Balanced Non-Adjacent Forms

Marc Joye

Zama, Paris, France

Abstract. Integers can be decomposed in multiple ways. The choice of a recoding technique is generally dictated by performance considerations. The usual metric for optimizing the decomposition is the Hamming weight. In this work, we consider a different metric and propose new modified forms (i.e., integer representations using signed digits) that satisfy minimality requirements under the new metric. Specifically, we introduce what we call *balanced non-adjacent forms* and prove that they feature a minimal Euclidean weight. We also present efficient algorithms to produce these new minimal forms. We analyze their asymptotic and exact distributions. We extend the definition to modular integers and show similar optimality results. The balanced non-adjacent forms find natural applications in fully homomorphic encryption as they optimally reduce the noise variance in LWE-type ciphertexts.

Keywords: Integer recoding · Lattice cryptography · Fully homomorphic encryption · Gadget decomposition · Noise control · Implementation.

1 Introduction

Let B be an integer ≥ 2. Every positive integer $k < B^n$ can be expressed uniquely as $k = \sum_{i=0}^{n-1} k_i\, B^i$ with $0 \leq k_i < B$. Integer B is referred to as the radix and integers k_i are called the digits. If the digit set is extended to $\{-(B-1), \dots, B-1\}$, integer k can also be written under the form

$$k = \sum_{i\geq 0} k'_i\, B^i \quad \text{with } -(B-1) \leq k'_i \leq B-1 \ .$$

The corresponding representation $(\dots, k'_2, k'_1, k'_0)_B$ is called a *modified radix-B form* for k. Modified radix-B forms are not unique. For example, $(2, 2)_4$ and $(1, -2, 2)_4$ are two modified radix-4 forms for 10.

Signed-digit representations. Minimal representations using signed digits find applications in the theory of arithmetic codes [39] and in fast arithmetic techniques [22]. In these applications, the minimality requirement relates to the Hamming weight of the representation (i.e., the number of nonzero digits). For radix 2, Reitwiesner [34] proved that the so-called non-adjacent form (NAF) is optimal in the sense that it has minimal Hamming weight among the modified radix-2 forms. The general case was later addressed by Clark and Liang [10]. They present a minimal representation for any signed-radix B. In that case, Arno and Wheeler [2] precisely estimated that the average proportion of nonzero digits is equal to $(B-1)/(B+1)$.

The main application of non-adjacent forms in cryptography resides in fast exponentiation [16]; in particular, in settings wherein the computation of an inverse is inexpensive like in elliptic curve groups [29]. Non-adjacent forms were further adapted to certain classes of elliptic curves by decomposing integers as power sums of the Frobenius endomorphism [21,25,36]. Another extension of the basic NAF representation is to consider a succession of w digits, at most one of them being nonzero [36,38,31]. Yet another extension of the basic NAF is to rely on certain digit sets of the form $\{0, 1, x\}$ with representations featuring the non-adjacency property [30]. Binary representations with respect to more general digit sets of the form $\{0, 1, x, y, \dots, z\}$ are investigated in [18]. Alternative minimal modified radix-B forms for fast exponentiation, but enabling to scan the exponent digits from the left to the right, are presented in [19,20,32].

Fully homomorphic encryption and noise propagation. A salient feature of known constructions for fully homomorphic encryption [35,14] is the presence of noise in the ciphertexts for security reasons. We refer the reader to [17] for an excellent survey on homomorphic encryption.

Here is a simple illustration. Multiplying a noisy ciphertext by a scalar may result in a ciphertext whose noise error exceeds tolerated bounds. A standard trick to control the noise growth is to decompose the scalar with respect to a (small) radix [4,3]. This technique has been applied in a number of fully homomorphic encryption schemes—at the heart of the encryption process or as an auxiliary tool for accompanying gadgets or procedures; see e.g. [6,27,4,15,1,13,7,9].

Let Enc denote a homomorphic encryption scheme. Imagine we need to evaluate $c \leftarrow k \cdot \mathrm{Enc}(x)$ for some scalar k and ciphertext $\mathrm{Enc}(x)$. Instead of directly computing $c \leftarrow k \cdot \mathrm{Enc}(x)$, letting $k = \sum_{i=0}^{n} k'_i\, B^i$ for a given radix B, we can alternatively obtain c as

$$c \leftarrow \sum_{i=0}^{n} k'_i \; \mathrm{Enc}(B^i\, x)$$

from the precomputed ciphertexts $\mathrm{Enc}(B^i\, x)$, for $0 \le i \le n$.

Let us look at the noise propagation in this second approach. For a random variable X, we respectively denote by $\mathbb{E}[X]$ and $\mathrm{Var}(X)$ its expectation and its variance. Suppose that scalar k with $|k| < B^n$ is drawn at random. Assuming that the noise is centered and that its variance is bounded by the same threshold σ^2 in $\mathrm{Enc}(x)$ and $\mathrm{Enc}(B^i\, x)$, the noise present in $c \leftarrow k \cdot \mathrm{Enc}(x)$ has its variance bounded by $(\mathrm{Var}(k) + \mathbb{E}[k]^2)\, \sigma^2 = \mathbb{E}[k^2]\, \sigma^2$ while the noise present in $c \leftarrow \sum_{i=0}^{n} k'_i\; \mathrm{Enc}(B^i\, x)$ has its variance bounded by $\left(\sum_{i=0}^{n} (\mathrm{Var}(k'_i) + \mathbb{E}[k'_i]^2)\right) \sigma^2 = \left(\sum_{i=0}^{n} \mathbb{E}[{k'_i}^2]\right) \sigma^2$. Observe that $\mathbb{E}[k^2] = \frac{1}{3}(B^n - 1)B^n$ if k is uniform over $\{-B^n + 1, \ldots, B^n - 1\}$ and that $k'_i \in \{-B + 1, \ldots, B - 1\}$ if $(k'_n, \ldots, k'_0)$ is a modified radix-B for k. Hence, $\sum_{i=0}^{n} \mathbb{E}[{k'_i}^2]$ is expected to be much smaller than $\mathbb{E}[k^2]$; this is all the more true that $\sum_{i=0}^{n} \mathbb{E}[{k'_i}^2]$ is small. It is therefore of interest to produce modified radix-B forms that minimize this latter bound in order to contain the noise propagation in ciphertexts.

Contributions. Not all signed-digit representations equally perform. We saw for example that $(2, 2)_4$ and $(1, -2, 2)_4$ are two valid radix-4 representations for 10. Another representation for 10 is $(1, -1, -2)_4$, This paper seeks for modified radix-B forms $(k'_n, \ldots, k'_0)$ that minimize the quantity $\sum_{i=0}^{n} {k'_i}^2$. Back to our example, the form $(1, -1, -2)_4$ is actually what we call a "balanced non-adjacent form" and, as will be shown, constitutes an optimal choice for decomposing 10 in radix 4.

Our main results are:

- We define a new modified radix-B form that we call *balanced non-adjacent form* (or BNAF in short). These forms get one's name from the usual NAF because for $B = 2$ they exhibit the non-adjacency property. We prove that every integer has a BNAF and that the BNAF is unique.
- We propose a simple criterion to check whether or not a modified radix-B form is a BNAF. Based on it, we present algorithms for producing BNAFs on various input formats.
- We introduce the metric of Euclidean weight for modified radix-B forms and prove that it is optimally met by BNAFs. Specifically, we show that among all possible modified radix-B forms $(k'_n, \ldots, k'_0)$ for a given integer k, the BNAF always minimizes the quantity $\sum_{i=0}^{n} {k'_i}^2$.
- We study statistical properties of BNAFs:
  1. We use Markov chains to model the asymptotic behavior of the BNAF recoding process and provide estimates for the occurrence probability for the digits in a random BNAF.
  2. We determine the exact distribution of BNAFs for n-digit random integers.

Outline of the paper. The rest of the paper is organized as follows. In the next section, we review some signed-digit representations. Section 3 is the core of the paper. We define the BNAF and prove important and useful properties the BNAF satisfies. In Section 4, we present a generic recoding algorithm. This algorithm is used to analyze the probability distribution of the BNAF representation under different settings. In Section 5, we extend the previous results to modular representations. Finally, we demonstrate a number of cryptographic applications in Section 6.

2 Balanced Signed-Digit Representations

Integers can always be recoded with digits in the set $\{-B_0, \dots, B-1-B_0\}$ for any integer $0 \le B_0 < B$. When $B_0 = \lfloor B/2 \rfloor$, the corresponding representation is known as *balanced* modified radix-B form [12, Chapter 9]. Remarkably, such a decomposition is at most one digit longer than the standard unsigned decomposition. In particular, any nonnegative integer $k < B^n$ can be recoded as a balanced radix-B form having at most $n+1$ digits.

2.1 Odd radices

If B is odd and $B_0 = \lfloor B/2 \rfloor$ then $-B_0 = -\frac{B-1}{2}$ and $B - 1 - B_0 = \frac{B-1}{2}$. In this case, the corresponding form with digits in the balanced set $\{-\frac{B-1}{2}, \dots, \frac{B-1}{2}\}$ for a nonnegative integer k is easily obtained. Analogously to obtaining the regular radix-B representation, the idea is to repeatedly divide by B, obtain the corresponding remainder, and repeat the process with the resulting quotient. The difference is that the remainders are chosen in the set $\{-\frac{B-1}{2}, \dots, \frac{B-1}{2}\}$ (instead of $\{0, \dots, B-1\}$).

Example 1. Take $B = 5$ and consider the integer $k = 93$. Starting with 93, we successively obtain $93 = 19\cdot5-2$, $19 = 4\cdot5-1$, $4 = 1\cdot5-1$, and $1 = 0\cdot5+1$. The balanced modified radix-5 for $k = 93$ is therefore $(1, -1, -1, -2)_5$.

2.2 Even radices

The previous methodology similarly applies to an *even* value for $B \ge 4$ [11, § 3]. It produces a valid modified radix-B representation, which is however not [fully] balanced. The so-obtained digits k'_i belong to the set $\{-\frac{B}{2}, \dots, \frac{B}{2} - 1\}$. This is the recoding typically used with the gadget decomposition; see e.g. [9, Algorithm 1] for B a power of two.

Example 2. With radix $B = 4$, an application to $k = 93$ yields the representation $(1, -2, -2, -1, 1)_4$. Note that the digits are in the set $\{-2, \dots, 1\}$.

When the radix B is even, one has $-\frac{B}{2} \equiv \frac{B}{2} \pmod{B}$. Digits $-\frac{B}{2}$ and $\frac{B}{2}$ can then be used interchangeably. By allowing remainders of $\frac{B}{2}$, one can obtain different representations using the digit set $\{-\frac{B}{2}, \dots, \frac{B}{2}\}$.

Example 3. Continuing with Example 2, one can check that $(1, 2, -1, 1)_4$ is another valid modified radix-4 representation for $k = 93$—but now with digits in the set $\{-2, \dots, 2\}$.

3 Balanced Non-Adjacent Forms

Here we introduce the balanced non-adjacent form and characterize its properties. This section mainly deals with integers; extensions to modular representations are covered in Section 5.

3.1 Definition

We start with the general definition.

Definition 1. *Let $B \ge 2$ be a radix. A modified radix-B representation*

$$(\dots, k'_2, k'_1, k'_0)$$

for an integer $k = \sum_i k'_i B^i$ is called a balanced non-adjacent form (BNAF) *if and only if, for all i,*

(C1) $|k'_i| \le \lfloor \frac{B}{2} \rfloor$;
(C2) $0 \le k'_i \cdot k'_{i+1} \le \lfloor \frac{B}{2} \rfloor \left(\lfloor \frac{B}{2} \rfloor - 1 \right)$ *when* $|k'_i| = \lceil \frac{B}{2} \rceil$.

