ANALYSIS OF GENERALIZED CONTINUED FRACTION ALGORITHMS OVER POLYNOMIALS

VALÉRIE BERTHÉ, HITOSHI NAKADA, RIE NATSUI, AND BRIGITTE VALLÉE

Abstract. We study and compare natural generalizations of Euclid's algorithm for polynomials with coefficients in a finite field. This leads to gcd algorithms together with their associated continued fraction maps. The gcd algorithms act on triples of polynomials and rely on two-dimensional versions of the Brun, Jacobi–Perron and fully subtractive continued fraction maps, respectively. We first provide a unified framework for these algorithms and their associated continued fraction maps. We then analyse various costs for the gcd algorithms, including the number of iterations and two versions of the bit-complexity, corresponding to two representations of polynomials (the usual and the sparse one). We also study the associated two-dimensional continued fraction maps and prove the invariance and the ergodicity of the Haar measure. We deduce corresponding estimates for the costs of truncated trajectories under the action of these continued fraction maps, obtained thanks to their transfer operators, and we compare the two models (gcd algorithms and their associated continued fraction maps). Proving that the generating functions appear as dominant eigenvalues of the transfer operator allows indeed a fine comparison between the models.

Keywords. Gcd algorithms; continued fractions; Laurent formal power series; finite fields; Jacobi–Perron algorithm; Brun algorithm; fully subtractive algorithm; cost function; bit-complexity; analytic combinatorics; generating function; Gaussian law; dynamical analysis.

Contents

Date: March 15, 2021.

This work was supported by the Agence Nationale de la Recherche through the project Codys (ANR-18-CE40-0007). The research of the second author was supported in part by Grant-in Aid for Scientific research (No. 20K03661) of Japan Society for the Promotion of Science (JSPS). The research of the third author was supported in part by Grant-in Aid for Scientific research (No. 20K03642) of JSPS.

1. Introduction

Gcd computation for (univariate) polynomials (with coefficients in a finite field) is a basic operation in computer algebra. This is the main tool for keeping a polynomial fraction under its irreducible form, but it also plays a central role in polynomial factorization for instance (see [14]). Euclid's algorithm completely solves the problem of gcd computation for two entries. However, there does not exist a canonical generalization of Euclid's algorithm when working with at least three entries. For the general case of d entries, one of the most natural and basic algorithms consists in performing a succession of $d-1$ phases, with each of them being the Euclid algorithm on two entries. It is described in Knuth's book [19] and its probabilistic study has been performed in [5].

1.1. **A unified framework for the three algorithms of interest.** Among all possible generalizations, we choose and compare three generalized Euclidean algorithms, inspired by classical multidimensional continued fraction maps, namely the Jacobi–Perron (JP), the Brun and the fully subtractive (FS) maps. These algorithms and their associated continued fraction maps are well studied in the number case; see for instance [30] and also [6] for the probabilistic analysis of the Brun map. They also have been studied in the present framework of formal power series in the case of the Jacobi–Perron algorithm [26, 27, 10, 16] and of the Brun algorithm [3, 17, 6], mostly for proving their convergence and for establishing the invariance of the Haar measure.

We introduce a unified framework for these three algorithms, first when they deal with triples of polynomials. There exist several possible realizations for these algorithms that work, for instance, with ordered or non-ordered inputs, and there is no real canonical way to define them. Here we have chosen a common model for the three algorithms, that may slightly differ from the previously studied forms, defined by working with a common set of inputs for the three algorithms. This common set is the set of triples $R := (R_1, R_2, R_3)$ of polynomials with coefficients in a finite field for which the two following conditions hold: R_3 is monic and $\deg R_3 > \max(\deg R_1, \deg R_2)$. The algorithms moreover share the same type of strategy, based on the role that is given to a specific component of R, which depends on the algorithm. This is the first component for the Jacobi–Perron algorithm, the second largest component for the Brun algorithm, and the

smallest component for the fully subtractive algorithm. The algorithm then divides the two other components by this specific component, and replaces these components by their remainders in the division by the specific component. After these two divisions, this specific component becomes the largest one, and it is thus placed at the third position.

This general description can then be extended as continued fractions when the algorithms deal with the set $\mathbb{L}^2$, where $\mathbb{L}$ denotes the set of Laurent formal power series with negative degree.

1.2. **The two models.** There are thus two models of interest, namely a discrete model (polynomials) and a continuous model (Laurent formal power series). The discrete model is defined by sets of triples of polynomials with fixed maximal degree, endowed with the uniform probability; we then study the probabilistic behaviour of costs of interest defined on this finite set, when the maximal degree of the triple of polynomials tends to infinity. The continuous model deals with the set $\mathbb{L}^2$ endowed with its Haar measure. We study costs that are defined on truncated continued fraction expansions of a series in $\mathbb{L}^2$ at depth n. We also consider in this case almost everywhere behaviour when the truncation degree n tends to infinity.

There are two types of costs under study. The first ones are "additive": these are the number of steps, the total number of divisions, the total possible number of monomials and the total number of non-zero monomials in the sequence of quotients. The other ones are related to two versions of the bit-complexity. We study these costs both in the discrete case (gcd algorithms on polynomials) and continuous case (continued fraction maps on Laurent formal power series) and compare their behaviours.

1.3. **Results.** In the one-dimensional case, the mean and the variance of many natural costs (for instance the number of steps) for the Euclidean algorithm acting on polynomials are known to be linear with respect to the maximal degree (see [11, 20]), and their distribution to be binomial [13, 18]. Furthermore, the usual bit-complexities have a mean and a variance that are quadratic with respect to the maximal degree and obey an asymptotic Gaussian law [20, 7]. These results, on polynomials, are obtained with tools of analytic combinatorics, mainly generating functions (see [11]). Furthermore, there is a close connection between the probabilistic behaviour for costs in the discrete model (polynomials) and in the continuous model (truncated trajectories under the action of the Gauss Artin map): Executions of the Euclidean algorithm on polynomials behave on average similarly to the way truncated trajectories behave almost everywhere. All these results parallel the results obtained for integers (see [2]), as highlighted in [32].

In the two-dimensional case, there are already results that have been obtained only in the continuous model, and in the case of two algorithms: the Jacobi–Perron algorithm [26, 27, 10, 16] and the Brun algorithm [3, 17, 6]. These articles study their convergence and their invariant measure.

The present paper provides a complete extension of the previous results to the two-dimensional case (except for bit-complexities where only average estimates are obtained), with methods that extend the previous approaches into the unified framework developed here. In particular, we make extensive use of the transfer operator underlying the system, allowing a fine comparison between the two models and a more transparent proof of the invariance of the Haar measure in the continuous model. However, technical difficulties arise: Firstly, even though there is a common framework for the three extended algorithms, their specificities must be highlighted and handled. Secondly, higher dimensions lead to a more complex analysis, where the singularities of the generating functions are more delicate to deal with. Asymptotic Gaussian laws for additive costs are proven in Theorems 1 and 3, whereas specific values of expectations and variances are exhibited in Tables 6 and 8. The bit-complexities are studied (only on average) in Propositions 7 and 10.

1.4. **Plan of the paper.** Section 2 introduces a general framework which is common for the three algorithms of interest and their associated continued fraction maps. It then discusses the convergence of the continued fraction maps. Section 3 provides a general introduction to the paper with a focus on analytic combinatorics. It describes the costs, the two models, the methodology that is used in each model, and finally the main results in an informal way. Section 4 is devoted

to the analysis of the three gcd algorithms in the discrete model. The main tool is here analytic combinatorics, and more precisely, bivariate generating functions, that enable in particular to exhibit asymptotic Gaussian laws. Section 5 considers costs for truncated trajectories (in the continuous model): it first describes the main properties of the associated continued fraction maps, introduces the transfer operator, and relates the (bi-variate) transfer operator to the bi-variate generating functions of Section 4.

1.5. **Notation.** Let q be a fixed power of a prime number p; we respectively let denote by

$$\mathbb{F}_q, \quad \mathbb{F}_q[X], \quad \mathbb{F}_q(X), \quad \mathbb{F}_q((X^{-1})), \quad \mathbb{L}$$

the finite field of cardinality q, the ring of polynomials with coefficients in $\mathbb{F}_q$, its fraction field, the field of Laurent series in X^{-1} with coefficients in $\mathbb{F}_q$, and the subset of Laurent series of negative degree, respectively. As usual, for $f \in \mathbb{F}_q((X^{-1}))$ with $f \neq 0$ of the form

$$f = \sum_{p \leq n} a_p X^p = a_n X^n + a_{n-1} X^{n-1} + \cdots, \qquad \text{with } a_n \neq 0,$$

the degree $\deg f$ of f, the absolute value $||f||$ of f, and the polynomial part $[f]$ of f are defined as

$$\deg f = n, \qquad ||f|| = q^{\deg f}, \qquad [f] = \sum_{0 \leq p \leq n} a_p X^p. \tag{1}$$

For $f = 0$, we define (as usual) $\deg 0 = -\infty$, $||0|| = 0$, and $[0] = 0$. We also denote by μ the Haar probability measure on $\mathbb{L}$.

Acknowledgements. We are very grateful to the referee for his constructive comments and his stimulating questions that helped us to improve this paper.

2. The three algorithms and their associated continued fraction maps

We wish to describe natural generalizations of the classical Euclid algorithm acting on triples of polynomials, together with their associated continued fraction maps that act on pairs of elements of $\mathbb{L}$. We consider here three algorithms, namely the Jacobi–Perron (JP), the Brun, and the fully subtractive (FS) algorithms. Two remarks should be made here: since there is no canonical standardized version for these algorithms, we choose versions that may enter in a precise and unified framework by being defined on a common set of inputs $\mathcal{R}$ described in (6). In particular, they do not exactly coincide with the versions of the Brun algorithm from [3, 17]. Furthermore, as it is not immediate to properly define additive versions (that use only additions and *not* multiplications or divisions) in the present context of formal power series with coefficients in a finite field, we only consider *multiplicative* versions[1].

We first recall the classical case (the usual Euclid algorithm) in Section 2.1. Then, in Sections 2.2 and 2.3, we announce a common framework for the three algorithms of interest, and their associated continued fraction maps. We describe in Section 2.4 the specific parameters with which each algorithm enters the general framework. The proof of this result is obtained via a precise description of each algorithm and their associated continued fraction map, done in the following three sections (Sections 2.5, 2.6, 2.7). We prove in Section 2.8 the convergence of the convergents for the three algorithms.

2.1. **The classical Euclid algorithm and continued fractions.** We first recall the classical Euclid algorithm on two polynomials, and its associated continued fraction map.

Gcd on polynomials. We consider the two sets $\underline{\mathcal{P}}$ and $\mathcal{P}$ defined as[2]

$$\begin{cases} \underline{\mathcal{P}} &= \{R = (R_1, R_2) \in \mathbb{F}_q^2[X] \mid \deg R_1 < \deg R_2\} \\ \mathcal{P} &= \{R = (R_1, R_2) \in \mathbb{F}_q^2[X] \mid \deg R_1 < \deg R_2,\ R_2 \text{ monic}\}. \end{cases} \tag{2}$$

The second one can be viewed as the projective space of the first one, namely $\mathcal{P} = \Pi(\underline{\mathcal{P}})$.

[1] This question is discussed in more details in [7].

[2] With the previous convention $\deg 0 = -\infty$, these sets contain pairs $(0, R_2)$ associated with any non-zero polynomial (possibly monic) R_2.

The Euclidean division defines a map $\mathrm{U}_G : \underline{\mathcal{P}} \to \underline{\mathcal{P}}$. On an input $R \in \underline{\mathcal{P}}$, the Euclidean division of R_2 by R_1 defines the *quotient* A and the *remainder* R_3 such that

$$R_2 = AR_1 + R_3, \quad A = \left[\frac{R_2}{R_1}\right], \quad R_3 = 0 \quad \text{or} \quad \deg R_3 < \deg R_1\,,$$

and leads to a new pair $\widehat{R} = (\widehat{R}_1, \widehat{R}_2) \in \underline{\mathcal{P}}$ defined as $\widehat{R}_1 = R_3$, $\widehat{R}_2 = R_1$ that satisfies, in matricial notation[3]

$$R = M(A)\,\widehat{R} \qquad \text{with} \quad M(A) := \begin{pmatrix} 0 & 1 \\ 1 & A \end{pmatrix}.$$

The map $R \mapsto \widehat{R}$ thus defines a map $\mathrm{U}_G : \underline{\mathcal{P}} \to \underline{\mathcal{P}}$. Any element $R' \in \underline{\mathcal{P}}$ with $R' = \lambda R$ and $\lambda \in \mathbb{F}_q^\star$ gives rise to the same matrix $M(A)$ as R; this matrix thus only depends on the class of R in the projective space $\Pi(\underline{\mathcal{P}}) = \mathcal{P}$. Then the map previously defined on $\underline{\mathcal{P}}$ may also be defined in $\mathcal{P}$.

The Euclid algorithm on $\mathcal{P}$ builds a finite sequence of quotients $(A_k)_k$, and a finite sequence $(\widehat{R}_k)_k$. It ends as soon as $R_k \in \mathcal{P}$ has its first component equal to zero. The second one provides the monic gcd of (R_1, R_2).

Then, the set $\mathcal{G}$ of possible quotients and the set $\mathcal{U}$ of possible gcd's are

$$\mathcal{G} = \{A \in \mathbb{F}_q[X] \mid \deg A \geq 1\}, \qquad \mathcal{U} = \{R \in \mathbb{F}_q[X] \mid R \text{ is monic}\}\,, \tag{3}$$

and the Euclid algorithm thus yields the following decomposition for the set $\mathcal{P}$

$$\mathcal{P} \sim \mathtt{Seq}(\mathcal{G}) \times \mathcal{U}, \tag{4}$$

where $\mathtt{Seq}(\mathcal{G})$ stands for the set of finite sequences of elements of $\mathcal{G}$. This is the starting point for the probabilistic study of the Euclid algorithm on polynomials, based on analytic combinatorics, see e.g. [11, 20, 32, 7] and Section 3.

Continued fraction map on $\mathbb{L}$**.** The Gauss Artin continued fraction map $\mathrm{T}_G : \mathbb{L} \to \mathbb{L}$ provides an extension to $\mathbb{L}$ of the projective version of the Euclid Algorithm. It is defined via the polynomial part function $[\cdot]$ defined in (1) as

$$\mathrm{T}_G(f) = \frac{1}{f} - \left[\frac{1}{f}\right], \quad \text{for } f \neq 0, \qquad \mathrm{T}_G(0) = 0\,. \tag{5}$$

This is the polynomial counterpart of the classical Gauss map. The trajectory of f under the action of T_G is defined as $(f, \mathrm{T}_G(f), \mathrm{T}_G^2(f), \cdots, T_G^n(f), \dots)$. It builds a sequence $(A_k)_k$ of quotients and a sequence $(M(A_k))_k$ of matrices with

$$A_k := \left[\frac{1}{\mathrm{T}_G^{k-1}(f)}\right], \qquad M(A_k) \,=\, \begin{pmatrix} 0 & 1 \\ 1 & A_k \end{pmatrix},$$

and produces the continued fraction expansion

$$f = \frac{1}{|\,A_1} + \frac{1}{|\,A_2} + \cdots\,.$$

This also defines the truncated trajectory at depth n, namely $(f, \mathrm{T}_G(f), \mathrm{T}_G^2(f), \cdots, \mathrm{T}_G^n(f))$, and the sequence

$$\begin{pmatrix} P_n \\ Q_n \end{pmatrix} := M(A_1)M(A_2)\cdots M(A_n)\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \text{with} \quad M(A) \,=\, \begin{pmatrix} 0 & 1 \\ 1 & A \end{pmatrix}.$$

The quotient $P_n/Q_n \in \mathbb{F}_q(X)$ is the n-th convergent of $f \in \mathbb{L}$.

Gcd and continued fraction map. The Gauss Artin continued fraction map provides an extension of the gcd algorithm on two polynomials. Indeed, when T_G is applied to an element $f = P/Q \in \mathbb{F}_q(X)$ with $\deg Q > \deg P$, the trajectory of f under the map T_G, namely $(f, \mathrm{T}_G(f), \cdots, \mathrm{T}_G^n(f), \cdots)$, thus reaches 0 after a finite number of steps. It produces the same sequence of quotients as the gcd algorithm on the pair (P, Q).

[3] Pairs (or triples) of polynomials are considered here as column vectors.

In the next two sections, we first draw a common framework, firstly for polynomials in Section 2.2, and secondly for continued fraction maps in Section 2.3. Then, Section 2.4 describes how the three algorithms of interest and their associated continued fraction maps actually enter this common framework. The convergence of these algorithms is discussed in Section 2.8.

2.2. **A common framework for generalized Euclid's algorithms.** We describe a common framework for polynomials.

Set of inputs. We consider the following two sets as possible sets of inputs for these algorithms:

$$\text{(6)} \qquad \begin{cases} \underline{\mathcal{R}} & = \{R := (R_1, R_2, R_3) \mid \deg R_3 > \max(\deg R_1, \deg R_2)\}, \\ \mathcal{R} & = \{R := (R_1, R_2, R_3) \mid \deg R_3 > \max(\deg R_1, \deg R_2), \quad R_3 \text{ monic }\}. \end{cases}$$

Here again, the second one can be viewed as the projective space of the first one, namely $\mathcal{R} = \Pi(\underline{\mathcal{R}})$. Let us stress the fact that the sets $\underline{\mathcal{R}}$ or $\mathcal{R}$ are *not* completely ordered. The algorithms that are of interest here deal with *not completely ordered* sets of inputs. Only the last component is the largest one, and the other components R_1 and R_2 are *not a priori* ordered. It is then useful to consider their possible order and to define the following two subsets of $\underline{\mathcal{R}}$ (and their analogs in $\mathcal{R}$), namely

$$\text{(7)} \qquad \underline{\mathcal{R}}_+ := \{R \in \underline{\mathcal{R}} \mid \deg R_1 \geq \deg R_2\}, \qquad \underline{\mathcal{R}}_- := \{R \in \underline{\mathcal{R}} \mid \deg R_1 < \deg R_2\},$$

together with the function η, called the *order sign*, that indicates the position of the element with largest degree, defined as

$$\text{(8)} \qquad \eta : \underline{\mathcal{R}} \to \{-1, +1\}, \qquad \eta(R) = +1 \Longleftrightarrow R \in \underline{\mathcal{R}}_+ .$$

If we were dealing with completely ordered sets, the function η would be constant, and thus useless.

Strategy. Each algorithm deals with a *specific* component of R. As the main operation will always be a division of the non-specific components by the specific one, the specific component will never be chosen as the component R_3 of largest degree. Then, the specific component will be always chosen amongst R_1 or R_2 and the strategy of the algorithm is described via a sign ϵ, that is called the *strategic sign*, and is defined by

$$\epsilon \in \{-1, +1\}, \qquad \epsilon(R) = +1 \Longleftrightarrow \big(R_1 \text{ is the specific component}\big).$$

This gives the following natural strategies:

- (a) For *positional* algorithms, the strategic sign ϵ depends only on the position, and not on the order sign η: the algorithm chooses, as its specific component, either always R_1 (with $\epsilon = +1 = \eta^2$), or always R_2 (with $\epsilon = -1 = -\eta^2$). The choice $\epsilon = +1 = \eta^2$ is indeed done by the Jacobi-Perron algorithm. It is natural to introduce another algorithm, called the *opposite* Jacobi–Perron algorithm, which always chooses R_2 as the specific component ($\epsilon = -1 = -\eta^2$). It plays a crucial role in the proof of Proposition 2.
- (b) For the other algorithms, the strategic sign ϵ depends on the order sign η which describes the ordering between R_1 and R_2. The natural choices are then
  - $\epsilon = +\eta$, performed by the Brun algorithm[4]. The Brun algorithm chooses, as the specific component, the second largest component. As the largest component is always R_3, the second largest component[5] is the component R_i for which $\deg R_i = \max(\deg R_1, \deg R_2)$. Then, with the definition of η, $\epsilon = \eta$ for the Brun algorithm.
  - The second choice is $\epsilon = -\eta$, performed by the fully subtractive algorithm. The fully subtractive algorithm chooses, as the specific component, the smallest component and performs the division of the other two components by the specific component. As the largest component is always R_3, the smallest component is the component[6] R_i

[4]In the literature, the Brun algorithm does not perform the division of all the components by the specific component (the second largest one). We consider here a variant of the Brun algorithm which performs all the divisions. However, this is a mild modification. Indeed, one does not create new partial quotients when performing the division of the smallest entry by the second largest one: only the case of equality of the degrees creates a constant term.

[5]There is a convention to be taken in case of equality of the degrees.

[6]There is again a convention to be taken in case of equality of the degrees.

for which $\deg R_i = \min(\deg R_1, \deg R_2)$. Then, with the definition of η, the equality $\epsilon = -\eta$ holds for the fully subtractive algorithm.

If we had considered completely ordered sets of inputs, there would have been only two possible strategies, defined by the two possible choices: $\epsilon = \pm 1$. However, the description of each algorithm would be different and would not fall within the framework described in the present paper (see also the discussion in Section 6).

Two phases. There are two phases in the algorithm: the *non-degenerate phase*, followed by the *degenerate phase* (that may be empty). One enters the degenerate phase as soon as the specific component becomes zero.

Non-degenerate phase. It performs a sequence of steps, each of them being defined by a *linear* mapping acting on $\underline{\mathcal{R}}$ (see (6)) as

$$\mathrm{U} : \underline{\mathcal{R}} \to \underline{\mathcal{R}}\,. \tag{9}$$

On the input $R \in \underline{\mathcal{R}}$, the map U outputs $\widehat{R} \in \underline{\mathcal{R}}$ after the following operations:

- The map U divides each non-specific component of a triple by the specific component, and gives rise to a pair (A, B) of quotients that satisfies either $\deg B > \deg A \geq 0$ or $A = 0$. The quotient pair (A, B) involves integer parts of quotients of components.
- The two cases (according to $\epsilon = \pm 1$) yield pairs $(A, B) = (A_\epsilon(R), B_\epsilon(R))$ satisfying

$$A_+(R) = \left[\frac{R_2}{R_1}\right], \quad B_+(R) = \left[\frac{R_3}{R_1}\right], \qquad A_-(R) = \left[\frac{R_1}{R_2}\right], \quad B_-(R) = \left[\frac{R_3}{R_2}\right]. \tag{10}$$

  The algorithm then replaces each non-specific component by its remainder in the associated division.
- After these divisions, the specific component, when it is non-zero, becomes the largest one (in terms of degrees). By applying possibly a permutation, it is placed at the third position. The result of the divisions by a quotient pair (A, B), followed by this possible permutation, defines a matrix $M_\epsilon(A, B)$, together with its inverse $N_\epsilon(A, B)$, with

$$M_+(A,B) := \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & A \\ 0 & 1 & B \end{pmatrix} \quad N_+(A,B) := \begin{pmatrix} -A & 1 & 0 \\ -B & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad (A,B) = (A_+, B_+)\,, \tag{11}$$

$$M_-(A,B) := \begin{pmatrix} 0 & 1 & A \\ 0 & 0 & 1 \\ 1 & 0 & B \end{pmatrix} \quad N_-(A,B) := \begin{pmatrix} 0 & -B & 1 \\ 1 & -A & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad (A,B) = (A_-, B_-)\,. \tag{12}$$

  There is again an important remark. The matrices *partially* re-order the triple R, but they do not aim to completely re-order it. We indeed wish to stay inside $\underline{\mathcal{R}}$, which is not completely ordered. We return to this question in the conclusion of the paper.
- In summary, for any $R \in \underline{\mathcal{R}}$, there exists a triple $(\epsilon, A_\epsilon, B_\epsilon)$ for which the (old) R is written in terms of the (new) $\widehat{R}$ as $R = M_\epsilon(A, B)\,\widehat{R}$, and the (new) $\widehat{R} = U(R)$ is written in terms of the (old) R as $\widehat{R} = N_\epsilon(A, B)\,R$.

Set of quotients in the non-degenerate phase. Each step of the non-degenerate phase uses the map U and builds a triple pair (ϵ, A, B) formed with a sign ϵ and a pair (A, B) of quotients. This set of possible quotients, denoted as $\mathcal{H}$, decomposes into two subsets, depending on whether the component A is zero or not. When $A = 0$, there is only one division, whereas there are two divisions for $A \neq 0$. This is why the two subsets are denoted as $\mathcal{H}^{\langle i \rangle}$, where the index i refers to the number of divisions that are actually performed. The sets $\mathcal{H}^{\langle i \rangle}$ depend on the algorithm, but the following always holds

$$\mathcal{H} = \mathcal{H}^{\langle 1 \rangle} \cup \mathcal{H}^{\langle 2 \rangle}, \qquad \mathcal{H}^{\langle 1 \rangle} \subset \{0\} \times \mathcal{G}, \qquad \mathcal{H}^{\langle 2 \rangle} \subset \mathcal{L}\,, \tag{13}$$

$$\text{with} \quad \mathcal{G} = \{B \in \mathbb{F}_q[X] \mid \deg B \geq 1\}, \qquad \mathcal{L} := \{(A, B) \in \mathbb{F}_q[X]^2 \mid \ \deg B > \deg A \geq 0\}\,. \tag{14}$$

Role of $\mathcal{R}$ versus $\underline{\mathcal{R}}$. As in the classical Euclid algorithm, the triple $(\epsilon, A_\epsilon, B_\epsilon)$ is the same for any input $R' = \lambda R$ with $\lambda \in \mathbb{F}_q^\star$. It only depends on the equivalence class of R in the projective space $\Pi(\underline{\mathcal{R}}) = \mathcal{R}$. Thus, the mapping $\mathrm{U} : \underline{\mathcal{R}} \to \underline{\mathcal{R}}$ defines a mapping, also called U, which now acts on $\Pi(\underline{\mathcal{R}}) = \mathcal{R}$. This mapping $\mathrm{U} : \mathcal{R} \to \mathcal{R}$ admits as inverse branches the linear mappings $M_\epsilon(A, B)$, for $(\epsilon, A, B) \in \mathcal{H}$.

