[Q = \sum_{j=k}^{2^{w-2}-1} \bar{\tau}^{j-k} y^{(j)} \cdot P\right]$
9.  return  $Q$                                               [Convert to affine coordinates]
```

and polynomial_basis perform the conversion of coordinates of the points between polynomial and normal bases.

Algorithm 5 is the version for fields with a slow inversion (such as large fields). It uses inversion-free coordinate systems and, since no halving formula is known for such coordinates, a doubling is used instead of a halving. This is not a problem, since using Projective or López-Dahab coordinates (see [4, Section 15.1]) a doubling followed by an application of τ^{-1} (which amounts to three square root extractions) is about twice as fast as a mixed-coordinate addition preceded by a basis conversion, hence the situation is as advantageous as the previous one. This also dispenses us with the need of using an auxiliary scalar y.

Although the digit set $\mathsf{P}\bar{\tau}(w)$ is not a w-NADS for all w, in the next subsection we show how to save the situation.

3.2.1. Stepping Down Window Size. Suppose we have a digit set $\mathcal{D}_w$, parametrized by an integer w, which is not a w-NADS for some values of w – but it is a NADS for some smaller values of w, and $\mathcal{D}_{v-1} \subset \mathcal{D}_v$ holds for all v. Then, Algorithm 1 may enter a loop for a few inputs; this can be caused by the appearance of "large" digits towards the end of the main loop of the recoding algorithm, so that the norm of the variable u gets too small in comparison to the chosen digit. In other words $|u| \leq \left|\frac{u-z_j}{\tau^w}\right| \leq \frac{|u|+|z_j|}{2^{w/2}}$. But, for other inputs the algorithm delivers the expected low density. How can we save it? One possible answer is to lower the value of w for the rest of the computation, so that the corresponding digit set is a NADS. We call this operation *stepping*

Algorithm 6. Windowed Integer Recoding With Termination Guarantee

INPUT: An element z from $\mathbb{Z}[\tau]$, a natural number $w \geq 1$ and a set of reduced residue systems $\mathcal{D}'_k \subset \mathcal{D}'_{k+1} \subset \ldots \subset \mathcal{D}'_w$ modulo τ^k, τ^{k+1}, …, τ^w respectively ($1 \leq k < w$), where $\mathcal{D}'_k \cup \{0\}$ is a k-NADS.
OUTPUT: A representation $z = \sum_{j=0}^{\ell-1} z_j \tau^j$ of length ℓ.

```
1.  $j \leftarrow 0$, $u \leftarrow z$, $v \leftarrow w$
2.  while $u \neq 0$ do
3.      if $\tau \mid u$ then
4.          $z_j \leftarrow 0$
5.      else
6.          Let $z_j \in \mathcal{D}'_v$ s.t. $z_j \equiv u \pmod{\tau^v}$
7.          if ( $|z_j| \geq |u|(2^{v/2} - 1)$ and $v > k$ ) then decrease $v$ and retry:
8.              $v \leftarrow v - 1$, go to Step 6
9.      $u \leftarrow u - z_j$, $u \leftarrow u/\tau$, $j \leftarrow j + 1$
10. $\ell \leftarrow j$
11. return $(\{z_j\}_{j=0}^{\ell-1}, \ell)$
```

down. The resulting recoding may have a slightly higher weight, but the algorithm is guaranteed to terminate. One possible implementation is presented as Algorithm 6.

Solinas is able to prove termination of his τ-adic w-NAF because his digits are minimal representatives in their classes and have norm bounded by $\frac{4}{7}2^w$. The presence of digits of non-minimal norm is a necessary condition for non-termination, but this is not sufficient. In fact, we have seen w-NADS with digits of norm larger than 2^w: the digit set from Example 2.10 and $\mathsf{P}\overline{\tau}(w)$ from Section 2.4.

Remark 3.6. Note that the digit set from Example 2.10, $\mathsf{MNR}(w)$ and $\mathsf{P}\overline{\tau}(w)$ all have the property that each set is contained in the corresponding sets with larger w – hence Algorithm 6 can be used. However, stepping down is not required for some of these digit sets.