Remark 1. Conditions (C1) and (C2) are exclusive when the radix is odd. For an odd radix B, the definition of a BNAF simplifies to $|k'_i| \leq \frac{B-1}{2}$ for all i.

For an even radix B, a pair of digits (k'_{i+1}, k'_i) with $|k'_{i+1}|, |k'_i| \leq \lfloor \frac{B}{2} \rfloor$ that satisfies Condition (C2) is said *admissible*. Interestingly, the definition of a BNAF for $B = 2$ coincides with the definition of the non-adjacent form (NAF): $k'_i \in \{-1, 0, 1\}$ and $k'_i \cdot k'_{i+1} = 0$.

We present below a generic algorithm that converts any modified-radix form $(k'_n, \dots, k'_0)$ where $k'_i \in \{-(B-1), \dots, B-1\}$ for an integer k with $|k| < B^n$ into a BNAF. The algorithm is valid for both even and odd radices.

Algorithm 1: Conversion algorithm

```
Input: Modified radix-$B$ form $(k'_n, \dots, k'_0)$ of an integer $k = \sum_{i=0}^{n} k'_i B^i$ with $|k| < B^n$
Output: BNAF$(k) \leftarrow (k'_n, \dots, k'_0)$
for $i = 0$ to $n-1$ do
    $\sigma_i \leftarrow \mathrm{sign}(k'_i)$
    if $\big(|k'_i| > \lfloor \frac{B}{2} \rfloor\big) \vee \big((|k'_i| = \lceil \frac{B}{2} \rceil) \wedge ((-\lfloor \frac{B}{2} \rfloor \leq \sigma_i \cdot k'_{i+1} \leq -1) \vee (\lfloor \frac{B}{2} \rfloor \leq \sigma_i \cdot k'_{i+1} \leq B-1))\big)$ then
        $k'_i \leftarrow k'_i - \sigma_i \cdot B$
        $k'_{i+1} \leftarrow k'_{i+1} + \sigma_i$
    end if
end for
return $(k'_n, \dots, k'_0)$
```

Lemma 1. *Algorithm 1 is correct.*

Proof. We have to prove that, on input a modified radix-B of an integer k with $|k| < B^n$, Algorithm 1 actually outputs a BNAF.

Iteration i of the for-loop at most modifies the values of digits k'_i and k'_{i+1}. As will become apparent, all along the algorithm, it holds that:

- $k'_i \in \{-B, \dots, B\}$ and $k'_{i+1} \in \{-B+1, \dots, B-1\}$ before entering iteration i;
- $|k'_i| \leq \lfloor \frac{B}{2} \rfloor$ after exiting iteration i and the corresponding value for k'_i is unchanged by the next iterations.

We introduce some notation for more clarity. Let $({k'_n}^{(0)}, \dots, {k'_0}^{(0)})$ denote the input in Algorithm 1 of a modified radix-B form of an integer $k = \sum_{j=0}^{n} {k'_j}^{(0)} B^j$, $|k| < B^n$. More generally, let $({k'_n}^{(i)}, \dots, {k'_0}^{(i)})$ denote the representation entering iteration i in Algorithm 1—remark that $({k'_n}^{(i+1)}, \dots, {k'_0}^{(i+1)})$ also denotes the representation exiting iteration i. Moreover, we note that the algorithm is such that $\sum_{j=0}^{n} k'_j B^j$ always keeps equal to k; i.e., $\sum_{j=0}^{n} {k'_j}^{(i+1)} B^j = k$ for all $0 \leq i \leq n-1$. We also have $\sigma_i = \mathrm{sign}({k'_i}^{(i)})$. The output of Algorithm 1 is $\sum_{j=0}^{n} {k'_j}^{(n)} B^j$.

With the new notation, the two above observations can be rewritten as:

- ${k'_i}^{(i)} \in \{-B, \dots, B\}$ and ${k'_{i+1}}^{(i)} \in \{-B+1, \dots, B-1\}$;
- $|{k'_i}^{(i+1)}| \leq \lfloor \frac{B}{2} \rfloor$ and ${k'_i}^{(j)} = {k'_i}^{(i+1)}$ for $i+1 < j \leq n-1$.

The first observation and the relation ${k'_i}^{(j)} = {k'_i}^{(i+1)}$ for $i+1 < j \leq n-1$ are easily verified by inspection of the algorithm. We now show that the relation $|{k'_i}^{(i+1)}| \leq \lfloor \frac{B}{2} \rfloor$ is verified. If $|{k'_i}^{(i)}| > \lfloor \frac{B}{2} \rfloor$ (i.e., ${k'_i}^{(i)} \in \{-B, \dots, -\lfloor \frac{B}{2} \rfloor - 1\} \cup \{\lfloor \frac{B}{2} \rfloor + 1, \dots, B\}$), it is updated as ${k'_i}^{(i+1)} = {k'_i}^{(i)} - \sigma_i \cdot B \in \{-B + \lfloor \frac{B}{2} \rfloor + 1, \dots, B -$

$\lfloor \frac{B}{2} \rfloor - 1\} \subseteq \{-\lfloor \frac{B}{2} \rfloor, \dots, \lfloor \frac{B}{2} \rfloor\}$. Observe also that when B is even and $|k'^{(i)}_i| = \lceil \frac{B}{2} \rceil = \lfloor \frac{B}{2} \rfloor$ and is updated then it becomes $k'^{(i+1)}_i = k'^{(i)}_i - \sigma_i \cdot B \in \{\pm\lfloor \frac{B}{2} \rfloor\}$. In all cases, the relation $|k'^{(i+1)}_i| \le \lfloor \frac{B}{2} \rfloor$ is therefore satisfied.

Since the for-loop iterates for i from 0 to $n-1$ and since $k'^{(j)}_i = k'^{(i+1)}_i$ for $j > i+1$, from the relation $|k'^{(i+1)}_i| \le \lfloor \frac{B}{2} \rfloor$, it follows that Condition (C1) holds true for all $0 \le i \le n-1$; i.e., $|k'^{(n)}_i| \le \lfloor \frac{B}{2} \rfloor$ for $0 \le i \le n-1$. It also holds true for the output k'_n because the algorithm keeps invariant $\sum_{j=0}^{n} k'^{(i+1)}_j B^j = k$; hence, the condition $|k| < B^n$ implies that the final output ${k'_n}^{(n)}$ lies in $\{0, 1, -1\}$ and thus $|{k'_n}^{(n)}| \le \lfloor \frac{B}{2} \rfloor$ because $B \ge 2$.

For even radices B, we need in addition to check that Condition (C2) is fulfilled. We so assume that B is even and so $\lfloor \frac{B}{2} \rfloor = \lceil \frac{B}{2} \rceil = \frac{B}{2}$. If the value of $k'^{(i)}_i$ entering iteration i satisfies $|k'^{(i)}_i| < \frac{B}{2}$ then this value remains unchanged until the end of the algorithm; as a consequence, the output $k'^{(n)}_i$ belongs to $\{-\frac{B}{2} + 1, \dots, \frac{B}{2} - 1\}$. If the value of $k'^{(i)}_i$ entering iteration i satisfies $|k'^{(i)}_i| > \frac{B}{2}$ then it is updated as $k'^{(i+1)}_i = k'^{(i)}_i - \sigma_i \cdot B \in \{-\frac{B}{2} + 1, \dots, \frac{B}{2} - 1\}$ and will no longer be modified by the subsequent iterations; as a consequence, the output $k'^{(n)}_i$ belongs to $\{-\frac{B}{2} + 1, \dots, \frac{B}{2} - 1\}$. For output values $k'^{(n)}_i \in \{-\frac{B}{2} + 1, \dots, \frac{B}{2} - 1\}$, Condition (C2) is always valid. Hence, the only case that needs to be analyzed is when the value of $k'^{(i)}_i$ entering iteration i is $\pm\frac{B}{2}$. There are four sub-cases according to the value of $\sigma_i \cdot {k'_{i+1}}^{(i)}$ entering iteration i.

- Sub-case 1: $k'^{(i)}_i \in \{\pm\frac{B}{2}\}$ and $\sigma_i \cdot {k'_{i+1}}^{(i)} \in \{-B+1, \dots, -\frac{B}{2} - 1\}$. In this case, as seen from Algorithm 1, k'_i and k'_{i+1} are not modified at iteration i. Hence, the final output value for k'_i is $k'^{(n)}_i = \frac{B}{2}$ if $\sigma_i = 1$ and $k'^{(n)}_i = -\frac{B}{2}$ if $\sigma_i = -1$. Furthermore, since ${k'_{i+1}}^{(i+1)} = {k'_{i+1}}^{(i)}$, if $\sigma_i = 1$ then ${k'_{i+1}}^{(i+1)} \in \{-B + 1, \dots, -\frac{B}{2} - 1\}$ and so will be changed at iteration $i+1$ to eventually become the final output value ${k'_{i+1}}^{(n)} = {k'_{i+1}}^{(i+1)} - \sigma_{i+1} \cdot B \in \{1, \dots, \frac{B}{2} - 1\}$, which satisfies Condition (C2). Likewise, if $\sigma_i = -1$ then ${k'_{i+1}}^{(i+1)} \in \{\frac{B}{2} + 1, \dots, B-1\}$ and so will be changed at iteration $i+1$ to eventually become ${k'_{i+1}}^{(n)} = {k'_{i+1}}^{(i+1)} - \sigma_{i+1} \cdot B \in \{-\frac{B}{2} + 1, \dots, -1\}$, which again satisfies Condition (C2).
- Sub-case 2: $k'^{(i)}_i \in \{\pm\frac{B}{2}\}$ and $\sigma_i \cdot {k'_{i+1}}^{(i)} \in \{-\frac{B}{2}, \dots, -1\}$. In this case, both k'_i and k'_{i+1} are updated at iteration i as $k'^{(i+1)}_i \leftarrow k'^{(i)}_i - \sigma_i \cdot B = -k'^{(i)}_i \in \{\mp\frac{B}{2}\}$ and ${k'_{i+1}}^{(i+1)} \leftarrow {k'_{i+1}}^{(i)} + \sigma_i$. Hence, at the end of iteration i, the resulting ${k'_{i+1}}^{(i+1)}$ belongs to $\{-\frac{B}{2} + 1, \dots, 0\}$ if $\sigma_i = 1$ and to $\{0, \dots, \frac{B}{2} - 1\}$ if $\sigma_i = -1$. In both cases, this value for ${k'_{i+1}}^{(i+1)}$ being such that $|{k'_{i+1}}^{(i+1)}| < \frac{B}{2}$, it won't be changed at iteration $i+1$. Moreover, as the resulting $k'^{(n)}_i = k'^{(i+1)}_i$ and ${k'_{i+1}}^{(n)} = {k'_{i+1}}^{(i+1)}$ have the same sign, the final output pair $({k'_{i+1}}^{(n)}, k'^{(n)}_i)$ satisfies Condition (C2).
- Sub-case 3: $k'^{(i)}_i \in \{\pm\frac{B}{2}\}$ and $\sigma_i \cdot {k'_{i+1}}^{(i)} \in \{0, \dots, \frac{B}{2} - 1\}$. In this case, the values of k'_i and k'_{i+1} won't be changed at iteration i nor at iteration $i+1$. The final output values are therefore $k'^{(n)}_i = \frac{B}{2}$ and ${k'_{i+1}}^{(n)} \in \{0, \dots, \frac{B}{2} - 1\}$ if $\sigma_i = 1$, and $k'^{(n)}_i = -\frac{B}{2}$ and ${k'_{i+1}}^{(n)} \in \{-\frac{B}{2} + 1, \dots, 0\}$ if $\sigma_i = -1$. Both cases satisfy Condition (C2).
- Sub-case 4: $k'^{(i)}_i \in \{\pm\frac{B}{2}\}$ and $\sigma_i \cdot {k'_{i+1}}^{(i)} \in \{\frac{B}{2}, \dots, B-1\}$. In this case, the values of k'_i and k'_{i+1} are changed at iteration i as $k'^{(i+1)}_i = -k'^{(i)}_i$ and ${k'_{i+1}}^{(i+1)} = {k'_{i+1}}^{(i)} + \sigma_i \in \{\frac{B}{2} + 1, \dots, B\}$ if $\sigma_i = 1$ and $\in \{-B, \dots, -\frac{B}{2} - 1\}$ if $\sigma_i = -1$. In turn, ${k'_{i+1}}^{(i+1)}$ will be changed at iteration $i+1$ to eventually become ${k'_{i+1}}^{(n)} = {k'_{i+1}}^{(i+1)} - \sigma_{i+1} \cdot B = {k'_{i+1}}^{(i+1)} - \sigma_i \cdot B$. The final output values are therefore such that $k'^{(n)}_i = -\frac{B}{2}$ and ${k'_{i+1}}^{(n)} \in \{-\frac{B}{2} + 1, \dots, 0\}$ if $\sigma_i = 1$, and $k'^{(n)}_i = \frac{B}{2}$ and ${k'_{i+1}}^{(n)} \in \{0, \dots, \frac{B}{2} - 1\}$ if $\sigma_i = -1$. Again Condition (C2) is satisfied.