As soon as the specific component becomes zero, one enters the degenerate phase, with a subset $\mathcal{D} \subset \mathcal{R}$. The subset $\mathcal{D}$ is formed with pairs of polynomials (instead of triples), and the algorithm acts on $\mathcal{D}$ as the usual Euclid algorithm. It stops when there remains only one non-zero component that is monic and that is the gcd of the triple R.

The non-degenerate phase of the algorithm finally entails the decomposition

$$\mathcal{R} = \mathtt{Seq}(\mathcal{H}) \times \mathcal{D}\,. \tag{15}$$

2.3. **A common framework for generalized continued fraction maps.** We observe that the triple (ϵ, A, B) only depends on the pair $(R_1/R_3, R_2/R_3)$ that belongs to $(\mathbb{L} \cap \mathbb{F}_q(X))^2$. It is then natural to:

- first, consider the projective version of the map U,
- second, extend this projective version to $\mathbb{L}^2$.

The "extending projective" process gives rise to the continued fraction map T, which admits as branches the homographies that are the projective counterparts of matrices $N_\epsilon(A, B)$. As the two components of $f = (f_1, f_2) \in \mathbb{L}^2$ are not a priori ordered, it will be useful to consider the following two subsets of $\mathbb{L}^2$, namely

$$\mathbb{L}^2_+ := \{(f_1, f_2) \in \mathbb{L}^2 \mid \deg f_1 \geq \deg f_2\}, \qquad \mathbb{L}^2_- := \{(f_1, f_2) \in \mathbb{L}^2 \mid \deg f_1 < \deg f_2\}, \tag{16}$$

and to use the function

$$\eta : \mathbb{L}^2 \to \{-1, +1\} \qquad \text{such that} \quad \eta(f_1, f_2) = +1 \quad \text{if and only} \quad (f_1, f_2) \in \mathbb{L}^2_+\,. \tag{17}$$

Each continued fraction map deals with a function ϵ which is defined in terms of η via the strategy of the algorithm (as previously for the case of polynomials). The value $\mathrm{T}(f)$ then depends on $\epsilon(f)$ and is defined via quotients (A, B) which depend on the sign ϵ and $f = (f_1, f_2)$, i.e.,

$$A_+(f) = \left[\frac{f_2}{f_1}\right], \quad B_+(f) = \left[\frac{1}{f_1}\right], \qquad A_-(f) = \left[\frac{f_1}{f_2}\right], \quad B_-(f) = \left[\frac{1}{f_2}\right], \tag{18}$$

and the continued fraction map T associates with the pair $f = (f_1, f_2)$

$$\mathrm{T}(f) \;=\; \begin{cases} \left(\dfrac{f_2}{f_1} - A_+(f),\, \dfrac{1}{f_1} - B_+(f)\right) & \text{if } \epsilon(f) = +1 \\ \left(\dfrac{1}{f_2} - B_-(f),\, \dfrac{f_1}{f_2} - A_-(f)\right) & \text{if } \epsilon(f) = -1. \end{cases}$$

With any quotient $(\epsilon, A, B) \in \mathcal{H}$, we associate the homography $h_{(\epsilon,A,B)}$ related to the matrix $M_\epsilon(A, B)$. The fundamental set relative to this quotient is defined as

$$\mathbb{L}^2_{(\epsilon,A,B)} := \{f = (f_1, f_2) \mid (\epsilon, A, B)(f) = (\epsilon, A, B)\} = h_{(\epsilon,A,B)}(\mathbb{L}^2)\,,$$

and the restriction of T to each $\mathbb{L}^2_{\epsilon,A,B}$ is a surjective homography $\mathbb{L}^2_{\epsilon,A,B} \to \mathbb{L}^2$ associated with the matrix $N_\epsilon(A, B)$. Moreover, the following equality holds:

$$\bigcup_{(\epsilon,A,B)\in\mathcal{H}} \mathbb{L}^2_{\epsilon,A,B} = \mathbb{L}^2\,.$$

This continued fraction map T provides (by construction) an extension of the projective version of the linear map U used in *non-degenerate phase* of the gcd algorithm. When T is applied to a pair

$$(f_1, f_2) \in \mathbb{F}_q(X)^2, \text{ with } f_1 = R_1/R_3,\ f_2 = R_2/R_3 \text{ and } R := (R_1, R_2, R_3) \in \mathcal{R},$$

the trajectory of (f_1, f_2) under the map T reaches, after a finite number of steps, a point (g_1, g_2) where at least one component g_1 or g_2 is zero. It produces the same sequence of quotients as the non-degenerate phase of the gcd algorithm on the triple $R = (R_1, R_2, R_3)$.

Degenerate phase. The gcd algorithm enters the degenerate phase (at a step of index n_0) as soon as there exists a zero component in $R = (R_1, R_2, R_3)$. Then, from index n_0 the gcd algorithm is now the usual gcd algorithm (on two entries). Equivalently, for the continued fraction map T on the input f, this means that $\mathrm{T}^{n_0}(f)$ is either of the form $(g_1, 0)$ (case (a)) or $(0, g_2)$ (case (b)). Then, from the index n_0, the continued fraction map coincides with the classical continued fraction map, and the sequence of quotients $(B_n)_{n>0}$ is the sequence of partial quotients in the continued fraction expansion of g_2 (case (a)) or g_1 (case (b)). The matrices used by the (degenerate) continued fraction map are

$$M_b(B) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & B \end{pmatrix} \quad [\text{case } (b)]; \qquad M_a(B) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & B \end{pmatrix} \quad [\text{case } (a)] \ .$$

Convergents. Consider an element $f = (f_1, f_2) \in \mathbb{L}^2$. There are two different cases, depending on whether the orbit $\mathrm{T}^n(f)$ enters the degenerate phase or not.

(i) The orbit $\mathrm{T}^n(f)$ stays in the non-degenerate phase. Let $M_{\epsilon_k}(A_k, B_k)$ be the matrix that is used in the k-th iteration of the non-degenerate continued fraction map T on the input f, for $k \geq 1$. The product $M_{[1..n]}$ of all the matrices used in the first n steps is thus

$$M_{[1..n]} = M_{\varepsilon_1}(A_1, B_1) M_{\varepsilon_2}(A_2, B_2) \cdots M_{\varepsilon_n}(A_n, B_n) \, . \tag{19}$$

(ii) The orbit $\mathrm{T}^n(f)$ enters the degenerate phase at a step of index n_0. This means that $\mathrm{T}^{n_0}(f)$ equals either $(0, g_2)$ (case (a)) or $(g_1, 0)$ (case (b)). Then, the algorithm continues with the classical Euclidean algorithm and the sequence $(B_{n_0+k})_{k\geq 1}$ of quotients is the sequence of partial quotients in the continued fraction expansion of g_2 or g_1. The matrix produced by each step of the (degenerate) continued fraction map is thus

$$M_a(B) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & B \end{pmatrix}, \qquad \text{or} \quad M_b(B) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & B \end{pmatrix} .$$

The product $M_{[1..n]}$ of all the matrices used in the first n steps, is thus

$$\begin{array}{lrcl} [\text{case } (a)] & M_{[1..n]} & = & M_{\varepsilon_1}(A_1, B_1) \cdots M_{\varepsilon_{n_0}}(A_{n_0}, B_{n_0}) M_a(B_{n_0+1}) \cdots M_a(B_n) \ ; \\ [\text{case } (b)] & M_{[1..n]} & = & M_{\varepsilon_1}(A_1, B_1) \cdots M_{\varepsilon_{n_0}}(A_{n_0}, B_{n_0}) M_b(B_{n_0+1}) \cdots M_b(B_n) \ . \end{array} \tag{20}$$

In both cases, the triple

$$\begin{pmatrix} P_{1,n} \\ P_{2,n} \\ Q_n \end{pmatrix} := M_{[1..n]} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{21}$$

gives rise to a pair $(P_{1,n}/Q_n, P_{2,n}/Q_n)$ that is called the *n-th convergent* of f. The convergents $(P_{1,n}/Q_n, P_{2,n}/Q_n)$ provide rational approximations of f, that converge toward f, as will be seen in Section 2.8.

Applying the non-degenerate phase of each of the three algorithms to the triple $(P_{1,n}, P_{2,n}, Q_n) \in \mathcal{R}$ produces the finite sequence $M_{\epsilon_k}(A_k, B_k)_{1\leq k\leq n}$. Moreover, the intermediate triples, defined as

$$\begin{pmatrix} P_{1,n,k} \\ P_{2,n,k} \\ Q_{n,k} \end{pmatrix} = M_{\varepsilon_{k+1}}(A_{k+1}, B_{k+1}) \cdots M_{\varepsilon_n}(A_n, B_n) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad (1 \leq k \leq n) \, ,$$

lead to the intermediate convergents

$$(P_{1,n,k}/Q_{n,k}, P_{2,n,k}/Q_{n,k}) \quad \text{of the rational pair} \quad (P_{1,n}/Q_n, P_{2,n}/Q_n).$$

2.4. **Each algorithm enters the common framework.** Next proposition shows how the common framework is adapted to each of the three algorithms of interest, namely Jacobi–Perron (in its two forms), Brun and fully subtractive.

Proposition 1. *Each algorithm of interest (gcd algorithm or continued fraction map) enters the common framework. Each algorithm is defined via a strategic sign ϵ that is expressed as a function of the order sign η, defined in* (8) *or* (17)*. The first line of Table 1 recalls the choice of the strategic function. Each algorithm uses at each iteration the set $\mathcal{H}$ of quotients that decomposes into two subsets $\mathcal{H}^{\langle 1\rangle}, \mathcal{H}^{\langle 2\rangle}$. Moreover, for the gcd algorithms, the set of inputs is $\mathcal{R}$, defined in* (6)*, the set of inputs of the degenerate phase is $\mathcal{D}$, and the following bijection holds:*

$$\mathcal{R} = \mathtt{Seq}(\mathcal{H}) \times \mathcal{D} \tag{22}$$

involving the set $\mathrm{Seq}(\mathcal{H})$ *of finite sequences of $\mathcal{H}$.*

Table 1 provides the combinatorial description of the four triples $\mathcal{T} = (\mathcal{H}^{\langle 2\rangle}, \mathcal{H}^{\langle 1\rangle}, \mathcal{D})$, in terms of the set $\mathcal{P}$ defined in (2) *together with the sets*

$$\mathcal{L} = \{(A,B) \mid \ \deg B > \deg A \geq 0\}, \quad \mathcal{G} = \{B \mid \deg B \geq 1\}, \quad \mathcal{U} = \{R \mid R \text{ monic}\}. \tag{23}$$

	Jacobi–Perron	Opposite Jacobi–Perron	Brun	Fully subtractive
ϵ	$+1$	-1	η	$-\eta$
$\mathcal{H}^{\langle 2\rangle}$	$\mathcal{L}$	$\mathcal{L}$	$\mathbb{F}_q^\star \times \mathcal{G}$	$\mathcal{L} + \big(\mathcal{L} \setminus \big(\mathbb{F}_q^\star \times \mathcal{G}\big)\big)$
$\mathcal{H}^{\langle 1\rangle}$	$\{0\} \times \mathcal{G}$	$\{0\} \times \mathcal{G}$	$\{0\} \times \mathcal{G} + \{0\} \times \mathcal{G}$	$\emptyset$
$\mathcal{D}$	$\mathcal{P} = \mathtt{Seq}(\mathcal{G}) \times \mathcal{U}$	$\mathtt{Seq}(\mathcal{G}) \times \mathcal{U}$	$\{0\} \times \mathcal{U}$	$\mathcal{P} + \mathcal{P} \setminus (\{0\} \times \mathcal{U})$

TABLE 1. For each algorithm, this table gives the combinatorial description of the triple $\mathcal{T} := (\mathcal{H}^{\langle 1\rangle}, \mathcal{H}^{\langle 2\rangle}, \mathcal{D})$. Here, the sum denotes the *combinatorial sum* associated with the union of disjoint copies. In our framework, the copies come from the two possible values taken by ϵ.

We observe in Table 1 that the sum of the last two columns is equal to the sum of the first two columns, or, alternatively, the sum of the last two columns provides twice the first column:

$$\left\{ \begin{array}{llll} \mathcal{H}_{FS}^{\langle 2\rangle} & +\mathcal{H}_{B}^{\langle 2\rangle} & = 2\mathcal{L} & = 2\mathcal{H}_{JP}^{\langle 2\rangle} \quad = \mathcal{H}_{JP}^{\langle 2\rangle} + \mathcal{H}_{JP^-}^{\langle 2\rangle} \\ \mathcal{H}_{FS}^{\langle 1\rangle} & +\mathcal{H}_{B}^{\langle 1\rangle} & = 2\,(\{0\} \times \mathcal{G}) & = 2\mathcal{H}_{JP}^{\langle 1\rangle} \quad = \mathcal{H}_{JP}^{\langle 1\rangle} + \mathcal{H}_{JP^-}^{\langle 1\rangle} \\ \mathcal{D}_{FS} & +\mathcal{D}_{B} & = 2\mathcal{P} & = 2\mathcal{D}_{JP} \quad = \mathcal{D}_{JP} + \mathcal{D}_{JP^-} \end{array} \right\}.$$

The following combinatorial relation between the four triples $\mathcal{T}_\sharp = (\mathcal{H}_\sharp^{\langle 2\rangle}, \mathcal{H}_\sharp^{\langle 1\rangle}, \mathcal{D}_\sharp)$ relative to each algorithm thus holds: $\mathcal{T}_B + \mathcal{T}_{FS} = 2\,\mathcal{T}_{JP} = \mathcal{T}_{JP} + \mathcal{T}_{JP^-}$.

We now prove this property that will be called in the sequel the Sum Property. The proof is based on the strategy of each algorithm and highlights the relations between the algorithms. It will be useful in the sequel of the paper, in particular in the beginning of Section 4 and during the whole Section 5.

Proposition 2. *The following relation, called the Sum Property, holds between the four triples $\mathcal{T}_\sharp = (\mathcal{H}_\sharp^{\langle 2\rangle}, \mathcal{H}_\sharp^{\langle 1\rangle}, \mathcal{D}_\sharp)$ relative to each algorithm:*

$$\mathcal{T}_B + \mathcal{T}_{FS} = 2\,\mathcal{T}_{JP} = \mathcal{T}_{JP} + \mathcal{T}_{JP^-}\,. \tag{24}$$

Proof. We consider the two following maps defined on $\mathcal{R}$

$$\mathrm{U}^{(1)} := \mathrm{U}_{JP} \times \mathrm{U}_{JP^-}, \qquad \mathrm{U}^{(2)} := \mathrm{U}_B \times \mathrm{U}_{FS}\,.$$

Each of them outputs a pair in $\mathcal{R}^2$ that respectively satisfies, with the choice of ϵ,

$$\mathrm{U}^{(1)}(R) = (\mathrm{U}_+(R), \mathrm{U}_-(R)), \qquad \mathrm{U}^{(2)}(R) = [\![\eta = 1]\!](\mathrm{U}_+(R), \mathrm{U}_-(R)) + [\![\eta = -1]\!](\mathrm{U}_-(R), \mathrm{U}_+(R))\,.$$

We choose two boolean functions α and β defined on $\mathcal{R}$, and we now consider the two corresponding maps from $\mathcal{R}$ to $\mathcal{R}$. The first one, $\mathrm{U}^{(1)}_\alpha$, built from $\mathrm{U}^{(1)}$ and α, outputs the first component of $\mathrm{U}^{(1)}$ when $\alpha = 1$, and, otherwise, the second one, which gives

$$\mathrm{U}^{(1)}_\alpha(R) = [\![\alpha = 1]\!]\ \mathrm{U}_+(R) + [\![\alpha = -1]\!]\ \mathrm{U}_-(R)\,.$$

The second one, $\mathrm{U}^{(2)}_\beta$, built from $\mathrm{U}^{(2)}$ and β, outputs the first component of $\mathrm{U}^{(2)}$ when $\beta = 1$, and, otherwise, the second one, which gives

$$\begin{array}{rll} \mathrm{U}^{(2)}_\beta(R) & = [\![\beta = 1]\!][\![\eta = 1]\!]\ \mathrm{U}_+(R) & +[\![\beta = -1]\!][\![\eta = 1]\!]\ \mathrm{U}_-(R) \\ & +\ [\![\beta = 1]\!][\![\eta = -1]\!]\ \mathrm{U}_-(R) & +[\![\beta = -1]\!][\![\eta = -1]\!]\ \mathrm{U}_+(R)\,. \end{array} \tag{25}$$

This second map thus satisfies the two relations

$$\begin{array}{rll} \mathrm{U}^{(2)}_\beta(R) & = [\![\beta\eta = 1]\!]\mathrm{U}_+(R) & +[\![\beta\eta = -1]\!]\mathrm{U}_-(R) \\ \mathrm{U}^{(2)}_\beta(R) & = [\![\beta = 1]\!]\ \mathrm{U}_B(R) & +[\![\beta = -1]\!]\ \mathrm{U}_{FS}(R)\,. \end{array} \tag{26}$$

The two maps $\mathrm{U}^{(1)}_\alpha$ and $\mathrm{U}^{(2)}_\beta$ are equal as soon as the two booleans α and β are related via the equality $\alpha = \beta \cdot \eta$. In this case, the first map uses the triple

$$\mathcal{T}^{(1)}_\alpha = [\![\alpha = 1]\!]\ \mathcal{T}_{JP} + [\![\alpha = -1]\!]\ \mathcal{T}_{JP^-} = \mathcal{T}_{JP} = \mathcal{T}_{JP^-}\,,$$

that does not depend on the choice of α, whereas the second one uses the triple

$$\mathcal{T}^{(2)}_\beta = [\![\beta = 1]\!]\ \mathcal{T}_B + [\![\beta = -1]\!]\ \mathcal{T}_{FS}\,.$$

With the two choices $\alpha = 1, \beta = \eta$, then $\alpha = -1, \beta = -\eta$, we obtain the two equalities

$$\mathcal{T}^{(2)}_\eta = [\![\eta = 1]\!]\ \mathcal{T}_B + [\![\eta = -1]\!]\ \mathcal{T}_{FS} = \mathcal{T}_{JP}, \qquad \mathcal{T}^{(2)}_{-\eta} = [\![\eta = -1]\!]\ \mathcal{T}_B + [\![\eta = 1]\!]\ \mathcal{T}_{FS} = \mathcal{T}_{JP}\,,$$

that finally entails the combinatorial equality $\mathcal{T}_B + \mathcal{T}_{FS} = 2\mathcal{T}_{JP}$, which was to be proved. □

We will now Prove Proposition 1. Before proving it, we first focus on some particular features of the three algorithms under study.

- First, the quotient pair (A, B) plays a different role in the Jacobi–Perron and in the fully subtractive algorithms, on the one side, and in the Brun algorithm, on the other side. As the Brun algorithm divides the largest component by the second largest one, the degree $\deg A$ of the quotient A always satisfies $\deg A \leq 0$.
- Second, the degenerate phase of the Brun algorithm is empty.

Note also that, unlike the classical real case for Jacobi–Perron algorithm (see e.g. [30]), there are no Markov admissibility restrictions on the pairs (A_k, B_k) of quotients that are produced by the algorithm. In other words, every sequence $(+, A_k, B_k)$, with the pairs of polynomials (A_k, B_k) in $\mathcal{H}$, is admissible.

Proposition 1 is proved in Sections 2.5, 2.6 and 2.7. Using the description of each algorithm of interest according to the literature –namely Jacobi–Perron or its opposite version in Section 2.5, Brun in Section 2.6, fully subtractive in Section 2.7–, we explain, in each case, how it indeed follows the strategy defined by its strategic sign. We also describe, in each case, the triples $(\mathcal{H}^{\langle 1\rangle}, \mathcal{H}^{\langle 2\rangle}, \mathcal{D})$, and check the equalities of Table 1.

2.5. **Jacobi–Perron and opposite Jacobi–Perron algorithm.** We first describe the Jacobi-Perron algorithm. The description of the opposite Jacobi–Perron algorithm follows by exchanging the two components R_1 and R_2. The specific component is either the first component or the second one (for the opposite case). We first describe the algorithm, and then, the associated continued fraction map.

Gcd algorithm. We start with an element $R^{(0)} \in \mathcal{R}$. At each step k, the specific component is the first component R_1 of the triple R. Hence, during each step of the non-degenerate phase, the first component R_1 is not zero, and the algorithm divides the last two components R_2 and R_3 by the specific component R_1. The largest component then becomes R_1, that we thus choose as the third component of the next triple $\widehat{R}$. A step of the Jacobi–Perron algorithm performs

$$\begin{cases} R_1 &= \widehat{R}_3 \\ R_2 &= A\,\widehat{R}_3 &+\widehat{R}_1 \\ R_3 &= B\,\widehat{R}_3 &+\widehat{R}_2 \end{cases} \qquad \text{with} \quad A = A_+(R) = \left[\frac{R_2}{R_1}\right], \qquad B = B_+(R) = \left[\frac{R_3}{R_1}\right].$$

We are always in the case where $\epsilon = +$.
In matricial notation, this can be written as $R = M_+(A,B)\,\widehat{R}, \quad \widehat{R} = N_+(A,B)\,.$

In the same vein, for the opposite version, the specific component is the second component R_2 of the triple R and the algorithm performs

$$\begin{cases} R_2 &= \widehat{R}_3 \\ R_1 &= A\,\widehat{R}_3 &+\widehat{R}_2 \\ R_3 &= B\,\widehat{R}_3 &+\widehat{R}_1 \end{cases} \qquad \text{with} \quad A = A_-(R) = \left[\frac{R_1}{R_2}\right], \qquad B = B_-(R) = \left[\frac{R_3}{R_2}\right].$$

We are always in the case where $\epsilon = -$.
In matricial notation, this gives $R = M_-(A,B)\,\widehat{R}, \quad \widehat{R} = N_-(A,B)\,.$

In both cases, the inequality between the two non-specific components ($\deg R_2 < \deg R_3$ for the Jacobi-Perron algorithm or $\deg R_1 < \deg R_3$ for the opposite version) always holds. Moreover, the following holds:

- For the Jacobi-Perron algorithm, consider the quotient $A = [R_2/R_1]$;
  - when $\deg R_1 \le \deg R_2$, the quotient A is not zero, and $0 \le \deg A < \deg B$;
  - when $\deg R_1 > \deg R_2$, the quotient A is equal to zero and only one division is performed.
- For the opposite version, consider the quotient $A = [R_1/R_2]$;
  - when $\deg R_2 \le \deg R_1$, the quotient A is not zero, and $0 \le \deg A < \deg B$;
  - when $\deg R_2 > \deg R_1$, the quotient A is equal to zero and only one division is performed.

In both cases, the sets $\mathcal{H}^{\langle i\rangle}$, the set $\mathcal{D}$ and the sign ϵ satisy

$$\begin{cases} \mathcal{H}^{\langle 1\rangle} &= \{0\} \times \mathcal{G} = \{0\} \times \{P \in \mathbb{F}_q[X] \,:\, \deg B \ge 1\}, \\ \mathcal{H}^{\langle 2\rangle} &= \mathcal{L} = \{(A,B) \mid \ 0 \le \deg A < \deg B\}, \\ \mathcal{D} &= \mathcal{P} = \mathtt{Seq}(\mathcal{G}) \times \mathcal{U}\,, \\ \epsilon &= +1 = \eta^2. \end{cases}$$

Furthermore, the Jacobi–Perron algorithm yields the bijection (15).

Continued fraction map. The associated continued fraction map T_{JP} is then defined on $\mathbb{L}^2$ as[7]

$$\mathrm{T}_{\mathrm{JP}}(f) \,=\, (0,0), \quad (f_1 = 0), \qquad \mathrm{T}_{\mathrm{JP}}(f) = \left(\frac{f_2}{f_1} - A(f),\, \frac{1}{f_1} - B(f)\right) \quad (f_1 \neq 0)\,,$$
$$\text{with} \qquad A(f) = A_+(f) = \left[\frac{f_2}{f_1}\right], \qquad B(f) = B_+(f) = \left[\frac{1}{f_1}\right].$$

We only describe the map for the Jacobi-Perron algorithm. The description of the opposite version will be clear. The continued fraction map T_{JP} provides an extension of the non-degenerate

[7]We use the following convention: the quantity $\left[\frac{f}{g}\right]$ is set to 0 if $g = 0$.

phase of the Jacobi–Perron gcd algorithm, in the following sense: when $\mathrm{T_{JP}}$ is applied to a pair $f = (f_1, f_2) \in \mathbb{F}_q(X)^2$ of the form

$$f_1 = R_1/R_3 \text{ and } f_2 = R_2/R_3 \text{ with } R = (R_1, R_2, R_3) \in \mathcal{R},$$

the trajectory of $f = (f_1, f_2)$ under $\mathrm{T_{JP}}$ is finite and provides the same sequence of quotients as the non-degenerate phase of Jacobi–Perron gcd algorithm on the triple $R = (R_1, R_2, R_3)$.