Remark 3.7. Checking an absolute value (or a norm) in Algorithm 6, Step 7 is expensive. Hence we need an alternative strategy. Let M_w be defined as M in Theorem 1 for the digit set we are considering, with parameter w. Consider an easy function that is bounded by the norm: for example, if $z = a + b\tau$, $\lambda(z) = \max\{\lceil |a + \frac{\mu}{2} b| \rceil^2, 2\lceil |\frac{\mu}{4} a + b| \rceil^2\}$. It is easy to check that $\lambda(z) \leq N(z)$ and that $\lambda(z) = 0$ iff $z = 0$. Therefore, if $\lceil \log_2(M_v) \rceil \geq \lfloor \log_2(\lambda(z)) \rfloor$ we step down to a new value of v with $\lceil \log_2(M_v) \rceil < \lfloor \log_2(\lambda(z)) \rfloor$. These checks are quickly computed only by using the bit lengths of a and b and performing additions, subtractions and bit shifts (but no expensive multiplication). The values $\lceil \log_2(M_v) \rceil$ are precomputed in an easy way.

Remark 3.8. In our experiments, the recodings done with the different digit sets have similar length, the average density is $1/(w + 1)$ (see also § 3.2.2), and stepping down only marginally increases the weight. Therefore the new digit sets bring their advantages with *de facto* no performance penalty.

3.2.2. Sublinearity. Algorithms 3, 4 and 5 all perform a scalar multiplication by alternating $2^{w-2} - 1$ "faster" operation blocks and (roughly) $n/(w + 1)$ "slower" operation blocks. In Algorithm 3 (with normal bases) these two block types are given by a halving and an addition respectively. In Algorithm 4, resp. 5 these two block types are given by a Frobenius operation plus a halving (resp. by an inverse Frobenius plus a doubling), and by a basis conversion followed by an addition (for both algorithms). In all cases we can see that computing the first block takes α times the time for computing the second block, where $\alpha \leq 1/2$.

We now determine asymptotically optimal values for w in these algorithms in terms of n, where n is assumed to be large. This will lead to large values w, such that $\mathsf{P}\overline{\tau}(w)$ is probably not a w-NADS. We will therefore have to use Algorithm 6 (or a variant of it). For the sake of simplicity, we do not decrease v step by step depending on the norm of $|z_j|$, but we use $v = w$ for $j < L$ and $v = 6$ for $j \geq L$, where the parameter L will be chosen below.

Let z be a random integer in $\mathbb{Z}[\tau]$ with $|z| \le |\tau|^n$. Here "random" means that for every positive integer m, every residue class modulo τ^m is equally likely. More precisely, we use the set of τ-adic integers equipped with its Haar measure as our probability space. Let $y = \sum_{j=0}^{L-1} z_j \tau^j$ where the z_j are calculated by Algorithm 6. Then $y \equiv z \pmod{\tau^L}$ and $|y| \le |\tau|^{2^{w-2}-1+L-1}(1-|\tau|^{-w})^{-1}$. Thus $|(z-y)/\tau^L| \le |\tau|^{n-L} + |\tau|^{2^{w-2}-2}(1-2^{-w/2})^{-1}$. The choice

$$L = n - 2^{w-2} + 2$$

implies that $|(z-y)/\tau^L| \le 3|\tau|^{n-L}$. The expected length of the $\mathcal{D}_6$-6-NAF of $(z-y)/\tau^L$ is therefore $n - L + O(1)$. Here, $\mathcal{D}_6 = \{0\} \cup \{\pm\bar{\tau}^k : 0 \le k < 16\}$. We conclude that the expected Hamming weight of the expansion constructed by Algorithm 6 is

$$\frac{L}{w+1} + \frac{n-L}{7} + O(1) \ .$$

Here, we use the well-known fact that a v-NAF of length m has expected Hamming weight $m/(v+1) + O(1)$.