This shows that Algorithm 1 is correct and produces a BNAF. □

3.2 Properties

Although an integer can have several modified radix-B representations, the next theorem states that it has exactly one BNAF.

Theorem 1. *Every integer has a unique BNAF.*

Proof. Lemma 1 shows that there exists a BNAF for every integer. Suppose that an integer k possesses two different BNAF representations: $k = \sum_i k_i' \, B^i$ and $k = \sum_i k_i'' \, B^i$. Let i^* denote the smallest index such that $k_{i^*}' \neq k_{i^*}''$. From $k = \sum_i k_i' \, B^i = \sum_i k_i'' \, B^i$, we deduce that $(k_{i^*}' - k_{i^*}'') \, B^{i^*} \equiv 0 \pmod{B^{i^*+1}}$ and thus $k_{i^*}' - k_{i^*}'' \equiv 0 \pmod{B}$.

The cases of odd radix and even radix are treated separately.

1. Consider the case of an odd radix B. Since $k_{i^*}' \neq k_{i^*}''$, the relation $k_{i^*}' - k_{i^*}'' \equiv 0 \pmod{B}$ implies $k_{i^*}' - k_{i^*}'' = \alpha \, B$ for some nonzero integer α. Since $|k_{i^*}'| \leq \lfloor \frac{B}{2} \rfloor = \frac{B-1}{2}$ and $|k_{i^*}''| \leq \lfloor \frac{B}{2} \rfloor = \frac{B-1}{2}$, it follows that $|k_{i^*}' - k_{i^*}''| \leq B - 1$. The equation $k_{i^*}' - k_{i^*}'' = \alpha \, B$ has therefore no nonzero solution α. A contradiction.
2. Consider now the case of an even radix B. Again, since by definition $|k_{i^*}'| \leq \lfloor \frac{B}{2} \rfloor = \frac{B}{2}$ and $|k_{i^*}''| \leq \lfloor \frac{B}{2} \rfloor = \frac{B}{2}$, and $k_{i^*}' \neq k_{i^*}''$, it thus follows from $k_{i^*}' - k_{i^*}'' \equiv 0 \pmod{B}$ that $k_{i^*}' = -k_{i^*}'' = \pm\frac{B}{2}$. Without loss of generality, we assume that $k_{i^*}' = \frac{B}{2}$ and $k_{i^*}'' = -\frac{B}{2}$. From $\sum_i k_i' \, B^i \equiv \sum_i k_i'' \, B^i \pmod{B^{i^*+2}}$, we deduce that $(k_{i^*+1}' - k_{i^*+1}'') \, B + (k_{i^*}' - k_{i^*}'') \equiv 0 \pmod{B^2}$. In turn, this yields $(k_{i^*+1}' - k_{i^*+1}'') \, B + B \equiv 0 \pmod{B^2}$ and therefore $k_{i^*+1}' - k_{i^*+1}'' + 1 \equiv 0 \pmod{B} \iff k_{i^*+1}'' \equiv k_{i^*+1}' + 1 \pmod{B}$. Furthermore, as $k_{i^*}' = \frac{B}{2}$ and $k_{i^*}'' = -\frac{B}{2}$, we must have $0 \leq k_{i^*+1}' \leq \frac{B}{2} - 1$ and $-\frac{B}{2} + 1 \leq k_{i^*+1}'' \leq 0$ to fulfill Condition (C2). From $k_{i^*+1}'' \equiv k_{i^*+1}' + 1 \pmod{B}$, the requirement $0 \leq k_{i^*+1}' \leq \frac{B}{2} - 1$ leads to $k_{i^*+1}'' \in \{1, \ldots, \frac{B}{2}\} \cup \{-\frac{B}{2}\}$, which contradicts the condition $-\frac{B}{2} + 1 \leq k_{i^*+1}'' \leq 0$.

Consequently, the BNAF is unique. □

Integers can have multiple modified radix-B forms. The notion of weight relates an integer to its representation and enables to qualitatively distinguish among different representations.

Definition 2. *Given a base $B \geq 2$, the* Euclidean weight *of an integer k is the smallest value W such that there is a modified radix-B form*

$$(k_n', \ldots, k_0') \quad \text{such that} \sum_{i=0}^{n} k_i' \, B^i = k \text{ and} \sum_{i=0}^{n} {k_i'}^2 = W$$

for digits k_i' with $|k_i'| < B$.

The next theorem exhibits the main feature of the BNAF representation. It states that the BNAF of an integer k, $\mathrm{BNAF}(k) = (k_n', \ldots, k_0')$, minimizes the quantity $\sum_{i=0}^{n} {k_i'}^2$. The BNAF representation is in that sense optimal.

Theorem 2. *If $(k_n', \ldots, k_0')$ denotes the BNAF of an integer k then the Euclidean weight of k is equal to $\sum_{i=0}^{n} {k_i'}^2$.*

Proof. The proof relies on Algorithm 1.

We use the notation used in the proof of Lemma 1. We let $({k_n'}^{(i)}, \ldots, {k_0'}^{(i)})$ denote the representation entering iteration i in Algorithm 1; we also let $\sigma_i = \mathrm{sign}({k_i'}^{(i)})$. By abuse of language, for the representation $({k_n'}^{(i)}, \ldots, {k_0'}^{(i)})$, we call the quantity

$$W^{(i)} = \sum_{j=0}^{n} \left({k_j'}^{(i)}\right)^2$$

the Euclidean weight of $({k_n'}^{(i)}, \ldots, {k_0'}^{(i)})$ in Algorithm 1. The weight of the output of Algorithm 1 is given by $W^{(n)}$.

We now show that $W^{(n)} \leq W^{(0)}$, namely that the Euclidean weight of the output is smaller than or equal to the Euclidean weight of the input. Specifically, we show that if the if-branching is executed at iteration i then there exists an index j with $i < j \leq n$ such that $W^{(j)} \leq W^{(i)}$; if it is not executed then obviously $W^{(i+1)} = W^{(i)}$.

If the if-branching is executed at iteration i, it follows that ${k'_i}^{(i+1)} = {k'_i}^{(i)} - \sigma_i\, B$ and ${k'_{i+1}}^{(i+1)} = {k'_{i+1}}^{(i)} + \sigma_i$. The other values are unchanged. This yields

$$\begin{aligned} W^{(i+1)} - W^{(i)} &= \sum\nolimits_{j=0}^{n} ({k'_j}^{(i+1)})^2 - \sum\nolimits_{j=0}^{n} ({k'_j}^{(i)})^2 \\ &= ({k'_i}^{(i+1)})^2 + ({k'_{i+1}}^{(i+1)})^2 - ({k'_i}^{(i)})^2 - ({k'_{i+1}}^{(i)})^2 \\ &= -\sigma_i B(2{k'_i}^{(i)} - \sigma_i B) + \sigma_i(2{k'_{i+1}}^{(i)} + \sigma_i) \\ &= -2B\,|{k'_i}^{(i)}| + B^2 + 2\sigma_i\, {k'_{i+1}}^{(i)} + 1 \ . \end{aligned} \tag{*}$$

The if-branching is executed at iteration i when (at least) one of the following conditions is met:

1. $|{k'_i}^{(i)}| > \lfloor \frac{B}{2} \rfloor$. There are several sub-cases.
   (a) B is even. Then, since $|{k'_i}^{(i)}| \geq \lfloor \frac{B}{2} \rfloor + 1 = \frac{B}{2} + 1$ and $\sigma_i\, {k'_{i+1}}^{(i)} \leq B - 1$, we get using (*)

   $$W^{(i+1)} - W^{(i)} \leq -2B(\tfrac{B}{2} + 1) + B^2 + 2(B-1) + 1 = -1 < 0\,,$$

   that is, $W^{(i+1)} < W^{(i)}$. We have $j = i + 1$.
   (b) B is odd and $\sigma_i\, {k'_{i+1}}^{(i)} \leq \lfloor \frac{B}{2} \rfloor$. When B is odd, we have $\lfloor \frac{B}{2} \rfloor = \frac{B-1}{2}$. This case is similar to the previous one. We obtain from (*)

   $$W^{(i+1)} - W^{(i)} \leq -2B(\tfrac{B-1}{2} + 1) + B^2 + 2(\tfrac{B-1}{2}) + 1 = 0 \ .$$

   We get $W^{(i+1)} \leq W^{(i)}$ and have $j = i + 1$.
   (c) B is odd and $\sigma_i\, {k'_{i+1}}^{(i)} > \lfloor \frac{B}{2} \rfloor$. First, we note that $\sigma_i\, {k'_{i+1}}^{(i)} > \lfloor \frac{B}{2} \rfloor$ supposes $i \leq n-2$ because for $i = n-1$ we have ${k'_{i+1}}^{(i)} \in \{0, 1, -1\}$ ($|k| < B^n$). If the condition $\sigma_i\, {k'_{i+1}}^{(i)} > \lfloor \frac{B}{2} \rfloor$ holds, this can only occur if $\sigma_i = 1$ and ${k'_{i+1}}^{(i)} > \lfloor \frac{B}{2} \rfloor$ or if $\sigma_i = -1$ and ${k'_{i+1}}^{(i)} < -\lfloor \frac{B}{2} \rfloor$. Moreover, since we have ${k'_{i+1}}^{(i+1)} = {k'_{i+1}}^{(i)} + \sigma_i$, we infer that $\sigma_{i+1} = \sigma_i$. The if-branching is executed at both iterations i and $i+1$. A double application of (*) yields

   $$\begin{aligned} &W^{(i+2)} - W^{(i)} \\ &\quad = (-2B\,|{k'_{i+1}}^{(i+1)}| + B^2 + 2\sigma_{i+1}\, {k'_{i+2}}^{(i+1)} + 1) + (-2B\,|{k'_i}^{(i)}| + B^2 + 2\sigma_i\, {k'_{i+1}}^{(i)} + 1) \\ &\quad \leq -2(B-1)\,|{k'_{i+1}}^{(i)}| - 2B + 2B^2 + 2 + 2(B-1) - 2B(\tfrac{B-1}{2} + 1) \\ &\quad \leq -2(B-1)\,|{k'_{i+1}}^{(i)}| + B(B-1) \\ &\quad \leq -2(B-1)(\tfrac{B-1}{2} + 1) + B(B-1) = 1 - B \\ &\quad < 0 \ . \end{aligned}$$

   Hence, we have $W^{(i+2)} < W^{(i)}$ and so $j = i + 2$.
2. $(|{k'_i}^{(i)}| = \lceil \frac{B}{2} \rceil) \wedge (-\lfloor \frac{B}{2} \rfloor \leq \sigma_i \cdot {k'_{i+1}}^{(i)} \leq -1)$. We assume w.l.o.g. that B is even because when, it is odd, the condition $|{k'_i}^{(i)}| = \lceil \frac{B}{2} \rceil$ implies $|{k'_i}^{(i)}| > \lfloor \frac{B}{2} \rfloor$, which is covered in the previous case. We so get from (*)

   $$W^{(i+1)} - W^{(i)} \leq -2B\,\tfrac{B}{2} + B^2 + 2(-1) + 1 = -1 < 0$$

   and thus $W^{(i+1)} < W^{(i)}$. We have $j = i + 1$.
3. $(|{k'_i}^{(i)}| = \lceil \frac{B}{2} \rceil) \wedge (\lfloor \frac{B}{2} \rfloor \leq \sigma_i \cdot {k'_{i+1}}^{(i)} \leq B - 1)$. Here too, we assume w.l.o.g. that B is even. We distinguish two sub-cases.