2.6. **The Brun algorithm.** The specific component is the second largest component (in terms of degree). We describe the algorithm and then, the associated continued fraction map.

Gcd algorithm. The specific component here is the second largest component with respect to the degree (provided it is not zero). When $R \in \mathcal{R}$, the specific component is either R_1 or R_2. In case of equality of the degrees, i.e., when $\deg R_1 = \deg R_2$, the choice of the specific component is made according to the sets $\mathcal{R}_\pm$ defined above in (7).
When the second largest component is equal to zero, this means that the two components R_1, R_2 are both equal to zero and the algorithm ends. The degenerate phase thus does not exist.
For each step, there are two cases according to the position of R with respect to sets $\mathcal{R}_\pm$.

- Assume first that R belongs to $\mathcal{R}_+$, i.e., $\deg R_1 \geq \deg R_2$ ($\eta(R) = +1$). The specific component is R_1 and the algorithm performs

$$\left\{\begin{array}{lcll} R_1 & = & \widehat{R}_3 & \\ R_2 & = & A\,\widehat{R}_3 & +\widehat{R}_1 \\ R_3 & = & B\,\widehat{R}_3 & +\widehat{R}_2 \end{array}\right. \qquad \text{with} \quad A = \left[\frac{R_2}{R_1}\right], \qquad B = \left[\frac{R_3}{R_1}\right].$$

  We are in the case $\epsilon = +1$.
  In matricial notation, this gives $R = M_+(A,B)\,\widehat{R}$, $\widehat{R} = N_+(A,B)\,R$.
  In this case, the quotients (A, B) belong to the set

$$\{(A,B) \mid A \in \mathbb{F}_q,\ \deg B \geq 1\} = \mathbb{F}_q \times \mathcal{G}.$$

  When $\deg R_1 > \deg R_2$, then $A = 0$, and only one division is performed. When $\deg R_1 = \deg R_2$, two divisions are performed and $A \in \mathbb{F}_q^\star$.
- Otherwise, when R belongs to $\mathcal{R}_-$, i.e., $\deg R_1 < \deg R_2$ ($\eta(R) = -1$), the specific component is R_2, there is only one division that is performed and the algorithm is written as

$$\left\{\begin{array}{lcll} R_1 & = & A\,\widehat{R}_3 & +\widehat{R}_2 = \widehat{R}_2 \\ R_2 & = & \widehat{R}_3 & \\ R_3 & = & B\,\widehat{R}_3 & +R_1 \end{array}\right. \qquad \text{with} \quad A = 0 = \left[\frac{R_1}{R_2}\right], \qquad B = \left[\frac{R_3}{R_2}\right].$$

  We are in the case $\epsilon = -1$.
  In matricial notation, this gives $R = M_-(A,B)\,\widehat{R}$, $\widehat{R} = N_-(A,B)\,R$.
  The quotients (A, B) belong to the set

$$\{(A,B) \mid A = 0,\ \deg B \geq 1\} = \{0\} \times \mathcal{G}.$$

In summary. The sets $\mathcal{H}^{\langle i\rangle}$, the set $\mathcal{D}$, and the sign ϵ satisfy

$$\left\{\begin{array}{ll} \mathcal{H}^{\langle 1\rangle} & = \{0\} \times \mathcal{G} + \{0\} \times \mathcal{G} \\ \mathcal{H}^{\langle 2\rangle} & = \mathbb{F}_q^\star \times \mathcal{G} \\ \mathcal{D} & = \mathcal{U} \\ \epsilon & = \eta \end{array}\right.$$

and Brun's algorithm yields the bijection (15).

Continued fraction map. As the equality $\epsilon = \eta$ always holds, the associated continued fraction $\mathrm{T_B}$ is defined on $\mathbb{L}^2$ as follows:

$$\mathrm{T_B}(f) = \left\{\begin{array}{ll} \left(\dfrac{f_2}{f_1} - A_+(f),\ \dfrac{1}{f_1} - B_+(f)\right) & \text{if } \eta(f) = +1\,, \\ \left(\dfrac{1}{f_2} - B_-(f),\ \dfrac{f_1}{f_2} - A_-(f)\right) & \text{if } \eta(f) = -1\,, \end{array}\right.$$

and if the involved denominator (f_1 or f_2) is equal to 0, then $\mathrm{T_B}(f) := (0,0)$.

The continued fraction map $\mathrm{T_B}$ provides an extension of the non-degenerate phase of the Brun gcd algorithm, in the following sense: when $\mathrm{T_B}$ is applied to a pair $(f_1, f_2) \in \mathbb{F}_q(X)^2$ of the form

$$f_1 = R_1/R_3 \text{ and } f_2 = R_2/R_3 \text{ with } R = (R_1, R_2, R_3) \in \mathcal{R},$$

the trajectory of (f_1, f_2) under $\mathrm{T_B}$ is finite and provides the same sequence of quotients as the (non-degenerate phase of the) Brun gcd algorithm on the triple $R = (R_1, R_2, R_3)$.

2.7. **The fully subtractive algorithm.** The specific component is the component of smallest degree.

Gcd algorithm. When the smallest component is non-zero, the algorithm chooses it as the specific component, divides the two largest components by the smallest one and, after such divisions, the smallest one becomes the largest one, and thus the third component. The degenerate phase begins when the smallest component is zero.

Non-degenerate phase. We first consider the non-degenerate phase where each of the two components of the pair (R_1, R_2) is non-zero, and, as for the Brun algorithm, we distinguish two cases according to the ordering of the degrees of the first two components.

– If $R \in \mathcal{R}_+$, i.e., $\deg R_1 \geq \deg R_2$ ($\eta(R) = +1$), the specific component is R_2 and the algorithm performs

$$\left\{ \begin{array}{lcll} R_1 & = & A\widehat{R}_3 & +\widehat{R}_2 \\ R_2 & = & \widehat{R}_3 & \\ R_3 & = & B\,\widehat{R}_3 & +\widehat{R}_1 \end{array} \right. \qquad \text{with} \qquad A = \left[\frac{R_1}{R_2}\right], \qquad B = \left[\frac{R_3}{R_2}\right].$$

We are in the case $\epsilon = -$.
In matricial notation, this yields $R = M_-(A, B)\,\widehat{R}$, $\widehat{R} = N_-(A, B)\,R$.
In this case, two divisions are performed, and the quotients (A, B) belong to the set

$$\mathcal{L} = \{(A, B) \mid \deg B > \deg A \geq 0\}.$$

– If now $R \in \mathcal{R}_-$, i.e., $\deg R_1 < \deg R_2$ ($\eta(R) = -1$), the specific component is R_1 and the algorithm performs

$$\left\{ \begin{array}{lcll} R_1 & = & \widehat{R}_3 & \\ R_2 & = & A\,\widehat{R}_3 & +\widehat{R}_1 \\ R_3 & = & B\,\widehat{R}_3 & +\widehat{R}_2 \end{array} \right. \qquad \text{with} \qquad A = \left[\frac{R_2}{R_1}\right], \qquad B = \left[\frac{R_3}{R_1}\right].$$

We are in the case $\epsilon = +$.
In matricial notation, this writes $R = M_+(A, B)\,\widehat{R}$, $\widehat{R} = N_+(A, B)\,R$.
In this case, two divisions are performed and the quotients (A, B) belong to the set

$$\{(A, B) \mid \deg B > \deg A \geq 1\} = \mathcal{L} \setminus \left(\mathbb{F}_q^\star \times \mathcal{G}\right).$$

Finally, the quotients (A, B) which occur in the non-degenerate phase belong to the set

$$\mathcal{H} = \mathcal{H}^{\langle 2\rangle} \cup \mathcal{H}^{\langle 1\rangle} \qquad \text{with} \quad \mathcal{H}^{\langle 1\rangle} = \emptyset, \quad \mathcal{H}^{\langle 2\rangle} = \mathcal{L} + \left(\mathcal{L} \setminus \left(\mathbb{F}_q^\star \times \mathcal{G}\right)\right).$$

Degenerate phase. The set $\mathcal{D}$ decomposes as $\mathcal{D}_+ \cup \mathcal{D}_-$ with $\mathcal{D}_\pm = \mathcal{D} \cap \mathcal{R}_\pm$, i.e.,

$$\left\{ \begin{array}{lll} \mathcal{D}_+ & = \{(R_1, R_3) \mid R_1 = 0 \ \text{ or } \ \deg R_1 < \deg R_3, \ \ R_3 \text{ monic}\} & = \mathcal{P} \\ \mathcal{D}_- & = \{(R_2, R_3) \mid R_2 \neq 0, \ \deg R_2 < \deg R_3, \ \ R_3 \ \text{ monic}\} & = \mathcal{P} \setminus (\{0\} \times \mathcal{U}). \end{array} \right.$$

The difference between the two cases is due to the inequality between $\deg R_1$ and $\deg R_2$ which may be strict or not (depending on whether we are in $\mathcal{R}_-$ or in $\mathcal{R}_+$).

In summary. The sets $\mathcal{H}^{\langle i\rangle}$, the set $\mathcal{D}$, and the sign ϵ are

$$\left\{ \begin{array}{ll} \mathcal{H}^{\langle 1\rangle} & = \emptyset \\ \mathcal{H}^{\langle 2\rangle} & = \mathcal{L} + \left(\mathcal{L} \setminus \left(\mathbb{F}_q^\star \times \mathcal{G}\right)\right) \\ \mathcal{D} & = \mathcal{P} + \mathcal{P} \setminus (\{0\} \times \mathcal{U}) \\ \epsilon & = -\eta \end{array} \right.$$

and the fully subtractive algorithm yields the bijection (15).

The continued fraction map. As the equality $\epsilon = -\eta$ always holds, the associated continued fraction T_{FS} is defined on $\mathbb{L}^2$ as follows:

$$\mathrm{T}_{\mathrm{FS}}(f_1, f_2) = \begin{cases} \left(\dfrac{1}{f_2} - B_-(f),\ \dfrac{f_1}{f_2} - A_-(f)\right) & \text{if } \eta(f) = +1\,, \\ \left(\dfrac{f_2}{f_1} - A_+(f),\ \dfrac{1}{f_1} - B_+(f)\right) & \text{if } \eta(f) = -1\,. \end{cases}$$

If the involved denominator (f_1 or f_2) is equal to 0, $\mathrm{T}_{\mathrm{FS}}(f_1, f_2) := (0, 0)$.
The continued fraction map T_{FS} provides an extension of the non-degenerate phase of the FS gcd algorithm, in the following sense: when T_{FS} is applied to a pair $f = (f_1, f_2) \in \mathbb{F}_q(X)^2$ of the form

$$f_1 = R_1/R_3 \text{ and } f_2 = R_2/R_3 \text{ with } R = (R_1, R_2, R_3) \in \mathcal{R},$$

the trajectory of $f = (f_1, f_2)$ under T_{FS} is finite and provides the same sequence of quotients as the non-degenerate phase of the FS gcd algorithm on the triple $R = (R_1, R_2, R_3)$.

2.8. **Convergence of the algorithms.** This section relates the behaviour of the algorithms on polynomials and on continued fractions. It proves that the convergents defined in (21) provide rational approximations of f that converge toward f. The convergence of the algorithms plays a central role in the context of simultaneous rational approximation.

Proposition 3. *The convergents $(P_{1,n}/Q_n, P_{2,n}/Q_n)$ defined in* (21) *provide rational approximations of $f = (f_1, f_2)$, and the following holds*

$$\left\| f_i - \frac{P_{i,n}}{Q_n} \right\| < \frac{1}{\|Q_n\|}. \tag{27}$$

This is a well-known property for the Jacobi–Perron algorithm (see [26, 27]) and for the Brun algorithm (see [3, 17]). We give here a unified proof that applies to each of the three algorithms.

Proof. We first recall that the third column of the matrix $M_{[1..n]}$ defined in (19) or (20) is equal to $(P_{1,n}, P_{2,n}, Q_n)$. We also consider the inverse matrix $N_{[1..n]}$ of $M_{[1..n]}$. We let denote by $m_{ij}^{(n)}$ the coefficient of $M_{[1,n]}$ at the i-th row and j -th column, and by $n_{ij}^{(n)}$ the coefficient of $N_{[1,n]}$ at the i-th row and j -th column. The product matrix $M_{[1..n]}$ involves as factors the matrices $M_{\epsilon_k}(A_k, B_k)$ (with $\epsilon_i = \pm 1$), or –depending on the case (a) or (b)– a sequence of matrices $M_a(B)$, or a sequence of matrices $M_b(B)$. In a similar way, the product matrix $N_{[1..n]}$ involves as factors the matrices $N_{\epsilon_k}(A_k, B_k)$ (with $\epsilon_i = \pm 1$), or –depending on the case (a) or (b)– a sequence of matrices $N_a(B) = M_a(B)^{-1}$, or a sequence of matrices $N_b(B) = M_b(B)^{-1}$. Moreover, since all these matrices have a determinant equal to ± 1, the coefficient $n_{ij}^{(n)}$ is, up to the sign, a determinant of the matrix $M_{[1..n]}$. More precisely, if the indices (i_1, i_2) and (j_1, j_2) satisfy $\{i_1, i_2, i\} = \{j_1, j_2, j\} = \{1, 2, 3\}$, one has

$$n_{ij}^{(n)} = \pm(m_{i_1 j_1}^{(n)} m_{i_2 j_2}^{(n)} - m_{i_1 j_2}^{(n)} m_{i_2 j_1}^{(n)})\,.$$

The proof is based on the following lemma:

Lemma 1. *The following relations hold*

$$\deg m_{33}^{(n)} = \deg Q_n, \quad \deg m_{ij}^{(n)} < \deg Q_n \quad \textit{for } (i,j) \neq (3,3), \quad \Delta_n := \max\left(\deg n_{ij}^{(n)}\right) \leq \deg Q_n\,.$$

Proof. It is based on the particular form of the matrices $M_\epsilon(A, B), N_\epsilon(A, B),\ M_a(B), M_b(B)$, $N_a(B), N_b(B)$ and is easily proven by induction on n. □

The estimate (27) is now deduced from the lemma: Starting with $f = (f_1, f_2) \in \mathbb{L}^2$, we let denote by $g = (g_1, g_2)$ the vector $g = T^n(f)$. One has:

$$\left\| f_i - \frac{P_{i,n}}{Q_n} \right\| = \left\| \frac{m_{i1}^{(n)} g_1 + m_{i2}^{(n)} g_2 + m_{i3}^{(n)}}{m_{31}^{(n)} g_1 + m_{32}^{(n)} g_2 + m_{33}^{(n)}} - \frac{m_{i3}^{(n)}}{m_{33}^{(n)}} \right\|$$

$$= \frac{\left\| g_1 \left(m_{i1}^{(n)} m_{33}^{(n)} - m_{31}^{(n)} m_{i3}^{(n)}\right) + g_2 \left(m_{i2}^{(n)} m_{33}^{(n)} - m_{32}^{(n)} m_{i3}^{(n)}\right) \right\|}{\|m_{33}^{(n)}\| \cdot \|m_{31}^{(n)} g_1 + m_{32}^{(n)} g_2 + m_{33}^{(n)}\|}$$

Lemma 1 entails the ultrametric equality

$$\|m_{31}^{(n)} g_1 + m_{32}^{(n)} g_2 + m_{33}^{(n)}\| = \|m_{33}^{(n)}\|$$

and the denominator is thus equal to $\|Q_n\|^2$. For the numerator and for $i = 1$, one has

$$m_{11}^{(n)} m_{33}^{(n)} - m_{31}^{(n)} m_{13}^{(n)}) = \pm n_{22}^{(n)}, \qquad m_{12}^{(n)} m_{33}^{(n)} - m_{32}^{(n)} m_{13}^{(n)} = \pm n_{21}^{(n)},$$

whereas, for $i = 2$, one has

$$m_{21}^{(n)} m_{33}^{(n)} - m_{31}^{(n)} m_{23}^{(n)} = \pm n_{12}^{(n)}, \qquad m_{22}^{(n)} m_{33}^{(n)} - m_{32}^{(n)} m_{23}^{(n)} = \pm n_{11}^{(n)}.$$

Then, Lemma 1 entails that the absolute value of the numerator is less than $\|Q_n\|$ and then

$$\left\| f_i - \frac{P_{i,n}}{Q_n} \right\| \leq \frac{\|Q_n\|}{\|Q_n\|^2} = \frac{1}{\|Q_n\|}.$$

□

3. Models, methodology and main results

The sequel of the paper is devoted to probabilistic analyses, in two different frameworks:

- the gcd algorithms (that act on $\mathcal{R}$) in Section 4; here, one deals with discrete inputs (polynomials), and the probabilistic model is called the *discrete model*;
- their associated continued fraction maps (that act on $\mathbb{L}^2$) in Section 5; here, one deals with continuous inputs, and the probabilistic model is called the *continuous model*.

First, Sections 3.1 and 3.2 describe the costs of algorithmic interest in each model (algorithms and continued fraction maps). Then, we focus on each model, the discrete one in Section 3.3 and the continuous model in Section 3.4. In each section, we make precise the probabilistic model, describe the methodogy that will be developed there, and state the main results obtained. We then focus in Section 3.5 on asymptotic Gaussian laws that play a central role in the paper. The last two sections perform a comparaison between the models (in Section 3.6) and the algorithms (in Section 3.7).

3.1. Main costs for the gcd algorithms.

We consider seven costs which intervene in a natural way in the complexity of the *non-degenerate phase* of the three algorithms under study.

Remark. We focus here on the non-degenerate phase for two reasons. First, we want to compare the probabilistic behaviours of truncated trajectories with the behaviour of the algorithms on polynomials, and, truncated trajectories only involve the degenerate phase with zero probability. Second, and this is related to the previous remark, the non-degenerate phase on polynomials has costs that are of strictly larger order than the degenerate phase. We return to this fact at the beginning of Section 4.5.

We first consider total costs C that give a measure of the total complexity of the degenerate phase. There are two main cases:

- The first case deals with total costs C that are called *additive*. They are defined via *step-costs* c that intervene in each step as a measure of the complexity of this step $R \mapsto \mathrm{U}(R)$ and that only depend on the quotient $Q(R) := (A, B)$ produced during this step. We then let $\widehat{c}(R) := c(Q(R))$. We give below five instances of such interesting costs c (see (28)).
- The second case deals with bit-complexites.

In the first case, we consider five step-costs c, defined on the pair (A, B) of quotients, namely

$$\begin{array}{ll} (i)\ \ c_0(A,B) = 1 & (ii)\ \ d(A,B) = 1 + [\![A \neq 0]\!] \\ (iii)\ \ d_B(A,B) = \deg B & \\ (iv)\ \ \delta(A,B) = \delta(A) + \delta(B) & (v)\ \ \nu(A,B) = \nu(A) + \nu(B). \end{array} \tag{28}$$

Here, $\delta(W)$ et $\nu(W)$ denote respectively the number of monomials which are possibly present in the polynomial W and $\nu(W)$ the number of monomials which are indeed present in the polynomial W. More precisely, for a non-zero polynomial $W \in \mathbb{F}_q[X]$ of the form

$$W = w_n X^n + w_{n-1} X^n + \cdots + w_1 X + w_0, \quad \text{with} \quad w_n \neq 0,$$

one has $\qquad \delta(W) = \deg(W) + 1 = n + 1, \qquad \nu(W) = \mathrm{Card}\{k \mid 1 \le k \le n,\ w_k \neq 0\}\,.$

Note that $\nu(W) \geq 1$ at soon as $W \neq 0$. One moreover sets $\delta(0) = \nu(0) = 0$.

The total costs associated with these step-costs are described as follows: one has for

(i) the number $S(R)$ of steps;
(ii) the total number of divisions;
(iii) the degree $\deg R_3$;
(iv) or (v) the total space that is needed for the storage of all the quotients with two possibilities depending on the representation of polynomials, the usual one for (iv) and the sparse one for (v).

On an input R, the total cost C is written as

$$C(R) = \sum_{i=0}^{S(R)-1} \widehat{c}(\mathrm{U}^i(R)), \qquad \text{with} \quad \widehat{c}(S) := c(Q(S))\,, \tag{29}$$

where $Q(S)$ is the pair (A, B) of quotients produced on S.

The last two costs (the bit-complexity costs) rely on two different notions of bit-complexity of a polynomial division of the form $R = AR' + R''$, namely

- the usual bit-complexity equals $\delta(A) \cdot \delta(R')$,
- the sparse[8] bit complexity equals $\nu(A) \cdot \delta(R')$.

Then, there are two versions of the total bit-complexity of the algorithm on the input R, namely the *total bit-complexity* and the *total sparse bit-complexity*, that are respectively equal to

$$\Phi_\delta(R) \,=\, \sum_{k=1}^{S} \delta(R_{k,3})\,(\delta(A_k) + \delta(B_k))\,, \quad \Phi_\nu(R) \,=\, \sum_{k=1}^{S} \delta(R_{k,3})\,(\nu(A_k) + \nu(B_k))\,,$$

where $(R_{k,1}, R_{k,2}, R_{k,3})_k$ stands for the sequence of polynomials in $\mathcal{R}$ produced by the algorithm. The additive formula $\deg R_{k,3} = \deg B_k + \deg R_{k-1,3}$ entails another form for these complexities that only involves $\delta(A_k), \nu(A_k), \delta(B_k), \nu(B_k)$. We will return later to this important remark (see Section 5.5).

Remark. These costs provide realistic measures of the complexity when the cardinality q of the field $\mathbb{F}_q$ is fixed. When the cardinality q varies, and in particular when $q \to \infty$, this cardinality must be taken into account and the costs $\delta, \nu, \Phi_\delta, \Phi_\nu$ are replaced by their "underlined" conterparts

$$\underline{\delta} := (\log q)\,\delta, \qquad \underline{\nu} := (\log q)\,\nu, \qquad \underline{\Phi}_\delta := (\log q)^2\,\Phi_\delta, \qquad \underline{\Phi}_\nu := (\log q)^2\,\Phi_\nu\,.$$

3.2. **Main costs for the continued fraction maps.** With $f = (f_1, f_2) \in \mathbb{L}^2$, we associate its trajectory and its truncated trajectory at level n defined as

$$(f, \mathrm{T}(f), \mathrm{T}^2(f), \cdots, \mathrm{T}^n(f), \cdots), \qquad (f, \mathrm{T}(f), \mathrm{T}^2(f), \cdots, \mathrm{T}^n(f))\,.$$

We consider total costs C_n that give a measure of the total complexity of the truncated trajectory. As previously, there are two main cases:

- The first case deals with total costs C_n that are called additive. They are defined via step-costs c that intervene in each step of the trajectory as a measure of the complexity of the generic step $f \mapsto \mathrm{T}(f)$ and only depend on the quotient $Q(f) := (A, B)$ produced in this step. One then lets $\widehat{c}(f) := c(Q(f))$. The instances of the step-costs c are the same as previously, and are described in (28).
- The bit-complexites which do not enter the previous case.

In the first case, the total costs associated with the step-costs given in (28) are described as follows: one has for

(i) the length n of the trajectory;
(ii) the total number of divisions performed;

[8] Sparse complexity is called fine complexity in [7]. We have chosen here the term "sparse" which better reflects what this cost aims to describe.

(*iii*) the degree $\deg Q_n$ of the denominator of the n-th convergent $(P_{1,n}/Q_n, P_{2,n}/Q_n)$ of the input f;
(*iv*) or (*v*) the space that is needed to store the continued fraction expansion, with two versions: the usual storage for (*iv*) or the sparse storage for (*v*).

On an input f, the total cost $C_n(f)$ is written as

$$C_n(f) = \sum_{i=0}^{n-1} \widehat{c}(\mathrm{T}^i(f))\,, \qquad \text{with} \quad \widehat{c}(g) := c(Q(g))\,, \tag{30}$$

where $Q(g)$ is the pair (A, B) of quotients produced on g.

In the second case, we are interested in the bit-complexities that are needed for computing the n-th convergent, depending on the representation (usual or sparse) of the quotients associated with cost $c = \delta$ or $c = \nu$, namely

$$\Phi_{c;n}(f) := \sum_{k=1}^{n} [c(A_k(f)) + c(B_k(f)]\, \delta(Q_{k-1}(f))\,.$$

Remark. As in the discrete model, these costs are well adapted when the cardinality q of the field $\mathbb{F}_q$ is fixed. When the cardinality q varies, and in particular when $q \to \infty$, we replace $\delta_n, \nu_n, \Phi_\delta, \Phi_\nu$ by their "underlined" conterparts

$$\underline{\delta}_n := (\log q)\, \delta_n, \qquad \underline{\nu}_n := (\log q)\, \nu_n, \qquad \underline{\Phi}_{\delta;n} := (\log q)^2\, \Phi_{\delta;n} \qquad \underline{\Phi}_{\nu;n} := (\log q)^2\, \Phi_{\nu,n}\,. \tag{31}$$

3.3. **The discrete model.** We first describe the model, then the methods, and finally the results.