The number of cheap operations (e.g., point halvings performed by, say, Algorithm 3) equals $2^{w-2} - 1$, the number of expensive operations (i.e., additions) is given by the Hamming weight of the expansion. With α defined as above, the overall costs of the curve operations (measured in additions) are given by

$$\alpha 2^{w-2} + \frac{L}{w+1} + \frac{n-L}{7} + O(1) = 2^{w-2}\left(\alpha + \frac{1}{7}\right) + \frac{n - 2^{w-2}}{w+1} + O(1) \ .$$

Balancing the two main terms gives

$$\hat{w} = \frac{1}{\log 2} W\left(\frac{7 \cdot 2^{\frac{21\alpha+10}{7\alpha+1}} \log 2}{7\alpha+1} n\right) - \frac{7\alpha+8}{7\alpha+1} \ ,$$

where $W(z)$ denotes the main branch of Lambert's W function, cf. Corless et al. [14]. Asymptotically, this is $\hat{w} = \log_2 n - \log_2 \log_2 n + 2 - \log_2\left(\alpha + \frac{1}{7}\right) + O\left(\frac{\log\log n}{\log n}\right)$. Thus we choose

$$w = \left\lfloor \log_2 n - \log_2 \log_2 n + 2 - \log_2\left(\alpha + \frac{1}{7}\right) \right\rfloor$$

and see that the expected number of curve additions asymptotically equals

$$\frac{n}{\log_2 n}\left(1 + c + O\left(\frac{\log\log n}{\log n}\right)\right) \tag{20}$$

with $\frac{1}{2} < c = 2^{-\{\log_2 n - \log_2 \log_2 n + 2 - \log_2(\alpha + \frac{1}{7})\}} \le 1$.

For Algorithms 4 and 5, the unit in the cost (20) is given by the cost of a group addition and a base conversion – the latter being comparable to a field multiplication. We thus have the following result:

Theorem 7. *Algorithms* 3, 4 *and* 5 *are sublinear scalar multiplication algorithms on Koblitz Curves with constant input-dependent memory consumption.*

Note that here *sublinear* refers to the number of group operations, and "constant memory consumption" refers to the number of registers required for temporary variables – each one taking of course $O(n)$ bits. Usual windowed methods with precomputations have, of course, similar time complexity but use storage for $2^{w-2} - 1$ points [36, 37] and thus $O(n2^w) = O(n^2/\log n)$ bits of memory. Algorithms 4 and 5 need $O(n^2)$ bits of field-dependent (but not point-dependent) data for base conversion (as in [33, 32]) that depends only on the field and thus can be stored statically (such as in ROM).

bits	Classic with $\mathsf{MNR}(w)$		Classic with $\mathsf{SNR}(w)$		Methods from [2]		Alg. 4	Alg. 5
	$\mathcal{A}$	$\mathcal{LD}$	$\mathcal{A}$	$\mathcal{LD}$	CMA	LMA		
163	390.77 (5)	392.93 (5)	355.98 (6)	384.07 (4)	305.75 (5)	300.35 (5)	382.27 (6)	390.32 (6)
233	590.93 (6)	513.36 (5)	535.71 (6)	496.25 (5)	427.69 (5)	407.71 (5)	562.88 (7)	493.91 (6)
283	950.84 (6)	594.41 (5)	862.43 (7)	577.91 (5)	605.06 (6)	568.96 (5)	884.20 (7)	598.66 (6)
409	2602.33 (8)	807.30 (5)	2377.83 (8)	791.21 (5)	1266.51 (6)	1144.68 (6)	2247.20 (9)	835.93 (7)
571	6809.33 (9)	1186.02 (6)	6244.88 (9)	1143.57 (6)	2746.00 (6)	2428.36 (6)	5817.70 (10)	1157.03 (7)

Table 3. Comparison of operation counts of different scalar multiplication algorithms for elliptic Koblitz Curves. The numbers between parentheses are the corresponding optimal window sizes w.

3.3. Cost Comparisons

It is interesting to determine the relative performance of various Koblitz curve scalar multiplication algorithms in different scenarios.