(a) $\sigma_i \cdot {k'_{i+1}}^{(i)} > \lfloor \frac{B}{2} \rfloor = \frac{B}{2}$. This case is analogous to Case 1c. We have $\sigma_{i+1} = \sigma_i$. A double application of (*) yields

$$\begin{aligned} W^{(i+2)} - W^{(i)} &= (-2B\,|{k'_{i+1}}^{(i+1)}| + B^2 + 2\sigma_{i+1}\,{k'_{i+2}}^{(i+1)} + 1) + (-2B\,|{k'_i}^{(i)}| + B^2 + 2\sigma_i\,{k'_{i+1}}^{(i)} + 1) \\ &\le -2(B-1)\,|{k'_{i+1}}^{(i)}| - 2B + 2B^2 + 2 + 2(B-1) - 2B(\tfrac{B}{2}) \\ &\le -2(B-1)(\tfrac{B}{2}+1) + B^2 = 2 - B < 0 \ . \end{aligned}$$

We have $W^{(i+2)} < W^{(i)}$ and $j = i+2$.

(b) $\sigma_i \cdot {k'_{i+1}}^{(i)} = \lfloor \frac{B}{2} \rfloor = \frac{B}{2}$. Again, this requires $\sigma_{i+1} = \sigma_i$. We thus have ${k'_{i+1}}^{(i)} = {k'_i}^{(i)} \in \{\pm\frac{B}{2}\}$ or, equivalently, ${k'_{i+1}}^{(i)} = {k'_i}^{(i)} = \sigma_i \frac{B}{2}$. We let i^* denote the smallest index such that $i^* > i+1$ and ${k'_{i^*}}^{(i)} \neq \sigma_i \frac{B}{2}$. Note that, for $B \neq 2$, $i^* \le n$ because ${k'_n}^{(l)} \in \{0, 1, -1\}$ for all $0 \le l \le n$ since $|k| < B^n$, and consequently, ${k'_n}^{(l)} \notin \{\pm\frac{B}{2}\}$. The same is true for $B = 2$ because if ${k'_{n-1}}^{(l)} \in \{\pm\frac{B}{2}\}$ for some $0 \le l \le n$ then ${k'_n}^{(l)} \neq {k'_{n-1}}^{(l)}$ since $|k| < B^n$. Moreover, note that $i^* \le n-1$ when $\sigma_i\,{k'_{i^*}}^{(i)} > \frac{B}{2}$ because $({k'_n}^{(l)}, {k'_{n-1}}^{(l)}, {k'_{n-2}}^{(l)}) = (d, \sigma_i\frac{B}{2}, \sigma_i\frac{B}{2})$ for some digit $d > \sigma_i \frac{B}{2}$ contradicts $|k| < B^n$.
We have ${k'_i}^{(i)} = {k'_{i+1}}^{(i)} = \cdots = {k'_{i^*-1}}^{(i)} = \sigma_i\frac{B}{2}$ and ${k'_{i^*}}^{(i)} \neq \sigma_i \frac{B}{2}$. At every subsequent iteration up to iteration i^*-1, it turns out that the if-branching is executed, which results in

$$({k'_{i^*}}^{(i^*)}, {k'_{i^*-1}}^{(i^*)}, \ldots, {k'_{i+1}}^{(i^*)}, {k'_i}^{(i^*)}) = \left({k'_{i^*}}^{(i)} + \sigma_i, -\sigma_i(\tfrac{B}{2}-1), \ldots, -\sigma_i(\tfrac{B}{2}-1), -\sigma_i\tfrac{B}{2}\right) \ .$$

The cases of $\sigma_i\,{k'_{i^*}}^{(i)} < \frac{B}{2}$ and $\sigma_i\,{k'_{i^*}}^{(i)} > \frac{B}{2}$ are treated separately.

i. $\sigma_i\,{k'_{i^*}}^{(i)} < \frac{B}{2}$. In this case, we get

$$\begin{aligned} W^{(i^*)} - W^{(i)} &= ({k'_{i^*}}^{(i)} + \sigma_i)^2 + \sum_{l=i+1}^{i^*-1}(-\sigma_i(\tfrac{B}{2}-1))^2 + (-\sigma_i\tfrac{B}{2})^2 - ({k'_{i^*}}^{(i)})^2 - \sum_{l=i}^{i^*-1}(\sigma_i\tfrac{B}{2})^2 \\ &= 2\sigma_i\,{k'_{i^*}}^{(i)} + 1 - (i^* - i - 1)(B-1) \\ &\le 2(\tfrac{B}{2}-1) + 1 - (B-1) = 0 \end{aligned}$$

since $i^* \ge i+2$. Hence, $W^{(i^*)} \le W^{(i)}$ and we have $j = i^*$.

ii. $\sigma_i\,{k'_{i^*}}^{(i)} > \frac{B}{2}$. In this case, the if-branching is also executed at iteration i^*. We get

$$\begin{aligned} W^{(i^*+1)} - W^{(i)} &= ({k'_{i^*+1}}^{(i)} + \sigma_i)^2 + ({k'_{i^*}}^{(i)} - \sigma_i(B-1))^2 + \sum_{l=i+1}^{i^*-1}(-\sigma_i(\tfrac{B}{2}-1))^2 + (-\sigma_i\tfrac{B}{2})^2 \\ &\qquad - ({k'_{i^*+1}}^{(i)})^2 - ({k'_{i^*}}^{(i)})^2 - \sum_{l=i}^{i^*-1}(\sigma_i\tfrac{B}{2})^2 \\ &= 2\sigma_i\,{k'_{i^*+1}}^{(i)} + 1 - 2\sigma_i(B-1)\,{k'_{i^*}}^{(i)} + (B-1)^2 - (i^*-i-1)(B-1) \\ &\le 2(B-1) + 1 - 2(B-1)(\tfrac{B}{2}+1) + (B-1)^2 - (B-1) \\ &= 3 - 2B < 0 \ . \end{aligned}$$

We so have $W^{(i^*+1)} < W^{(i)}$ and $j = i^*+1$.

Theorem 1 teaches that the BNAF is unique. This means that, given an integer k and any modified radix-B representation of that integer, Algorithm 1 always returns the same modified radix-B form for k, namely the BNAF of k. Moreover, we just saw that the BNAF has an Euclidean weight that is smaller than or equal to the Euclidean weight of the input modified radix-B form. This implies that the BNAF representation has minimal Euclidean weight. In other words, if $(k'_n, \ldots, k'_0)$ is the BNAF of k, $|k| < B^n$, then the Euclidean weight of k is $W = \sum_{i=0}^{n} {k'_i}^2$. □

4 Recoding Algorithm

4.1 Description

The BNAF of an integer k can be obtained directly from the definition by repeatedly dividing k by B (integer division); if Conditions (C1) or (C2) (cf. Definition 1) are invalidated then a correction is applied to the resulting digit (and as well as to k). The "mod" operator in $K \bmod B$ indicates the unique value in $\{0, \dots, B-1\}$ that is congruent to K modulo B.

Algorithm 2: BNAF recoding

```
Input: Integer $k \neq 0$
Output: BNAF$(k) \leftarrow (k'_n, \dots, k'_0)$ with $k'_i \in \{-\lfloor \frac{B}{2} \rfloor, \dots, \lfloor \frac{B}{2} \rfloor\}$ s.t. $\sum_{i=0}^{n} k'_i B^i = k$
$K \leftarrow k$; $i \leftarrow 0$
while $(K \neq 0)$ do
    $k'_i \leftarrow K \bmod B$; $K \leftarrow (K - k'_i)/B$
    if $(k'_i > \lfloor \frac{B}{2} \rfloor) \vee \big((k'_i = \lceil \frac{B}{2} \rceil) \wedge ((K \bmod B) \geq \lfloor \frac{B}{2} \rfloor)\big)$ then
        $k'_i \leftarrow k'_i - B$; $K \leftarrow K + 1$
    end if
    $i \leftarrow i + 1$
end while
return $(k'_{i-1}, \dots, k'_0)$
```

Remark 2. As indicated in Remark 1, Conditions (C1) and (C2) are exclusive when radix B is odd. In this case, the if-branching in Algorithm 2 simply reads as **if** $(k'_i > \lfloor \frac{B}{2} \rfloor)$ **then**.

4.2 Stochastic analysis

This section studies the asymptotic behavior of the digits resulting from the BNAF recoding.

If $k'_i = K \bmod B$ then $(K - k'_i)/B = \lfloor K/B \rfloor$. Hence, we see that the output value of k'_i is at each iteration completely determined by the current values of $K \bmod B$ and of $\lfloor K/B \rfloor \bmod B$. A pair of digits (e, d) is therefore sufficient to describe each possible case.

It is useful to introduce some notation. We define the quantities K_i that keep track of the successive values of K entering the for-loop at iteration i, and represent $K_i \bmod B^2$ as the pair (e_i, d_i) with $0 \leq d_i, e_i < B$ such that $K_i \bmod B^2 = e_i\, B + d_i$. By construction, we have

$$\begin{cases} K_0 = k \\ K_{i+1} = \frac{K_i - k'_i}{B} & \text{for } 0 \leq i \leq n-1 \end{cases} .$$

Depending on the values of $K_i \bmod B^2 = (e_i, d_i)$, there are different cases to consider. This is detailed in Table 1.

From the last column in Tables 1(a) and 1(b), we remark that the knowledge of (e_i, d_i) only enables to obtain d_{i+1}. We have

$$d_{i+1} = K_{i+1} \bmod B = \frac{(K_i - k'_i) \bmod B^2}{B} = \frac{e_i B + d_i - k'_i}{B} \pmod B$$
$$= \begin{cases} e_i & \text{if } k'_i = d_i \\ e_i + 1 \pmod B & \text{if } k'_i = d_i - B \end{cases} .$$

Table 1. Output digit k'_i according to the pair (e_i, d_i).