Discrete model on $\mathcal{R}$. The *size* of a triple $R = (R_1, R_2, R_3) \in \mathcal{R}$ is chosen as the maximum degree of its three components, namely $\deg R_3$. The finite set $\mathcal{R}_m$

$$\mathcal{R}_m := \{R := (R_1, R_2, R_3) \mid m = \deg R_3 > \max(\deg R_1, \deg R_2), \quad R_3 \text{ monic } \} \tag{32}$$

gathers triples of polynomials with size m, and is endowed with the uniform probability $\mathbb{P}_m$.

We wish to study the probabilistic behaviour of each of the seven costs $C \in \{c_0, d, d_B, \delta, \nu, \Phi_\delta, \Phi_\nu\}$ defined on $\mathcal{R}$ and related to the behaviour of a gcd algorithm, notably when the size m of the input tends to ∞. More precisely, we consider the restriction C_m of C to each $\mathcal{R}_m$, and we let denote its expectation by $\mathbb{E}[C_m]$ and its variance by $\mathbb{V}[C_m]$.

Methods. [Analytic combinatorics] We use methods from analytic combinatorics, which we briefly describe here informally. The reference is the book of Flajolet and Sedgewick [11].

We consider each subset $\mathcal{X}$ of Table 1. There is a natural notion of size, denoted as $||\cdot||$ and defined here as the maximum degree. There is also a cost c defined on $\mathcal{X}$, as described in Section 3.1. We associate with $\mathcal{X}$ its (plain) generating function (gf) and its bivariate generating function (bgf), defined as

$$X(z) := \sum_{x \in \mathcal{X}} z^{||x||} = \sum_{n \geq 0} X_n z^n \qquad X_c(z, u) := \sum_{x \in \mathcal{X}} z^{||x||} u^{c(x)} = \sum_{n,k} X_{n,k} z^n u^k\,,$$

where the coefficients X_n and $X_{n,k}$ are respectively the number of elements of size n, and the number of elements of size n with cost equal to k. They are respectively denoted as $[z^n]X(z)$ and $[z^n u^k]X(z, u)$.

There are two steps in analytic combinatorics: the symbolic step and the analytical step. The symbolic step views a generating function as a formal object, and builds gf's with a symbolic dictionary. The analytic step views a generating function as a function of one complex variable z, and uses an analytic dictionary, between the (dominant) singularity of a gf and the asymptotics of its coefficients.

The symbolic step is the main tool for obtaining Tables 2 and 3. It deals with the gf's with their initial combinatorial definition and transfers the main combinatorial operations into operations on gf's. These combinatorial operations are the combinatorial (disjoint) sum $\mathcal{X} + \mathcal{Y}$, the product $\mathcal{X} \times \mathcal{Y}$, the class $\mathrm{Seq}(\mathcal{X})$ of the finite sequences built on $\mathcal{X}$. They translate as $X(z) + Y(z)$,

$X(z) \cdot Y(z)$ and finally as $1/(1 - X(z))$. These are instances provided in Table 2. This finally transfers the combinatorial equality (15) into the equality (36), i.e.,

$$\mathcal{R} = \mathtt{Seq}(\mathcal{H}) \times \mathcal{D} \text{ into } R(z) = \frac{D(z)}{1 - H(z)}.$$

There are also extensions for bgf's, notably when they are relative to *additive costs* in the sense of Section 3.1. Then the bivariate generating function of the total cost is a quasi-inverse of the bivariate generating function of the step cost. We thus obtain Eqn (49).

The analytic step considers a generating function as a function of the complex variable z, and deals with it. The study of its dominant singularity (the singularity with the smallest module), and notably, its location and its nature, provide asymptotic information on the coefficients of the gf's or bgf's. Here, all the generating functions are rational fractions and their dominant singularities are always dominant simple poles. Then, the following simple analytic transfers hold

$$\begin{array}{lll} X(z) \text{ has a simple dominant pole at } z = \rho & \Longrightarrow & [z^m]X(z) \sim_{m\to\infty} B \cdot \rho^{-m} \\ X(z,u) \text{ has a simple dominant pole at } z = \rho(u) & \Longrightarrow & [z^m]X(z,u) \sim_{m\to\infty} B(u)\rho(u)^{-m} \end{array}$$

for some constants $B, B(u)$.

As the expectation $\mathbb{E}[u^{C_m}]$ is written as $\mathbb{E}[u^{C_m}] = \frac{[z^m]X(z,u)}{[z^m]X(z)}$, the analytic step yields the estimate

$$\mathbb{E}[u^{C_m}] \sim \frac{B(u)}{B} \left[\frac{\rho}{\rho(u)}\right]^m , \tag{33}$$

that may give interesting hints on the distribution of C_m. With extra hypotheses on the previous estimates (33) (mainly, good analytic properties, a good knowledge about remainder terms, and a so-called condition of "variability" on the function $\rho/\rho(u)$), we will see in Section 3.5 that this indeed entails that the distribution is *asymptotically Gaussian*. We return to this notion in Section 3.5.

Results. Our results are as follows (see Theorem 1 and Proposition 7):

- For each of the three algorithms and for each additive cost C associated with a step-cost c, the expectations $\mathbb{E}[C_m]$ and variances $\mathbb{V}[C_m]$ are asymptotically linear with respect to the size m, and involve constants $M_c = M_{c,\sharp}, V_c = V_{c,\sharp}$, with

  $$\mathbb{E}[C_m] \sim M_c\, m, \qquad \mathbb{V}[C_m] \sim V_c\, m\,.$$

  Moreover, the expectation of each bit-complexity cost Φ_δ or Φ_ν is asymptotically quadratic with respect to the size m, and satisfy

  $$\mathbb{E}[\Phi_{\delta,m}] \sim M_\delta \frac{m^2}{2}, \qquad \mathbb{E}_m[\Phi_{\nu,m}] \sim M_\nu \frac{m^2}{2}\,.$$

  We provide precise expressions for the dominant constants $M_c = M_{\sharp,c}$ and $V_c = V_{\sharp,c}$ in terms of the pair $(\sharp, c)$ formed by the algorithm and the cost (see Table 6 and Proposition 7).
- Second, for each of the three algorithms and for each of the four additive costs C associated with step-cost $c \in \{c_0, d, \delta, \nu\}$, an asymptotic Gaussian law holds (see Section 3.5 and Theorem 1).

3.4. **The continuous model.** We first describe the model, then the methods, and finally the results.

Continuous model on $\mathbb{L}^2$**.** Here, the set $\mathbb{L}^2$ is endowed with its Haar measure, which involves the Haar measure μ on $\mathbb{L}$ and is here $\mu \otimes \mu$. We consider a continued fraction map T and a cost $c \in \{c_0, d_B, d, \delta, \nu\}$. This gives rise to a cost C_n defined on the n-th truncated trajectory, as in Section 3.2. We are interested in the behaviour of the cost $C_n(f)$ on the truncated trajectory at level n, when the level n of truncature tends to ∞.

Methods. [Dynamical analysis] We view each continued fraction map as a dynamical system. We associate with this dynamical system its transfer operators, here the density transformer (i.e., the Perron–Frobenius operator) defined in (74), together with its bivariate version defined in (78).

Here, the branches of map T are homographies, whose Jacobian has a constant absolute value. This entails that the map T is $\mu\otimes\mu$ invariant and the quotients define independent and identically distributed random variables. Furthermore, the (bi-variate) transfer operator that underlies the dynamical system is closely related to the (bi-variate) generating function of the discrete model.

Results. Our results are as follows (see Theorem 3 and Propositions 9 and 10):

- Each of the three continued fraction maps $\mathrm{T}=\mathrm{T}_\sharp$ is $\mu\otimes\mu$ preserving and ergodic, with an entropy $\mathcal{E}(\mathrm{T})$.
- The cost $C_n(f)$ satisfies the following: For each additive cost C, the cost $(1/n)\,C_n(f)$ admits almost everywhere a *constant limit*, and for each bit-complexity cost C, the cost $(1/n^2)\,C_n(f)$ also admits almost everywhere a *constant limit*. In both cases, this limit is denoted by $\widetilde{M}_c$.
- For each continued fraction map and each additive cost c, the following relation holds between the two constants of the expectations (the discrete constant and the continuous constant):
$$3M_c=\widetilde{M}_c\cdot\mathcal{E}(\mathrm{T})\,;$$
it involves the entropy $\mathcal{E}(\mathrm{T})$ that is furthermore equal to $3\,\mathbb{E}_{\mu\otimes\mu}[d_B]$.

- For each continued fraction map, and for each cost $c\in\{d_B,d,\delta,\nu\}$, the associated cost C_n follows a Gaussian law, except in the case (FS,d) where the cost d (the number of divisions that are performed) is constant and equal to 2 (see Theorem 3). Moreover, we provide expressions for the constants $\widetilde{V}_c$ involved in the variance (see Table 8).

3.5. **Gaussian laws, exact or asymptotic.** We begin with some generalities on Gaussian laws, that may be exact, or asymptotic.

We first consider a sequence $(C_n)_n$ of random variables. It is well known that, if C_n is a sum $X_1+X_2+\cdots+X_n$ of n random variables independent and of the same distribution as X, then the equality $\mathbb{E}[u^{C_n}]=\big(\mathbb{E}[u^X]\big)^n$ holds and C_n follows an *exact* Gaussian law. This situation happens in the continuous model, as described in Section 3.4.

There are many cases where a sequence $(C_m)_m$ of variables has a distribution that does not lead to an exact power for $\mathbb{E}[u^{C_m}]$, but only to a "Quasi-Powers" phenomenon (as called by Hwang [15] who introduced it). In this Quasi-Powers framework, the estimate
$$\mathbb{E}[u^{C_m}]=K(u)\,A(u)^m\,(1+O(\epsilon_m))\,,\qquad \epsilon_m\to 0$$
holds on a complex neighbourhood of $u=1$, with analytic functions $K(u),A(u)$ and a uniform remainder $O(\epsilon_m)$. This situation is omnipresent in analytic combinatorics. It indeed occurs, here, in the discrete model, for most of the additive costs under study. When a supplementary condition, called the *admissibility condition*, is fulfilled by the first two derivatives of the function $A(u)$ at $u=1$, it may be proven that C_m follows an asymptotic Gaussian law. We now recall the definition in the framework of the discrete model.

Definition. *Consider the set $\mathcal{R}$ of inputs, the sequence $(\mathcal{R}_m)$ of subsets of the inputs of size m endowed with the uniform probability $\mathbb{P}$ and a cost $C:\mathcal{R}\to\mathbb{R}^+$.*
The sequence $(C_m)_m$ of the restrictions of C to $\mathcal{R}_m$ is said to follow an asymptotic Gaussian law if there exist

- *two sequences of real numbers (a_m), (b_m),*
- *a sequence (r_m) of functions $r_m\colon\mathbb{R}\to\mathbb{R}$, with $\lim_{m\to\infty}\sup\{r_m(y)\mid y\in\mathbb{R}\}=0$*

for which the following holds:
$$\mathbb{P}\left[R\in\mathcal{R}_m\;\mid\;\frac{C_m(R)-a_m}{\sqrt{b_m}}\le y\right]=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y}e^{-t^2/2}dt+r_m(y).$$
The expectation $\mathbb{E}[C_m]$ and the variance $\mathbb{V}[C_m]$ of the cost C_m then satisfy
$$\mathbb{E}[C_m]\sim a_m,\quad \mathbb{V}[C_m]\sim b_m.$$

In Section 4.6, we will provide a theorem, called Theorem A, that is ready for use in our framework and entails that the Quasi-Powers framework holds. If moreover the admissibility condition on $A(u)$ is fulfilled, we then obtain an asymptotic Gaussian law.

3.6. **Comparison between the models.** Even though the two probabilistic models are different, they study, in both cases, the cost of trajectories, as seen from (29) and (30). In the discrete model, the trajectories deal with discrete inputs (polynomial triples of $\mathcal{R}$) and they stop as soon as they reach an element of $\mathcal{D}$, after a finite number of steps $S = S(R)$ which is a random variable, depending on the input $R \in \mathcal{R}$. In the continuous model, the trajectories deal with continuous inputs (pairs $f \in \mathbb{L}^2$), and they are truncated in a deterministic way, after a given number n of steps, which does not depend on the inputs.

However, we establish the following strong relation between the two models, that already holds in the one-dimensional case, both in the real and the polynomial case (see for instance [32]):

> *The finite trajectories of the discrete model behave on average exactly in the same way as the truncated trajectories of the continuous model behave everywhere.*

This relation is strongly based on the relation between the bivariate generating function $H_c(z, u)$ relative to the step-cost c, and the transfer operator $\mathbf{H}_{c,u}$ associated with the same cost: as proved in (79), the dominant eigenvalue of the operator $\mathbf{H}_{c,u}$ equals $H_c(1/q^3, u)$.

3.7. **Comparison between the algorithms.** We use the estimates shown in Tables 6 and 8 to compare the three algorithms under study.

- The constants $M_c, \widetilde{M}_c, V_c, \widetilde{V}_c$ are rational fractions in q with integer coefficients.
- The constants M_c or $\widetilde{M}_c$ are in $\Theta(1)$ when $q \to \infty$ and their limit (for $q \to \infty$) depends neither on the algorithm nor on the model, with
$$M_{c_0} \sim 1, \quad M_{d_B} \sim \widetilde{M}_{d_B} \sim 2, \quad M_d \sim \widetilde{M}_d \sim 1, \quad M_\delta \sim \widetilde{M}_\delta \sim M_\nu \sim \widetilde{M}\nu \sim 3.$$
- The constants for the variance V_c in the discrete model exhibit two possible behaviours: the fully subtractive and the Jacobi–Perron algorithms have all their variance constants V_c in $\Theta(1/q)$ when $q \to \infty$, whereas they are in $\Theta(1)$ for the Brun algorithm. Moreover, the behaviour of the variances of the Jacobi–Perron and the fully subtractive algorithms are very close. They both satisfy
$$qV_{c_0} \sim 1, \quad qV_\delta \sim 2, \quad qV_\nu \sim 3,$$
and they slightly differ for the cost d, with
$$qV_d \sim 5 \quad \text{(JP)}, \qquad qV_d \sim 8 \quad \text{(FS)}.$$
- The variance constants $\widetilde{V}_c$ in the continuous model are in $\Theta(1/q)$ with various possible limits for $q\widetilde{V}_c$. However, there are two exceptions: first, for the FS algorithm, as the cost d is constant and equal to 2 (there are always two "real"divsions), the constant $\widetilde{V}_d$ is zero; second, for the Brun algorithm, the constant $\widetilde{V}_{d_B}$ is in $\Theta(1/q^2)$, a sort of reminiscence of the discrete case, where the constant V_{d_B} is zero.

4. Probabilistic analysis of the three gcd algorithms

The present section is devoted to the discrete case, and we analyse the behaviour of the three gcd algorithms on polynomials. As already mentioned, this is the first analysis conducted in this case. We use here analytic combinatorics, and, as recalled in Section 3.3, the main tools are generating functions (gf in short), first plain gf's, then bivariate gf's, and even trivariate gf's.

We first recall the analytic combinatoric framework for the classical Euclid algorithm in Section 4.1. We continue with three sections that implement the symbolic step (as described in Section 3.3). After Section 4.2 that describes the plain generating functions, the next three sections are devoted to bivariate generating functions (bgf's). Our final object of interest are the total costs defined in Section 3.1 and we are mainly interested in their bgf's, obtained in Proposition 6; however, we proceed by steps: first basic bgf's in Section 4.3 with Proposition 4, then bgf's

associated with a step-cost in Section 4.4 with Proposition 5, and finally bgf's associated with a total cost in Section 4.5. We indeed view most of them as trivariate generating functions (tgf's).

The next three sections are devoted to the analytic step of analytic combinatorics (see Section 3.3). We first state in Section 4.6 that an asymptotic Gaussian law holds for additive costs (see Section 3.5 for a definition); the proof of the result is done in the next two sections. Section 4.7 performs the general computations needed, then Section 4.8 applies the previous computations to each pair (algorithm, cost) and obtain constants that intervene in the mean values and in the variances.

The section ends with a study of the bit complexities in Section 4.9.

4.1. **Generating functions for the classical Euclid algorithm.** We deal with the set

$$\mathcal{P} = \{(R_1, R_2) \in \mathbb{F}_q^2[X] \mid \deg R_1 < \deg R_2, R_2 \text{ monic}\},$$

and the size of a pair R is the maximum degree of its two components, namely $\deg R_2$. The subset of the elements of $\mathcal{P}$ of size n (i.e., $\deg R_2 = n$) has cardinality

$$q^n \cdot \left(1 + \sum_{m=0}^{n-1} (q-1)q^m\right) = q^{2n},$$

and the generating function $P(z)$ of $\mathcal{P}$ is

$$P(z) := \sum_{n\geq 0} q^{2n} z^n = \frac{1}{1-q^2 z}\,.$$

The set $\mathcal{G}$ of possible quotients and the set $\mathcal{U}$ of possible gcd's are described in (3) and their generating functions are

$$G(z) = (q-1)\left(\frac{1}{1-qz} - 1\right) = \frac{(q-1)qz}{1-qz}, \qquad U(z) = \frac{1}{1-qz}\,. \tag{34}$$

Due to (4), and the symbolic method, Euclid's algorithm decomposes the generating function $P(z)$ of the inputs as

$$P(z) = \frac{U(z)}{1-G(z)}\,. \tag{35}$$

4.2. **Plain generating functions for the three algorithms.** The *size* of a triple $R = (R_1, R_2, R_3) \in \mathcal{R}$ is the maximum degree of its three components, namely $\deg R_3$. The finite set $\mathcal{R}_m$ gathers triples of polynomials of size m as

$$\mathcal{R}_m := \{R := (R_1, R_2, R_3) \mid m = \deg R_3 > \max(\deg R_1, \deg R_2), \quad R_3 \text{ monic }\}\,.$$

Its cardinality is equal to

$$q^m \cdot \left(1 + \sum_{k=0}^{m-1} (q-1)q^k\right)^2 = q^{3m}\,,$$

and the generating function $R(z)$ of $\mathcal{R}$ is thus

$$R(z) := \sum_{m\geq 0} q^{3m} z^m = \frac{1}{1-q^3 z}\,.$$

With the bijection (15), together with the principles of the symbolic method, the generating function $R(z)$ of the inputs decomposes as the product of two terms, namely

$$R(z) = \frac{1}{1-q^3 z} = \frac{D(z)}{1-H(z)}\,. \tag{36}$$

Here, the generating function $D(z)$ of the set $\mathcal{D}$ is easily obtained from its combinatorial description given in Table 1, that is transferred, via the symbolic step, into the third line of Table 2. Moreover,

the generating function $H(z)$ of the set $\mathcal{H}$ of quotients (where the size of a quotient (A,B) is the largest degree of the pair, i.e., $\deg B$) involves the generating function $L(z)$ of the set $\mathcal{L}$:

(37)
$$L(z) = \sum_{(A,B)\in\mathcal{L}} z^{\deg B} = \sum_{n\geq 1}(q-1)q^n z^n \sum_{m=0}^{n-1}(q-1)q^m = (q-1)\sum_{n\geq 1}(qz)^n(q^n-1) = G(zq) - G(z)\,.$$

Finally, in the three cases, $H(z)$ is expressed with $G(z)$ and $G(zq)$, as seen from the first two lines of Table 2. In the three cases, we may check by computation the equality (36), that itself entails the following two equalities at $z = 1/q^3$:

(38)
$$H(1/q^3) = 1, \qquad (1/q^3)H'(1/q^3) = D(1/q^3)\,.$$

	Jacobi–Perron	Brun	Fully subtractive
$H(z)$	$L(z)+G(z)$ $= G(zq)$	$(q+1)G(z)$ $= (q+1)G(z)$	$2L(z)-(q-1)G(z)$ $= 2G(zq)-(q+1)G(z)$
$D(z)$	$P(z)$	$U(z)$	$2P(z)-U(z)$

TABLE 2. Expressions of the plain generating functions in terms of $L(z)$ and $G(z)$. As seen from the Sum Property of Proposition 2, the sum of the last two columns is equal to twice the first column.

4.3. **Basic bivariate generating functions.** The plain generating function $H(z)$ of $\mathcal{H}$, relative to the size $\deg B = \max(\deg B, \deg A)$, does not take into account the role played by the polynomial A. We thus introduce a further variable u: for $A \neq 0$, it marks the total degree, i.e., $\deg A + \deg B$, and for $A = 0$, it marks $\deg B$. The associated bivariate generating functions are

(39)
$$H^{(2)}(z,u) = \sum_{\substack{(A,B)\in\mathcal{H}\\ A\neq 0}} z^{\deg B}\, u^{\deg B+\deg A}, \qquad H^{(1)}(z,u) = \sum_{\substack{(A,B)\in\mathcal{H}\\ A=0}} z^{\deg B}\, u^{\deg B}\,.$$

Due to Proposition 1 which relates the sets $\mathcal{H}^{\langle i\rangle}$ to sets $\mathcal{L}$ and $\mathcal{G}$, we are led to the bivariate generating functions $L(zu,u)$ and $G(zu)$ where the generating function $L(z,u)$ involves the set $\mathcal{L}$ defined in (23) and satisfies

$$L(z,u) := \sum_{(A,B)\in\mathcal{L}} z^{\deg B} u^{\deg A} = \sum_{n\geq 1}(q-1)q^n z^n \sum_{m=0}^{n-1}(q-1)q^m u^m = \frac{(q-1)^2}{qu-1}\sum_{n\geq 1}(qz)^n((qu)^n-1)\,.$$

Then, $L(z,u)$ is expressed as a finite difference which involves the function G computed in (34)

(40)
$$L(z,u) = (q-1)\frac{G(zqu)-G(z)}{qu-1}\,, \qquad L(z) = L(z,1) = G(zq) - G(z)\,.$$

Then, Proposition 1 entails the expressions of the two bivariate generating functions $H^{(i)}(z,w)$ in terms of $L(zw,w)$ and $G(zw)$ that are provided in Table 3.

	Jacobi–Perron	Brun	Fully subtractive
$H^{\langle 2\rangle}(z,w)$	$L(zw,w)$	$(q-1)G(zw)$	$2L(zw,w)-(q-1)G(zw)$
$H^{\langle 1\rangle}(z,w)$	$G(zw)$	$2G(zw)$	0

TABLE 3. Expressions of $H^{\langle i\rangle}(z,w)$ in terms of $L(zw,u)$ and $G(zw)$. As seen from the Sum Property of Proposition 2, the sum of the last two columns is equal to twice the first column.

With Table 3 and (40), we derive the following result:

Proposition 4. [Expressions of the basic bgf's] *The following matricial expressions hold for the bivariate bgf's $H^{\langle i\rangle}(z,w)$ $(i=1,2)$, namely*

$$\begin{bmatrix} H_{\rm JP}^{\langle i\rangle}(z,w) \\ H_{\rm B}^{\langle i\rangle}(z,w) \\ H_{\rm FS}^{\langle i\rangle}(z,w) \end{bmatrix} = \Gamma^{\langle i\rangle}(w) \begin{bmatrix} G(zqw^2) \\ G(zw) \end{bmatrix}, \tag{41}$$

expressed[9] *in terms of the matrices $\Gamma^{\langle i\rangle}(w)$, defined as*

$$\Gamma^{\langle 2\rangle}(w) := \left(\frac{q-1}{qw-1}\right) \begin{bmatrix} 1 & -1 \\ 0 & (qw-1) \\ 2 & -(qw+1) \end{bmatrix}, \qquad \Gamma^{\langle 1\rangle}(w) := \begin{bmatrix} 0 & 1 \\ 0 & 2 \\ 0 & 0 \end{bmatrix}. \tag{42}$$

4.4. **Bivariate generating functions associated with a step-cost.** We are now ready to study the bgf $H_c(z,u)$ relative to a step-cost of interest $c \in \{c_0, d, d_b, \delta, \nu\}$, defined in (28). With a cost c defined on the pair (A,B) with $c(A,B) = c_A(A) + c_B(B)$ and $c(0) = 0$, we associate

$$H_c(z,u) := \sum_{(A,B)\in\mathcal{H}} z^{\deg B} u^{c_A(A)+c_B(B)}.$$

Costs $c = c_0, d_B$. According to (39), the bivariate generating functions $H_c(z,u)$ of interest are, first, for the step-cost $c = c_0$ (corresponding to the cost S) and for the cost $c = d_B$

$$H_1(z,u) = uH(z), \quad H_{d_B}(z,u) = H(uz).$$

Costs $c = d, \delta, \nu$. The bivariate generating functions relative to the number of divisions d or the number of monomials (δ or ν) that appear in the pair (A,B) are

$$H_d(z,u) = u^2 H^{\langle 2\rangle}(z) + uH^{\langle 1\rangle}(z), \quad H_\delta(z,u) = u^2 H^{\langle 2\rangle}(z,u) + uH^{\langle 1\rangle}(z,u),$$

$$H_\nu(z,u) = u^2 H^{\langle 2\rangle}(z,t) + uH^{\langle 1\rangle}(z,t), \qquad \text{with} \quad t := u\left(\frac{q-1}{q}\right) + \left(\frac{1}{q}\right).$$

Indeed, for the costs ν, δ that count the number of monomials, the multiplication of the $H^{\langle i\rangle}$ terms, by u or u^2 respectively, corresponds to the leading term (which is non-zero), whereas, for the cost ν, the variable u inside $H^{(i)}(z,u)$ is indeed replaced by the variable $t = u(q-1)/q + (1/q)$ which is associated with the Bernoulli law on $\mathbb{F}_q$ defined by

$$\mathbb{P}[x=0] = \tfrac{1}{q}, \quad \mathbb{P}[x\neq 0] = \tfrac{q-1}{q}.$$

Using now *trivariate generating functions* (tgf's), we define a *unified* framework for the bgf's relative to all these five step-costs. We indeed consider the two trivariate gf's

$$I(z,u,w) := uH(zw), \qquad J(z,u,w) := u^2 H^{\langle 2\rangle}(z,w) + uH^{\langle 1\rangle}(z,w); \tag{43}$$

we associate furthermore with a cost c, a function γ_c defined as

$$\gamma_c(u) = (u,1) \ \ (c=1), \quad \gamma_c(u) = (1,u) \ \ (c=d_B), \quad \gamma_c(u) = (u, w_c(u)) \ \ (c = d,\delta,\nu), \tag{44}$$

that involves itself the function $w_c(u)$ that satisfies

$$w_d(u) = 1, \qquad w_\delta(u) = u \qquad w_\nu(u) = u\tfrac{q-1}{q} + \tfrac{1}{q}. \tag{45}$$

This leads to the equalities

$$\text{for costs } c_0, d_B\text{:} \quad H_c(z,u) = I(z,\gamma_c(u)), \qquad \text{for costs } d,\delta,\nu\text{:} \quad H_c(z,u) = J(z,\gamma_c(u)),$$

and using now Proposition 4, we obtain the following result.