It is to be expected that, for the same values of w, Algorithms 4 and 5 perform better than techniques storing precomputations. The precomputation stage in the latter case takes one addition and some Frobenius operations per precomputation. Using $\mathsf{P}\bar{\tau}(w)$ these additions can be replaced with cheaper operations (halvings or doublings depending on the coordinate system), whereas in Algorithms 4 and 5 the cost of the basis conversion associated to each addition in the main loop is relatively small. In all cases, the increase in recoding weight is marginal.

On the other hand, using $\mathsf{SNR}(w)$, in § 3.1.2 we show how to save time by merging field inversions or using faster addition formulas. Finally, in [2] two further algorithms derived from Yao's scalar multiplication technique are introduced, that are particularly attractive for small inputs. Table 3 updates and extends the table from [2]. The operation counts have been recalculated after the precomputation stage has been improved and taking into account our base conversion timings.

By $\mathcal{A}$ we denote the use of affine coordinates in the main loop of the scalar multiplication, by $\mathcal{LD}$ the use of López-Dahab coordinates throughout. In each case, the fastest precomputation method has been adopted (cf. [2]). The two columns under $\mathsf{MNR}(w)$ represent scalar multiplications using Solinas' digit set, under $\mathsf{SNR}(w)$ operation counts for scalar multiplication using the short-NAF representatives are given, with the precomputation strategies described in § 3.1.2. The next two columns report performance data for the methods from [2] and the last two columns refer to Algorithms 4 and 5.

The value of w, that is always chosen to be optimal with respect to performance, is found between parentheses. It is important to keep in mind that the methods described in the first 6 columns require storage for 2^{w-2} points (even somewhat more in the cases of the algorithms from [2]) with $w \approx \log_2 n - \log_2 \log_2 n$, hence their storage requirements grow essentially quadratically with n. On the other hand, Algorithms 4 and 5 require to store just three intermediate values, and thus their total storage requirements that depend on the input point grow linearly with the degree of the extension field. This and the fact that their performance is comparable to the other best methods, makes Algorithms 4 and 5 especially interesting for their asymptotic behaviour (for example in computer algebra systems).

The method in [9] is also sublinear, but its applicability still has to be assessed – the authors warn that the involved constants may be quite large. Another approach is presented in [5].

References

1. E. Al-Daoud, R. Mahmod, M. Rushdan, and A. Kilicman, *A new addition formula for elliptic curves over GF*(2^n), IEEE Trans. on Computers **51** (2002), no. 8, 972–975.

2. R. Avanzi, *Delaying and Merging Operations in Scalar Multiplication: Applications to Curve-Based Cryptosystems*, Selected Areas in Cryptography: 13th International Workshop, SAC 2006, Montreal, Quebec, Canada, August 17–18, 2006, Revised Selected Papers (Eli Biham and Amr M. Youssef, eds.), Lecture Notes in Comput. Sci., vol. 4356, Springer, Berlin, 2007, pp. 203–219.

3. R. Avanzi, M. Ciet, and F. Sica, *Faster scalar multiplication on Koblitz curves combining point halving with the Frobenius endomorphism*, Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004 (Feng Bao, Robert H. Deng, and Jianying Zhou, eds.), Lecture Notes in Comput. Sci., vol. 2947, Springer, 2004, pp. 28–40.
4. R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, and K. Nguyen, *Handbook of elliptic and hyperelliptic curve cryptography*, CRC Press Series on Discrete Mathematics and its Applications, vol. 34, Chapman & Hall/CRC, Boca Raton, FL, 2005.
5. R. Avanzi, V. Dimitrov, C. Doche, and F. Sica, *Extending scalar multiplication using double bases*, Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings (Xuejia Lai and Kefei Chen, eds.), Lecture Notes in Comput. Sci., vol. 4284, Springer, 2006, pp. 130–144.