(a) Odd radix B

State	(e_i, d_i)	k'_i	(e_{i+1}, d_{i+1})
a-1.	$\begin{cases} 0 \le d_i \le \frac{B-1}{2} \\ 0 \le e_i < B \end{cases}$	d_i	$(*, e_i)$
a-2.	$\begin{cases} \frac{B-1}{2} < d_i < B \\ 0 \le e_i < B \end{cases}$	$d_i - B$	$(*, (e_i + 1) \bmod B)$

(b) Even radix B

State	(e_i, d_i)	k'_i	(e_{i+1}, d_{i+1})
b-1.	$\begin{cases} 0 \le d_i < \frac{B}{2} \\ 0 \le e_i < B \end{cases}$	d_i	$(*, e_i)$
b-2.	$\begin{cases} d_i = \frac{B}{2} \\ 0 \le e_i < \frac{B}{2} \end{cases}$	$\frac{B}{2}$	$(*, e_i)$
b-3.	$\begin{cases} d_i = \frac{B}{2} \\ \frac{B}{2} \le e_i < B \end{cases}$	$-\frac{B}{2}$	$(*, (e_i + 1) \bmod B)$
b-4.	$\begin{cases} \frac{B}{2} < d_i < B \\ 0 \le e_i < B \end{cases}$	$d_i - B$	$(*, (e_i + 1) \bmod B)$

We also remark that if the radix-B digits forming k are uniformly random over $\{0, \ldots, B-1\}$ then so is e_{i+1}; in other words, e_{i+1} can take any value in $\{0, \ldots, B-1\}$ with a probability of $\frac{1}{B}$.

When B is odd (and thus $\lfloor \frac{B}{2} \rfloor = \frac{B-1}{2}$), from Table 1(a), we have $k'_i \leftarrow d_i \in \{0, \ldots, \frac{B-1}{2}\}$ or $k'_i \leftarrow d_i - B \in \{-\frac{B-1}{2}, \ldots, -1\}$. We therefrom infer that each digit is equiprobable in the recoding and so has an occurrence probability of $\Pr[k'_i = d] = \frac{1}{B}$ for any $d \in \{0, \ldots, \lfloor \frac{B}{2} \rfloor\}$.

When B is even, we obtain from Table 1(b) the following transition probabilities. For State b-1, since $d_{i+1} = e_i$ and $0 \le e_i < B$, there is a probability of $\frac{B/2}{B} = \frac{1}{2}$ to stay in State b-1 and a probability of $\frac{1/2}{B} = \frac{1}{2B}$ to transition to State b-2, and similarly of $\frac{1}{2B}$ to State b-3. The probability to transition to State b-4 from State b-1 is thus of $1 - \frac{1}{2} - \frac{1}{2B} - \frac{1}{2B} = \frac{B-2}{2B}$. For State b-2, since $d_{i+1} = e_i$ and $0 \le e_i < \frac{B}{2}$, the transition is necessarily to State b-1. For State b-3, since $d_{i+1} = e_i + 1 \bmod B$ and $\frac{B}{2} \le e_i < B$, there is a probability of $\frac{B/2-1}{B/2} = \frac{B-2}{B}$ to transition to State b-4 and a probability of $\frac{1}{B/2} = \frac{2}{B}$ to transition to State b-1 (i.e., when $e_i = B - 1$). Finally, for State b-4, since $d_{i+1} = e_i + 1 \bmod B$ and $0 \le e_i < B$, there is a probability of $\frac{B/2-1}{B} = \frac{B-2}{2B}$ to stay in State b-4 (i.e., when $e_i \in \{\frac{B}{2}, \ldots, B-2\}$), there is a probability of $\frac{1/2}{B}$ to transition either to State b-2 or b-3 (i.e., when $e_i = \frac{B}{2} - 1$), and there is a probability of $\frac{B/2}{B} = \frac{1}{2}$ to transition to State b-1 (i.e., when $e_i \in \{0, \ldots, \frac{B}{2} - 2\} \cup \{B-1\}$).

Schematically, we have the automaton depicted in Figure 1.
The corresponding Markov matrix $\mathbf{P}$ where element (i, j) denotes the probability for transitioning from State b-i to State b-j $(1 \le i, j \le 4)$ is given by

$$\mathbf{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2B} & \frac{1}{2B} & \frac{B-2}{2B} \\ 1 & 0 & 0 & 0 \\ \frac{2}{B} & 0 & 0 & \frac{B-2}{B} \\ \frac{1}{2} & \frac{1}{2B} & \frac{1}{2B} & \frac{B-2}{2B} \end{pmatrix} .$$

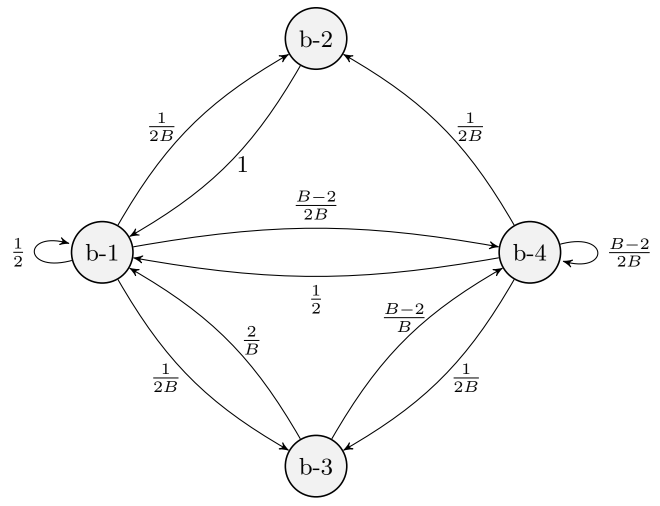

Fig. 1. Transition probabilities among the different states for an even radix B.

The companion stationary probability vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)$ satisfies $\boldsymbol{\pi}\,\mathbf{P} = \boldsymbol{\pi}$ subject to $\sum_{1\le j\le 4} \pi_j = 1$. We find

$$\boldsymbol{\pi} = \left(\frac{B^2+B+2}{2B(B+1)}, \frac{1}{2(B+1)}, \frac{1}{2(B+1)}, \frac{B-2}{2B}\right) .$$

We can now estimate the occurrence probability of each digit. This is made explicit in the next proposition.

Proposition 1. *Let $B \ge 2$ be a radix. Then a digit $k'_i \in \{-\lfloor \frac{B}{2} \rfloor, \dots, \lfloor \frac{B}{2} \rfloor\}$ from a uniformly random radix-B BNAF features the following distribution*

$$\Pr[k'_i = d] = \begin{cases} \frac{B+2}{B+1}\frac{1}{B} & \textit{if } d = 0 \textit{ and } B \textit{ is even} \\ \frac{1}{2(B+1)} & \textit{if } d \in \{-\lfloor \frac{B}{2} \rfloor, \lfloor \frac{B}{2} \rfloor\} \textit{ and } B \textit{ is even} \\ \frac{1}{B} & \textit{otherwise} \end{cases}$$

where $d \in \{-\lfloor \frac{B}{2} \rfloor, \dots, \lfloor \frac{B}{2} \rfloor\}$.

Proof. We already showed that $\Pr[k'_i = d] = \frac{1}{B}$ when B is odd. We henceforth assume that B is even. Excluding the digit $d_i = 0$, States b-1 and b-4 in Table 1(b) are symmetric. We so deduce $\pi_1 = \Pr[k'_i = 0] + \pi_4$ and thus $\Pr[k'_i = 0] = \frac{B^2+B+2}{2B(B+1)} - \frac{B-2}{2B} = \frac{B+2}{B(B+1)}$. From Table 1(b), we also get $\Pr[k'_i = \frac{B}{2}] = \pi_2 = \frac{1}{2(B+1)}$ and $\Pr[k'_i = -\frac{B}{2}] = \pi_3 = \frac{1}{2(B+1)}$. For the remaining case (i.e., $d \notin \{-\frac{B}{2}, 0, \frac{B}{2}\}$), we infer from States b-1 and b-4 in Table 1(b) (excluding 0) that every digit is equiprobable and thus $(\pi_1 - \Pr[k'_i = 0]) + \pi_4 = (B-2)\Pr[k'_i = d \mid d \notin \{-\frac{B}{2}, 0, \frac{B}{2}\}] \iff \Pr[k'_i = d \mid d \notin \{-\frac{B}{2}, 0, \frac{B}{2}\}] = \frac{2\pi_4}{B-2} = \frac{1}{B}$. ◻

As a corollary, we can easily deduce the corresponding variance.

Corollary 1. *Let $B \ge 2$ be a radix. Then a digit k'_i in a uniformly random radix-B BNAF satisfies*

$$\mathbb{E}[k'_i] = 0 \quad \textit{and} \quad \mathrm{Var}(k'_i) = \begin{cases} \frac{1}{12}(B^2 - 1) & \textit{if } B \textit{ is odd} \\ \frac{1}{12}\frac{(B+2)(B^2-B+1)}{B+1} & \textit{if } B \textit{ is even} \end{cases} .$$

Proof. This is immediate. We use the identity $\sum_{t=1}^{T} t^2 = \frac{1}{6}T(T+1)(2T+1)$. From its definition, since $\mathbb{E}[k'_i] = \sum_{-\lfloor \frac{B}{2} \rfloor \le d \le \lfloor \frac{B}{2} \rfloor} \Pr[k'_i = d]\, d = 0$ by symmetry, the variance is given by $\mathrm{Var}(k'_i) = \sum_{-\lfloor \frac{B}{2} \rfloor \le d \le \lfloor \frac{B}{2} \rfloor} \Pr[k'_i = d]\, d^2 = 2\sum_{d=1}^{\lfloor \frac{B}{2} \rfloor} \Pr[k'_i = d]\, d^2$. If B is odd, we get $\mathrm{Var}(k'_i) = 2\frac{1}{B}\sum_{d=1}^{\frac{B-1}{2}} d^2 = \frac{B^2-1}{12}$. If B is even then we get $\mathrm{Var}(k'_i) = 2\frac{1}{B}\sum_{d=1}^{\frac{B}{2}-1} d^2 + 2\frac{1}{2(B+1)}(\frac{B}{2})^2 = \frac{1}{12}(B-1)(B-2) + \frac{1}{4}\frac{B^2}{B+1} = \frac{(B+2)(B^2-B+1)}{12(B+1)}$. ◻

4.3 Exact distribution

In this section, we consider the set of nonnegative integers whose standard radix-B representation consists of at most n digits; that is, the set $\{0,\dots,B^n-1\}$. The previous analysis shows that endowing this set with the uniform probability measure results in a digit distribution for the BNAF satisfying

$$\Pr[k'_i] \sim \begin{cases} \frac{1}{B} & \text{when } B \text{ is odd} \\ \frac{B+2}{B+1}\frac{1}{B} & \text{when } B \text{ is even} \end{cases} \quad \text{as } n \to \infty \enspace .$$