[9] The Sum Property of Proposition 2 is now expressed on the lines of the matrices.

Proposition 5. [Expressions for the bgf's relative to a step-cost] *The bgf $H_c(z,u)$ is viewed in all the cases as a trivariate generating function (tgf) $H(z,u,w)$ that is expressed in terms of the tgf's I and J defined in* (43) *and of the function γ_c defined in* (44).
Consider the matrix $\Gamma(u,w)$ that is expressed in terms of matrices $\Gamma^{\langle i\rangle}(u,w)$ defined in (42) *as*[10]

$$\Gamma(u,w) = (qw-1)(u^2\Gamma^{\langle 2\rangle}(w) + u\Gamma^{\langle 1\rangle}(w))\,,$$

$$\text{namely} \qquad \Gamma(u,w) = \begin{bmatrix} u^2(q-1) & -u^2(q-1)+u(qw-1) \\ 0 & u^2(qw-1)(q-1)+2u(qw-1) \\ 2(q-1)u^2 & -(q-1)u^2(qw+1) \end{bmatrix}, \tag{46}$$

and for which the Sum Property of Proposition 2 holds on the lines. Then,

- *For costs c_0, d_B, the bgf $H_c(z,u)$ involves the tgf I and equals $I(z,\gamma_c(u))$. The tgf $I(z,u,w)$ satisfies*

$$\begin{bmatrix} I_{\rm JP}(z,u,w) \\ I_{\rm B}(z,u,w) \\ I_{\rm FS}(z,u,w) \end{bmatrix} = u\frac{\Gamma(1,1)}{q-1}\begin{bmatrix} G(zqw) \\ G(zw) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & (q+1) \\ 2 & -(q+1) \end{bmatrix}\begin{bmatrix} G(zqw) \\ G(zw) \end{bmatrix}. \tag{47}$$

- *For costs d, δ, ν, the bgf $H_c(z,u)$ involves the tgf J and equals $J(z,\gamma_c(u))$. The tgf $J(z,u,w)$ satisfies*

$$\begin{bmatrix} J_{\rm JP}(z,u,w) \\ J_{\rm B}(z,u,w) \\ J_{\rm FS}(z,u,w) \end{bmatrix} = \frac{\Gamma(u,w)}{qw-1}\begin{bmatrix} G(zqw^2) \\ G(zw) \end{bmatrix}. \tag{48}$$

4.5. **Bivariate generating functions for total additive costs.** We now focus on the total costs C that occur during the execution of the non-degenerate phase of the algorithm.

We *do not* consider costs of the degenerate phase. Such a study would involve a bivariate generating function $D(z,u)$ that would be a variation of the plain generating function $D(z)$ of the degenerate phase. We already know that $1/q^3$ is the unique pole of $R(z)$ (see (36)) whereas $D(z)$ has a dominant pole located at $1/q^2$ (in the Jacobi–Perron or in the fully subtractive cases) or $1/q$ (in the Brun case), that is always larger than $1/q^3$. This entails that the growth order of the asymptotic costs of the degenerate phase will be strictly smaller than those of the degenerate phase.

As the cost C is the total cost associated with the step-cost c, the symbolic method recalled in Section 3.3 proves that the bgf $R_C(z,u)$ is obtained by replacing in the expression (36) the quasi-inverse $D(z)/(1-H(z))$ by the quasi-inverse $D(z)/(1-H_c(z,u))$. As $H_c(z,u)$ is itself viewed as a tgf $H(z,u,w)$, this leads to the expression of $R_c(z,u)$ as a tgf $R(z,u,w)$:

$$R_C(z,u) = \frac{D(z)}{1-H_c(z,u)}, \qquad R(z,u,w) = \frac{D(z)}{1-H(z,u,w)} \tag{49}$$

where $H(z,u,w)$ is described in Proposition 5. The denominator $1-H(z,u,w)$ thus plays a central role, as it will bring the dominant singularity. It depends on both the algorithm and the cost.

We first obtain a general form for this denominator that is always a polynomial (in z) of degree at most two. The present section then continues with five different classes that occur in a more precise description of the denominator $1-H(z,u,w)$. Finally, Proposition 6 describes the tgf's $R(z,u,w)$ in these five cases. This will be the final step of the symbolic study.

Bivariate generating functions $R_c(z,u)$ and the Sum Property. Before dealing with precise computations, we first provide a general point of view. With each step-cost c and each algorithm $\sharp$, we thus associate a bivariate generating function

$$R_{C,\sharp}(z) = \frac{D_\sharp(z)}{1-H_{c,\sharp}(z,u)} = \frac{D_\sharp(z)}{1-H_\sharp(z,u,w)}\,.$$

[10] The Sum Property of Proposition 2 is now expressed on the lines of the matrices.

In fact, further analysis (see Theorem A) shows that only the following denominators are involved

$$1 - H_{c,JP}(z,u) = 1 - H_{JP}(z,u,w),$$
$$1 - H_{c,B}(z,u) = 1 - H_B(z,u,w), \quad 1 - H_{c,FS}(z,u) = 1 - H_{FS}(z,u,w)\,,$$

and the Sum Property holds for these denominators. For each pair $(c,\sharp)$, the analysis of the cost (see Theorem A) involves more precisely the functions $u \mapsto \rho_{c,\sharp}(u)$ for which the equality $1 - H_{c,\sharp}(\rho_{c,\sharp}(u),u) = 1$ holds, via their inverses $\rho^{-1}_{c,\sharp}$. This explains why the Sum Property does not really intervene in the study of the discrete model[11].

General form for $R_C(z,u)$**.** As was already the case for $H(z,u,w)$, there are two different cases depending on whether the step-cost c belongs to $\{1,d_B\}$ or $\{d,\delta,\nu\}$.

Costs d,δ,ν. One has, with (48) and (46),

$$\begin{bmatrix} 1-H_{\rm JP}(z,u,w)\\ 1-H_{\rm B}(z,u,w)\\ 1-H_{\rm FS}(z,u,w)\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix} - \frac{\Gamma(u,w)}{qw-1}\begin{bmatrix}G(zqw^2)\\ G(zw)\end{bmatrix}.$$

Then, by recalling the expression of G as $G(z) = (q-1)qz/(1-qz)$, the trivariate generating functions (relative to costs d,δ,ν) satisfy the following matricial relations

$$(qw-1)(1-zq^2w^2)(1-zqw)\begin{bmatrix} 1-H_{\rm JP}(z,u,w)\\ 1-H_{\rm B}(z,u,w)\\ 1-H_{\rm FS}(z,u,w)\end{bmatrix}$$
$$= (qw-1)(1-zq^2w^2)(1-zqw)\begin{bmatrix}1\\1\\1\end{bmatrix} - (q-1)\Gamma(u,w)\begin{bmatrix}zq^2w^2(1-zqw)\\ zqw(1-zq^2w^2)\end{bmatrix}.$$

The bgf $R_C(z,u)$ (relative to the step-costs d,δ,ν) is thus written as

$$R_c(z,u) = R(z,\gamma_c(u)), \quad \text{with} \quad R(z,u,w) = D(z)\frac{(qw-1)(1-zq^2w^2)(1-zqw)}{Q(u,w)(z)}\,. \tag{50}$$

Here, the denominator $Q(u,w)(z) = Q_\sharp(u,w)(z)$ is a polynomial of degree at most two in z, i.e.,

$$Q_\sharp(u,w)(z) = (qw-1)(1-zq^2w^2)(1-zqw) - E_\sharp(u,w)zq^2w^2(1-zqw) - F_\sharp(u,w)zqw(1-zq^2w^2)\,,$$

where $E_\sharp$ and $F_\sharp$ are the coefficients of the $\sharp$ line of the matrix $(q-1)\Gamma(u,w)$ described in (46).

Costs c_0, d_B. In the same vein, for the costs c_0 and d_B, the bgf $H_c(z,u)$ is viewed as a trivariate generating function that satisfies, with (47),

$$(1-zq^2w)(1-zqw)\begin{bmatrix} 1-H_{\rm JP}(z,u,w)\\ 1-H_{\rm B}(z,u,w)\\ 1-H_{\rm FS}(z,u,w)\end{bmatrix} = (1-zq^2w)(1-zqw)\begin{bmatrix}1\\1\\1\end{bmatrix} - u\Gamma(1,1)\begin{bmatrix}zq^2w(1-zqw)\\ zqw(1-zq^2w)\end{bmatrix}.$$

The bivariate generating function $R_C(z,u)$ (relative to step-costs c_0, d_B) is thus written as

$$R_C(z,u) = R(z,\gamma_c(u)) \quad \text{with} \quad R(z,u,w) = D(z)\frac{(1-zq^2w)(1-zqw)}{Q(u,w)(z)} \tag{51}$$

where the denominator $Q(u,w)(z) = Q_\sharp(u,w)(z)$ is a polynomial of degree at most two in z, i.e.,

$$Q(u,w)(z) = Q_\sharp(u,w)(z) = (1-zq^2w)(1-zqw) - uE_\sharp\, zq^2w(1-zqw) - uF_\sharp\, zqw(1-zq^2w)\,,$$

where $E_\sharp$ and $F_\sharp$ are the coefficients of the $\sharp$ line of the matrix $M(1,1)$ described in (46).

Degree of polynomial $Q(u,w)$**.** There are cases where the polynomial $Q_\sharp(u,w)$ involved in (50) or in (51) is in fact of degree 1. This situation arises when one of the two coefficients $E_\sharp$ and $F_\sharp$ is zero, and there is a simplification with the numerator. This is the case for the five step-costs of the Brun algorithm (with $E_\sharp = 0$), or for the step-costs c_0 or d_B of the JP algorithm (with $F_\sharp = 0$). These seven cases lead to trivariate gf's where the denominator is a polynomial of degree 1 in z and are then gathered in a subclass called the *linear* class (LC in short), whereas the other

[11] We will see that the Sum Property directly intervenes in the analysis of the continuous model.

cases (all step-costs for the FS algorithm and step-costs d, δ, ν for the JP algorithm) are gathered in a subclass called the *quadratic* class (QC in short).

Linear class. There are three subclasses, called LC1, LC2 and LC3.

LC1. For the costs c_0 and d_B of the Brun algorithm, the polynomial $Q_B(u,w)(z)$ factorizes as $Q_B(u,w)(z) = (1 - zq^2w)(1 - zqw(1 + uF_B)$ with $F_B = (q^2 - 1)$ and

$$R_B(z,u,w) = \frac{D_B(z)(1-zqw)}{1 - zqw(1 + u(q^2-1))}, \qquad \widetilde{Q}(u,w)(z) = 1 - zqw(1 + u(q^2-1))\,. \tag{52}$$

LC2. For the costs c_0 and d_B of the JP algorithm, the polynomial $Q_{JP}(u,w)(z)$ factorizes as $Q_{JP}(u,w)(z) = (1 - zqw)(1 - zq^2w(1 + uE_{JP})$ with $E_{JP} = (q-1)$ and

$$R_{JP}(z,u,w) = \frac{D_{JP}(z)(1-zq^2w)}{1 - zq^2w(1 + u(q-1))}, \qquad \widetilde{Q}(u,w)(z) = 1 - zq^2w(1 + u(q-1))\,. \tag{53}$$

LC3. For the costs d, δ, ν of the Brun algorithm, one has $F_B(u,w) = (q-1)(qw-1)[u^2(q-1)+2u]$, the polynomial $Q_B(u,w)(z)$ factorizes as

$$\begin{aligned} Q_B(u,w)(z) &= (qw-1)(1 - zq^2w^2)[1 - z\,qw - (u^2(q-1)^2 + 2u(q-1))zqw)] \\ &= (qw-1)(1 - zq^2w^2)[1 - zqw(1 + u^2(q-1)^2 + 2u(q-1)] \end{aligned}$$

$$\text{and} \quad R_B(z,u,w) = \frac{D_B(z)(1-zqw)}{1 - zqw\,(1 + u(q-1))^2}, \qquad \widetilde{Q}(u,w)(z) = 1 - zqw(1 + u(q-1))^2\,. \tag{54}$$

Quadratic class. In the quadratic subclass, there are also two subcases. The first class, called QC1, gathers the costs d, δ, ν for algorithms FS and JP, whereas the second class, called QC2, gathers the costs c_0, d_B for the FS algorithm; see Table 4 for a summary.

QC1. We begin with (50). The polynomial $Q(u,w)(z)$ is a polynomial of the second degree in the variable z, i.e.,

$$Q(u,w)(z) = \widehat{b}(u,w)z^2 - z\widehat{a}(u,w) + (qw-1) \tag{55}$$

whose coefficients are

$$\begin{cases} \widehat{b}(u,w) & = \widehat{b}_\sharp(u,w) & = (qw)^3(qw - 1 + E_\sharp(u,w) + F_\sharp(u,w))\,, \\ \widehat{a}(u,w) & = \widehat{a}_\sharp(u,w) & = (qw)^2(qw - 1 + E_\sharp(u,w)) + (qw)(qw - 1 + F_\sharp(u,w)) \end{cases}$$

In each algorithm (JP or FS), each coefficient $\widehat{a}_\sharp(u,w)$ or $\widehat{b}_\sharp(u,w)$ is written as

$$\widehat{a}_\sharp(u,w) = \widehat{F}(w) + \widehat{G}(w)K_\sharp(u), \qquad \widehat{b}(u,w) = \widehat{L}(w)O_\sharp(u)\,,$$

where $\widehat{F}, \widehat{G}, \widehat{L}$ satisfy

$$\widehat{F}(w) = qw(qw-1)(qw+1), \quad \widehat{L}(w) = (qw)^3(qw-1), \quad \widehat{G}(w) = qw(qw-1)\,.$$

Moreover, the polynomials $K_\sharp$ and $O_\sharp$ satisfy

$$\text{(JP case)} \quad K_{\mathrm{JP}}(u) = u^2(q-1)^2 + u(q-1), \quad O_{\mathrm{JP}}(u) = 1 + u(q-1) \tag{56}$$

$$\text{(FS case)} \quad K_{\mathrm{FS}}(u) = u^2(q-1)^2, \quad O_{\mathrm{FS}}(u) = 1 - (q-1)^2u^2\,. \tag{57}$$

Then $(qw-1)$ is a common factor of the three terms in (55). We then work with the following reduced version of $Q(u,w)(z)$

$$\widetilde{Q}(u,w)(z) = \widetilde{Q}_\sharp(u,w)(z) := b_\sharp(u,w)z^2 - za_\sharp(u,w) + 1 \tag{58}$$

where the coefficients

$$a_\sharp(u,w) := F(w) + G(w)K_\sharp(u), \quad b\sharp(u,w) = L(w)O_\sharp(u)\,, \tag{59}$$

involve K, O defined previously together with

$$F(w) = qw(qw+1), \quad G(w) = qw, \quad L(w) = (qw)^3\,. \tag{60}$$

QC2. We begin with (51). The polynomial $Q(u,w)(z)$ is a polynomial of the second degree in the variable z, i.e.,

$$Q(u,w)(z) = Q_\sharp(u,w)(z) = (1-zq^2w)(1-zqw) - 2u(q-1)zq^2w(1-zqw) + u(q^2-1)zqw(1-zq^2w)\,.$$

We then set $\widetilde{Q}(u,w)(z) = Q(u,w)(z)$. This gives

$$\widetilde{Q}(u,w)(z) = \widetilde{Q}_\sharp(u,w)(z) := b_\sharp(u,w)z^2 - za_\sharp(u,w) + 1\,, \tag{61}$$

$$\text{where} \qquad a_\sharp(u,w) := F(w) + G(w)K(u), \quad b_\sharp(u,w) = L(w)O(u)\,, \tag{62}$$

$$F(w) = qw(q+1), \quad G(w) = qw, \quad L(w) = q^3w^2, \quad O(u) = 1-u(q-1)^2, \quad K(u) = u(q-1)^2\,. \tag{63}$$

Now, in all the cases (LC and QC), one checks that $1/q^3$ is the smallest root of $\widetilde{Q}_\sharp(u,w)(z)$ for $(u,w) = (1,1)$.

The study is summarized in the following proposition:

Proposition 6. [Expression of the bgf's relative to a total cost] *For any of the five step-costs c, the bgf $(R_C(z,u)/D(z))$ of the total cost C is a rational fraction with respect to z viewed as a trivariate generating function (tgf) $R(z,u,w)$ where $(u,w) = \gamma_c(u)$ with γ_c defined in* (44).

- *For the subclass QC1 (the Jacobi–Perron and the fully subtractive algorithms for the costs d, δ, ν), the tgf is given in* (50) *and it involves the polynomial $\widetilde{Q}(u,w)(z)$ given in* (58).
- *For the subclass QC2 (costs c_0, d_B for the FS algorithm), the tgf is given in* (51) *and it involves the polynomial $\widetilde{Q}(u,w)$ given by* (61).
- *For the subclasses of the linear class (the Brun algorithm for all the costs, and the Jacobi–Perron algorithm for the step-costs c_0 and d_B), the tgf's are given in* (52), (54) *and* (53).

Table 4 describes the result according to the pair (algorithm, cost).

Step-cost c	Jacobi–Perron	Brun	Fully subtractive
Cost c_0	LC2 Eqn (53)	LC1 Eqn (52)	QC2 Eqn (51), (61)
Degree d_B	LC2 Eqn (53)	LC1 Eqn (52)	QC2 Eqn (51), (61)
Cost d	QC1 Eqn (50), (58)	LC3 Eqn (54)	QC1 Eqn (50) (58)
Cost δ	QC1 Eqn (50), (58)	LC3 Eqn (54)	QC1 Eqn (50), (58)
Cost ν	QC1 Eqn (50), (58)	LC3 Eqn (54)	QC1 Eqn (50), (58)

TABLE 4. Distribution of the five subclasses – three linear subclasses (LC1, LC2, LC3) and two quadratic subclasses (QC1, QC2) – according to the pair formed by an algorithm and a cost. The number of the equation provides the expression of the bivariate generating function and the polynomial $\widetilde{Q}$.

4.6. **Asymptotic Gaussian law for additive costs.** We have now a precise expression for the bgf $R_C(z,u)$ associated with a total cost C defined from the step cost $c \in \{c_0, d, d_B, \delta, \nu\}$. This bgf is a rational function. The symbolic step ends, and we now begin the analytic step of analytic combinatorics.

Section 3.5 reminds us of what is an asymptotic Gaussian law and why asymptotic Gaussian laws are expected for costs C, in a "Quasi-Powers" framework. This is the case here, and the following theorem, which is the main result of this section, indeed proves that such asymptotic Gaussian laws occur, as soon as the step cost is not d_B.

Theorem 1. *Consider the input set*

$$\mathcal{R} := \{R := (R_1, R_2, R_3) \mid \deg R_3 > \max(\deg R_1, \deg R_2), \quad R_3 \text{ monic}\}$$

together with its subsets $\mathcal{R}_m$ of size m, the three algorithms of interest, and the four additive costs $c \in \{c_0, d, \delta, \nu\}$. For each algorithm, and each total cost C associated with c, an asymptotic Gaussian law holds.
Furthermore, the expectation $\mathbb{E}[C_m]$ and the variance $\mathbb{V}[C_m]$ are asymptotically linear with respect to the size m, and, for any pair $(\sharp, c)$, there exist constants $M_c = M_{\sharp,c}$ and $V_c = V_{\sharp,c}$ such that the expectations and the variances of the cost c satisfy

$$\mathbb{E}[C_m] \sim M_c \cdot m \qquad \mathbb{V}[C_m] \sim V_c \cdot m\,.$$

The constants M_c and V_c are given in Table 6.

We have excluded the cost d_B. The cost d_B is constant on each $\mathcal{R}_m$ and equal to m. It cannot be asymptotically Gaussian as confirmed by the results below.

The proof of Theorem 1 is based on a general theorem of analytic combinatorics, given in Theorem IX. 9 (page 656 of [11]). The general statement of this theorem is given when the bivariate generating function of the cost is meromorphic. This theorem is itself a consequence of the Quasi-Powers theorem due to Hwang [15]. Here, we are in a specific case where the bivariate generating function is a rational fraction whose denominator is a *polynomial of degree at most two* in z. This is why we provide a specific theorem, called Theorem A, that is a particular case of Theorem IX. 9 in [11] and that is ready for use here.

We first recall the following notation: for a function $u \mapsto f(u)$ that satisfies $f(1) = 1$, we let

$$\mathfrak{M}(f) := f'(1), \qquad \mathfrak{V}(f) := f''(1) + f'(1) - f'(1)^2\,. \tag{64}$$

Theorem A. [Adaptation of Theorem IX.9 in [11]] *We consider a total cost C associated with a step cost c. We assume that its bivariate generating function $R_C(z, u)$ is also a trivariate generating function $R(z, u, w)$ of the form*

$$R_C(z, u) = R(z, u, w) = \frac{N(w)(z)}{\widetilde{Q}(u, w)(z)}\,,$$

where $N(w)(z)$ is a rational fraction in z and $\widetilde{Q}(u, w)(z)$ is a non-constant polynomial of degree at most two in z, both having coefficients that are polynomials in u and w.
We assume that the following holds:

- *(i) The plain generating function $R(z) = R(z, 1, 1)$ has a denominator $\widetilde{Q}(1, 1)(z)$ which has two distinct roots (when it is of degree 2), and the smallest root is located at $z = \rho$. The numerator $N(w)(z)$ is analytic and takes non-zero values on a domain of the form*

  $$\{(z, w) \mid z \text{ in a neighbourhood of } \{z \mid |z| < \rho\}, w \text{ in a neighbourhood of } w = 1\}.$$

- *(ii) There exists a non-constant analytic function $\rho(u, w)$ analytic at $(1, 1)$ that defines on a real neighbourhood of $(1, 1)$ the smallest root of the polynomial $\widetilde{Q}(u, w)(z)$. We denote by $\rho_c(u)$ the function equal to $\rho(\gamma_c(u))$ where the function γ_c is defined in (44).*
- *(iii) The admissibility condition holds : the term $\mathfrak{V}\left(\frac{\rho}{\rho_c(u)}\right)$ is non-zero.*

Then, an asymptotic Gaussian law holds for the restriction C_m of cost C to $\mathcal{R}_m$. Moreover, the expectation and the variance of the cost C_m are linear with respect to m and satisfy

$$\mathbb{E}[C_m] = m\,\mathfrak{M}\left(\frac{\rho}{\rho_c(u)}\right) + O(1), \qquad \mathbb{V}[C_m] = m\,\mathfrak{V}\left(\frac{\rho}{\rho_c(u)}\right) + O(1). \tag{65}$$

We will prove that the bgf's $R_C(z, u)$ associated with a total cost C defined from the four step-costs $c \in \{c_0, d, \delta, \nu\}$ fullfill all the hypotheses of the previous theorem. This will entail Theorem 1. Even though we know that the cost $c = d_B$ cannot be Gaussian, we perform the computation also in this case, since it will be partly needed in the analysis of the continuous model.

Hypothesis (i) clearly holds with $\rho = 1/q^3$ in our five cases $(QC1, QC2, LC1, LC2, LC3)$, according to Formulae (50), (51), (52), (54), (53). It remains to perform two tasks:

- The first task studies the function

$$g_c(u) = \frac{\rho}{\rho_c(u)}\,, \qquad \text{where } \rho_c \text{ is defined via the equality } \; H_c(\rho_c(u), u) = 1\,, \tag{66}$$

  and is already mentioned in the beginning of Section 4.5. One performs a general computation of the constants $\mathfrak{M}[g_c]$ and $\mathfrak{V}[g_c]$ which appear in (65).
- The second task computes these constants fore each pair (algorithm, cost) and checks that condition *(iii)* holds (except in the case of the cost d_B).