6. R. Avanzi, C. Heuberger, and H. Prodinger, *Minimality of the Hamming weight of the τ-NAF for Koblitz curves and improved combination with point halving*, Selected Areas in Cryptography: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11–12, 2005, Revised Selected Papers (Preneel B. and Tavares St., eds.), Lecture Notes in Comput. Sci., vol. 3897, Springer, Berlin, 2006, pp. 332–344.
7. ———, *Scalar multiplication on Koblitz curves. Using the Frobenius endomorphism and its combination with point halving: Extensions and mathematical analysis*, Algorithmica **46** (2006), 249–270.
8. ———, *On Redundant τ-adic Expansions and Non-Adjacent Digit Sets.*, Selected Areas in Cryptography: 13th International Workshop, SAC 2006, Montreal, Quebec, Canada, August 17–18, 2006, Revised Selected Papers (Eli Biham and Amr M. Youssef, eds.), Lecture Notes in Comput. Sci., vol. 4356, Springer, Berlin, 2007, pp. 285–301.
9. R. Avanzi and F. Sica, *Scalar multiplication on Koblitz curves using double bases*, Progress in Cryptology - VIETCRYPT 2006, First International Conference on Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers (Phong Q. Nguyen, ed.), Lecture Notes in Comput. Sci., vol. 4341, Springer, 2006, pp. 131–146.
10. R. Avanzi and N. Thériault, *Effects of Optimizations for Software Implementations of Small Binary Field Arithmetic.*, WAIFI 2007: International Workshop on the Arithmetic of Finite Fields (Claude Carlet and Berk Sunar, eds.), Lecture Notes in Comput. Sci., vol. 4547, Springer, Berlin, 2007, pp. 69–84.
11. R. Avanzi, N. Thériault, and Z. Wang, *Rethinking Low Genus Hyperelliptic Jacobian Arithmetic over Binary Fields: Interplay of Field Arithmetic and Explicit Formulæ*, Preprint, 2007.
12. G. Avoine, J. Monnerat, and Th. Peyrin, *Advances in alternative non-adjacent form representations*, Progress in cryptology—INDOCRYPT 2004, Lecture Notes in Comput. Sci., vol. 3348, Springer, Berlin, 2004, pp. 260–274.
13. I. F. Blake, V. K. Murty, and G. Xu, *A note on window τ-NAF algorithm*, Inform. Process. Lett. **95** (2005), 496–502.
14. R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, *On the Lambert W function*, Adv. Comput. Math. **5** (1996), no. 4, 329–359.
15. J.-S. Coron, D. M'Raïhi, and C. Tymen, *Fast generation of pairs $(k,[k]P)$ for Koblitz elliptic curves*, Selected Areas in Cryptography, 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16-17, 2001, Revised Papers (Serge Vaudenay and Amr M. Youssef, eds.), Lecture Notes in Comput. Sci., vol. 2259, Springer, Berlin, 2001, pp. 151–164.
16. W. J. Gilbert, *Radix representations of quadratic fields*, J. Math. Anal. Appl. **83** (1981), no. 1, 264–274.
17. C. Heuberger, *Redundant τ-adic Expansions II: Non-Optimality and Chaotic Behaviour*, Tech. Report 2008-4, Graz University of Technology, 2008, available at: `http://www.math.tugraz.at/fosp/pdfs/tugraz_0093.pdf`.
18. C. Heuberger and H. Prodinger, *Analysis of alternative digit sets for nonadjacent representations*, Monatsh. Math. **147** (2006), 219–248.
19. IEEE Std 1363-2000, *IEEE standard specifications for public-key cryptography*, IEEE Computer Society, August 29 2000.
20. I. Kátai and B. Kovács, *Canonical Number Systems in Imaginary Quadratic Fields*, Acta Math. Hungar. **37** (1981), 159–164.
21. I. Kátai and J. Szabó, *Canonical Number Systems for Complex Integers*, Acta Sci. Math. (Szeged) **37** (1975), 255–260.

22. E. W. Knudsen, *Elliptic Scalar Multiplication Using Point Halving*, Advances in Cryptology - ASIACRYPT '99, International Conference on the Theory and Applications of Cryptology and Information Security, Singapore, November 14-18, 1999, Proceedings (Kwok-Yan Lam, Eiji Okamoto, and Chaoping Xing, eds.), Lecture Notes in Comput. Sci., vol. 1716, Springer, Berlin, 1999, pp. 135–149.