For a finite value of n, when B is even, the exact digit distribution for 0 and $\pm B/2$ oscillates around the asymptotic value according to the digit index. Indeed, if $({S_1}^{(i)},{S_2}^{(i)},{S_3}^{(i)},{S_4}^{(i)})$ denotes the probability vector wherein component ${S_j}^{(i)}$ represents the probability of being in State b-j at iteration i in Algorithm 2 then

$$({S_1}^{(i+1)},{S_2}^{(i+1)},{S_3}^{(i+1)},{S_4}^{(i+1)}) = ({S_1}^{(i-1)},{S_2}^{(i-1)},{S_3}^{(i-1)},{S_4}^{(i-1)})\,\mathbf{P}$$

where $\mathbf{P}$ is the Markov matrix; see Section 4.2. For a uniformly random integer $k \in \{0,\dots,B^n-1\}$, letting $\mathrm{BNAF}(k) = (k'_n,\dots,k'_0)$, the least significant digit k'_0 is equiprobable amongst the possible values except for $\pm\frac{B}{2}$. Namely, we have $\Pr[k'_0 = d] = \frac{1}{B}$ if $d \in \{-\frac{B}{2}+1,\dots,\frac{B}{2}-1\}$ and $\Pr[k'_0 = d] = \frac{1}{2B}$ if $d \in \{\pm\frac{B}{2}\}$. We so deduce that $({S_1}^{(0)},{S_2}^{(0)},{S_3}^{(0)},{S_4}^{(0)}) = (\frac{1}{2},\frac{1}{2B},\frac{1}{2B},\frac{B-2}{2B})$. By induction, we find

$$\begin{aligned}({S_1}^{(i)},{S_2}^{(i)},{S_3}^{(i)},{S_4}^{(i)}) &= ({S_1}^{(0)},{S_2}^{(0)},{S_3}^{(0)},{S_4}^{(0)})\,\mathbf{P}^i \\ &= \Big(\frac{1}{2}+\frac{B^i+(-1)^{i+1}}{B^{i+1}(B+1)},\frac{B^{i+1}-(-1)^{i+1}}{2B^{i+1}(B+1)},\frac{B^{i+1}-(-1)^{i+1}}{2B^{i+1}(B+1)},\frac{B-2}{2B}\Big) \enspace .\end{aligned}$$

Hence, since $\Pr\big[k'_i = \frac{B}{2}\big] = {S_2}^{(i)}$ and $\Pr\big[k'_i = -\frac{B}{2}\big] = {S_3}^{(i)}$, we obtain

$$\Pr\big[k'_i = \tfrac{B}{2}\big] = \Pr\big[k'_i = -\tfrac{B}{2}\big] = \frac{1}{2(B+1)} - (-1)^{i+1}\frac{1}{2B^{i+1}(B+1)} \enspace .$$

In the same way, from the symmetry between State b-1 without 0 and State b-4, we have ${S_1}^{(i)} + \Pr\big[k'_i = 0\big] = {S_4}^{(i)}$ and thus

$$\Pr[k'_i = 0] = \frac{1}{B} + \frac{B^i+(-1)^{i+1}}{B^{i+1}(B+1)} = \frac{B+2}{B(B+1)} + (-1)^{i+1}\frac{1}{B^{i+1}(B+1)} \enspace .$$

Furthermore, as the BNAF distribution is computed over the finite set of integers in $\{0,\dots,B^n-1\}$, the exact digit distribution for the leading digits also differs from the asymptotic distribution. Owing to the BNAF definition, when B is even, the largest integer $k^* \in \{0,\dots,B^n-1\}$ having the most significant of its BNAF equal to 0 has for BNAF $(k'_n,k'_{n-1},k'_{n-2},k'_{n-3},\dots) = (0,\frac{B}{2},\frac{B}{2}-1,\frac{B}{2},\frac{B}{2}-1,\dots)$; that is, the BNAF starts with a leading 0 followed by a succession of the digits $(\frac{B}{2},\frac{B}{2}-1)$. We need to distinguish the cases of n even or n odd. If n is even then $\mathrm{BNAF}(k^*) = (0,\frac{B}{2},\frac{B}{2}-1,\frac{B}{2},\frac{B}{2}-1,\dots,\frac{B}{2},\frac{B}{2}-1)$, which corresponds to $k^* = \sum_{i=0}^{(n-2)/2}(\frac{B}{2}B+\frac{B}{2}-1)\,B^{2i} = \frac{1}{2}(B^n-1)+\frac{1}{2}\frac{B^n-1}{B+1} = \frac{1}{2}B^n\frac{B+2}{B+1} - \frac{B+2}{2(B+1)}$. If n is odd then $\mathrm{BNAF}(k^*) = (0,\frac{B}{2},\frac{B}{2}-1,\frac{B}{2},\frac{B}{2}-1,\dots,\frac{B}{2})$ and thus $k^* = \frac{B}{2}B^{n-1} + \sum_{i=0}^{(n-3)/2}((\frac{B}{2}-1)B+\frac{B}{2})\,B^{2i} = \frac{1}{2}(B^n-1)+\frac{1}{2}\frac{B^n+1}{B+1} = \frac{1}{2}B^n\frac{B+2}{B+1} - \frac{B}{2(B+1)}$. As a consequence, it follows that

$$\Pr[k'_n = 0] = \frac{\frac{1}{2}B^n\frac{B+2}{B+1} - \frac{B+2}{2(B+1)} + 1}{B^n} = \frac{1}{2}\frac{B+2}{B+1} + \frac{1}{2B^n} - \frac{1}{2B^n(B+1)}$$

when n is even, and

$$\Pr[k'_n = 0] = \frac{\frac{1}{2}B^n\frac{B+2}{B+1} - \frac{B}{2(B+1)} + 1}{B^n} = \frac{1}{2}\frac{B+2}{B+1} + \frac{1}{2B^n} + \frac{1}{2B^n(B+1)}$$

when n is odd. Because $k \in \{0, \ldots, B^n - 1\}$, k'_n must be 0 or 1. In turn, we have $\Pr[k'_n = 1] = 1 - \Pr[k'_n = 0] = \frac{1}{2}\frac{B}{B+1} - \frac{1}{2B^n} + \frac{1}{2B^n(B+1)}$ when n is even, and $\Pr[k'_n = 1] = \frac{1}{2}\frac{B}{B+1} - \frac{1}{2B^n} - \frac{1}{2B^n(B+1)}$ when n is odd. The BNAF definition also prohibits k'_{n-1} to be equal to $-\frac{B}{2}$ when B is even—otherwise we would have $k'_n \le 0$, which contradicts the range definition $0 \le k \le B^n - 1$. As a result, since $-\frac{B}{2} \equiv \frac{B}{2} \pmod{B}$, the proportion of $\frac{B}{2}$ for k'_{n-1} grows accordingly. We so have $\Pr\big[k'_{n-1} = -\frac{B}{2}\big] = 0$ and $\Pr\big[k'_{n-1} = \frac{B}{2}\big] = \big(\frac{1}{2(B+1)} - (-1)^n \frac{1}{2B^n(B+1)}\big) + \big(\frac{1}{2(B+1)} - (-1)^n \frac{1}{2B^n(B+1)}\big) = \frac{1}{B+1} - (-1)^n \frac{1}{B^n(B+1)}$.

The case of an odd radix B is easier to deal with. We immediately have $\Pr[k'_i = d] = \frac{1}{B}$ for any $d \in \{-\lfloor \frac{B}{2} \rfloor, \ldots, \lfloor \frac{B}{2} \rfloor\}$, $0 \le i \le n-1$. This clearly appears from Algorithm 2; see Remark 2. For the most significant digit k'_n, it can only take values in $\{0, 1\}$ due to range restrictions—recall that input integer $k \in \{0, \ldots, B^n - 1\}$. Specifically, we have $k'_n = 0$ if $k \in \{0, \ldots, \frac{B^n-1}{2}\}$ and $k'_n = 1$ if $k \in \{\frac{B^n+1}{2}, \ldots, B^n - 1\}$. We so get $\Pr[k'_n = 0] = \frac{(B^n+1)/2}{B^n}$ and $\Pr[k'_n = 1] = \frac{(B^n-1)/2}{B^n}$.

Putting it all together, we proved the following result.

Theorem 3. *Given a radix $B \ge 2$, let $(k'_n, \ldots, k'_0)$ represent the BNAF of an n-digit integer uniformly drawn at random in $\{0, \ldots, B-1\}^n$. Then*

- *for an even radix B:*

$$\Pr[k'_i = d \mid 0 \le i \le n-1] = \begin{cases} \frac{1}{B}\frac{B+2}{B+1} + (-1)^{i+1}\frac{1}{B^{i+1}(B+1)} & \text{if } d = 0 \\ \frac{1}{2(B+1)} - (-1)^{i+1}\frac{1}{2B^{i+1}(B+1)} & \text{if } d \in \{\pm\frac{B}{2}\} \text{ and } i \ne n-1 \\ \frac{1}{(B+1)} - (-1)^{n}\frac{1}{B^{n}(B+1)} & \text{if } d = \frac{B}{2} \text{ and } i = n-1 \\ 0 & \text{if } d = -\frac{B}{2} \text{ and } i = n-1 \\ \frac{1}{B} & \text{otherwise} \end{cases}$$

and

$$\Pr[k'_n = d] = \begin{cases} \frac{1}{2}\frac{B+2}{B+1} + \frac{1}{2B^n} + (-1)^{n+1}\frac{1}{2B^n(B+1)} & \text{if } d = 0 \\ \frac{1}{2}\frac{B}{B+1} - \frac{1}{2B^n} - (-1)^{n+1}\frac{1}{2B^n(B+1)} & \text{if } d = 1 \\ 0 & \text{otherwise} \end{cases} ;$$

- *for an odd radix B:*

$$\Pr[k'_i = d \mid 0 \le i \le n-1] = \frac{1}{B} \quad \text{and} \quad \Pr[k'_n = d] = \begin{cases} \frac{1}{2} + \frac{1}{2B^n} & \text{if } d = 0 \\ \frac{1}{2} - \frac{1}{2B^n} & \text{if } d = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $d \in \{-\lfloor \frac{B}{2} \rfloor, \ldots, \lfloor \frac{B}{2} \rfloor\}$. □

5 Extensions

The balanced non-adjacent forms can be extended to modular representations.

Definition 3. *Let $B \ge 2$ be a radix. If there exists a BNAF $(k'_{n-1}, \ldots, k'_0)$ such that*

$$k \equiv \sum_{i=0}^{n-1} k'_i B^i \pmod{B^n}$$

then $(k'_{n-1}, \ldots, k'_0)$ is called a BNAF modulo B^n *for k.*

BNAF representations modulo B^n apply to integers in $\mathbb{Z}/B^n\mathbb{Z}$. They equally apply to discretized torus elements in $B^{-n}\mathbb{Z}/\mathbb{Z}$ by noting that $B^{-n}\mathbb{Z}/\mathbb{Z} \cong \frac{\mathbb{Z}/B^n\mathbb{Z}}{B^n}$. Indeed, a torus element $\tau \in B^{-n}\mathbb{Z}/\mathbb{Z}$ can always be rewritten as $\tau = k \cdot B^{-n}$ where $k \equiv \sum_{i=0}^{n-1} k'_i\, B^i \pmod{B^n}$. It is interesting to note that modular BNAFs do not need one more digit in their encoding.

The next two theorems are straightforward generalizations of Theorem 1 and Theorem 2.

Theorem 4. *Every integer k has a BNAF modulo B^n. This BNAF, say $(k'_{n-1}, k'_{n-2}, \dots, k'_0)$, is unique—unless B is even and $k'_{n-1} \in \{\pm\frac{B}{2}\}$, in which case $(-k'_{n-1}, k'_{n-2}, \dots, k'_0)$ is also a BNAF modulo B^n for k.* □

Theorem 5. *If $(k'_{n-1}, \dots, k'_0)$ is a BNAF modulo B^n of an integer k then the Euclidean weight of $k \pmod{B^n}$ is equal to $\sum_{i=0}^{n-1} {k'_i}^2$.* □

If k is an integer in $\{0, \dots, B^n - 1\}$ and if $(k'_n, \dots, k'_0) \leftarrow \mathrm{BNAF}(k)$, Theorem 3 tells that $k'_n \in \{0, 1\}$. More precisely, a close inspection of the proof shows that $k'_n = 0$ whenever

- $0 \le k \le \frac{B^n-1}{2}$ if B is odd, or
- $0 \le k \le \frac{B^n(B+2)-B-1-(-1)^n}{2(B+1)}$ if B is even.