We begin with the first task, performed in the next section. The second task will be performed in Section 4.8.

4.7. **General expressions of constants relative to the mean and variance.** We compute the constants $\mathfrak{M}[g_c]$ and $\mathfrak{V}[g_c]$ which appear in (65). There are five different cases, according to the three linear subclasses and the two quadratic subclasses. We begin with the linear class.

(A) ***Linear class***. There is only one root located at $\rho(u, w)$, with

$$\begin{array}{lllll}
\text{(LC1)} & \rho(u,w)^{-1} = qw(1+u(q^2-1)), & g_{c_0}(u) = q^{-2}(1+u(q^2-1)), & & g_{d_B}(u) = u\,. \\
\text{(LC2)} & \rho(u,w)^{-1} = q^2w(1+u(q-1)) & g_{c_0}(u) = q^{-1}(1+u(q-1)), & & g_{d_B}(u) = u\,. \\
\text{(LC3)} & \rho(u,w)^{-1} = qw\,(1+u(q-1))^2 & g_c(u) = q^{-2}w_c(u)(1+u(q-1))^2\,, & &
\end{array}$$

where $w_c(u)$ is defined in (45).

We then compute $\mathfrak{M}[g_c]$ and $\mathfrak{V}[g_c]$ in each case (see Table 6). We already check that $\mathfrak{V}[g_{d_B}]$ is indeed equal to 0 in cases (LC1) and (LC2).

(b) ***Quadratic class***. There is a common framework for the two cases QC1 and QC2. As we are interested in the function $g_c(u)$ which involves the function $\rho_c(u)^{-1}$, we deal with *reciprocical* polynomials which moreover involve pairs (u, w) equal to $\gamma_c(u)$. Omitting the explicit references to the cost and the algorithm, we are then led to the polynomial

$$\widehat{Q}(z, u) = z^2 - a(u)z + b(u)\,, \qquad a(u) := a(\gamma_c(u)), \quad b(u) := b(\gamma_c(u)$$

where $a(u, w)$ and $b(u, w)$ are previously defined in (59) and (62). We first study the function f for which the polynomial $\widehat{Q}$ satisfies $\widehat{Q}(f(u), u) = 0$. We then return to the function $g = \rho \cdot f$ with $\rho = 1/q^3$.

We consider the derivatives of the polynomial $\widehat{Q}(z, u)$:

$$\begin{array}{ll}
\widehat{Q}'_z(z,u) = 2z - a(u), & \widehat{Q}'_u(z,u) = -za'(u) + b'(u) \\
\widehat{Q}''_{z^2}(z,u) = 2, & \widehat{Q}''_{u^2}(z,u) = -za''(u) + b''(u), \quad \widehat{Q}''_{zu}(z,u) = -a'(u)\,.
\end{array}$$

In particular, at point $(q^3, 1)$, the derivatives involve the function

$$\sigma(u) := a(u) - \frac{b(u)}{q^3}\,, \tag{67}$$

and are equal to

$$\begin{array}{lll}
\widehat{Q}'_z(q^3,1) = 2q^3 - a(1), & \widehat{Q}'_u(q^3,1) = -q^3a'(1) + b'(1) = -q^3\sigma'(1) & \\
\widehat{Q}''_{z^2}(q^3,1) = 2, & \widehat{Q}''_{u^2}(q^3,1) = -q^3\sigma''(1), & \widehat{Q}''_{zu}(q^3,1) = -a'(1)\,.
\end{array}$$

At $u = 1$, the polynomial $\widehat{Q}(f(u), u)$ equals $\widehat{Q}(q^3, 1)$. As q^3 is a simple root of $\widehat{Q}(z, 1)$, this entails that the derivative $\widehat{Q}'_z(q^3, 1) = 2q^3 - a(1)$ is not equal to zero.

The relation $\widehat{Q}(f(u), u) = 0$ holds. Taking the first derivative leads to the equality

$$f'(u)\,\widehat{Q}'_z(f(u), u) + \widehat{Q}'_v(f(u), u) = 0\,.$$

This gives at point $(q^3,1)$ the relation $f'(1)\,(2q^3-a(1))-q^3a'(1)+b'(1)=0\,.$
This determines the value of $f'(1)$, and then, the value of $\mathfrak{M}[g]$:

$$\mathfrak{M}[g]=g'(1)=\frac{\sigma'(1)}{2q^3-a(1)}\,. \tag{68}$$

The second derivative of the function $u\mapsto \widehat{Q}(f(u),u)$ is equal to

$$f''(u)\,\widehat{Q}'_z(f(u),u)+f'(u)^2\,\widehat{Q}''_{z^2}(f(u),u)+2f'(u)\,\widehat{Q}''_{zu}(f(u),u)+\widehat{Q}''_{v^2}(f(u),u)=0\,.$$

This gives at $(q^3,1)$ the relation $f''(1)(2q^3-a(1))+2f'(1)^2-2a'(1)f'(1)-q^3\sigma''(1)=0.$
This determines the value of $g''(1)$, namely

$$g''(1)(2q^3-a(1))=\sigma''(1)-2q^3g'(1)^2+2a'(1)g'(1)\,,$$

and entails the value of $(2q^3-b(1))\mathfrak{V}[g]$, equal to

$$(2q^3-a(1))\,\mathfrak{V}[g]=\sigma''(1)+g'(1)^2(a(1)-4q^3)+g'(1)(2q^3-a(1)+2a'(1))\,, \tag{69}$$

which gives

$$(2q^3-a(1))^3\,\mathfrak{V}[g]=\sigma''(1)(2q^3-a(1))^2+\sigma'(1)^2(a(1)-4q^3)+\sigma'(1)(2q^3-a(1)+2a'(1))(2q^3-a(1))\,. \tag{70}$$

We thus have obtained in (68) and (70) the expressions of the two constants $\mathfrak{M}[g]$ and $\mathfrak{V}[g]$ in terms of the functions $a(u)$ and $b(u)$. These functions come from functions $a(u,w)$ and $b(u,w)$ that are described in (59) and (62). This ends the proof of the first task.

4.8. **Explicit computations for the expectations and the variances.** It remains to compute these values $\mathfrak{M}[g_c]$ and $\mathfrak{V}[g_c]$ for each pair (algorithm, cost) and check that they are positive for $c\neq d_B$. This has already been done for the class (LC). This is now the goal of this section for the class (QC).
The vectorial function γ_c has two components, denoted[12] by $\gamma_c^{(u)}$ and $\gamma_c^{(w)}$. It is completely defined by the two derivatives

$$\theta_u^{(c)}:=\frac{\partial}{\partial u}\gamma_c^{(u)},\qquad \theta_w^{(c)}=\frac{\partial}{\partial u}\gamma_c^{(w)} \tag{71}$$

that are constant functions, equal respectively to

$$\begin{array}{lll}\theta_u^{(d)}=\theta_u^{(c_0)}=\theta_u^{(\delta)}=\theta_u^{(\nu)}=1, & \theta_u^{(d_B)}=0,\\ \theta_w^{(c_0)}=\theta_w^{(d)}=0, & \theta_w^{(\delta)}=\theta_w^{(d_B)}=1, & \theta_w^{(\nu)}=\frac{q-1}{q}\,.\end{array}$$

Then, for any function $(u,w)\mapsto x(u,w)$ and any cost c, the derivatives of the function $u\mapsto x(\gamma_c(u))$ satisfy

$$[x(\gamma_c(u))]'=\theta_u^{(c)}\,\frac{\partial}{\partial u}x(u,w)\big|_{(u,w)=\gamma_c(u)}+\theta_w^{(c)}\,\frac{\partial}{\partial w}x(u,w)\big|_{(u,w)=\gamma_c(u)}\,.$$

This applies in particular to the functions $a(u):=a(\gamma_c(u)), b(u):=b(\gamma_c(u))$.

The expressions (68) and (69) involve the five values

$$a(1),a'(1),a''(1),b'(1),b''(1)\,,$$

that are thus expressed with the five functions F,G,K,L,O and their derivatives together with θ_u and θ_w, as described in Table 5(a). The functions F,G,K,L,O and their derivatives are described in Table 5(b) below.

With these two tables, we compute the values of the mean values M_c and of the variances V_c that are provided in Table 6. And, in each case (except for the cost d_B) we check that the polynomials involved in V_c are strictly positive for $q\geq 2$.

[12] In Section 5.4, the second component of γ_c will be denoted as β_c.

$$
\begin{array}{ll}
a(1) & = F(1) + G(1)K(1) \\
a'(1) & = \theta_w F'(1) + \theta_w G'(1)K(1) + \theta_u G(1)K'(1) \\
b'(1) & = \theta_w L'(1)O(1) + \theta_u L(1)O'(1) \\
a''(1) & = \theta_w^2 F''(1) + \theta_w^2 G''(1)K(1) + 2\theta_u\theta_w G'(1)K'(1) + \theta_u^2 G(1)K''(1) \\
b''(1) & = \theta_w^2 L''(1)O(1) + 2\theta_u\theta_w L'(1)O'(1) + \theta_u^2 L(1)O''(1).
\end{array}
$$

JP and FS case (costs d, δ, ν) $F(w) = qw(1+qw)$ $L(w) = (qw)^3$ $G(w) = qw$	 $F(1) = q(q+1)$ $L(1) = q^3$ $G(1) = q$	 $F'(1) = q(2q+1)$ $L'(1) = 3q^3$ $G'(1) = q$	 $F''(1) = 2q^2$ $L''(1) = 6q^3$ $G''(1) = 0$
JP case (costs d, δ, ν) $K(u) = u^2(q-1)^2 + u(q-1)$ $O(u) = 1 + u(q-1)$	 $K(1) = q(q-1)$ $O(1) = q$	 $K'(1) = (q-1)(2q-1)$ $O'(1) = (q-1)$	 $K''(1) = 2(q-1)^2$, $O''(1) = 0$
FS case (costs d, δ, ν) $K(u) = u^2(q-1)^2$ $O(u) = 1 - (q-1)^2u^2$	 $K(1) = (q-1)^2$ $O(1) = 1-(q-1)^2$ $= q(2-q)$	 $K'(1) = 2(q-1)^2$ $O'(1) = -2(q-1)^2$	 $K''(1) = 2(q-1)^2$, $O''(1) = -2(q-1)^2$
FS case (costs c_0, d_B) $F(w) = qw(1+q)$ $L(w) = q^3w^2$ $G(w) = qw$ $K(u) = u(q-1)^2$ $O(u) = 1 - (q-1)^2u$	 $F(1) = q(q+1)$ $L(1) = q^3$ $G(1) = q$ $K(1) = (q-1)^2$ $O(1) = 1-(q-1)^2$ $= q(2-q)$	 $F'(1) = q(q+1)$ $L'(1) = 2q^3$ $G'(1) = q$ $K'(1) = (q-1)^2$ $O'(1) = -(q-1)^2$	 $F''(1) = 0$ $L''(1) = 2q^3$ $G''(1) = 0$ $K''(1) = 0$, $O''(1) = 0$

TABLE 5. (a) Top: General expressions of the derivatives of the functions $a(u) = a(\gamma_c(u))$ and $b(u) = b(\gamma_c(u))$. (b) Below: Expressions of the derivatives of the functions F, G, K, L, O.

Remark. All the dominant constants (mean values and variances) are rational fractions in q with integer coefficients. All the mean values are in $\Theta(1)$ for $q \to \infty$, and the constants involved depend only on the cost, not on the algorithm. They respectively satisfy

$$M_{c_0} \sim 1 \quad (c = 1), \quad M_d \sim 2 \quad (c = d), \quad M_\delta, M_\nu \sim 3 \quad (c = \delta, \nu).$$

The variance constants V_c exhibit two possible behaviours. For the Jacobi–Perron and the fully subtractive, they are in $\Theta(1/q)$ when $q \to \infty$, whereas they are in $\Theta(1)$ for the Brun algorithm. Moreover, the behaviours of the variances for the Jacobi–Perron and fully subtractive algorithms are very close. They both satisfy

$$qV_{c_0} \sim 1, \quad qV_\delta \sim 2, \quad qV_\nu \sim 3,$$

and they slightly differ for the cost d, with

$$qV_d \sim 5 \quad \text{(JP)}, \qquad qV_d \sim 8 \quad \text{(FS)}.$$

4.9. **Average values for the bit-complexities.** The bivariate generating functions $R_{\Phi_c}(z, u)$ for the bit-complexities Φ_c involve, for $c = \delta, \nu$, the cumulative generating function

$$\widehat{H}_c(z) := \frac{\partial}{\partial u} H_c(z, u)\big|_{u=1}$$

under the form

$$R_{\Phi_c}(z, u) = \frac{1}{1 - H(z)} \cdot u\widehat{H}_c(z) \cdot T(zu) = \frac{1}{1 - H(z)} \cdot \widehat{H}_c(z) \cdot u \cdot \left(\frac{1}{1 - H(uz)}\right) \cdot D(uz)\,.$$

	Jacobi–Perron	Brun	Fully subtractive
M_{c_0}	$\frac{q-1}{q}$	$\frac{q^2-1}{q^2}$	$\frac{q^2-1}{q(q+2)}$
M_d	$\frac{(2q+1)(q-1)}{q(q+1)}$	$\frac{2(q-1)}{q}$	$\frac{2(q^2-1)}{q(q+2)}$
M_δ	$\frac{3q^2+q-1}{q(q+1)}$	$\frac{3q-2}{q}$	$\frac{3q^2+4q-2}{q(q+2)}$
M_ν	$\frac{3(q-1)}{q}$	$\frac{3(q-1)}{q}$	$\frac{3(q-1)}{q}$
M_{d_B}	1	1	1

	Jacobi–Perron	Brun	Fully subtractive
V_{c_0}	$\frac{q-1}{q^2}$	$\frac{(q^2-1)^2}{q^4}$	$\frac{(q^2-1)(2q^2+q+2)}{q^2(q+2)^3}$
V_d	$\frac{(5q^3+5q^2+3q+1)(q-1)}{q^2(q+1)^3}$	$2\frac{(q-1)^2}{q^2}$	$\frac{4(2q^2+q+2)(q^2-1)}{q^2(q+2)^3}$
V_δ	$\frac{2q^5-q^4-q^3+q+1}{q^2(q-1)(q+1)^3}$	$\frac{(q-1)(4q-1)}{q^2}$	$\frac{2(q^5+q^4-2q^2-2q+4)}{q^2(q-1)(q+2)^3}$
V_ν	$3\frac{q-1}{q^2}$	$6\frac{(q-1)^2}{q^2}$	$3\frac{q-1}{q^2}$
V_{d_B}	0	0	0

TABLE 6. Top: Expressions for the constants M_c (expectations) for the five additive costs c. Below: Expressions for the constants V_c (variances) for the five additive costs c. We have explained in Section 4.5 why one cannot expect precise relations between the first column and the sum of the last two columns.

Indeed, consider an input for which the algorithm performs j iterations, and, for $k \in [1..j]$ and $c \in \{\delta, \nu\}$, consider the cost $c(A_k, B_k) \cdot u^{\delta(R_{k,3})}$ defined on the set $\mathcal{H}^j \times \mathcal{D}$. Its generating function is, for a given pair (k, j),

$$H^{k-1}(z) \cdot \widehat{H}_c(z) \cdot u \cdot H^{j-k}(uz) \cdot D(uz)\,.$$

We then take the sum over all the indices (k, j) with $k \le j$.

The associated cumulative series satisfies

$$\widehat{R}_{\Phi_c}(z) := \frac{\partial}{\partial u} R_{\Phi_c}(z,u)\big|_{u=1} = \frac{\widehat{H}_c(z)}{1-H(z)} \cdot \left(R(z) + zD'(z) + zH'(z) \cdot \frac{R(z)}{1-H(z)} \right),$$

and its dominant part is given by

$$\frac{\widehat{H}_c(z)}{1-H(z)} \cdot zH'(z) \cdot \frac{R(z)}{1-H(z)} = z\,\frac{\widehat{H}_c(z)H'(z)}{D^2(z)} \cdot R^3(z)\,. \tag{72}$$

Now, the equality $(1/q^3)H'(1/q^3) = D(1/q^3)$ (see (38)) entails the following:

$$M_{\Phi_\delta} = \frac{1}{2q^3}\frac{H'(1/q^3)}{D(1/q^3)}\,M_\delta = \frac{1}{2}\,M_\delta, \qquad M_{\Phi_\nu} = \frac{1}{2q^3}\frac{H'(1/q^3)}{D(1/q^3)}\,M_\nu = \frac{1}{2}\,M_\nu\,.$$

We have then proven:

Proposition 7. *Consider any of the two bit-complexity costs Φ_δ or Φ_ν. Their expectations on $\mathcal{R}_m$ satisfy*

$$\mathbb{E}_m[\Phi_\delta] \sim M_\delta \cdot \frac{m^2}{2}, \qquad \mathbb{E}_m[\Phi_\nu] \sim M_\nu \cdot \frac{m^2}{2},$$

where M_δ and M_ν are the constants that appear in Table 6.

Remark. Observe that the three algorithms have the same expectation for the cost ν and for the sparse bit-complexity; we found no explanation for this surprising fact.

5. Probabilistic analysis in the continuous model

This section is devoted to the analysis in the continuous model and deals with continued fraction maps. When the set $\mathbb{L}$ of Laurent formal power series with negative degree is endowed with its (normalized) Haar measure μ, the Gauss Artin continued fraction map $\mathrm{T}_G : \mathbb{L} \to \mathbb{L}$ (defined in (5)) is μ-preserving (see e.g. [4]). Here, $\mathbb{L}^2$ is endowed with the Haar measure $\mu \otimes \mu$ and we study the three continued fraction maps $\mathrm{T}_\sharp$ (where $\sharp$ refers to JP, B or FS according to the algorithm) acting on $\mathbb{L}^2$ endowed with $\mu \otimes \mu$. As previously, we provide a unified proof for the ergodic and probabilistic study of the continuous case that applies to each continued fraction map.

We have shown in Section 2 that the continued fraction maps, described respectively in Section 2.5, 2.6 and 2.7, are defined via branches (or inverse branches) that are homographies of the same form. Our unified framework deals here with the transfer operator that is defined in Section 5.2, and then extended in Section 5.3. The transfer operator uses in its definition a change of variables formula that is first stated in Section 5.1. We prove here that the absolute value of the Jacobian of each branch is *constant*. This entails (see Theorem 2) that each $\mathrm{T}_\sharp$ is $\mu \otimes \mu$ preserving and that the sequences of quotients $(\epsilon_k, A_k, B_k)_k$ form sequences of independent and identically distributed random variables. This also entails a close relation between transfer operators and associated (bivariate) generating functions. Precise computations of the expectations and the variances are performed in Section 5.4. We then study the bit-complexity costs in Section 5.5. Finally, Section 5.6 deals with the possible values taken by the degrees of the convergents.

There are various metric results in the continuous model in the literature. However, as already said, the corresponding algorithms are not exactly the ones that are studied here, but they are clearly in the same spirit. In particular, the properties stated in Theorem 2 have already been established in the Jacobi–Perron and in the Brun cases (see [26, 27] and [17], respectively). However, the fact that the Haar measure is invariant for continued fraction maps (in the context of formal power series with coefficients in a finite field) is proven "by hands", and not clearly related to nice properties of the transfer operator. Observe that a similar situation arises in many beta-numeration contexts for formal power series (see [4, 22, 34]), where the invariance of the Haar measure is also proven "by hands" and not clearly related with properties of the Jacobian of the inverse branches. One of the interests of the transfer approach developed here is to provide simple proofs for Haar measure invariance.

5.1. A change of variables formula.

We state here a change of variables formula that plays a central role in the definition of the transfer operator, studied in Section 5.2.

Proposition 8. *Consider the Haar measure $\mu \otimes \mu$ on $\mathbb{L}^2$ and the homographies $h_{(\pm 1, A, B)}$ associated respectively with the matrices $M_{\pm 1}(A, B)$. Then, for any Borel subset $F \in \mathbb{L}^2$ and for any $\epsilon = \pm 1$, the measure $\mu \otimes \mu(h_{\epsilon,A,B}((F))$ satisfies*

$$\mu \otimes \mu(h_{\epsilon,A,B}(F)) = \|\mathrm{Jac}(h_{\epsilon,A,B})\|\, \mu \otimes \mu(F) = \frac{1}{q^{3 \deg B}}\, \mu \otimes \mu(F), \tag{73}$$

where $\mathrm{Jac}(g)$ *stands for the Jacobian of the application g.*

There are two different proofs for the previous proposition.

Proof A. There exists a general change of variables formula that holds in the ultrametric case. It is cited for instance in [8], see also [31]. This general formula entails the first equality in (73). Moreover, when h is associated with a matrix $M_\epsilon(A, B)$, its Jacobian $\mathrm{Jac}(h)$ admits a particularly simple expression. First, the Jacobian of the homography $h_{(\varepsilon,A,B)}$ is related to its denominator

$D[h_{(\varepsilon,A,B)}]$, via the equality $\mathrm{Jac}(h_{(\varepsilon,A,B)}(g)) = D^{-3}[h_{(\varepsilon,A,B)}](g)$ (see also [33, Proposition 5.2]). Second, this denominator $D[h_{(\varepsilon,A,B)}](g)$ is of the form

$$B(h) + g_2 \quad (\epsilon = +1), \qquad B(h) + g_1 \quad (\epsilon = -1)\,.$$

In both cases, as $g_i \in \mathbb{L}$ and $\deg B \geq 1$, the ultrametric equality $||B(h) + g_i|| = ||B(h)|| = q^{\deg B}$ holds and entails the equality

$$||\mathrm{Jac}(h_{(\varepsilon,A,B)})(g)|| = ||D^{-3}[h_{(\varepsilon,A,B)}](g)|| = q^{-3\deg B}.$$

Proof B. It is done "by hands" and it is based on the following assertions, which provide a proof of the change of variables formula for the case of simple transformations. This can be formulated in dynamical terms along the following two assertions. Assertion (i) is a classical statement for the beta-transformation S_β and Assertion (ii) is a classical statement for the Gauss map and its proof can be found e.g. in [4, 7].

(i) For any $\beta \in \mathbb{F}_q((X^{-1}))$, $\deg\beta > 0$, the map $S_\beta\colon \mathbb{L} \to \mathbb{L}$, $f \mapsto \beta f - [\beta f]$ is μ-measure preserving and for any $A \in \mathbb{F}_q[X]$, $\deg A < \deg\beta$ (it also holds for $A = 0$) and for any Borel subset E of $\mathbb{L}$, one has

$$\mu(\{f \in \mathbb{L} \mid [\beta f] = A,\, S_\beta(f) \in E\}) = \frac{1}{q^{\deg\beta}}\mu(E).$$

(ii) For any $B \in \mathbb{F}_q[X]$, $\deg B \geq 1$, and for any Borel subset E of $\mathbb{L}$, one has

$$\mu(\{f \in \mathbb{L} \mid \left[\frac{1}{f}\right] = B,\, \frac{1}{f} - B \in E\} = \frac{1}{q^{2\deg B}}\mu(E).$$

Proof. Let us prove Assertion (i). As S_β is an endomorphism of $\mathbb{L}$ onto itself (i.e., $S_\beta(f+g) = S_\beta(f) + S_\beta(g)$), it is Haar measure preserving. We let denote by $(a_{-1}, a_{-2}, \ldots, a_{-k})(f)$ the k-uple of coefficients of f with index running from -1 to $-k$. Suppose that $\deg\beta = k > 0$. For each $A \in \mathbb{F}_q[X]$, $\deg A < k$, we can find a k-uple $(a_{-1}, a_{-2}, \ldots, a_{-k}) \in \mathbb{F}_q^k$ such that $[\beta f] = A$ if and only if $(a_{-1}, a_{-2}, \ldots, a_{-k})(f) = (a_{-1}, a_{-2}, \ldots, a_{-k})$. It is enough to show the assertion for a cylinder set of the form

$$E = \{f \in \mathbb{L} \mid (a_{-1}, a_{-2}, \ldots, a_{-\ell})(f) = (c_{-1}, c_{-2}, \ldots, c_{-\ell})\}$$

with $(c_{-1}, c_{-2}, \ldots, c_{-\ell}) \in \mathbb{F}_q^\ell$, $\ell \geq 1$. There exists $(a_{-(k+1)}, a_{-(k+2)}, \ldots, a_{-(k+\ell)}) \in \mathbb{F}_q^\ell$ such that

$$\begin{aligned} & \{f \in \mathbb{L} \mid [\beta f] = A,\, (a_{-1}, a_{-2}, \ldots, a_{-\ell})(S_\beta(f)) = (c_{-1}, \ldots, c_{-\ell})\} \\ = \;& \{f \in \mathbb{L} \mid [\beta f] = A,\, (a_{-(k+1)}, a_{-(k+2)}, \ldots, a_{-(k+\ell)})(f) = (c_{-1}, \ldots, c_{-\ell})\}. \end{aligned}$$

This entails the equality $\mu(\{f \in \mathbb{L} \mid [\beta f] = A,\, S_\beta(f) \in E\}) = \frac{1}{q^k}\mu(E) = \frac{1}{q^{\deg\beta}}\mu(E)$. □

We now prove Proposition 8.