23. N. Koblitz, *Elliptic curve cryptosystems*, Math. Comp. **48** (1987), no. 177, 203–209.

24. N. Koblitz, *CM-curves with good cryptographic properties*, Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings (Joan Feigenbaum, ed.), Lecture Notes in Comput. Sci., vol. 576, Springer, Berlin, 1992, pp. 279–287.

25. J. López and R. Dahab, *Improved algorithms for elliptic curve arithmetic in GF*(2^n), Tech. Report IC-98-39, Relatório Técnico, October 1998.

26. D. W. Matula, *Basic digit sets for radix representation*, J. Assoc. Comput. Mach. **29** (1982), no. 4, 1131–1143.

27. W. Meier and O. Staffelbach, *Efficient multiplication on certain nonsupersingular elliptic curves*, Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings (Ernest F. Brickell, ed.), Lecture Notes in Comput. Sci., vol. 740, Springer, Berlin, 1993, pp. 333–344.

28. V. S. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings (Hugh C. Williams, ed.), Lecture Notes in Comput. Sci., vol. 218, Springer, Berlin, 1986, pp. 417–426.

29. J. A. Muir and D. R. Stinson, *Alternative digit sets for nonadjacent representations*, Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14-15, 2003, Revised Papers (Mitsuru Matsui and Robert J. Zuccherato, eds.), Lecture Notes in Comput. Sci., vol. 3006, Springer, Berlin, 2004, pp. 306–319.

30. ———, *Alternative digit sets for nonadjacent representations*, SIAM J. Discrete Math. **19** (2005), 165–191.

31. National Institute of Standards and Technology, *Digital signature standard*, FIPS Publication, vol. 186-2, February 2000.

32. K. Okeya, T. Takagi, and C. Vuillaume, *Short memory scalar multiplication on Koblitz curves*, Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings (J.R. Rao and B. Sunar, eds.), Lecture Notes in Comput. Sci., vol. 3659, Springer, Berlin, 2005, pp. 91–105.

33. D. J. Park, S. G. Sim, and Pil Joong Lee, *Fast scalar multiplication method using change-of-basis matrix to prevent power analysis attacks on koblitz curves*, Information Security Applications 4th International Workshop, WISA 2003, Jeju Island, Korea, August 25-27, 2003, Revised Papers (Kijoon Chae and Moti Yung, eds.), Lecture Notes in Comput. Sci., vol. 2908, Springer, 2004, pp. 474–488.

34. R. Schroeppel, *Elliptic curve point ambiguity resolution apparatus and method*, International Application Number PCT/US00/31014, filed 9 November 2000.

35. R. Schroeppel, *Point halving wins big*, Talk at the ECC 2001 Workshop, October 29–31, 2001, University of Waterloo, Ontario, Canada.

36. J. A. Solinas, *An improved algorithm for arithmetic on a family of elliptic curves*, Advances in Cryptology — CRYPTO '97. 17th annual international cryptology conference. Santa Barbara, CA, USA. August 17–21, 1997. Proceedings (B. S. Kaliski, jun., ed.), Lecture Notes in Comput. Sci., vol. 1294, Springer, Berlin, 1997, pp. 357–371.

37. ———, *Efficient arithmetic on Koblitz curves*, Des. Codes Cryptogr. **19** (2000), 195–249.

Roberto Maria Avanzi
Faculty of Mathematics and Horst Görtz Institute for IT Security
Ruhr-University Bochum
Germany
e-mail: `roberto.avanzi AT ruhr-uni-bochum.de`

Clemens Heuberger
Institut für Mathematik B
Technische Universität Graz
Austria
e-mail: clemens.heuberger AT tugraz.at

Helmut Prodinger
Department of Mathematics
University of Stellenbosch
South Africa
e-mail: hproding AT sun.ac.za