In all cases, we therefore have $k'_n = 0$ if $0 \le k \le \lfloor\frac{B^n}{2}\rfloor$. We define $\bar{k} = B^n - k$ and let $(\bar{k}'_n, \dots, \bar{k}'_0) \leftarrow \mathrm{BNAF}(\bar{k})$. If $\lfloor\frac{B}{2}\rfloor + 1 \le k \le B^n - 1$, we have by symmetry $\bar{k}'_n = 0$ since $0 \le B^n - k \le \lfloor\frac{B^n}{2}\rfloor$ for $\lfloor\frac{B}{2}\rfloor + 1 \le k \le B^n - 1$. Note also that $\mathrm{BNAF}(-\bar{k}) = -\mathrm{BNAF}(\bar{k})$. The BNAF modulo B^n of an integer k can therefore be defined as

$$\begin{cases} \mathrm{BNAF}(k \bmod B^n) & \text{if } k \bmod B^n \le \lfloor\frac{B^n}{2}\rfloor \\ \mathrm{BNAF}\big((k \bmod B^n) - B^n\big) & \text{otherwise} \end{cases} \tag{**}$$

where the BNAF is obtained as per Algorithm 2. This alternative definition makes the modular BNAF unique, except when B is even and $k \equiv \frac{B^n}{2} \bmod B^n$. In this latter case, there are two BNAFs modulo B^n: $\big(\frac{B}{2}, 0, \dots, 0\big)$ and $\big(-\frac{B}{2}, 0, \dots, 0\big)$.

Another benefit of the formulation (**) is that the distribution of the resulting BNAF digits is centered, provided that when B is even and $k \equiv \frac{B^n}{2} \pmod{B^n}$ one of the two forms $\big(\frac{B}{2}, 0, \dots, 0\big)$ or $\big(-\frac{B}{2}, 0, \dots, 0\big)$ is returned at random; see Theorem 6. The corresponding algorithm is detailed in Algorithm 3. Given an integer k and a power n, it outputs the BNAF modulo B^n of k. The algorithm makes use of an internal routine `random()` that returns a uniformly random bit.

Algorithm 3: Modular BNAF recoding

```
Input: Integer $k \neq 0$ and $n \ge 1$
Output: $(k'_{n-1}, \dots, k'_0)$ with $k'_i \in \{-\lfloor\frac{B}{2}\rfloor, \dots, \lfloor\frac{B}{2}\rfloor\}$ s.t. $\sum_{i=0}^{n-1} k'_i\, B^i \equiv k \pmod{B^n}$
$K \leftarrow k \bmod B^n$; $i \leftarrow 0$
if $(K > \lfloor\frac{B^n}{2}\rfloor) \vee \big((K = \lceil\frac{B^n}{2}\rceil) \wedge (\texttt{random}() = 1)\big)$ then
    $K \leftarrow K - B^n$
end if
while $(K \neq 0)$ do
    $k'_i \leftarrow K \bmod B$; $K \leftarrow (K - k'_i)/B$
    if $(k'_i > \lfloor\frac{B}{2}\rfloor) \vee \big((k'_i = \lceil\frac{B}{2}\rceil) \wedge ((K \bmod B) \ge \lfloor\frac{B}{2}\rfloor)\big)$ then
        $k'_i \leftarrow k'_i - B$; $K \leftarrow K + 1$
    end if
    $i \leftarrow i + 1$
end while
return $(k'_{i-1}, \dots, k'_0)$
```

We state exactly the occurrence probability of each digit in the modular BNAF produced by Algorithm 3. We also state their expectation and variance.

Theorem 6. *Given a radix $B \geq 2$ and a power n, let $(k'_{n-1}, \ldots, k'_0)$ represent the BNAF modulo B^n of an integer uniformly drawn at random in $\{0, \ldots, B^n - 1\}$ as per Algorithm 3. Then*

– *for an even radix B:*

$$\Pr[k'_i = d \mid 0 \leq i \leq n-1] = \begin{cases} \frac{1}{B}\frac{B+2}{B+1} + (-1)^{i+1}\frac{1}{B^{i+1}(B+1)} & \text{if } d = 0 \\ \frac{1}{2(B+1)} - (-1)^{i+1}\frac{1}{2B^{i+1}(B+1)} & \text{if } d \in \{\pm\frac{B}{2}\} \\ \frac{1}{B} & \text{otherwise} \end{cases};$$

– *for an odd radix B:*

$$\Pr[k'_i = d \mid 0 \leq i \leq n-1] = \frac{1}{B}$$

where $d \in \{-\lfloor\frac{B}{2}\rfloor, \ldots, \lfloor\frac{B}{2}\rfloor\}$.

Proof. The theorem is a direct consequence of Theorem 3 using (**) and noting that $k'_n = 0$. When B is even, the digits $\frac{B}{2}$ and $-\frac{B}{2}$ are equiprobable for k'_{n-1} because of the random choice for $k \equiv \frac{B^n}{2} \pmod{B^n}$. We so infer from Theorem 3 that $\Pr\big[k'_{n-1} = \frac{B}{2}\big] = \Pr\big[k'_{n-1} = -\frac{B}{2}\big] = \frac{1}{2(B+1)} - (-1)^n\frac{1}{2B^n(B+1)}$ when B is even. The other cases are immediate. □

Corollary 2. *Given a radix $B \geq 2$ and a power n, let $(k'_{n-1}, \ldots, k'_0)$ represent the BNAF modulo B^n of an integer uniformly drawn at random in $\{0, \ldots, B^n - 1\}$ as per Algorithm 3. Then any digit k'_i in the representation $(k'_{n-1}, \ldots, k'_0)$ satisfies*

$$\mathbb{E}[k'_i] = 0 \quad \text{and} \quad \mathrm{Var}(k'_i) = \begin{cases} \frac{1}{12}\left(B^2 - 1\right) & \text{if } B \text{ is odd} \\ \frac{1}{12}\frac{(B+2)(B^2-B+1)}{B+1} - \frac{(-1)^{i+1}}{4B^{i-1}(B+1)} & \text{if } B \text{ is even} \end{cases}.$$

Proof. The proof is analogous to that of Corollary 1. We have $\mathbb{E}[k'_i] = 0$ because the distribution is centered. For the variance, it then follows that $\mathrm{Var}(k'_i) = 2\sum_{d=1}^{\lfloor\frac{B}{2}\rfloor}\Pr[k'_i = d]\, d^2$. If B is odd, we get $\mathrm{Var}(k'_i) = 2\frac{1}{B}\sum_{d=1}^{\frac{B-1}{2}} d^2 = \frac{B^2-1}{12}$ and if B is even, $\mathrm{Var}(k'_i) = 2\frac{1}{B}\sum_{d=1}^{\frac{B}{2}-1} d^2 + 2\big(\frac{1}{2(B+1)} - (-1)^{i+1}\frac{1}{2B^{i+1}(B+1)}\big)(\frac{B}{2})^2 = \frac{1}{12}\frac{(B+2)(B^2-B+1)}{B+1} - \frac{(-1)^{i+1}}{4B^{i-1}(B+1)}$. □

6 Applications

As already mentioned in the introduction, a salient feature of lattice cryptosystems, notably those based on LWE [33] and its variants [37,24,5,23,8], is the presence of noise in the ciphertexts.

The noise can have different natures. It can be a parameter that is defined at setup time to guarantee a certain security level. It can be a quantity that evolves over time due to ciphertext evaluations as in fully homomorphic encryption. Finally, it can be algorithmic as the result of approximate computations or numerical errors.

Gadget decomposition. Informally, an LWE ciphertext $\boldsymbol{c}$ can be seen as a $(d+1)$-dimensional vector such that its dot product with the key $\boldsymbol{t} = (-\boldsymbol{s}, 1)$ (also seen as a $(d+1)$-dimensional vector) equals the input plaintext μ plus some small noise error e. The noise term is typically removed by rounding. We write $\boldsymbol{c} \leftarrow \mathsf{LWE}_{\boldsymbol{s}}(\mu) \in (\mathbb{Z}/q\mathbb{Z})^{d+1}$. Ciphertext $\boldsymbol{c}$ can be multiplied by a small scalar k to give an encryption of $k\mu$. In order to support multiplication by an arbitrary scalar k, the multiplier needs first to be decomposed.

We follow the presentation of [28]; see also [17, Section 3]. Given a radix B and a level ℓ, the so-called gadget vector is given by $\boldsymbol{g} = (1, B, \ldots, B^{\ell-1}) \in (\mathbb{Z}/q\mathbb{Z})^{\ell}$ so that for any vector $\boldsymbol{v} \in (\mathbb{Z}/q\mathbb{Z})^{\ell}$ the product

$\boldsymbol{v} \cdot \boldsymbol{g}^{\intercal}$ yields a scalar k in $\mathbb{Z}/q\mathbb{Z}$. We also consider the associated inverse transformation $g^{-1}\colon \mathbb{Z}/q\mathbb{Z} \to (\mathbb{Z}/q\mathbb{Z})^{\ell}$ such that for any scalar $k \in \mathbb{Z}/q\mathbb{Z}$, we have $g^{-1}(k) \cdot \boldsymbol{g}^{\intercal} = k$ and $g^{-1}(k)$ is "small". Explicitly, this inverse transformation replaces the input scalar by a (signed) radix-B expansion:

$$g^{-1}(k) = (k_0, \dots, k_{\ell-1}) \quad \text{such that } k \equiv \textstyle\sum_{i=0}^{\ell-1} k_i\, B^i \pmod{q} \ .$$

From the basic LWE encryption scheme, an "extended" encryption scheme $\widehat{\mathsf{LWE}}$ is built by LWE-encrypting individually each component of plaintext vector $\mu\, \boldsymbol{g}$: $\widehat{\mathsf{LWE}}_{\boldsymbol{s}}(\mu) \leftarrow (\mathsf{LWE}_{\boldsymbol{s}}(\mu), \mathsf{LWE}_{\boldsymbol{s}}(B\mu), \dots, \mathsf{LWE}_{\boldsymbol{s}}(B^{\ell-1}\mu))$. The LWE encryption of $k\,\mu$ can then be obtained from $\widehat{\mathsf{LWE}}_{\boldsymbol{s}}(\mu)$ as $\mathsf{LWE}_{\boldsymbol{s}}(k\mu) \leftarrow g^{-1}(k) \cdot \widehat{\mathsf{LWE}}_{\boldsymbol{s}}(\mu)^{\intercal}$.