Proof. We give the proof in the case $\epsilon = -1$ for the map T defined in Section 2.2. It is enough to show the assertion when F is of the form $E_1 \times E_2$, where E_1 and E_2 are Borel subsets of $\mathbb{L}$. By Fubini's theorem, we have

$$\begin{aligned} & \mu\otimes\mu(\{(f_1,f_2) \mid \eta(f_1,f_2) = -1,\; A_- = A,\, B_- = B,\, \mathrm{T}(f_1,f_2) \in E_1 \times E_2\}) \\ = \;& \iint_{\langle +,A,B\rangle} \mathbf{1}_{E_1\times E_2}\left(\frac{1}{f_2} - B, \frac{f_1}{f_2} - A\right) d\mu\otimes\mu \\ = \;& \int_{[1/f_2]=B}\left(\int_{[f_1/f_2]=A} \mathbf{1}_{E_1\times E_2}\left(\frac{1}{f_2} - B, \frac{f_1}{f_2} - A\right) d\mu(f_1)\right) d\mu(f_2) \\ = \;& \int_{[1/f_2]=B} \mathbf{1}_{E_1}\left(\frac{1}{f_2} - B\right)\left(\int_{[f_1/f_2]=A} \mathbf{1}_{E_2}\left(\frac{f_1}{f_2} - A\right) d\mu(f_1)\right) d\mu(f_2). \end{aligned}$$

For any f_2 with $[1/f_2] = B$, we now use Assertion (i) for the internal integral with $\beta = 1/f_2$ and thus $\deg \beta = \deg B$, and then Assertion (ii). This gives

$$\begin{aligned} &= \frac{1}{q^{\deg B}} \mu(E_2) \int_{[1/f_2]=B} \mathbf{1}_{E_1}\left(\frac{1}{f_2} - B\right) d\mu(f_2) \\ &= \frac{1}{q^{\deg B}} \mu(E_2) \frac{1}{q^{2\deg B}} \mu(E_1) = \frac{1}{q^{3\deg B}} \mu\otimes\mu(E_1 \times E_2). \end{aligned}$$

□

5.2. **The transfer operator.** Let $\mathrm{T} = \mathrm{T}_\sharp$ be one of the three continued fraction maps of interest, with $\sharp \in \{\mathrm{JP}, \mathrm{B}, \mathrm{FS}\}$. We recall that the set of quotients $\mathcal{H}_\sharp$ depends on the continued fraction map. With a given quotient $(\varepsilon, A, B) \in \mathcal{H}_\sharp$, one associates the *inverse branch* $h_{\sharp,(\varepsilon,A,B)}$ of $\mathrm{T}_\sharp$. This is the map acting on $\mathbb{L}^2$ satisfying $h_{\sharp,(\varepsilon,A,B)}(g) = f$ if and only if $\mathrm{T}_\sharp(f) = g$ with the produced quotient being (ε, A, B). We know that the following holds for the three continued fraction maps:

- the inverse branch $h_{\sharp,(\varepsilon,A,B)}$ is the homography associated with the matrix $M_\epsilon(A, B)$;
- the map $h_{(\varepsilon,A,B)} \colon \mathbb{L}^2 \to h_{(\varepsilon,A,B)}(\mathbb{L}^2)$ is a bijection.

We now omit the reference to the algorithm. We consider the density transformer $\mathbf{H}$ associated with the map T. This operator, also called the Perron–Frobenius operator, was introduced early in the study of dynamical systems as the dual of the Koopman operator $f \mapsto f \circ T$ for non-invertible maps. Then Ruelle [28] introduced a more general notion of transfer operators; this was further adapted to various contexts, notably to the continued fraction case by Mayer [23]. For more on transfer operators, see for instance [1].
Here, the transfer operator acts on $L^1_{\mu\otimes\mu}(\mathbb{L}^2)$ and it associates with a density ϕ the new density that holds on $\mathbb{L}^2$ after one iteration of the map T. As the inverse branches are full, the transfer operator is defined for $\phi \in L^1(\mu\otimes\mu)$ and $g \in \mathbb{L}^2$ as

$$\mathbf{H}[\phi](g) = \sum_{(\varepsilon,A,B)\in\mathcal{H}} \|\mathrm{Jac}(h_{(\varepsilon,A,B)})(g)\| \; \phi \circ h_{(\varepsilon,A,B)}(g)\,. \tag{74}$$

With Proposition 8, the norm of the Jacobian is a constant function, and the operator is

$$\mathbf{H}[\phi](g) = \sum_{(\varepsilon,A,B)\in\mathcal{H}} \frac{1}{q^{3\deg B}} \; \phi \circ h_{(\varepsilon,A,B)}(g)\,.$$

Then, the constant function $\phi \equiv \mathbf{1}$ on $\mathbb{L}^2$ is an eigenfunction for the operator $\mathbf{H}$. The associated eigenvalue involves the generating function $H(z)$ introduced in Section 4.2, and this yields

$$\mathbf{H}[\mathbf{1}] = \left(\sum_{(\varepsilon,A,B)\in\mathcal{H}} q^{-3\deg B}\right)\mathbf{1} = H(1/q^3)\,\mathbf{1} = \mathbf{1}\,, \tag{75}$$

where the last equality $H(1/q^3) = 1$ comes from the fact that the rational fraction $1/(1 - H_\sharp(z))$ has a pole at $z = 1/q^3$. This entails the invariance of $\mu\otimes\mu$ under the action of T.

Moreover, for any triple $(\epsilon, A, B) \in \mathcal{H}_\sharp$ and any Borel set $F \in \mathbb{L}^2$, Proposition 8 entails the equality

$$\mu\otimes\mu\left(\{f \mid (\epsilon, A, B)(f) = (\varepsilon, A, B), \quad f \in \mathrm{T}^{-1}(F)\}\right) = \mu\otimes\mu(h_{(\epsilon,A,B)}(F)) = \frac{1}{q^{3\deg B}}\mu\otimes\mu(F)\,.$$

We then conclude, by induction on the number of partial quotients considered, that the quotients $(\epsilon_n, A_n, B_n)_n$ are independent and define identically distributed random variables on $\mathbb{L}^2$. We have then proven the following.

Theorem 2. *Each continued fraction maps $\mathrm{T}_\sharp$, with $\sharp \in \{JP, B, FS\}$, is $\mu\otimes\mu$-preserving. The sequence $(\varepsilon_n, A_n, B_n)_{n\geq 1}$ is formed with independent and identically distributed random variables with respect to the probability measure $\mu\otimes\mu$ and the map $\mathrm{T}_\sharp$ is ergodic with respect to $\mu\otimes\mu$.*

5.3. **Additive costs in the continuous model.** The following result deals with the cost C_n along the n-th truncated trajectory associated with a step-cost c. This step-cost c gives rise to a cost $\widehat{c}$ defined on $\mathbb{L}^2$, via the equality $\widehat{c}(f) = c(Q(f))$, where $Q(f)$ is the quotient pair (A, B) computed by T on f. As already seen in Section 3.2, the following equality holds:

$$C_n(f) = \widehat{c}(f) + \widehat{c}(\mathrm{T}(f)) + \cdots + \widehat{c}(\mathrm{T}^k(f)) + \cdots + \widehat{c}(\mathrm{T}^{n-1}(f))\,. \tag{76}$$

This cost C_n has two characteristics:

- The equality (76) is written as an ergodic sum: the ergodic theorem (and here in fact, the strong law of large numbers) entails a convergence that holds almost everywhere.
- It is also written as a sum of i.i.d. variables, and the central limit theorem applies.

Theorem 3. *The following holds for any continued fraction maps* $\mathrm{T}_\sharp$ *with* $\sharp \in \{JP, B, FS\}$ *and any step-cost* c *defined in* (28)*:*

(*i*) *Any cost* C_n *defined in* (76) *satisfies*

$$\frac{1}{n}\, C_n(f) \to \widetilde{M}_c\,, \qquad (a.e\ in\ \mathbb{L}^2)\,, \qquad with \quad \widetilde{M}_c = \mathbb{E}_{\mu\otimes\mu}[\widehat{c}]\,.$$

(*ii*) *The cost* C_n *is written as a sum of independent and identically distributed random variables, whose expectation and variance, respectively denoted as* $\widetilde{M}_c$ *and* $\widetilde{V}_c$*, are expressed via the generating function* $u \mapsto \ell_c(u) = H_c(1/q^3, u)$*. The function* $H_c(z, u)$ *was introduced in Section 4.5 as the bivariate generating function of the cost* c*. The following equalities hold:*

$$\widetilde{M}_c = \mathbb{E}_{\mu\otimes\mu}[\widehat{c}] = \mathfrak{M}[\ell_c] = \ell_c'(1), \qquad \widetilde{V}_c = \mathbb{V}_{\mu\otimes\mu}[\widehat{c}] = \mathfrak{V}[\ell_c] = \ell_c''(1) + \ell_c'(1) - \ell_c'(1)^2\,, \tag{77}$$

and they involve the cost $\widehat{c}$ *defined from the step-cost* c *as* $\widehat{c}(f) = c(Q(f))$*.*

(*iii*) *When the cost* c *is not constant, the variance* $\mathbb{V}_{\mu\otimes\mu}[\widehat{c}]$ *is positive, the central limit theorem applies, and the cost* C_n *asymptotically follows a Gaussian law, with*

$$\mathbb{E}[C_n] = n\,\widetilde{M}_c, \qquad \mathbb{V}[C_n] = n\,\widetilde{V}_c\,.$$

Remark. Assertion (*iii*) does not apply to the pair (FS, d) where the cost d is constant and equal to 2.

Proof. (*i*) First, each (generic) cost $C_n(f)$ along the n-th truncated trajectory associated with a step-cost c defined on $\mathbb{L}^2$ is written as the sum

$$C_n(f) = \widehat{c}(f) + \widehat{c}(\mathrm{T}(f)) + \cdots + \widehat{c}(\mathrm{T}^k(f)) + \cdots + \widehat{c}(\mathrm{T}^{n-1}(f))\,,$$

that involves the associated cost $\widehat{c}$ defined on $\mathbb{L}^2$. The quantity $(1/n)\, C_n(f)$ is thus an ergodic sum, and the ergodic theorem entails

$$\lim_{n\to\infty} \frac{1}{n} C_n(f) = \mathbb{E}_{\mu\otimes\mu}[\widehat{c}] = \int_{\mathbb{L}^2} \widehat{c}(f)\, d\mu\otimes\mu \qquad \text{(a.e)}\,.$$

(*ii*) Due to Theorem 2, the cost C_n is the sum of independent and identically distributed random variables. In order to study the distribution of each elementary random variable, we consider a perturbation of the transfer operator defined in (74). Such operators have been introduced by Ruelle [28] in the study of the thermodynamic formalism, and are now strongly used in the dynamic analysis method.
We associate with the step-cost c the operator $\mathbf{H}_{c,u}(= \mathbf{H}_{c,u,\sharp})$, that depends on a complex variable u, and is defined as

$$\begin{aligned} \mathbf{H}_{c,u}[\phi](g) &= \sum_{(\varepsilon,A,B)\in\mathcal{H}} \|\mathrm{Jac}(h_{(\varepsilon,A,B)})(g)\|\, u^{c(A,B)}\, \phi\circ h_{(\varepsilon,A,B)}\,(g) \\ &= \sum_{(\varepsilon,A,B)\in\mathcal{H}} (q^{-3\deg B})\, u^{c(A,B)}\, \phi\circ h_{(\varepsilon,A,B)}\,(g)\,. \end{aligned} \tag{78}$$

As previously, the constant function $\phi \equiv \mathbf{1}$ on $\mathbb{L}^2$ is an eigenfunction for the operator $\mathbf{H}_{c,u}$, and the associated eigenvalue involves the bivariate generating function $H_c(z,u)$ introduced in Section 4.2 at point $z = 1/q^3$ as

$$\mathbf{H}_{c,u}[\mathbf{1}] = \sum_{(\varepsilon,A,B)\in\mathcal{H}} (q^{-3\deg B})\, u^{c(A,B)}\, \mathbf{1} = \mathbb{E}_{\mu\otimes\mu}[u^{\widehat{c}}]\, \mathbf{1} = H_c(1/q^3, u)\, \mathbf{1}\,. \tag{79}$$

This entails the equality $\mathbb{E}_{\mu\otimes\mu}[u^c] = H_c(1/q^3, u)$, and thus, taking the derivatives with respect to u at $u = 1$ of the function $\ell_c(u) := H_c(1/q^3, u)$ leads to the two equalities that involve the quantities defined in (64) and described in (77).

With a fixed step-cost c and an algorithm $\sharp$, we then associate the operator $\mathbf{H}_{\sharp,c,u}$. The Sum Property of Proposition 2 now extends to the operators, and the following equality between the three operators holds:

$$2\,\mathbf{H}_{JP,c,u} = \mathbf{H}_{B,c,u} + \mathbf{H}_{FS,c,u}\,.$$

As these three operators admit the same eigenfunction $\phi \equiv \mathbf{1}$ on $\mathbb{L}^2$, this also entails the Sum Property for their associated eigenvalues $u \mapsto \ell_c(u)$ and thus for the (first or second) derivatives of the functions ℓ_c. Then, the Sum Property extends to the associated $\mathfrak{M}[\ell_c]$ but *not*[13] to the associated $\mathfrak{V}[\ell_c]$ that satisfy

$$2\,\mathfrak{M}[\ell_{JP,c}] = \mathfrak{M}[\ell_{B,c}] + \mathfrak{M}[\ell_{FS,c}], \qquad \mathfrak{V}[\ell_{JP,c}] = \mathfrak{V}[\ell_{B,c}] + \mathfrak{V}[\ell_{FS,c}] + 2\mathfrak{M}[\ell_{B,c}]\cdot\mathfrak{M}[\ell_{FS,c}]\,.$$

Then, the equalities hold

$$2\,\widetilde{M}_{JP,c} = \widetilde{M}_{B,c} + \widetilde{M}_{FS,c} \qquad 2\widetilde{V}_{JP,c} = \widetilde{V}_{B,c} + \widetilde{V}_{FS,c} - 2\left(\widetilde{M}^2_{JP,c} - \widetilde{M}_{B,c}\cdot\widetilde{M}_{FS,c}\right)\,. \tag{80}$$

(*iii*) With $p(h) := q^{-3\deg B}$, the Cauchy-Schwartz inequality leads to

$$\sum p(h) u^{c(h)} \le \Big(\sum p(h) u^{2c(h)}\Big)^{1/2} \Big(\sum p(h)\Big)^{1/2}\,,$$

and the equality holds if and only if the terms are collinear. This only happens when the cost $c(h)$ does not depend on the branch h.

□

5.4. **Computation of expectations and variances.** We now perform precise computations for the constants $\widetilde{M}_c = \mathfrak{M}[\ell_c]$ and $\widetilde{V}_c = \mathfrak{V}[\ell_c]$ that are involved in (77). In particular, for any additive cost c, we compare the constant $\widetilde{M}_c$ with the constant M_c which occurs in the analysis of the gcd algorithm that deals with finite trajectories. We prove that a finite trajectory behaves on average in the same way as a truncated trajectory behaves almost everywhere. We also provide a precise expression of the constants $\widetilde{V}_c$ which intervene in the variances. The computations are summarized in Table 8.

Proposition 9. *The following holds for any continued fraction map* $\mathrm{T} = \mathrm{T}_\sharp$*, where* $\sharp$ *refers to JP, B or FS according to the algorithm, and for any additive cost* $c \in \{d, d_B, \delta, \nu\}$*:*

(*i*) *The entropy of the dynamical system* $(\mathbb{L}^2, \mathrm{T}, \mu\otimes\mu)$ *is expressed in terms of the degree* d_B *of the quotient* B *and satisfies*

$$\mathcal{E}(\mathrm{T}) = 3\,\mathbb{E}_{\mu\otimes\mu}[d_B]\,.$$

(*ii*) *The relation*

$$3\,\widetilde{M}_c = M_c\cdot\mathcal{E}(\mathrm{T})$$

holds between the expectation $\widetilde{M}_c = \mathbb{E}_{\mu\otimes\mu}[c]$ *in the continuous model, the dominant constant* M_c *described in* (65) *which appears in the analysis of the discrete model and the entropy.*

[13] This is due to the term $\ell'(1)^2$ inside $[\ell]$.

(*iii*) *The relation* (83) *holds between the function* ℓ_c *defined in Theorem 3*(*ii*)*, the functions* ρ_c^{-1} *and* σ_c *that arise in the analysis of the discrete model, and the second component* β_c *of the vectorial function* γ_c *defined in* (44)*, together with a function* Δ_c *described in Table 7.*

(*iv*) *The following relations hold*

$$\widetilde{M}_c = \mathfrak{M}[\ell_c] = \sigma_c'(1)\,\Delta_c(1)\ ,$$
$$\widetilde{V}_c = \mathfrak{V}[\ell_c] = \sigma_c''(1)\,\Delta_c(1) - \sigma_c'(1)^2\Delta_c(1)^2 + \sigma_c'(1)\left[\Delta_c(1) + 2\theta_w^{(c)}\,\Delta_c'(1)\right]\ .$$

(*v*) *The constants associated with each pair (algorithm, cost) are shown in Table 8. Except in the case* (FS, d)*, the variance constants* $\widetilde{V}_c$ *involve polynomials that are (strictly) positive for* $q \geq 2$*.*

Proof. (*i*) Theorem 2 shows that the process which emits at each discrete time k the triple (ϵ_k, A_k, B_k) is memoryless. Denote by $p(h)$ the probability of emitting the quotient (ϵ, A, B). Then, the entropy is just equal to

$$\mathcal{E}(\mathrm{T}_\sharp) := \sum_{h\in\mathcal{H}_\sharp} p(h)\log_q p(h) = 3\sum_{(\epsilon,A,B)\in\mathcal{H}_\sharp} \frac{1}{q^{3\deg B}}\deg B = \mathbb{E}_{\mu\otimes\mu}[3\deg B] = 3\widehat{H}_{\deg B}(1/q^3)$$

where $\widehat{H}_c(z)$ is the cumulative gf of the cost c, equal (by definition) to $(\partial/\partial u)H_c(z,u)|_{u=1}$. In the case of cost d_B, Section 2 proves the equality $H_{d_B}(z,u) = H(zu)$ that entails

$$\frac{1}{3}\mathcal{E}(T) = \widehat{H}_{\deg B}(1/q^3) = (1/q^3)H'(1/q^3)\,.$$

Furthermore, the equality $(1/q^3)H'(1/q^3) = D(1/q^3)$ (see (38)) implies that the entropy is also equal in each case to $3D(1/q^3)$.

(*ii*) The constant M_c which occurs in the analysis of the discrete model is equal to $\mathfrak{M}[g_c]$ where g_c appears in (66). The function g_c is related itself to another function ρ_c defined by the implicit equality $H_c(\rho_c(u), u) = 1$ via the relation $\rho_c(u) = (q^3 g_c(u))^{-1}$. Taking the derivative with respect to u leads to the equality

$$\rho_c'(u)\frac{\partial}{\partial z}H_c(\rho_c(u),u) + \frac{\partial}{\partial u}H_c(\rho_c(u),u) = 0\,.$$

Now, at $u = 1$, the equality $g_c(1) = 1$ and the relation $\rho_c'(1) = -(1/q^3)g_c'(1)$ holds. Furthermore, the equality $H_c(z,1) = H(z)$ holds and the partial derivative $(\partial/\partial z)\,H_c(z,1)\big|_{z=1/q^3}$ is just equal to $H'(1/q^3)$. Using now (*i*), and the relation with the entropy, we conclude with the equality

$$-g_c'(1)\frac{1}{q^3}H'(1/q^3) + \ell_c'(1) = 0, \qquad \mathfrak{M}[g_c]\,\mathcal{E}(T) = 3\,\mathfrak{M}[\ell_c]\,.$$

(*iii*) We could compute the second derivative of the relation $H_c(\rho_c(u), u) = 1$ at $u = 1$. However, we prefer to adopt a more direct method, which directly deals with the polynomial $\widetilde{Q}$ introduced in Section 4.5, and more precisely with its reciprocal polynomial $\widehat{Q}$ also introduced there.

Quadratic class. For the subclass QC1, one has

$$\text{(QC1)}\quad 1 - H\left(\tfrac{1}{z}, u, w\right) = \frac{1}{z^2}\frac{\widehat{Q}(u,w)(z)}{(1-\frac{qw}{z})(1-\frac{q^2w^2)}{z})} = \frac{\widehat{Q}(u,w)(z)}{(z-qw)(z-q^2w^2)}\,,$$

where $\widehat{Q}(u,w)$ is the reciprocal of $\widetilde{Q}(u,w)$ given in (58). In the same vein, for the subclass QC2, one has

$$\text{(QC2)}\quad 1 - H\left(\tfrac{1}{z}, u, w\right) = \frac{1}{z^2}\frac{\widehat{Q}(u,w)(z)}{(1-\frac{qw}{z})(1-\frac{q^2w)}{z})} = \frac{\widehat{Q}(u,w)(z)}{(z-qw)(z-q^2w)}\,,$$

where $\widehat{Q}(u,w)$ is the reciprocal of $\widetilde{Q}(u,w)$ given in (61).

Linear class. There are analog formulae that involve the function $\rho(u,w)$, namely

$$\text{(LC1), (LC3)}\quad 1-H\left(\tfrac{1}{z},u,w\right)=\frac{1-\frac{1}{z}\rho^{-1}(u,w)}{1-\frac{1}{z}qw},\qquad \text{(LC2)}\quad 1-H\left(\tfrac{1}{z},u,w\right)=\frac{1-\frac{1}{z}\rho^{-1}(u,w)}{1-\frac{1}{z}q^2w}\,,$$

or, equivalently

$$\text{(LC1), (LC3)}\quad 1-H\left(\tfrac{1}{z},u,w\right)=\frac{z-\rho^{-1}(u,w)}{z-qw},\qquad \text{(LC2)}\quad 1-H\left(\tfrac{1}{z},u,w\right)=\frac{z-\rho^{-1}(u,w)}{z-q^2w}\,.$$

We now focus on the point (z,u,w) where $z=q^3$ and $(u,w)=\gamma_c(u)$ where γ_c is defined in (44). The second component of the vectorial function γ_c was previously denoted as $\gamma_c^{(w)}$ at the beginning of Section 4.8, and its derivative was denoted there by $\theta_w^{(c)}$ (see Eqn. (71)). This function $\gamma_c^{(w)}$ is presently denoted, in a lighter way, by β_c, so that the equality $\beta_c'(u)=\theta_w^{(c)}$ holds.
For the (whole) quadratic class, the polynomial $\widehat{Q}(q^3,\gamma_c(u))$ equals $q^3(q^3-\sigma_c(u))$ where the function σ_c is defined in (67). This gives for this class

$$\text{(81)}\qquad \text{(QC1)}\quad 1-\ell_c=\frac{q^3-\sigma_c}{(q^2-\beta_c)(q-\beta_c^2)}\,,\qquad \text{(QC2)}\quad 1-\ell_c=\frac{q^3-\sigma_c}{(q^2-\beta_c)(q-\beta_c)}\,.$$

For the linear subclass, the quantity $1-\ell_c$ is equal respectively to

$$\text{(82)}\qquad \text{(LC1), (LC3)}\quad 1-\ell_c=\frac{q^3-\rho_c^{-1}}{q(q^2-\beta_c)},\qquad \text{(LC2)}\quad 1-\ell_c=\frac{q^3-\rho_c^{-1}}{q^2(q-\beta_c)}\,.$$

A unified formula. In all the cases, the following equality holds:

$$\text{(83)}\qquad 1-\ell_c=(q^3-\sigma_c)\;\Delta_c\circ\beta_c\;.$$

Here, the function σ_c is defined in (67) for the quadratic subclass and equal to ρ_c^{-1} for the linear subclass. The function β_c is the second component of the vectorial function γ_c. The function $w\mapsto\Delta_c(w)$ only depends on the subclass and is defined in Table 7.

Subclass	σ_c	Expression of $w\mapsto\Delta_c(w)$	$\Delta_c'(1)$
QC1	Eqn (67)	$[(q^2-w)(q-w^2)]^{-1}$	$(2q^2+q-3)\,(q^2-1)^{-2}\,(q-1)^{-2}$
QC2	Eqn (67)	$[(q^2-w)(q-w)]^{-1}$	$(q^2+q-2)\,(q^2-1)^{-2}\,(q-1)^{-2}$
LC1	ρ_c^{-1}	$[q(q^2-w)]^{-1}$	$q^{-1}(q^2-1)^{-2}$
LC2	ρ_c^{-1}	$[q^2(q-w)]^{-1}$	$q^{-2}(q-1)^{-2}$
LC3	ρ_c^{-1}	$[q(q^2-w)]^{-1}$	$q^{-1}(q^2-1)^{-2}$

TABLE 7. Expressions of the functions σ_c, Δ_c and $\Delta_c'(1)$ for each subclass.