For an LWE ciphertext $\boldsymbol{c} \leftarrow \mathsf{LWE}_{\boldsymbol{s}}(\mu)$, we let $\mathrm{Err}(\boldsymbol{c})$ denote the noise error present in $\boldsymbol{c}$. We can then write $\langle \boldsymbol{c}, \boldsymbol{t}\, \rangle = \mu + \mathrm{Err}(\boldsymbol{c})$, where $\boldsymbol{t} = (-\boldsymbol{s}, 1)$. We define $\boldsymbol{C}_i \coloneqq \mathsf{LWE}_{\boldsymbol{s}}(B^i\mu)$ and $\boldsymbol{\mathcal{C}}' \coloneqq g^{-1}(k) \cdot \widehat{\mathsf{LWE}}_{\boldsymbol{s}}(\mu)^{\intercal}$. We so obtain

$$\begin{aligned}
\langle \boldsymbol{\mathcal{C}}', \boldsymbol{t}\, \rangle &= \big\langle \textstyle\sum_{i=0}^{\ell-1} k_i\, \mathsf{LWE}_{\boldsymbol{s}}(B^i\mu), \boldsymbol{t}\, \big\rangle = \textstyle\sum_{i=0}^{\ell-1} k_i\, \langle \mathsf{LWE}_{\boldsymbol{s}}(B^i\mu), \boldsymbol{t}\, \rangle \\
&= \textstyle\sum_{i=0}^{\ell-1} k_i (B^i\, \mu + \mathrm{Err}(\boldsymbol{C}_i)) = \big(\textstyle\sum_{i=0}^{\ell-1} k_i B^i\big)\mu + \textstyle\sum_{i=0}^{\ell-1} \mathrm{Err}(\boldsymbol{C}_i)) \\
&= k\,\mu + \textstyle\sum_{i=0}^{\ell-1} k_i\, \mathrm{Err}(\boldsymbol{C}_i) \ .
\end{aligned}$$

The noise present in $\boldsymbol{\mathcal{C}}'$ only amounts to $\mathrm{Err}(\boldsymbol{\mathcal{C}}') = \sum_{i=0}^{\ell-1} k_i\, \mathrm{Err}(\boldsymbol{C}_i)$—this has to be compared with the noise $k\, \mathrm{Err}(\boldsymbol{c})$ present in $\boldsymbol{c}' \leftarrow k\, \mathsf{LWE}_{\boldsymbol{s}}(\mu)$. Hence, using the gadget decomposition, the error only grows logarithmically in q instead of linearly. Furthermore, the gadget decomposition can accommodate any digit expansion. As a consequence, selecting the BNAF for $g^{-1}(k)$ further improves the situation since the variance $\mathrm{Var}(\mathrm{Err}(\boldsymbol{\mathcal{C}}'))$ is then minimal.

We note that the gadget decomposition applies to all LWE-type encryption schemes. It also extends naturally to vectors and matrices as done for example in the GSW encryption scheme [15] for the multiplication of ciphertexts.

Key switching. LWE-type ciphertexts under a given key can be converted into ciphertexts under another key in different parameter sets thanks to a key switching procedure [6, § 1.2]. Its implementation requires key-switching keys: they essentially consist of an encryption of the key components of the original $\boldsymbol{s} = (s_1, \dots, s_d)$ with respect to the new key $\boldsymbol{s}'$. More precisely, using the previous notation, the d key switching keys are given $\mathsf{ksk}[j] \leftarrow \widehat{\mathsf{LWE}}_{\boldsymbol{s}'}(s_j)$, $1 \le j \le d$. An input LWE ciphertext $\boldsymbol{c} \leftarrow \mathsf{LWE}_{\boldsymbol{s}}(\mu) \coloneqq (a_1, \dots, a_d, b)$ is then turned into the ciphertext

$$\boldsymbol{c}' \leftarrow (0, \dots, 0, b) - \sum_{j=1}^{d} g^{-1}(a_j)\, \mathsf{ksk}[j] \ .$$

We are back to a setting similar to the previous one. Letting $g^{-1}(a_j) = (a_{j,0}, \dots, a_{j,\ell-1})$, $\mathsf{ksk}[j] = (\mathsf{ksk}[j]_0, \dots, \mathsf{ksk}[j]_{\ell-1})$ and $\boldsymbol{t}\,' = (-\boldsymbol{s}', 1)$, we can check that

$$\begin{aligned}
\langle \boldsymbol{c}', \boldsymbol{t}\,' \rangle &= \langle (0, \dots, 0, b), \boldsymbol{t}\,' \rangle - \textstyle\sum_{j=1}^{d} \langle g^{-1}(a_j)\, \mathsf{ksk}[j], \boldsymbol{t}\,' \rangle \\
&= b - \sum_{j=1}^{d} \Big(a_j\, s_j + \textstyle\sum_{i=0}^{\ell-1} a_{j,i}\, \mathrm{Err}(\mathsf{ksk}[j]_i) \Big) \\
&= \langle \boldsymbol{c}, \boldsymbol{t}\, \rangle - \sum_{j=1}^{d} \sum_{i=0}^{\ell-1} a_{j,i}\, \mathrm{Err}(\mathsf{ksk}[j]_i)\, ,
\end{aligned}$$

that is, an LWE encryption of μ under key $\boldsymbol{s}'$—provided that the noise keeps small.

Here too, it is crucial to adopt the BNAF representation for the gadget decomposition. The gain quickly becomes significant as the error not only is amplified by the number ℓ of levels but also by the dimension d of the input ciphertext (typically of the order of $10^3 \approx 2^{10}$ at a 128-bit security level).

Fast Fourier transform. Crandall and Fagin [11, § 3] empirically observe that when floating-point FFTs are employed, it is advantageous to make use of balanced representations. Indeed, they tend to reduce the convolution errors attendant to floating-point arithmetic, including those resulting from round-off errors. Algorithm 3 produces decompositions that are perfectly balanced: they on average have a zero mean; see Corollary 2. FFT techniques are well suited to module lattices as a way to reduce the computation time in lattice cryptography [26]. They for example play a central role in the fast bootstrapping procedure of FHEW [13] or of TFHE [9].

References

1. Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 297–314. Springer, 2014. doi:10.1007/978-3-662-44371-2_17.
2. Steven Arno and Ferrell S. Wheeler. Signed digit representations of minimal Hamming weight. *IEEE Transactions on Computers*, 42(8):1007–1110, 1993. doi:10.1109/12.238495.
3. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012. doi:10.1007/978-3-642-32009-5_50.
4. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory*, 6(3):13:1–13:36, 2014. Earlier version in ITCS 2012. doi:10.1145/2633600.
5. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584. ACM Press, 2013. doi:10.1145/2488608.2488680.
6. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing*, 43(2):831–871, 2014. doi:10.1137/120868669.
7. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, 2017. doi:10.1007/978-3-319-70694-8_15.
8. Jung Hee Cheon and Damien Stehlé. Fully homomophic encryption over the integers revisited. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 513–536. Springer, 2015. doi:10.1007/978-3-662-46800-5_20.
9. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020. doi:10.1007/s00145-019-09319-x.
10. W. Edwin Clark and J. J. Liang. On arithmetic weight for a general radix representation of integers. *IEEE Transactions on Information Theory*, 19(6):823–826, 1973. doi:10.1109/TIT.1973.1055100.
11. Richard Crandall and Barry Fagin. Discrete weighted transforms and large-integer arithmetic. *Mathematics of Computation*, 62(205):305–324, 1994. doi:10.1090/S0025-5718-1994-1185244-1.
12. Richard Crandall and Carl Pomerance. *Prime Numbers: A Computational Perspective*. Springer, 2001. doi:10.1007/978-1-4684-9316-0.
13. Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015. doi:10.1007/978-3-662-46800-5_24.
14. Craig Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010. doi:10.1145/1666420.1666444.
15. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013. doi:10.1007/978-3-642-40041-4_5.
16. Daniel M. Gordon. A survey of fast exponentiation methods. *Journal of Algorithms*, 27(1):129–146, 1998. doi:10.1006/jagm.1997.0913.
17. Shai Halevi. Homomorphic encryption. In Y. Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 219–276. Springer, 2017. doi:10.1007/978-3-319-57048-8_5.

18. David Jao, S. Ramesh Raju, and Ramarathnam Venkatesan. Digit set randomization in elliptic curve cryptography. In J. Hromkovič et al., editors, *Stochastic Algorithms: Foundations and Applications (SAGA 2007)*, volume 4665 of *Lecture Notes in Computer Science*, pages 105–117. Springer, 2007. doi:10.1007/978-3-540-74871-7_10.
19. Marc Joye and Sung-Ming Yen. Optimal left-to-right binary signed-digit exponent recoding. *IEEE Transactions on Computers*, 49(7):740–748, 2000. doi:10.1109/12.863044.
20. Marc Joye and Sung-Ming Yen. New minimal modified radix-r representation with applications to smart cards. In D. Naccache and P. Paillier, editors, *Public-Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 375–384. Springer, 2002. doi:10.1007/3-540-45664-3_27.
21. Neil Koblitz. CM-curves with good cryptographic protocols. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 279–287. Springer, 1992. doi:10.1007/3-540-46766-1_22.
22. Israel Koren. *Computer Arithmetic Algorithms.* A K Peters/CRC Press, 2nd edition, 2002. doi:10.1201/9781315275567.
23. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015. doi:10.1007/s10623-014-9938-4.
24. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60(6):43:1–43:35, 2013. doi:10.1145/2535925.
25. Willi Meier and Othmar Staffelbach. Efficient multiplication on certain nonsupersingular elliptic curves. In E. F. Brickell, editor, *Advances in Cryptology – CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 333–344. Springer, 1993. doi:10.1007/3-540-48071-4_24.
26. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. doi:10.1007/s00037-007-0234-9.
27. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012. doi:10.1007/978-3-642-29011-4_41.
28. Daniele Micciancio and Yuriy Polyakov. Bootstrapping in FHEW-like cryptosystems. Cryptology ePrint Archive, Report 2020/086, 2020. URL: https://ia.cr/2020/086.
29. François Morain and Jorge Olivos. Speeding up the computations on an elliptic curve using addition-subtraction chains. *RAIRO – Theoretical Informatics and Applications*, 24(6):531–543, 1990. doi:10.1051/ita/1990240605311.
30. James A. Muir and Douglas R. Stinson. Alternative digit sets for nonadjacent representations. In M. Matsui and R. J. Zuccherato, editors, *Selected Areas in Cryptography (SAC 2003)*, volume 3006 of *Lecture Notes in Computer Science*, pages 306–319. Springer, 2004. doi:10.1007/978-3-540-24654-1_22.
31. James A. Muir and Douglas R. Stinson. Minimality and other properties of the width-w nonadjacent form. *Mathematics of Computation*, 75(253):369–384, 2005. doi:10.1090/S0025-5718-05-01769-2.
32. Baodong Qin, Ming Li, Fanyu Kong, and Daxing Li. New left-to-right minimal weight signed-digit radix-r representation. *Computers & Electrical Engineering*, 35(1):150–158, 2008. doi:10.1016/j.compeleceng.2008.09.007.
33. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009. doi:10.1145/1568318.1568324.
34. George W. Reitwiesner. Binary arithmetic. *Advances in Computers*, 1:231–308, 1960. doi:10.1016/S0065-2458(08)60610-5.
35. Ronald L. Rivest, Len Adleman, and Michael L. Detouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, editors, *Foundations of Secure Computation*, pages 165–179. Academic Press, 1978. Available at https://people.csail.mit.edu/rivest/pubs.html#RAD78.
36. Jerome A. Solinas. Efficient arithmetic on Koblitz curves. *Designs, Codes and Cryptography*, 19(2/3):195–249, 2000. doi:10.1023/A:1008306223194.
37. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In M. Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009. doi:10.1007/978-3-642-10366-7_36.
38. Tsuyoshi Takagi, Sung-Ming Yen, and Bo-Ching Wu. Radix-r non-adjacent form. In K. Zhang and Y. Zheng, editors, *Information Security*, volume 3225 of *Lecture Notes in Computer Science*, pages 99–110. Springer, 2004. doi:10.1007/978-3-540-30144-8_9.
39. Jacobus H. van Lint. *Introduction to Coding Theory*, volume 86 of *Graduate Texts in Mathematics*. Springer, 3rd edition, 1999. doi:10.1007/978-3-642-58575-3.