(iv) Then, in all the cases, due to (83), the expectation $\mathfrak{M}[\ell_c]$ and the variance $\mathfrak{V}[\ell_c]$ are expressed with possibly the first two derivatives of the function σ_c, together with the derivatives of the function $w\mapsto\Delta_c(w)$ and the derivative $\theta_w^{(c)}$ of the second component β_c of γ_c. All these values are taken at $u=1$. We now omit the reference to the cost c in the next computations.
Taking the first derivative of (83) gives:

$$-\ell'(u)=\theta_w(q^3-\sigma(u))\,\Delta'(\beta(u))-\sigma'(u)\,\Delta(\beta(u))\,.$$

This leads to the equality $\ell'(1)=\sigma'(1)\Delta(1)$, and

$$\text{(84)}\qquad \text{(QC)}\quad \mathfrak{M}[\ell]=\frac{\sigma'(1)}{(q-1)(q^2-1)},\quad \text{(LC1), (LC3)}\quad \mathfrak{M}[\ell]=\frac{\sigma'(1)}{q(q^2-1)},\quad \text{(LC2)}\quad \mathfrak{M}[\ell]=\frac{\sigma'(1)}{q^2(q-1)}\,.$$

We now compute the constants $\mathfrak{V}[\ell]$ involved in (77) that are relative to the variances.

The second derivative at $u = 1$ of (83) just involves three terms taken at $u = 1$, namely

$$-\ell''(u) = -\theta_w\, \sigma'(u)\, \Delta'(\beta(u)) - \sigma''(u)\, \Delta(\beta(u)) - \theta_w\, \sigma'(u)\, \Delta'(\beta(u))\,.$$

This leads to the equality

$$\ell''(1) = \sigma''(1)\, \Delta(1) + 2\theta_w\, \sigma'(1)\Delta'(1)\,.$$

Finally, the term $\mathfrak{V}[\ell]$ for the variance satisfies:

$$\mathfrak{V}[\ell] = \ell''(1) - \ell'(1)^2 + \ell'(1) = \sigma''(1)\, \Delta(1) - \sigma'(1)^2\Delta(1)^2 + \sigma'(1)\, [\Delta(1) + 2\theta_w\, \Delta'(1)]\ . \tag{85}$$

	Jacobi–Perron	Brun	Fully subtractive
Entropy $\mathcal{E}$	$\dfrac{3q}{q-1}$	$\dfrac{3q^2}{q^2-1}$	$\dfrac{3q(q+2)}{q^2-1}$
$\widetilde{M}_{d_B}$	$\dfrac{q}{q-1}$	$\dfrac{q^2}{q^2-1}$	$\dfrac{q(q+2)}{q^2-1}$
$\widetilde{M}_{d}$	$\dfrac{2q+1}{q+1}$	$\dfrac{2q}{q+1}$	2
$\widetilde{M}_{\delta}$	$\dfrac{3q^2+q-1}{q^2-1}$	$\dfrac{q(3q-2)}{q^2-1}$	$\dfrac{3q^2+4q-2}{q^2-1}$
$\widetilde{M}_{\nu}$	3	$3\dfrac{q}{q+1}$	$3\dfrac{q+2}{q+1}$

	Jacobi–Perron	Brun	Fully subtractive
$\widetilde{V}_{d_B}$	$\dfrac{q}{(q-1)^2}$	$\dfrac{q^2}{(q^2-1)^2}$	$\dfrac{q(2q^2+q+2)}{(q^2-1)^2}$
$\widetilde{V}_{d}$	$\dfrac{q}{(q+1)^2}$	$\dfrac{2(q-1)}{(q+1)^2}$	0
$\widetilde{V}_{\delta}$	$\dfrac{q(5q^2+7q+1)}{(q^2-1)^2}$	$\dfrac{(2q-1)(q^2-2q+2)}{(q^2-1)^2}$	$\dfrac{q(8q^2+q+8)}{(q^2-1)^2}$
$\widetilde{V}_{\nu}$	$\dfrac{6}{q}$	$\dfrac{3(3q-2)}{(q+1)^2}$	$\dfrac{3(3q^2+2q+4)}{q(q+1)^2}$

TABLE 8. Constants $\mathfrak{M}[\ell_c]$ for the mean (top) and $\mathfrak{V}[\ell_c]$ for the variance (below) for the three algorithms and the costs of interest. We check in each table the properties described in (80), namely
– for the means, the relation $2\widetilde{M}_{c,JP} = \widetilde{M}_{c,B} + \widetilde{M}_{c,FS}$,
– for the variances, the relation $2\widetilde{V}_{JP,c} = \widetilde{V}_{B,c} + \widetilde{V}_{FS,c} - 2\left(\widetilde{M}^2_{JP,c} - \widetilde{M}_{B,c}\cdot\widetilde{M}_{FS,c}\right)$.

Remark. There are strong similarities between the present formulae (84) and (85) (here in the continuous model) and formulae (68) and (70) that stand in the discrete model (at least in the (QC) subclass).

(v) The constants relative to each pair (algorithm, cost) are summarized in Table 8 below. We have checked that the polynomials involved are always positive for $q \geq 2$ (except for the pair (FS, d) as already mentioned).

Remark. As in the discrete case, all the dominant constants (mean values and variances) are elements of $\mathbb{Z}(q)$. All the mean values are in $\Theta(1)$ for $q \to \infty$, and the involved constants only depend on the cost, not on the algorithm. This is the same as in the discrete case, and they respectively satisfy

$$\widetilde{M}_{d_B} \sim 1, \quad \widetilde{M}_{d} \sim 2, \quad \widetilde{M}_{\delta}, \widetilde{M}_{\nu} \sim 3.$$

The variance constants are in $\Theta(1/q)$ with two exceptions: first, the constant $\widetilde{V}_d$ is zero for the FS algorithm, as expected; second, for the pair (Brun, d_B), the variance is in $\Theta(1/q^2)$, a sort of reminiscence of the discrete case, where it is zero.

□

5.5. **The case of bit-complexities.** For each cost $c \in \{\delta, \nu\}$, we consider two notions of bit-complexities. First, the bit-complexity, as defined in Section 3.2, satisfies for $c \in \{\delta, \nu\}$

$$\begin{aligned}\Phi_{c,n}(f) &= \textstyle\sum_{i=1}^{n} (c(A_i) + c(B_i))\, \delta(Q_{i-1}) \\ &= (c(A_1) + c(B_1)) + \textstyle\sum_{i=2}^{n-1} (c(A_i) + c(B_i)) \left(1 + \sum_{j=1}^{i-1} \deg B_j\right).\end{aligned}$$

It involves the trajectory $(f, T(f), \cdots, T^n(f))$ with quotients (ϵ_k, A_k, B_k) and the convergents

$$\begin{pmatrix} P_{1,k} \\ P_{2,k} \\ Q_k \end{pmatrix} = M_{\epsilon_k}(A_k, B_k) \cdots M_{\epsilon_1}(A_1, B_1) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

For a polynomial Q_n, let $\widehat{Q}_n$ stands for Q_n divided by its leading coefficient. We may also consider the execution of the gcd algorithm on the triple $(P_{1,n}, P_{2,n}, \widehat{Q}_n)$ and get a sequence of quotients $(\epsilon'_k, A'_k, B'_k)$ that correspond to the execution of the algorithm on $(P_{1,n}, P_{2,n}, \widehat{Q}_n)$. The quotients $(\epsilon'_k, A'_k, B'_k)$ are just the quotients $(\epsilon_{n-k}, A_{n-k}, B_{n-k})$ produced in the reverse order and the sequence of polynomials produced by the algorithm is $(P_{1,n,k}, P_{2,n,k}, \widehat{Q}_{n,k})_{1 \le k \le n}$, where the triples $(P_{1,n,k}, P_{2,n,k}, Q_{n,k})$ are intermediate convergents defined for $1 \le k \le n-1$ as

$$\begin{pmatrix} P_{1,n,k} \\ P_{2,n,k} \\ Q_{n,k} \end{pmatrix} = M_{\epsilon'_{k+1}}(A'_{k+1}, B'_{k+1}) \cdots M_{\epsilon_n}(A'_n, B'_n) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} P_{1,n,n} \\ P_{2,n,n} \\ Q_{n,n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

It turns out that

$$\begin{aligned}\Phi_c(P_{1,n}, P_{2,n}, \widehat{Q}_n) &= \textstyle\sum_{i=1}^{n} (c(A'_i) + c(B'_i))\, \delta(\widehat{Q}_{n,i}) \\ &= \textstyle\sum_{i=1}^{n-1} (c(A'_i) + c(B'_i)) \left(1 + \sum_{j=i+1}^{n} \deg B'_j\right) + (c(A'_n) + c(B'_n)) \\ &= \textstyle\sum_{i=2}^{n} (c(A_i) + c(B_i)) \left(1 + \sum_{j=1}^{i-1} \deg B_j\right) + (c(A_1) + c(B_1)).\end{aligned}$$

In both cases, the sequence $(A_n, B_n)_{n \ge 1}$ is formed with independent and identically distributed random variables with respect to $\mu \otimes \mu$, and we have for $i < j$

$$\mathbb{E}_{\mu\otimes\mu}[c(A_i) \deg B_j] = \mathbb{E}_{\mu\otimes\mu}[c(A)]\, \mathbb{E}_{\mu\otimes\mu}[d_B] \qquad \mathbb{E}_{\mu\otimes\mu}[c(B_i) \deg B_j] = \mathbb{E}_{\mu\otimes\mu}[c(B)]\, \mathbb{E}[d_B].$$

We now use the following result from [7] that we recall below:

Proposition [7, Proposition 3] *Let (V_n) and (W_n) be stationary and ergodic sequences of non-negative valued random variables on a probability space $(\Omega, \mathcal{F}, P)$ with finite expectations μ_V and μ_W, respectively. For P-a.e. $\omega \in \Omega$, we have*

$$\lim_{n\to\infty} \frac{1}{n^2} \sum_{k=1}^{n} V_k \sum_{j=k+1}^{n} W_j = \frac{1}{2} \mu_V \mu_W.$$

Once applied to our framework, we obtain the following result

Proposition 10. *For $c \in \{\delta, \nu\}$, the following holds for the two notions of bit-complexity previously defined:*

$$\lim_{n\to\infty} \frac{1}{2n^2}\, \Phi_{c,n}(f) = \lim_{n\to\infty} \frac{1}{2n^2}\, \Phi_c(P_{1,n}, P_{2,n}, Q_n) = \mathbb{E}_{\mu\otimes\mu}[c]\, \mathbb{E}_{\mu\otimes\mu}[d_B]\,.$$

5.6. **On the degree of convergents.** We wish to study the possible values of $\deg Q_k$ when k varies. We are then interested in the indices N for which there exists some index k for which

$$\deg Q_k(f) = \sum_{i=1}^{k} \deg B_i(f) = N\,.$$

If the degrees $\deg B_i$ were all equal to a constant d_B, the integers N of interest would form an arithmetical sequence with ratio d_B, and their density would be equal to $1/d_B$. Even though the degree d_B is not a constant random variable, the following result exhibits such a (limit) density.

Proposition 11. *We consider the sequence of sets*

$$F_N := \{f \in \mathbb{L}^2 \mid \exists k \quad \deg Q_k(f) = N\}\,.$$

For the Jacobi–Perron and Brun algorithms, the measure of F_N is independent of N, and is equal to $1/\mathbb{E}_{\mu\otimes\mu}(d_B)$. For the fully subtractive algorithm, this measure depends on N and satisfies

$$\lim_{N\to\infty} \mu\otimes\mu(F_N) = \frac{1}{\mathbb{E}_{\mu\otimes\mu}(d_B)}.$$

Proof. We recall first that $\deg Q_k = \sum_{i=1}^{k} \deg B_i$ for the three algorithms.

In the case of the Jacobi–Perron and Brun continued fraction maps, the sequence $(\deg B_i)_{i\geq 1}$ is an i.i.d. sequence with geometric distribution of parameter $1/q$ (for Jacobi–Perron) and $1/q^2$ (for Brun). One has indeed, for any $i \geq 1$ and any $\ell \geq 1$

$$\mu\otimes\mu\left\{f \in \mathbb{L}^2 \mid \deg B_i = \ell\right\} = \frac{q-1}{q^\ell}, \quad \text{(JP)}, \qquad \mu\otimes\mu\left\{f \in \mathbb{L}^2 \mid \deg B_i = \ell\right\} = \frac{q^2-1}{q^{2\ell}} \quad \text{(Brun)}\,.$$

Then, $\deg Q_k$ is a sum of k independent geometric variables of the same parameter q^{-1} or q^{-2}, respectively. This implies that the probability that $\deg Q_k = N$ for some k is equal to $(q-1)/q$ or $(q^2-1)/q^2$, respectively; it is thus independent of N.

For the fully subtractive algorithm, the distribution of $\deg B_i$ is not geometric. First of all, we note that $\deg Q_k = N$ implies $k \leq N$. We let

$$\nu_N = \mu\otimes\mu(F_N), \qquad \alpha_k = \mu\otimes\mu\left\{f \in \mathbb{L}^2 \mid \deg B_i = k\right\}, \qquad \bar{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n, \cdots)$$

One has $\nu_1 = \alpha_1$. With $\bar{\alpha}$, we define the bi-infinite matrix Z by blocks as $Z = \begin{pmatrix} \bar{\alpha} \\ I_\infty \end{pmatrix}$ and denote by $z_{ij}^{(n)}$, for $n \geq 1$ the coefficient of the power Z^n at the i-th row and the j-th column.

It is first clear that $z_{11}^{(N)} = \nu_N$. We now prove by induction the equality

$$z_{1j}^{(N)} = \mu\otimes\mu\left(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_k = N + j - 1 \text{ and } \deg Q_{k-1} < N\}\right)\,.$$

Suppose that the claim holds for $N \geq 1$. The following matricial equality holds:

$$z_{1j}^{(N+1)} = z_{11}^{(N)}\alpha_j + z_{1\,j+1}^{(N)}.$$

Because $(A_n, B_n)_{n\geq 1}$ is an i.i.d. sequence, it turns out that $z_{1j}^{(N+1)}$ is equal to

$$\begin{aligned}
&\mu\otimes\mu\left(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_k = N \text{ and } \deg B_{k+1} = j\}\right)\\
&+ \mu\otimes\mu\left(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_k = N + j \text{ and } \deg Q_{k-1} < N\}\right)\\
&= \mu\otimes\mu\left(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_k = N, \text{ and } \deg Q_{k+1} = (N+1) + j - 1\}\right)\\
&+\mu\otimes\mu\left(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_k = N + j \text{ and } \deg Q_{k-1} < N\}\right)\\
&= \mu\otimes\mu(\{f \in \mathbb{L}^2 \mid \exists k\,(1 \leq k \leq N) \deg Q_{k+1} = (N+1) + j - 1 \text{ and } \deg Q_{k-1} < N\})
\end{aligned}$$

where we should note that $k = N+1$ is possible only when $\deg Q_N = N$.

The matrix Z is an aperiodic irreducible stochastic matrix. We refer to [29, Chapter 5] for the detail of the theory of non-negative matrices of countable infinite states. By the Perron–Frobenius theorem, there exists a row stochastic eigenvector $\mathbf{p}$ such that each row of Z^n converges to $\mathbf{p}$ as $n \to \infty$. The first coordinate of $\mathbf{p}$ is the α that we need. The fact that $\mathbb{E}_{\mu\otimes\mu}[d_B] < \infty$ implies that the Markov chain associated with Z is positive recurrent. Moreover, the solution of the equation $\mathbf{u}Z = \mathbf{u}$ with $u_1 = 1$ can be computed inductively, and this shows that the normalizing constant

of $\mathbf{u}$ is equal to the inverse of $\mathbb{E}_{\mu\otimes\mu}[d_B]$. This proves the third assertion of the proposition, and ends the proof □

6. Conclusion and open problems

To our knowledge, this is the first study that provides a unified analysis of multidimensional gcd algorithms and their associated continued fraction maps for polynomials and formal power series with coefficients in a finite field. This analysis is based on the use of the transfer operator, which firstly provides a simple explanation for the invariance of the Haar measure. Secondly, the generating functions appear as dominant eigenvalues of the transfer operator. In this way, the main cost characteristics (expectation, variance) appear in a natural manner and are easily computed and compared.

The following features are central for developing our unified framework:

(i) the algorithms are based on the choice of a *specific component*;
(ii) they perform the *division* of the *other two* components by the specific component;
(iii) the algorithms deal with *partially ordered* inputs.

The present formalism covers classical algorithms (in the case of real numbers) such as the Jacobi–Perron, the Brun and the fully subtractive algorithms.

What about other algorithms? There exist other classical algorithms for real numbers, as the Selmer algorithm and the Poincaré algorithm, and we consider their possible adaptations to the context of positive characteristic.

– The Poincaré algorithm, described in [25], subtracts the smallest entry to the second largest one, and the second largest one to the largest one. It does not fit into the present framework *at all*, since it *does not use* the notion of a *specific* component, and it seems difficult to decompose a step of this algorithm into two steps of a "specific flavour".

– The Selmer algorithm subtracts the smallest entry to the largest one [30]. It may be adapted in order to enter the present framework. Even though the classical version of the Selmer algorithm uses subtractions, one can deal with its multiplicative version, more natural in the context of positive characteristic. Then, the multiplicative version performs *one* division of the largest component by the smallest component. Even though there is *only one* division, the framework described by Condition (ii) may be extended. Generally speaking, our analyses may be relatively easily adapted to algorithms that that *do not perform all* the divisions.

Concerning Condition (iii), the analysis will be more difficult in the case of *totally ordered inputs*. As already mentioned in Section 2.2 just after Eqns (11) and (12), we need indeed to completely re-order the input after the divisions. This involves a possible extra permutation μ (between R_1 and R_2) which acts *after* the divisions, whereas the permutation described by η (in the present study) *just observes* the ordering of the pair *before* the divisions.

Higher dimensions. In the general unified framework of the paper (i.e., partially ordered subsets), it seems possible to design a similar framework for higher dimensions ($n \geq 3$). Consider a set of $(n+1)$–uples of polynomials $R = (R_1, R_2, \ldots, R_{n+1})$, partially ordered, with $\deg R_{n+1} > \max(R_1, R_2, \ldots, R_n)$. Consider also a specific component, always different from R_{n+1} , defined by $\epsilon \in [1..n]$. It can be chosen by its position, with a given fixed ϵ (as in the case of the Jacobi-Perron algorithm where $\epsilon = 1$). There are then n positional algorithms of the Jacobi-Perron flavour. However, they are however all of the same vein, and they share the same behaviour. The specific component can also be chosen in terms of the rank $\eta \in [1..n]$ of the specific component inside the n-uple $(R_1, R_2, \ldots, R_n)$. The generalized Brun strategy is defined by $\epsilon = \eta = 1$, whereas the generalized fully subtractive strategy is defined by $\epsilon = \eta = n$. For $n+1 > 3$, there are intermediate strategies attached with each choice $\epsilon = \eta = i \in [1..n]$. Similar algorithms have been considered for subtractive algorithms in the real number case [9, 12].

The execution of such an algorithm will be considered as a succession of phases, as already seen for the analysis of the *ordered version* of the Brun algorithm in n dimensions in the real case [6]. As in the present study, we expect the first phase –the only one which is not degenerate and which

deals with the total dimension– to be the dominant one in terms of complexity and costs. The bivariate generating functions of the main costs during the first phase will be written as rational fractions with respect to z, with a denominator being a polynomial of degree at most n (in the n-dimensional case); the other phases, the degenerate ones, deal with dimensions smaller than n, and will have a negligible complexity compared to the first phase.

References

[1] V. Baladi, *Positive transfer operators and decay of correlations*, Advanced Series in Nonlinear Dynamics **16**, World Scientific Publishing Co., 2000.

[2] V. Baladi and B. Vallée, *Euclidean algorithms are Gaussian*, Journal of Number Theory **110** (2005), 331–386.

[3] H. Benamar, A. Chandoul, *Convergence of the Brun algorithm over the field of formal power series*, Journal of Number Theory **129** (2009), 621–631

[4] V. Berthé and H. Nakada, *On continued fraction expansions in positive characteristic: equivalence relations and some metric properties*, Expo. Math. **18** (2000), 257–284.

[5] V. Berthé, L. Lhote and B. Vallée, *Probabilistic analyses of the plain multiple gcd algorithm*, J. Symbolic Comput. **74** (2016), 425–474.

[6] V. Berthé, L. Lhote and B. Vallée, *The Brun gcd algorithm in high dimensions is almost always subtractive*, J. Symbolic Comput. **85** (2018), 72–107.

[7] V. Berthé, H. Nakada, R. Natsui and B. Vallée, *Fine costs for Euclid's algorithm on polynomials and Farey maps*, Advances Appl. Math. **54** (2014), 25–65.

[8] N. Bourbaki, *Variétés différentielles et analytiques. Fascicule de résultats*, Actualités Scientifiques et Industrielles **1347**, Hermann, Paris 1967.

[9] H. Bruin, R. Fokkink, C. Kraaikamp, *The convergence of the generalised Selmer algorithm*, Israel J. Math. **209** (2015), 803–823.

[10] K.Q. Feng, F. R. Wang, *The Jacobi-Perron algorithm on function fields*, Algebra Colloq. **1** (1994), 149–158.

[11] P. Flajolet and R. Sedgewick, *Analytic combinatorics*, Cambridge University Press, Cambridge, 2009.

[12] R. Fokkink, C. Kraaikamp, H. Nakada, *On Schweiger's problems on fully subtractive algorithms*, Israel J. Math. **186** (2011), 285–296.

[13] C. Friesen and D. Hensley, *The statistics of continued fractions for polynomials over a finite field*, Proc. Amer. Math. Soc.**124** (1996), 2661–2673.

[14] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, (2003).

[15] H.-K. Hwang, *On convergence rates in the central limit theorems for combinatorial structures*, European J. Combinatorics **19** (1998) 329–343.

[16] K. Inoue, *On the exponential convergence of Jacobi-Perron algorithm over* $\mathbb{F}(X)^d$, JP J. Algebra Number Theory Appl. **3** (2003), 27–41.

[17] K. Inoue and H. Nakada, *The modified Jacobi-Perron algorithm over* $F_q(X)^d$, Tokyo J. Math. **26** (2003), 447–470.

[18] A. Knopfmacher, J. Knopfmacher, *The exact length of the Euclidean algorithm in* $\mathbb{F}_q[X]$, Mathematika **35** (1988), 297–304.

[19] D. E. Knuth, *Seminumerical Algorithms*, 3rd Edition. Vol. 2 of The Art of Computer Programming, 1988. Addison-Wesley.

[20] L. Lhote and B. Vallée, *Gaussian laws for the main parameters of the Euclid algorithms*, Algorithmica **50** (2008), 497–554.

[21] B. Li and J. wu, *Beta-expansion and continued fraction expansion over formal Laurent series*, Finite Fields Appl. **14** (2008), 635–647.

[22] B. Li, J. Wu, J. Xu, *Metric properties and exceptional sets of* β*-expansions over formal Laurent series*, Monatsh. Math. **155** (2008), 145–160.

[23] D. H. Mayer, *On the thermodynamic formalism for the Gauss map*, Comm. Math. Phys. **130** (1990), 311–333.

[24] R. E. A. C. Paley and H. D. Ursell, *Continued fractions in several dimensions*, Math. Proc. Cambridge Philos. Soc. **26** (1930), 127–144.

[25] A. Nogueira, *The three dimensional Poincaré continued fraction algorithm*, Israël Journal of Mathematics **90** (1995), 373–401.

[26] R. Paysant-Leroux and E. Dubois, *Algorithme de Jacobi-Perron dans un corps de séries formelles*, C. R. Acad. Sci. Paris Sér. A-B **272** (1971), A564–A566.

[27] R. Paysant-Leroux and E. Dubois, *Etude métrique de l'algorithme de Jacobi-Perron dans un corps de séries formelles*, C. R. Acad. Sci. Paris **275** (1972), 683–686.

[28] D. Ruelle, *Thermodynamic formalism*, Addison Wesley, 1978.

[29] E. Seneta, *Nonnegative matrices and Markov chains*. Second edition. Springer Series in Statistics. Springer-Verlag, New York, 1981.

[30] F. Schweiger, *Multidimensional continued fractions*, Oxford Science Publications, Oxford University Press, Oxford, 2000.

[31] A. C. M. van Rooij, *Non-Archimedean functional analysis*, Monographs and Textbooks in Pure and Applied Math. **51**, Marcel Dekker, Inc., New York, 1978.

[32] B. Vallée, *Euclidean Dynamics*, Discrete Contin. Dyn. Syst. **15** (2006), 281–352.
[33] W. A. Veech, *Interval exchange transformations*, J. Analyse Math. **33** (1978), 222–272.
[34] J. Wu, *On the sum of degrees of digits occurring in continued fraction expansions of Laurent series*, Math. Proc. Cambridge Philos. Soc. **138** (2005), 9–20.

IRIF–Univ. Paris Diderot – Paris 7 & CNRS–Case 7014, 75205 Paris Cedex 13, France
Email address: berthe@irif.fr

Department of Mathematics, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
Email address: nakada@math.keio.ac.jp

Department of Mathematics, Japan Women's University, 2-8-1 Mejirodai, Bunkyou-ku, Tokyo, 112-8681, Japan
Email address: natsui@fc.jwu.ac.jp

GREYC–Université de Caen–Bd. Maréchal Juin, 14032 Caen Cedex, France
Email address: Brigitte.Vallee@unicaen.fr